AugToAct: Scaling Complex Human Activity Recognition with Few Labels

Abu Zaher Md Faridee
University of Maryland, Baltimore County
faridee1@umbc.edu

Md Abdullah Al Hafiz Khan
Philiphs Research North America
hafiz.khan@philips.com

Nilavra Pathak
Expedia Group
nilavrapthk7@gmail.com

Nirmalya Roy
University of Maryland, Baltimore County
nroy@umbc.edu

Abstract

Human activity recognition (HAR) from wearable sensor data has recently gained widespread adoption in a number of fields. However, recognizing complex human activities, postural and rhythmic body movements (e.g. dance, sports) is challenging due to the lack of domain-specific labeling information, the perpetual variability in human movement kinematics profiles due to age, sex, dexterity and the level of professional training. In this paper, we propose a deep activity recognition model to work with limited labeled data, both for simple and complex human activities. To mitigate the intra and inter-user spatio-temporal variability of movements, we posit novel data augmentation and domain normalization techniques. We depict a semi-supervised technique that learns noise and transformation invariant feature representation from sparsely labeled data to accommodate intra-personal and inter-user variations of human movement kinematics. We also postulate a transfer learning approach to learn domain invariant feature representations by minimizing the feature distribution distance between the source and target domains. We showcase the improved performance of our proposed framework, AugToAct, using a public HAR dataset. We also design our own data collection, annotation and experimental setup on complex dance activity recognition steps and kinematics movements where we achieved higher performance metrics with limited label data compared to simple activity recognition tasks.

CCS Concepts

• **Human-centered computing → Ubiquitous and mobile computing design and evaluation methods**; **Empirical studies in ubiquitous and mobile computing**; • **Computing methodologies → Supervised learning by classification**; **Dimensionality reduction and manifold learning**; **Transfer learning**; **Semi-supervised learning settings**; **Neural networks**.

Keywords

Activity Recognition, Convolutional Neural Network, Data Augmentation, Affine Transformation, Domain Adaptation, Semi supervised Learning, Wearable Sensors, Transfer learning, Dance

ACM Reference Format:
Abu Zaher Md Faridee, Md Abdullah Al Hafiz Khan, Nilavra Pathak, and Nirmalya Roy. 2019. AugToAct: Scaling Complex Human Activity Recognition with Few Labels. In *16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous), November 12–14, 2019, Houston, TX, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3360774.3360831

1 Introduction

The recent proliferation of low-cost wearable sensors and IoT devices have created a plethora of exciting new opportunities for Human Activity Recognition (HAR) applications embedded with innovative machine learning approaches. This research effort is breaking new grounds in various fields such as health-care, sports analytics, fitness monitoring and entertainment. The inbuilt inertial sensors such as accelerometer, gyroscope, magnetometer (commonly known as Inertial Measurement Unit or IMU) provide a wide range of spatio-temporal features in capturing the human body movements and kinematics than visual/depth sensing which suffers from user privacy concerns and overlapped/occluded field of view. This flexibility comes at a price though, as the raw features in NLP or computer vision domain often carry more contextual information, unlike inertial data streams. For example, the co-occurrence of the words can be utilized to learn the latent language representation, which can greatly reduce the number of samples required for downstream supervised learning task; whereas similar aspects are not always present in inertial data streams. This requirement on data labeling is exacerbated by the fact that even simple human activity datasets tend to be highly personal and heterogeneous due to the variations in age, sex, and physical condition across different users let alone the specialized activity datasets as observed in dance and sports.

While simple activities of daily living (ADLs) (e.g., walking, running, standing, sitting, and many more) are easier to label due to their longer performance duration, typically ranging in minutes to hours, the same may not be true for complex human activities. Therefore, building a reliable classifier model for recognizing, learning and assessing complex human activities such as the ones observed in dance and sports in the presence of limited label is non-trivial. Motivated by this, we take the example of building a

MobiQuitous, November 12–14, 2019, Houston, TX, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7283-1/19/11…$15.00
https://doi.org/10.1145/3360774.3360831

classifier for recognizing the dance micro-steps of a minute long dance choreography [9] where each micro-step (label) lasts only a few seconds. However, due to a wide variety of possible choreography within a single dance genre, each dance session requires the onerous step of relabeling the micro-steps even though they are thematically very similar. Also, unlike ADLs, dance moves are learned over time. Therefore, the labeling recorded for a novice may not be well justified for redeploying to classify the performance of the same performer after a few months of skill learning and training (which is also true for sports). Same would be true when a classifier is being trained on a well-defined performance of the instructor but then deployed to classify the performance of a beginner. In effect, the ripple of low generalizability of the complex activities dictates that they cannot be annotated without the help of the domain experts (e.g., dance/sports instructors/students) which makes the data annotation process computationally expensive and physically laborious. As such, there is a profound need for designing adaptable and scalable algorithms that help classify complex human activities without relying on large amounts of hand-crafted data and are able to operate with a large number of users without much supervision.

Over the last few years, deep neural networks (DNN) have become the most adopted methodology for supervised classification. They are less dependent on clever feature engineering and have shown strong generalization [58] ability compared to traditional supervised methods [22, 31]. However, their performance is also highly dependent on the quality of the datasets [25], without which they tend to overfit. Such problems arising from the lack of quality labeled data can be tackled in a number of ways. Data augmentation in the form of Gaussian noise addition, affine transformation, and permutation on the spatial domain can be used to synthetically increase the variations in training samples. This approach works as properly parameterized augmentation transformations can approximate the minute variations noticed when humans perform their activities [49], thus helping to cover unexplored input space and increase the generalizability [13] of the trained model. Limited training samples can lead to a constricted feature representation learning, which can be expanded by exploiting the distributions of the unlabeled data [8], circumventing the cost of manual labeling. However, semi-supervised learning (SSL) assumes the labeled and unlabeled data come from the same distribution [45] therefore reducing its effectiveness when domain shift is introduced [54] for making simple human activity models adaptable in presence of cross-user, device type and instance diversity [18] etc. Transfer learning and domain adaptation techniques [18, 54] have been shown to provide respectable performance gains with minimal training labels in the target domain. But such architectures still necessitate ample amount of labels in the source domain, which might not be possible in a lot of use cases. Development of a semi-supervised transfer learning architecture which can work with minimal labels in both source and target domain for complex human activity recognition tasks is therefore of great prospect.

Our proposed work, *AugToAct*, is directly aligned towards such goals, in which we combine augmentation transformations with deep semi-supervised learning to infer complex activities with the minimal labels in both source and target domains. We showcase how the *AugToAct* framework offers the highest amount of flexibility in activity classification tasks, both in terms of required labeled samples and the complexity of tasks. In addition to evaluating our model on simple ADLs, we design an experiment where we collect data on a complex set of dance moves on four subjects with varying levels of training and try to recognize the dance moves of the less experienced subject by training on the partially labeled moves on the instructor. To our knowledge, this is the first work that successfully combines data augmentation with semi-supervised transfer learning in activity recognition literature, particularly for complex activity such as dance. The key contributions of this paper are summarized as follows:

- *Deep Semi-Supervised Activity Recognition:* We propose a semi-supervised activity recognition technique that retains high classification accuracy in ADL classification with only a fraction of the labeled samples. Our model can retain 90% accuracy with only 25% labeled samples and 80% accuracy with even 6% labeled samples. We combine affine augmentation transformation on input data with an end-to-end convolutional autoencoding architecture to learn noise and transformation invariant features from the unlabeled data without much of the labeled information.
- *Deep Semi-Supervised Domain Adaptation:* We extend our SSL technique with domain adaptation which helps classify activities of multiple users while training on a single user with limited labeled information. Our end-to-end solution only assumes that the source and target domain labels are unchanged and requires no further feature engineering or parameter optimization.
- *Empirical Evaluation on Simple and Complex Human Activity:* We evaluate the effectiveness of *AugToAct* on public ADL datasets using a fraction of the labeled data both in the source and target domain and report superior performance compared to state of the art. To demonstrate the effectiveness of our approach on a complex HAR task, we design an experiment around dance micro activity recognition and collect data with four synchronous IMU sensor data streams, one on each limb of the dancer. We recognize complex dance choreography steps of novice students by training the model on a few labeled samples of the dance performance of a teacher. We noted higher performance gain of *AugToAct* compared to simple HAR benchmarks attesting the promise of our proposed architecture. We can retain high accuracy (89% in the target domain) with only 50% labelled data from both the source and target domain.

2 Related Works

In this section, we review relevant literature on data augmentation, semi-supervised learning and transfer learning as these three concepts form the core of our proposed methodology. We also try to pinpoint the key differences between our approach and those in the literature.

2.1 Data Augmentation

Data augmentation is a very simple and effective technique that boosts the accuracy of classifiers when very limited data labels are available. It helps to achieve peak performance in several deep learning algorithms [14, 21]. Recent works also show that data augmentation is not only valuable in input space, but also in the learned feature space [39]. Traditionally a staple in computer vision and audio/signal processing domain, data augmentation is having a renewed attention in time series augmentation tasks [10, 24]. Recent works show that data augmentation can significantly boost the performance in complex cases of activity recognition such as monitoring Parkinson's disease with wearable sensors [49] and combating domain shift in the form of software and hardware heterogeneity [30]. However, redundant and overly aggressive augmentation can slow down training and introduce biases into the data-set [15] and label preserving augmentation for inertial data can be challenging without domain knowledge. As such, in our proposed work we set the augmentation parameters sparingly and instead of performing augmentation on the whole data-set, we apply them in each batch of sliding windows in an online manner. This not only ensures that different part of the long activity sequence can achieve different augmentation variations but also average out any aggressive or bad augmentation at the beginning of the training stage.

2.2 Semi Supervised Learning

The lack of labeled data and associated cost of labelling has driven a great deal of research that exploit more easily available unlabeled data to perform feature learning among which Semi-supervised [45] and Self-taught [40] Learning has garnered most attention over the years. Semi-supervised learning assumes that labeled and unlabeled data come from the same distribution whereas self-taught learning makes no such assumption. Authors of [3] use sparse coding (a variant of self-taught learning) to derive over-complete basis vectors as features from unlabeled inertial streams. The authors of DeActive [16] have employed K-means clustering to derive codewords as an alternative to sparse coding for unsupervised feature extraction. However, they heavily relied on hand crafted feature engineering in the pre-processing step and their goal was more aligned with active learning methodology vs semi-supervised setting. Self-learning [41] and graph based [42] approaches have also garnered interests in unlabeled HAR tasks but these approaches often treat the feature learning and classification as separate tasks in which the correlation between the labeled and unlabeled data might get unexploited. Recent analysis [34] have shown that with proper setup semi-supervised learning can often match the performance of a supervised setup. Authors of [57] have demonstrated the effectiveness of semi-supervised learning in HAR context where they employ a set of stacked denoising auto-encoder with short-cut connections between encoder and decoder network (defined as Ladder architecture) while sharing the encoder parameters with the classifier and show that low-level features of the neural network gain the most from using the unlabeled data. Alternatively, Deep Auto-Set [50] employ a conventional auto-encoder instead of layer-wise pre-training and they switch to a more flexible set prediction objective. The work, however, does not employ any comparative study of using any fraction of the labeled samples. While our proposed method also employs a shared auto-encoding pipeline with the classifier, we also apply several affine augmentation transformation in addition to Gaussian noise with the objective of reconstructing the original non-transformed signal. We argue that compared to reconstructing from Gaussian noise-corrupted state, reconstructing the original signal from the affine-transformed state is a much harder objective, which forces our network to do better unsupervised feature learning.

2.3 Transfer Learning

Transfer learning and domain adaptation techniques have been traditionally applied in NLP [4] and computer vision [7] fields with great success. Recently, these techniques have garnered much interests in activity recognition domain [6] as well. Based on the primary mode of knowledge transfer, such techniques can be categorized into three types [37]: (i) *instance-based* methods use instance reweighting techniques to perform knowledge transfer [5, 44], (ii) *parameter-based* methods use clustering on the target domain after training source model on labeled data [56, 59], and (iii) *feature-based* methods learn a feature transformation between domains when the distance can be minimized [12, 36, 52]. Our proposed semi-supervised transfer learning work falls under the final category. Most of the recent feature transformation based domain adaptation uses deep learning architectures. Deep Domain Confusion (DDC) [47], Joint Adaptation Network (JAN) [28], Joint Distribution Adaptation (JDA) [27] are popular domain adaptation architectures in computer vision domain that tries to minimize maximum mean discrepancy (MMD) distance between the final deep layers. Domain Adversarial Neural Network (DANN) [11], Adversarial Discriminative Domain Adaptation (ADDA) [46], on the other hand, try to find domain invariant features from the source and target data-set with adversarial training. Although fewer in number, there have been several important domain adaptations works in HAR domain. Authors of [32] analyzed CNN based transfer learning approaches for wearable human activity recognition and showed that lower layer features are more transferable. Works of [53] suggested a transfer learning method to exploit the intra-affinity of classes to perform intra-class knowledge transfer. Alternatively, [1] follow a generative approach where they use variational auto-encoders to capture a stochastic feature space to transfer between sensors without any new labeled information and argue that stochastic features are more robust against additive noise than deterministic features of CNN. Almost all of the methods have an assumption that the source domain model is trained with adequate labeled data which might not be ideal for a real-world use case; our proposed architecture is specially designed to work in such situations.

There have been only a few works that examine SSL and transfer learning together where both the source and target domain have few labeled samples. Authors of [38] present a flexible semi-supervised transfer learning framework with the primary goal of protecting sensitive data samples against adversarial attack, and as a result, optimizing raw target domain classification performance is not the primary concern. More recently, authors of [60] systematically explore the use cases where SSL can provide tangible benefits over transfer learning and concludes that SSL based algorithms work best when the source domain is quite different from the target domain. However, they consider transfer learning in the context of only fine-tuning the weights of the model. Again, none of the models set any constraint on the labeled samples in the source domain which we are trying to address in our work.

3 Methodology

Figure 1 shows a general overview of our proposed *AugToAct* framework. Our semi-supervised learning idea is rooted in the working principle of denoising auto-encoders [51], in which reconstruction of the clean signal from Gaussian noise-corrupted input can lead to a much better unsupervised feature learning. During our data collection and analysis process of the complex activity recognition task (dance, detailed in a later section) we made a few interesting observations. The intra-class variability of the ADL tasks is caused by a few factors. In addition to the sensor noise, such variability can be introduced by the local vibrations of the sensors, placement of the sensors around the wrong axis or ideal point on the body across sessions. Moreover, the same ADL can be performed with a different state of body and mind by the same person (e.g. normal walking vs slower walking due to fatigue). We observed that such variation can manifest as a non-linear affine transformation on the original signal on the temporal domain and can be approximated with such augmentation transformation. As such, we introduce such augmentations on the input stream of the auto-encoder on the fly during training and force the autoencoder to reconstruct the original signal. As a result, the unsupervised feature extracted during the process is not only random noise invariant but also transformation (rotation, scaling, magnitude warp) invariant. The extracted features are shared with the classifier at training time which leads to a much simpler architecture, unlike layer-wise pre-training approaches of stacked denoising auto-encoders [51]. The modifications needed to adapt our SSL architecture into a transfer learning setup is shown in Figure 2. In that case, we first train this semi-supervised module on the source domain with limited labeled data. We then train another copy of the module with the unlabeled target data while enforcing another constraint in training which dictates that the feature weights on the corresponding source and target module need to have a similar distribution. We ensure such constraint by trying to minimize the *Jensen-Shannon Divergence* between the distribution of activation of the corresponding features in the source and target domain which in turn through back-propagation forces the weights of the layers to become similar as training goes on. Such network architecture is flexible enough that it can accommodate between an unsupervised, semi-supervised or full-supervised mode for both the source and target domain. We have the option to fine-tune parameters to put more or less emphasis on supervised, unsupervised or transfer learning component during training if required. In the following sections, we specify the details of each of the building blocks of our architecture with greater detail.

3.1 Supervised Activity Recognition

The supervised part of the architecture consists of 3 convolutional layers with increasing number of filters and decreasing kernel and receptive field shape in each successive layer. This is followed by two fully connected layers, the final layer being a soft-max layer. It is trained to minimize a *Categorical Cross Entropy Loss* ($\mathcal{L}_{cce_s}$), with the labeled training data. The architecture is very similar to our previous works in [9] which has proved to be a very strong and stable classifier complex HAR classification task. Dropout and batch-normalization (BN) are frequently used in convolutional neural network (CNN) architectures, to prevent over-fitting and ensure stable convergence of the optimizer by preventing internal co-variate shift [17], respectively. However, we have chosen to not use the Dropout in this architecture as the introduction of online augmentation already have a strong regularization effect and only apply *L2* regularization on the weight and bias parameters. In our experience, the augmentation and batch-normalization operations also seem to interfere with each other during the training process making it unstable and harder to converge. As such we remove batch-normalization altogether. To keep the layers normalized we opt to use self normalizing activation functions (SELU) [20] instead of conventional RELU activations.

3.2 Augmentation Techniques for Activity Recognition

Before we do any feature extraction during either of the unsupervised or supervised learning process, the input data stream first goes through the data augmentation module. We apply the following primary type of augmentation in IMU data streams.

Jittering: Random noise is added to the accelerometer data-stream to simulate local vibrations. We pick random noise values from a normal distribution with a very small standard deviation.

Scaling: We randomly scale the magnitude of the data to simulate the increased/reduced intensity the activity can be performed by the subjects, for example, higher magnitude of movement while walking faster compared to a normal walk.

Rotation: Applying random rotation to the data stream can help simulate several cases. Smaller values can account for the minute intra-class variations in angular velocity while larger rotation values can account for the occasional large local angular vibrations and misaligned sensor placement.

Time Warping: To account for the change in pace while performing an activity, we apply time warping by smoothly varying the distance between the samples by using linear interpolation.

To simplify scaling and rotation operation, we generate a random affine transformation matrix. Given a point $p = (p_x, p_y, p_z)$ and a unit vector $u = (u_x, u_y, u_z)$, where $u_x^2 + u_y^2 + u_z^2 = 1$, the matrix, $R_{(\theta,u)}$ for a rotation by an angle of θ about an axis in the direction of u can be calculated by Rodrigues' rotation formula [33]. The final rotation transformation can be calculated as follows:

$$R_{(\theta,u)}p = \tag{1}$$

$$\begin{bmatrix} \cos\theta + u_x^2(1-\cos\theta) & u_x u_y(1-\cos\theta) - u_z\sin\theta & u_x u_z(1-\cos\theta) + u_y\sin\theta \\ u_y u_x(1-\cos\theta) + u_z\sin\theta & \cos\theta + u_y^2(1-\cos\theta) & u_y u_z(1-\cos\theta) - u_x\sin\theta \\ u_z u_x(1-\cos\theta) - u_y\sin\theta & u_z u_y(1-\cos\theta) + u_x\sin\theta & \cos\theta + u_z^2(1-\cos\theta) \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

Similarly, to scale p by a vector $v = (v_x, v_y, v_z)$ we use the following formula which gets us the final scaled output:

$$S_v p = \begin{bmatrix} v_x & 0 & 0 \\ 0 & v_y & 0 \\ 0 & 0 & v_z \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{2}$$

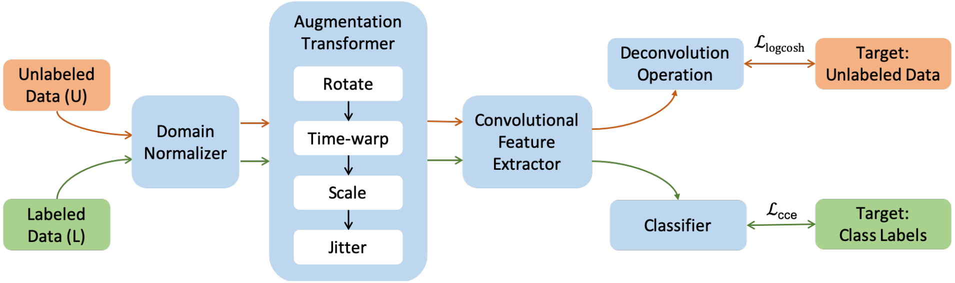

Figure 1: Primary semi-supervised learning setup in *AugToAct*.

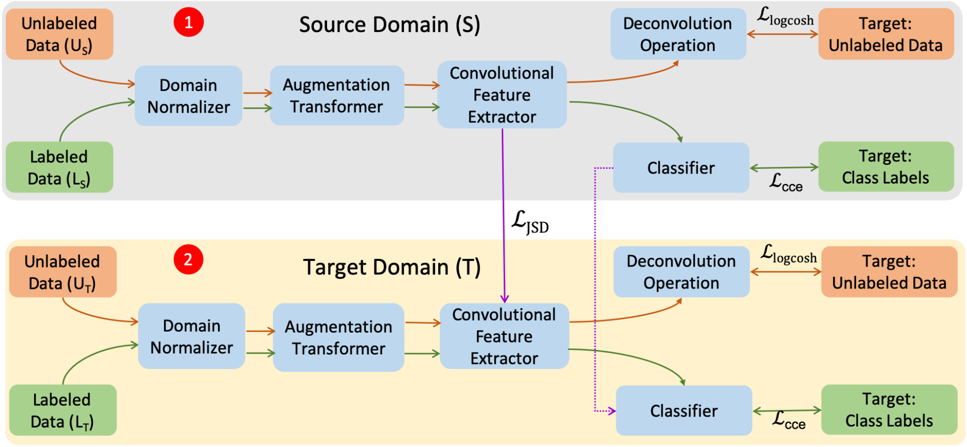

Figure 2: Full semi-supervised transfer learning setup in *AugToAct*.

We empirically discovered the following order of transformations to be the most stable one: rotation ⇒ time-warp ⇒ scaling ⇒ jitters. Doing the rotations, scaling and magnitude warping earlier and jitters later in the pipeline helps avoid extreme variations. Separating out the affine transformations between rotation and scaling ensures that we are able to maintain fine control over the parameters of the transformation while also able to simulate complex rotation, scale/shear operation at the same time. These transformations are applied online on each incoming batch of input sliding window samples and the parameters are chosen from random distributions (detailed in Section 4.2). Augmentation parameters are kept same across single sensor channels in a batch, but in case of multi-sensor setup, we chose to have differential augmentation which is more realistic as variations are most of the time local to the sensors.

3.3 Unsupervised Feature Learning

We employ a convolutional auto-encoding pipeline for the unsupervised feature learning part of our design. Similar to the supervised counterpart, the network consists of 3 convolutional layers with increasing number of filters and decreasing filter size and receptive field. They are now followed by 3 deconvolutional layers that mirrors the dimensions of the convolutional layers. As such the input and the output has the same size. The region between the 3rd convolution layer and 1st deconvolutional layer is called *bottleneck*. We use the augmented samples as the input to this network with the expectation of reconstructing the original samples in the output of the final deconvolution layer with a *Logcosh Loss* ($\mathcal{L}_{logcosh_s}$). We choose *Logcosh Loss* over *Mean Squared Loss (L2)* for its easier optimization characteristics [2]. By passing the augmented data through such layer arrangement and bottleneck and forcing to reconstruct the un-augmented input samples we allow the layer to learn the noise and transformation invariant feature representation from the unlabeled data, which is a more robust feature representation than what is normally learned by using a simple denoising auto-encoder. It is to be noted that the features extracted by such process are generally very domain-specific which is very helpful when the classification task is in the same domain but falls short when a domain shift is encountered. To handle those cases, we introduce the transfer learning pipeline described in the next section. During the main training phase, the first 3 convolutional layers are shared between auto-encoder and classifier and the auto-encoder and classifiers are trained simultaneously. We employ an alternate training policy where we switch between supervised and unsupervised data batches and optimize each sub-network. Compared to joint training, where supervised and unsupervised batches need to be properly aligned, such policy offers more flexibility in terms tuning the relative importance between unsupervised and supervised loss by looking on the ratio between the number of unsupervised and supervised data samples.

3.4 Transfer Learning

We bring transfer learning pipeline to handle domain shift in the data streams. In our experiments we have primarily focused on handling cross-user diversity where we train on partially labeled ADL data of one/few users and then adapt the model to work with ADL data streams of a larger number of users with minimal/no labels. For controlled comparison we assume the same type (model) of device being used to collect the all data. Our transfer learning module consists of two instances of the augmented semi-supervised module (*source* and *target* network) described in the previous section. Before training starts, we normalize both source and target data by their domain-specific mean and variance calculated from the unlabeled data pool of the respective domains. Performing domain-specific normalization on the input feature space ensures partial domain alignment [26] which helps the transfer learning algorithm achieve better convergence. The learning process assumes two discrete training phases. We first train the *source* network with partially labeled data from the source domain. On the second phase, *target* network is instantiated under the following conditions:

- The weight of the *target* softmax layer is initialized with the values from the *source* softmax layer
- In addition to the *Logcosh Loss* ($\mathcal{L}_{logcosh_t}$) and *Categorical Cross-entropy Loss* ($\mathcal{L}_{cce_t}$), we also add a new loss to minimize the *Jensen-Shannon Divergence* ($\mathcal{L}_{JS_i}, i = 1 \ldots 3$) between the activations of source and target convolutional outputs

For discrete probability distributions P and Q, *Jensen-Shannon divergene* (D_{JS}) is a symmetrized and smoothed version of the *Kullback-Leibler Divergence*, $D_{KL}(P \parallel Q)$. The *Kullback-Leibler* divergence from P to Q is defined to be:

$$D_{KL}(P \parallel Q) = -\sum_i P(i) \log \frac{Q(i)}{P(i)} \tag{3}$$

and thus D_{JS} is defined by:

$$D_{JS}(P \parallel Q) = \frac{1}{2} D_{KL}(P \parallel M) + \frac{1}{2} D_{KL}(Q \parallel M) \tag{4}$$

where $M = \frac{1}{2}(P + Q)$. We then pass the unlabeled/partially labeled data instances of the target domain through the newly instantiated network and evaluate the performance of our transfer model. In all cases of loss minimization, we use *Adam* optimizer [19]. The objective function for the source network is defined as,

$$\mathcal{L}_{Source} = \alpha_s \mathcal{L}_{cce_s} + \beta_s \mathcal{L}_{logcosh_s} \tag{5}$$

and for the target network

$$\mathcal{L}_{Target} = \alpha_t \mathcal{L}_{cce_t} + \beta_t \mathcal{L}_{logcosh_t} + \gamma \mathcal{L}_{JS_1} + \delta \mathcal{L}_{JS_2} + \kappa \mathcal{L}_{JS_3} \tag{6}$$

where $\alpha_s, \beta_s, \alpha_t, \beta_t, \gamma, \delta$ and κ are relative weight hyper-parameters between the losses that we also optimize with cross validation. If we need to transfer again to a new domain only the *target* network will require re-training with the incoming data stream.

4 Experiments

In the following sections we discuss the details of the datasets including the specifics of the dance choreography data collection and subsequent pre-processing steps. We then articulate the details of the implementation, discuss the experimentation results and perform evaluations in both semi-supervised and transfer learning settings.

4.1 Datasets

We showcase the effectiveness of our proposed framework on two data-sets:

1. We use *Heterogeneous Activity Recognition Data-set (HHAR)* [43] as a representative of simple ADL recognition tasks containing a single sensor stream with 3 inertial channels.
2. We conduct our own experiment to create a *Dance Activity Recognition Data-set*, which represents a complex HAR task, with 4 inertial sensors, 3 inertial channel per sensor.

We discuss the details of the two data-sets in the following sections.

Heterogeneous Activity Recognition Data-set (HHAR): This is a publicly available data-set that contains six different locomotive activities, with accelerometer data from 8 smart-phones and 4 smart-watches collected at 200Hz. The data was collected from 9 users, with the smart-phones placed in a waist pouch, and smart-watches mounted on each arm. HHAR data consists of 6 in-the-wild locomotive activities (biking, sitting, standing, walking, stair-sup, stairs-down), collected by 15 users, with phones mounted on 6 different body positions (chest, head, shin, thigh, upper arm waist) and a forearm-mounted smart-watch. For our study, we only use the smart-phone data-set with 3 inertial channels. For semi-supervised learning task, we individually calculate the accuracy score for each user. For transfer learning task, perform *leave-all-but-one-user-out* cross-validation, where we train on only one user and then test on the rest of the user and finally take the average of the results.

Dance Activity Recognition (DAR) Data-set: Recognizing dance activity is fundamentally different from recognizing and learning the traditional ADLs. Dancing requires grace and finesse, and involves repetitive movements of the fingers, hands, forearm, elbow, arm, legs, toes, waist, heads, etc., in a rhythmic fashion. It also reflects the delicacy and rhythm of different postures along with the cognitive ability and physical fitness of an individual. This makes DAR a perfect proxy for a complex human activity recognition task. In our experiment, we chose to study a classical Indian dance style: *Lasya* which is a subcategory of Manipuri [55] dance form; the dance is noted for its gentle, smooth and subtle limb movements. We designed a specific dance script for *Lasya* which a beginner would learn during the first few dance sessions which contain ten micro-steps, as described by their primary limb movements in Table 2. The transition between the dance steps can create complications as it introduces ambiguity in the label boundaries. Depending on the arrangements between the micro-steps, the transition can also vary. We made a simple assumption and considered the transition to be a part of the dance steps and labeled accordingly (after appending the transition parts to the micro-steps). Since dance involves different movements of limbs to perform distinct steps, it warrants more than one sensor to capture the user's actions with required accuracy [9], in this case, we used actigraph (model *wGT3X-BT*)[1] sensors placed in each of the limbs. We collected data for 20 trials out of which the first 10 trials were conducted as such that the participants danced only the specific micro-steps repetitively. The remaining 10 trials

[1] https://www.actigraphcorp.com/

Table 1: Percentage of samples per class in the DAR dataset

Class Label	Percentage	Class Label	Percentage
Step 1	7.96%	Step 6	5.16%
Step 2	3.55%	Step 7	12.19%
Step 3	11.79%	Step 8	13.02%
Step 4	12.74%	Step 9	12.65%
Step 5	9.11%	Step 10	11.77%

Table 2: Description of the dance labels in the DAR dataset

Class Label	Description
Step 1	Waving both hands from left to right
Step 2	Stepping left leg forward
Step 3	Clockwise rapid rotational Movement
Step 4	Taking two forward steps with extended arms
Step 5	Anti-clockwise rotational movement
Step 6	Move both wrists to left side
Step 7	Clockwise step-by-step slow rotation
Step 8	Anti-clockwise step-by-step slow rotation
Step 9	Clockwise rapid tiptoe rotation
Step 10	Anti-clockwise rapid tiptoe rotation

were recorded as a sequence of all micro-steps, e.g. the way they would naturally dance in a full routine. Such redundant ordering ensures that the classifier can learn each micro-steps without associating the transitions patterns with the class labels. We followed the same method for both the instructor and the learners. To gather realistic data we tried to emulate a classroom environment as much as possible. The learners' data were collected when the teacher was primarily performing the moves and the learners were following the instructor. This motivated the learners to try to match the speed of the instructor but at the same time introducing more mistakes than what they would normally do when they would perform the same moves in a much slower pace.

ActiGraph wGT3X-BT has a tri-axis accelerometer sensor, that gives us acceleration data for x, y and z axis at the desired sampling frequency (in this case 100 Hz) along with the UNIX time-stamp of each of the readings. The dance routines lasted for roughly one minute and each of the dance steps taking up between six to fourteen seconds (they are not of equal lengths). We recorded each dance session using a video camera and annotated each micro-step of the dance session by synchronizing the video with the accelerometer data stream. We synchronized the signals from each sensor, the video and the timestamps associated with them using *ELAN* software [23]. Because of the initial clock synchronization, all sensor samples are also properly aligned with each other in the end. The raw sample size of the training and testing data-set for both the instructor and the students is shown in Table 3. Due to the variable lengths between dance steps, we ended up with a little bit imbalanced distribution of the class labels, the average distribution of the class labels are shown in Table 1.

Table 3: Training and test set size in the DAR dataset

	Training	Testing	Validation
Instructor	56352x144	19560x144	19560x144
Student	112136x144	37632x144	37632x144

4.2 Implementation Details

We conducted our experiments on a Linux Server running Ubuntu 18.04 running on Intel Core i7-6850K CPU and 64GB DDR4 RAM and 4 Nvidia 1080Ti Graphics cards with 44GB of total VRAM. Python was used for all coding tasks. For the signal processing, filtering and shallow learning tasks we used libraries such as scikit-learn and scipy. For deep learning tasks we used Tensorflow with Keras frontend and the codes were written to run in parallel in all 4 GPUs. After the extraction (also synchronization, annotation in case of DAR data-set) of the data, we split the data between train and test set before applying any kind of pre-processing to ensures absolute zero overlappings between training and test samples. We followed standard 60/20/20 train/test/validation split. Then we divide the accelerometer data into 128 sample window with a sliding window offset of 16 which results in 87.5% overlap between the windows.

A sample window of 128 or smaller is preferred as during the training phase, the augmentation operations are performed online on each incoming samples. Having a smaller window ensures more varied augmentation on the input data stream because of the random nature of the augmentations. For a similar reason, a smaller batch size while training is preferred and we set out the batch size to 32. In order to perform *Jitter* augmentation, we apply Gaussian noise to the batch, with a standard deviation of 0.05. To apply *Magnitude Warping* and *Time Warping* on the batch, we first create random cubic spline curves with a standard deviation of 0.2 and a maximum frequency of 2. We then use this carves to shift the magnitude or temporal values in the samples. Rotation is done by randomly generating a rotation matrix for the axes. The rotation value is picked from a *von Mises distribution* [29] with 0 mean and kappa value of 0.5 which helps keep the majority of the values within a sensible range of $[-\pi/2, \pi/2]$. All these parameters were chosen to not introduce any extreme changes in the augmented batches compared to the original batches and we found these values worked similarly well for both data-set, suggesting that these parameters might be generalizable for most HAR use cases. We optimized the deep learning model hyper-parameters with *Randomized Search* search on the validation sets and performed the final evaluation on the held-out test data using precision, accuracy and recall metrics. Our optimal model parameters are listed in Table 4.

5 Results

Since our proposed framework AugToAct contains a number of modules performing augmentation, semi-supervised and transfer learning tasks, in order to evaluate their individual contribution we perform an ablation study. In the following section, we separately evaluate the semi-supervised learning module with and without the effect of augmentation. We then take the best performing combination of the operations and we use that for the transfer learning pipeline evaluation.

Table 4: Hyper-parameters of AugToAct model

Hyper-parameters	HHAR Data-set	DAR Data-set
Convolution layers	3	3
Convolution filters	64, 128, 256	64, 128, 256
Convolution filter shapes	15x1, 7x1, 5x1	15x1, 7x1, 5x1
Deconvolution layers	4	4
Deconvolution filters	256, 128, 64, 3	256, 128, 64, 12
Deconvolution filter shapes	5x1, 7x1, 15x1, 1x1	5x1, 7x1, 15x1, 1x1
Fully-connected (FC) layers	2	2
FC neurons	64, 6	64, 10
Batch size	32	32

5.1 Evaluation of Semi-Supervised Learning Module

We evaluate our SSL module by varying the ratio of labeled and unlabeled data. We start by using 0.63% labeled data and 99.38% unlabeled data and gradually increase the labeled data usage to 100%. At the same time, we also train the network with and without augmentation. We also have a baseline where we only use the labeled fraction. The resulting evaluation on HHAR and Dance Activity dataset is shown in Table 5. As we can see, using unlabeled data with augmentation always provide an accuracy boost. The results are more prominent in DAR data-set where it is possible to retain 90% accuracy with only 25% labeled data if augmentation and SSL are also used together. This is around 6% more accuracy gain over using only the labeled samples. Similarly, 80% accuracy can be retained with just 6.25% labeled samples (8.6% gain). On the HHAR data-set though, the gains are much more modest. Here 66.38% accuracy can be retained with only 0.63% labeled data with augmentation and semi-supervised learning, which is still around 3% accuracy boost over baseline. In addition to showing the effectiveness the interplay between such augmentation schemes in a semi-supervised setup, this benchmark also showcases the stark contrast between the classification challenges of complex activities (e.g. dance) and simple activities of daily living (ADL). With DAR data-set we consider 12 channel (4 sensors) input data stream vs 3 channel in HHAR, which gives us more features to make the classification. But as we observed, dance activities are still harder to classify than the ADL even with more feature involved and the accuracy falls more sharply compared to ADL data when few labels are available, providing more justification for the usage of the semi-supervised framework. Furthermore, such a result also showcases the opportunity to exploit partially labeled data in a domain adaptation scenario, which we discuss in the next section.

Table 5: Accuracy comparison of SSL and augmentation module with different ratio of labeled and unlabeled samples on DAR and HHAR dataset

Labeled	Unlabeled	Aug.	Accuracy (DAR)	Accuracy (HHAR)
0.63%	99.38%	Yes	**0.2308**	**0.6638**
0.63%	99.38%	No	0.1407	0.6462
0.63%	0%	Yes	0.2233	0.6538
0.63%	0%	No	0.1618	0.6343
1.25%	98.75%	Yes	**0.3673**	**0.7154**
1.25%	98.75%	No	0.2561	0.7037
1.25%	0%	Yes	0.3412	0.6912
1.25%	0%	No	0.2435	0.6827
6.25%	93.75%	Yes	**0.8088**	**0.7621**
6.25%	93.75%	No	0.7565	0.7578
6.25%	0%	Yes	0.7885	0.7501
6.25%	0%	No	0.7237	0.7436
12.50%	87.50%	Yes	**0.8812**	**0.8046**
12.50%	87.50%	No	0.7978	0.8013
12.50%	0%	Yes	0.8804	0.7913
12.50%	0%	No	0.8012	0.7851
25.00%	75%	Yes	**0.9056**	**0.8634**
25.00%	75%	No	0.8332	0.8589
25.00%	0%	Yes	0.8964	0.8578
25.00%	0%	No	0.8484	0.8552
37.50%	62.50%	Yes	**0.9070**	**0.9175**
37.50%	62.50%	No	0.8416	0.9113
37.50%	0%	Yes	0.8922	0.9054
37.50%	0%	No	0.8500	0.8921
50.00%	50%	Yes	**0.9056**	**0.9221**
50.00%	50%	No	0.8576	0.9189
50.00%	0%	Yes	0.8964	0.9178
50.00%	0%	No	0.8551	0.9156
62.50%	37.50%	Yes	**0.909**	**0.9371**
62.50%	37.50%	No	0.8559	0.9365
62.50%	0%	Yes	0.9040	0.9344
62.50%	0%	No	0.8602	0.9325
75.00%	25.00%	Yes	**0.9183**	**0.9439**
75.00%	25.00%	No	0.9056	0.9415
75.00%	0%	Yes	0.9183	0.9379
75.00%	0%	No	0.8888	0.9377
87.50%	12.50%	Yes	**0.9259**	**0.9507**
87.50%	12.50%	No	0.9259	0.9476
87.50%	0%	Yes	0.9242	0.9457
87.50%	0%	No	0.9183	0.9445
100.00%	0%	Yes	**0.9343**	**0.9589**
100.00%	0%	No	0.9301	0.9527

5.2 Evaluation of Transfer Learning Module

To evaluate our transfer learning module, while at the same time show the interaction between transfer learning and semi-supervised learning module, we train the source model with 50% labeled data and 50% unlabeled data with augmentation transformation enabled. We then vary the amount of labeled target data from 0% to 100% and show accuracy. A 3 layer CNN is treated as the baseline which

Table 6: Accuracy comparison of AugToAct and other transfer learning approaches on the DAR dataset.

labeled Data%	0%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Base CNN	N/A	0.6026	0.6410	0.7438	0.7641	0.7803	0.8128	0.8205	0.8359	0.8359	0.8385	0.8513	0.8538
AugToAct	**0.6279**	**0.6648**	**0.7167**	**0.7953**	**0.8325**	**0.8523**	**0.8842**	**0.8887**	**0.8948**	**0.8978**	**0.9004**	**0.9106**	**0.9138**
HDCNN	0.5761	0.6197	0.6781	0.7487	0.7592	0.7949	0.7976	0.8326	0.8230	0.8333	0.8428	0.8536	0.8685
DDC	0.5623	0.6180	0.6749	0.7277	0.7205	0.7641	0.7949	0.8205	0.8285	0.8354	0.8385	0.8487	0.8667

can only utilize the labeled fraction of the target domain. We compare the results with two other transfer learning approaches, HDCNN [18] and DDC [48], the results of which are shown in Table 6. We did not include any comparison with a few other transfer learning algorithms such as JDA [27] and TCA [35], as HDCNN [18] has been shown to provide superior performance to those and therefore we directly compare AugToAct's performance against HDCNN. We can see that our AugToAct model can produce better results than all other approaches in most cases. Interestingly, with 100% labeled data in the target domain, AugToAct can achieve 6% higher accuracy than the baseline CNN network, proving that our architecture can provide an improvement over classical deep architectures even if there is redundant labeling information available.

6 Conclusion and Future Works

In this paper we investigated the effectiveness of our *AugToAct* framework, a novel and flexible augmented, semi-supervised transfer learning framework. The modular architecture can be adapted to work in a variety of classification and domain adaptation tasks. We experimentally showcased that our architecture is especially suitable in classifying complex human activity recognition tasks. In the future, we want to keep investigating the further improvement of the network parameter optimization process. The data augmentation methodology described is still somewhat unguided, as we enumerated the optimal values of the parameters through experimentation. Although these values seemed generalizable for a number of simple and complex HAR tasks, in future, we wish to automate the process of finding the optimal parameter for the augmentation operation. In this paper, we only validated methodology work on two HAR datasets. In the future, we wish to further test the generalizability of the model on a few more data-sets which offer more variety of human activities. We intend to include more users with varying training and dexterity level into our study to further test the scalability issues of complex human activity recognition. Our experiment is also not designed around to work with unseen labels in the target domain, a weakness we plan to address in future work.

Acknowledgment

This research is partially supported by the NSF CAREER Award # 1750936, ONR under grant N00014-18-1-2462, and Alzheimer's Association, Grant/Award # AARG-17-533039. The authors would like to especially thank Fatema Hasan, who provided us with valuable consultations on the intricacies of the dance forms with endless patience and passion. We would also like to extend our gratitude to the team of volunteers for lending us their time and effort during the dance data collection process.

References

[1] Ali Akbari and Roozbeh Jafari. 2019. Transferring activity recognition models for new wearable sensors with deep generative domain adaptation. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks.* ACM, 85–96.

[2] Vasileios Belagiannis, Christian Rupprecht, Gustavo Carneiro, and Nassir Navab. 2015. Robust optimization for deep regression. In *Proceedings of the IEEE International Conference on Computer Vision.* 2830–2838.

[3] Sourav Bhattacharya, Petteri Nurmi, Nils Y. Hammerla, and Thomas Plötz. 2014. Using unlabeled data in a sparse-coding framework for human activity recognition. *Pervasive and Mobile Computing* 15 (2014), 242–262.

[4] John Blitzer, Ryan T. McDonald, and Fernando Pereira. 2006. Domain Adaptation with Structural Correspondence Learning. In *EMNLP 2006, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 22-23 July 2006, Sydney, Australia.* 120–128.

[5] Rita Chattopadhyay, Qian Sun, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. 2012. Multisource domain adaptation and its application to early detection of fatigue. *TKDD* 6, 4 (2012), 18:1–18:26.

[6] Diane J. Cook, Kyle D. Feuz, and Narayanan Chatapuram Krishnan. 2013. Transfer learning for activity recognition: a survey. *Knowl. Inf. Syst.* 36, 3 (2013), 537–556.

[7] Lixin Duan, Ivor W. Tsang, and Dong Xu. 2012. Domain Transfer Multiple Kernel Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 3 (2012), 465–479.

[8] Dumitru Erhan, Yoshua Bengio, Aaron C. Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why Does Unsupervised Pre-training Help Deep Learning? *Journal of Machine Learning Research* 11 (2010), 625–660.

[9] Abu Zaher Md Faridee, Sreenivasan Ramasamy Ramamurthy, H. M. Sajjad Hossain, and Nirmalya Roy. 2018. HappyFeet: Recognizing and Assessing Dance on the Floor. In *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications, HotMobile 2018, Tempe, AZ, USA, February 12-13, 2018.* 49–54.

[10] Germain Forestier, François Petitjean, Hoang Anh Dau, Geoffrey I Webb, and Eamonn Keogh. 2017. Generating synthetic time series to augment sparse datasets. In *Data Mining (ICDM), 2017 IEEE International Conference on.* IEEE, 865–870.

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2016. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research* 17 (2016), 59:1–59:35.

[12] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012.* 2066–2073.

[13] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. 2016. *Deep Learning.* MIT Press.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 770–778.

[15] Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. 2019. Population Based Augmentation: Efficient Learning of Augmentation Policy Schedules. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA.* 2731–2741.

[16] HM Sajjad Hossain, MD Abdullah AL Hafiz Khan, Nirmalya Roy, and Shimei Pan. 2018. DeActive: Scaling Activity Recognition with Active Deep Learning. *Proceedings of the ACM on Interactive, Multimedia, Wearable and Ubiquitous Technologies (IMWUT 2018)* (2018).

[17] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning.* 448–456.

[18] Md Abdullah Hafiz Khan, Nirmalya Roy, and Archan Misra. 2018. Scaling human activity recognition via deep learning-based domain adaptation. *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom) 2018* (2018).

[19] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[20] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. In *Advances in neural information processing systems.* 971–980.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[22] Nicholas D. Lane and Petko Georgiev. 2015. Can Deep Learning Revolutionize Mobile Sensing?. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications, HotMobile 2015, Santa Fe, NM, USA, February 12-13, 2015*. 117–122.

[23] Hedda Lausberg and Han Sloetjes. 2009. Coding gestural behavior with the NEUROGES-ELAN system. *Behavior research methods* 41, 3 (2009), 841–849.

[24] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. 2016. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*.

[25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[26] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. 2017. Revisiting Batch Normalization For Practical Domain Adaptation. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*.

[27] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. 2013. Transfer Feature Learning with Joint Distribution Adaptation. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. 2200–2207.

[28] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. 2017. Deep Transfer Learning with Joint Adaptation Networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 2208–2217.

[29] Kanti V Mardia and Peter E Jupp. 2009. *Directional statistics*. Vol. 494. John Wiley & Sons.

[30] Akhil Mathur, Tianlin Zhang, Sourav Bhattacharya, Petar Velickovic, Leonid Joffe, Nicholas D. Lane, Fahim Kawsar, and Pietro Liò. 2018. Using deep data augmentation training to address software and hardware heterogeneities in wearable and smartphone sensing devices. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN 2018, Porto, Portugal, April 11-13, 2018*. 200–211.

[31] Francisco Javier Ordóñez Morales and Daniel Roggen. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* 16, 1 (2016), 115.

[32] Francisco Javier Ordóñez Morales and Daniel Roggen. 2016. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. ACM, 92–99.

[33] Richard M. Murray, Zexiang Li, and Shankar Sastry. 1994. *A mathematical introduction to robotics manipulation*. CRC Press.

[34] Avital Oliver, Augustus Odena, Colin A. Raffel, Ekin Dogus Cubuk, and Ian J. Goodfellow. 2018. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*. 3239–3250.

[35] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. [n. d.]. Domain adaptation via transfer component analysis. *Neural Networks, IEEE Transactions, 2011* ([n. d.]).

[36] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. 2011. Domain Adaptation via Transfer Component Analysis. *IEEE Trans. Neural Networks* 22, 2 (2011), 199–210.

[37] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* 22, 10 (2010), 1345–1359.

[38] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. 2017. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[39] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 77–85.

[40] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. 2007. Self-taught learning: transfer learning from unlabeled data. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*. 759–766.

[41] Maja Stikic, Kristof Van Laerhoven, and Bernt Schiele. 2008. Exploring semi-supervised and active learning for activity recognition. In *12th IEEE International Symposium on Wearable Computers (ISWC 2008), September 28 - October 1, 2008, Pittsburgh, PA, USA*. 81–88.

[42] Maja Stikic, Diane Larlus, and Bernt Schiele. 2009. Multi-graph Based Semi-supervised Learning for Activity Recognition. In *13th IEEE International Symposium on Wearable Computers (ISWC 2009), 4-7 September 2009, Linz, Austria*. 85–92.

[43] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 127–140.

[44] Ben Tan, Yu Zhang, Sinno Jialin Pan, and Qiang Yang. 2017. Distant Domain Transfer Learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. 2604–2610.

[45] Philippe Thomas. 2009. Semi-Supervised Learning by Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien (Review). *IEEE Trans. Neural Networks* 20, 3 (2009), 542.

[46] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial Discriminative Domain Adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2962–2971.

[47] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep Domain Confusion: Maximizing for Domain Invariance. *CoRR* abs/1412.3474 (2014).

[48] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474* (2014).

[49] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. ACM, 216–220.

[50] Alireza Abedin Varamin, Ehsan Abbasnejad, Qinfeng Shi, Damith Chinthana Ranasinghe, and Seyed Hamid Rezatofighi. 2018. Deep Auto-Set: A Deep Auto-Encoder-Set Network for Activity Recognition Using Wearables. In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MobiQuitous 2018, 5-7 November 2018, New York City, NY, USA*. 246–253.

[51] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research* 11, Dec (2010), 3371–3408.

[52] Jindong Wang, Yiqiang Chen, Shuji Hao, Wenjie Feng, and Zhiqi Shen. 2017. Balanced Distribution Adaptation for Transfer Learning. In *2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, USA, November 18-21, 2017*. 1129–1134.

[53] Jindong Wang, Yiqiang Chen, Lisha Hu, Xiaohui Peng, and Philip S Yu. 2017. Stratified Transfer Learning for Cross-domain Activity Recognition. *arXiv preprint arXiv:1801.00820* (2017).

[54] Jindong Wang, Yiqiang Chen, Lisha Hu, Xiaohui Peng, and Philip S. Yu. 2018. Stratified Transfer Learning for Cross-domain Activity Recognition. In *2018 IEEE International Conference on Pervasive Computing and Communications, PerCom 2018, Athens, Greece, March 19-23, 2018*. 1–10.

[55] Wikipedia. 2017. Manipuri Dance — Wikipedia, The Free Encyclopedia.

[56] Yi Yao and Gianfranco Doretto. 2010. Boosting for transfer learning with multiple sources. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*. 1855–1862.

[57] Ming Zeng, Tong Yu, Xiao Wang, Le T Nguyen, Ole J Mengshoel, and Ian Lane. 2017. Semi-supervised convolutional neural networks for human activity recognition. In *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 522–529.

[58] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[59] Zhongtang Zhao, Yiqiang Chen, Junfa Liu, Zhiqi Shen, and Mingjie Liu. 2011. Cross-People Mobile-Phone Based Activity Recognition. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*. 2545–2550.

[60] Hong-Yu Zhou, Avital Oliver, Jianxin Wu, and Yefeng Zheng. 2018. When Semi-Supervised Learning Meets Transfer Learning: Training Strategies, Models and Datasets. *CoRR* abs/1812.05313 (2018).