DeepContext: Parameterized Compatibility-Based Attention CNN for Human Context Recognition

Abdulaziz Alajaji, Walter Gerych, Kavin Chandrasekaran, Luke Buquicchio,
Emmanuel Agu, Elke Rundensteiner
Worcester Polytechnic Institute, United States
{asalajaji, wgerych, kchandrasekaran, ljbuquicchio, emmanuel, rundenst}@wpi.edu

***Abstract*—The ubiquity of sensor-rich smartphones has increased interest in mobile context-aware sensing applications in domains such as ambient assisted living, remote health care, and sports injury detection. Recognizing the user's current context by analyzing their smartphone's sensor data is a critical problem for such applications. One of the major technical challenges for context recognition is reliable feature extraction due to coarse-grained labeling. In sensor data coarse-grained labeling, only certain parts of smartphone sensor data are truly representative of the assigned label, while their exact duration and location within the segment are unknown. To address this, we propose *DeepContext*, a deep learning based network architecture for recognizing a smartphone user's current context. *DeepContext* uses a Convolutional Neural Network (CNN) with parameterized compatibility-based attention to discover and focus on important parts of smartphone sensor data, mitigating coarse-grained weak labels and extracting salient discriminative features. *DeepContext* uses a joint-learning fusion strategy that utilizes both domain-specific handcrafted features and features that are autonomously generated by a Convolutional NeuralNetwork (CNN) . We demonstrate that *DeepContext* consistently outperforms prior state-of-the-art context recognition and human activity recognition deep learning models on smartphone context sensor data gathered from 100 participants by nearly 5% in Balanced Accuracy.**

***Index Terms*—Ubiquitous and mobile computing, Context-aware computing, Human context recognition, Deep learning.**

I. Introduction

Human Context Recognition (HCR) is the task of detecting a person's current situation including their location, physical state, and other semantic information [1]. Accurate HCR is an important problem in context-aware applications targeting a wide variety of domains including smart homes [24], assisted living [25], fitness tracking [22], military deployment [21], and mobile health [14], [17], [22], [28]. In healthcare, accurate HCR can facilitate passive context-specific patient assessments and continuous monitoring, decreasing operational costs [17]. Historically, HCR systems are used to determine user context utilizing data from custom body-worn sensors [36]. However, wearing such dedicated hardware and maintaining them (such as keeping their batteries charged) imposes a significant burden on users. Fortunately, smartphones have recently become popular for context-aware applications [12], [27], [30] as they are now ubiquitously owned (over 3.2 billion people globally [29]) and are often equipped with a wide variety of built-in sensors such as accelerometers, gyroscopes and light sensors [12].

This work is supported by the Computer Science Dept. at Worcester Polytechnic Institute and the DARPA WASH project, grant HR00111780032-WASH-FP-031.

Semantic Contexts for Health Assessment Testing: As part of our DARPA-funded Warfighter Analytics for Smartphone Healthcare (WASH) project, our group is developing methods to determine a user's context from sensor data gathered passively from their smartphone. We define a person's context as the tuple: <Physical State, Phone Prioception, App Usage, Social>, as described in Table (I). We focus on recognizing specific user contexts in which high-specificity health assessments for Traumatic Brain Injury (TBI) and infectious diseases can be performed on monitored smartphone users. For example, if our sensing application accurately recognizes a user's context to be $< *, \text{Phone In Hand}, *, * >$ (with "*" denoting a wild card), then additional tests to assess whether their hand is shaking (tremors) can be performed by analyzing data from their phone's accelerometer and gyroscope sensors. Shaking Hands (tremors) is a symptom of TBI and other diseases [18]. Examples of TBI and infection disease tests that could potentially be performed in specific user contexts recognized by HCR systems are listed in Table (II). In this work we do not focus on the detection of these ailments, but rather on detecting the contexts from which ailment tests can later be run.

Item	Potential Values
Physical Activity	{Walking, Running, Sitting, ...}
Phone Prioception	{*In Hand, In Bag, On Table, In Pocket*}
Social	{Alone, With People}
App Usage	{Multimedia, Texting, Games ...}

TABLE I: WASH Human Behavioral Context

Challenges of Context Detection: HCR systems typically assume strict (context) labels for supervised learning [30], wherein the labeled training data is indeed a true representative of its assigned label. In reality, the data labels in HCR datasets are typically coarse-grained, in the sense that only a subset of smartphone sensor data that is assigned a certain label exhibits patterns truly representative of that label.

State-of-Art Strategies: Prior studies have focused on the related problem of recognizing ambulatory human activities (e.g., sitting, walking, running, etc.), also called Human Activity Recognition (HAR), though they typically classify the sensor data into only one out of k possible labeled activities [10],

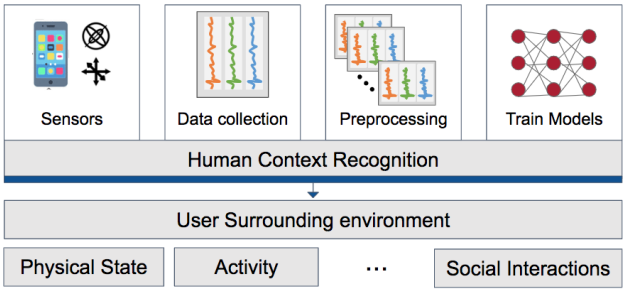


Fig. 1: HCR using smartphone sensor data.

[13]. However, while human context includes the person's current activity, it is also critical to include other semantic information such as their location and social situation. While there exist a few HCR methods that aim to classify human behavioral context [27], [30], they still do not address coarse-grained labeling.

Our solution: In this paper, we present *DeepContext*, a HCR system that uses neural networks to recognize the smartphone user contexts in which the TBI and Infectious diseases tests can be performed (See the Background section). *DeepContext* has two major innovations. First, *DeepContext* employs a joint-learning fusion strategy that utilizes both domain-specific handcrafted features and features that are autonomously generated by a Convolutional Neural Network (CNN). Second, *DeepContext* addresses the problem of coarse-grained labels by discovering and giving higher importance to the most salient regions of the sensor data. These regions are expected to correspond to a higher predictive value for specific contexts. This allows our model to overcome potentially noisy inputs, which is achieved by *DeepContext's* parametrized compatibility-based attention mechanism. Also, as many of our target activities can be performed concurrently (e.g., walking and talking on the phone), *DeepContext* formulates the HCR problem as a multi-label classification problem in a manner similar to Vaizman *et al* [30].

II. Related Work

Various deep-learning architectures have been proposed for Human Activity Recognition (HAR) from smartphone sensor data [8], [19], [23], [34], [35]. However, these architectures classify sensor data into only one of k possible labeled activities [10], [13]. Moreover, these conventional human activity recognition methods are not suitable for real-world problems since most of them assume that the sensor is placed at a fixed location on the body (hip, wrist or waist) [31]. Inspired by recent advancements in using multiple modalities for deep-learning models in other domains including computer vision and natural language processing, multiple-modality feature learning on sensor data has been found to be an effective way of learning more discriminative features and more generalizable models that can fit real-world settings [23], [30].

Vaizman *et al* recently gathered and studied a dataset containing a large collection of self-reported labels, fusing several smartphone and smartwatch modalities, classifying human behavioral context using shallow machine learners with handcrafted features [30]. The two leading deep learning methods are: 1) ExtraSensory: multi-layer perceptron context recognition architecture using handcrafted features and 2) DeepSense: generic deep learning-based activity recognition model using raw sensor data. These two methods do not address the challenge of coarse-grained labeling or weakly supervised learning.

Numerous attention mechanisms techniques have been proposed to improve classification accuracy and providing explainability in document classification, machine translation and recently for object detection and localization in images [11]. Wang et al [32] used an attention mechanism for human activity recognition from accelerometer data, addressing the same weak supervision problem as *DeepContext*, but applied it to recognizing a relatively smaller set of mutually exclusive labels. *DeepContext's* attention model is similar to that of Wang *et al* [32] but uses a parameterized compatibility-based attention model on multi-sensor CNNs. We also propose a new way of incorporating an attention mechanism on multiple sensors by first using a separate-and-merge [35] CNN and applying attention layers on features generated by single-sensor CNNs as well as on features generated by CNNs that analyzed the merged sensor outputs. The *DeepContext* multi-sensor fusion framework is also motivated by the ability to learn cross-sensor correlations using deep-learning on multiple modalities for ubiquitous computing [23].

III. Background

A. Background on the WASH Contexts

To explain our context definition (Table (I)), an individual's *Physical Activity* refers to what activity they are currently performing including ambulation activities such as walking or sitting, as well as complex activities such as eating, using the toilet or watching TV [13]. *Phone Prioception* describes the position or pocket in which the smartphone is currently being carried including whether the phone is in the user's bag, pocket, or on the table. Detecting a phone's prioception is important as signal patterns captured by the smartphone for the same activity (e.g., walking) may vary for different phone placements [16]. A phone's prioception could also be used to infer user-specific information such as their stride length [4] that can be used in TBI and infectious disease gait tests. *App usage* information can provide insights into an individual's behavior and health state. For instance, a decrease in the usage of social apps may indicate that the individual has begun to isolate themselves, a sign of TBI [3]. *Social* state of the individual (alone, with friends or co-workers) measures their degree of social interaction and isolation, and provides insights on whether the individual might have mental health ailments such as depression [6]. The behaviors and movement patterns of TBI patients are similar to those with depression.

B. Attention Mechanisms

Attention mechanisms are motivated by how humans pay visual attention only to specific regions of a picture or correlating words in a sentence [2], [33], [37]. Although some attention mechanisms are mainly used during post-hoc analysis of neural networks, several trainable attention mechanisms have been effective not only in increasing the neural network model's performance, but also in explaining the final predictions by facilitating the visualization of attention scores. There are two main types of attention mechanisms: 1) hard attention and 2) soft attention [33]. Hard attention is a stochastic process and often cannot be trained through back-propagation. Thus, the distribution of attention scores has to be assumed and fixed a priori [33]. Soft attention uses a probabilistic distribution function to apply attention scores to the source input [33], which makes it more suitable for sensor data, where a fixed distribution of scores, or the size and number of attention regions to focus on cannot be assumed a priori. Our attention mechanism is inspired by a promising model proposed by Jetley *et al*, an end-to-end trainable attention mechanism for CNN for the task of object detection and localization [11].

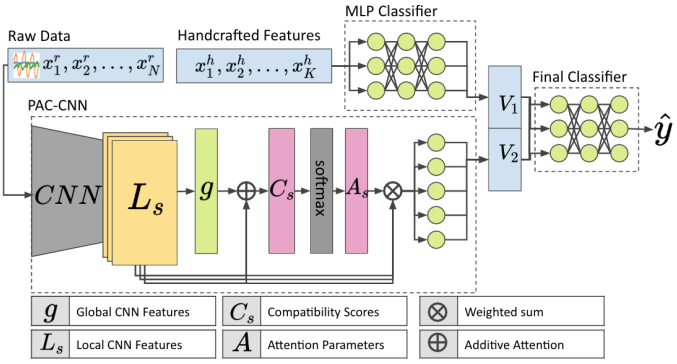
Traumatic Brain Injury	
Ailment Test	**Test Context**
Worse Reaction Time	<Interacting with Phone, in Hand, *, *>
Increased Light Sensitivity	<*, in Hand, *, *>
Unilateral Pupil Dilation	<Interacting w/ Phone, in Hand, Texting, *> <Interacting w/ Phone, in Hand, Video chat, *>
Hands Shaking	<*, in Hand, *, *>
Slurred Speech	<Talking into Phone, *, *, *>
Infectious Diseases	
Ailment Test	**Test Context**
Increased Cough Frequency	<Coughing, *, *, *>
Increased Sneezing	<Sneezing,*, *, *>
Resting Heart Rate	<Sitting, in Pocket, *, *>
Increased Toilet use Frequency	<Using Toilet, *, *, *>
Change in respiration	<Sleeping, on Table, *, *> <Exercising, *, *, *>
Both TBI and Infectious Disease	
Ailment Test	**Test Context**
Increase In Activity Transition Time	<Lying down, Phone In Pocket, *, *> <Sitting, Phone In Pocket, *, *> <Standing, Phone In Pocket, *, *>
Change in Sleep Quality	<Sleeping, *, *, *>
Change in Gait	<Walking, Phone in Pocket/Hand, *, *>

TABLE II: Context-specific ailment tests to detect TBI and infectious diseases and relevant human contexts.

C. Weakly supervised learning

Traditionally, in supervised learning tasks such as classification and regression, predictive models are trained on a large number of annotated training examples. A training example comprises of 1) an input feature vector (or instance), and an associated label (or ground-truth). Weakly supervised learning is categorized into three typical types: 1) incomplete supervision: utilizing unlabeled training data, 2) inexact supervision: only coarse-grained labels are provided, and 3) inaccurate supervision: where the labels are not always true [38]. In numerous tasks, it is difficult to gather strictly supervised information due to the costly data-labeling process. Thus, designing models that can work under weak supervision is desirable [32], [38]. *DeepContext* addresses coarse-grained labels in the HCR dataset.

IV. *DeepContext*



N: Raw Data segment size. K: Handcrafted Features dimension

Fig. 2: *DeepContext* architecture.

A. Overview

Our deep learning architecture for Human Context Recognition (*DeepContext*) is comprised of two CNNs that jointly learn from raw smartphone sensor data and handcrafted features in parallel, fusing their outputs. Fig. 2 shows the overall architecture of *DeepContext*. This joint learning fusion approach enables our model to learn not only discriminative features from handcrafted features and raw sensor data, but also from a shared representation, discovering complex cross-modality correlations. Moreover, the attention mechanism utilized enables *DeepContext* to learn salient features, giving higher weights (importance) to regions of the raw sensor data that contain predictive features for context recognition. Figure (3) shows *DeepContext's* classification pipeline. Sensor data is initially segmented using sliding windows to generate training instances, which are then input to CNN layers that extract feature vectors that are utilized for context prediction later in the pipeline [15]. The design of our CNN feature extractor follows a separate-and-merge strategy proposed in [35], where data generated by each sensor is first passed into a single-sensor CNN model that learns local interactions within each sensor. The outputs of individual single-sensor CNNs are then concatenated together to form a cross-modality representation

that is then passed to additional CNN layers to learn global cross-sensor interactions.

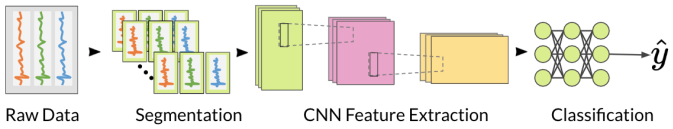


Fig. 3: Classification Pipeline for Raw Sensor data - showing a CNN feature generation approach.

B. Parameterized Compatibility-Based Attention Convolution Neural Network (PAC-CNN)

The context labels that subjects assign to smartphone sensor data during data gathering studies is often coarse-grained, making it challenging to create reliable context classifiers. Specifically, only relatively small regions of data that a user has assigned a given context label (e.g. walking) may actually be truly representative of that context. *DeepContext's* attention mechanism tries to learn the most relevant regions of the sensor data, which exhibit patterns that predict specific contexts. The intuition behind the design of its attention mechanism is similar to that proposed by Jetley *et al* [11].

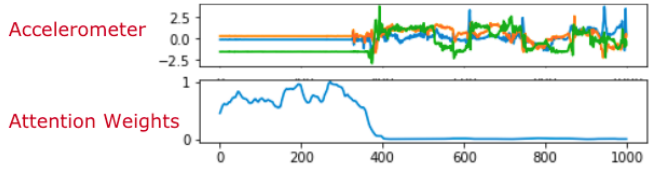


Fig. 4: The attention mechanism assigns more importance to regions of data that contain salient context-specific features extracted from raw sensor data. For instance, the attention mechanism learns that the left side of the accelerometer signal better represents *Phone on Table* context and assigns it higher weights.

In Fig (4), the attention model ignores parts of the sensor data when trying to classify the *"Phone on the table"* context. The model learns predictive patterns and increases their influence, while simultaneously suppressing irrelevant and potentially noisy parts of the data. As more data is utilized in training the model, it learns representations that are more generalizable and work better in real-world settings. The important regions detected within the data form saliency maps that could be analyzed to interprete classifier outputs, improve its performance and potentially facilitate the data-labeling process [32].

$\mathcal{L}^s = \{\ell_1^s, \ell_2^s, \cdots, \ell_n^s\}$ are intermediate (local) features extracted by convolutional layer $s \in \{1, 2, \ldots, S\}$, where l_s^i is extracted from the ith node out of a total of n nodes, each corresponding to one spatial location in the local feature vector $\mathcal{L}^s$.

In order to adapt the attention mechanism of Jetley *et al* [11] that was designed for images, to fit the multiple-modality nature of smartphone sensor data, we considered s to be various intermediate layers in the seperate-and-merge [35] CNN pipeline.

The flattened (global) feature vector G generated by the fully connected layer is combined with the final set of CNN-extracted (local) features. The attention mechanism tries to learn a compatibility score $\mathcal{C}\left(\hat{\mathcal{L}}^s, \boldsymbol{g}\right) = \{c_1^s, c_2^s, \ldots c_n^s\}$ between the local features $\mathcal{L}^s$ and the global feature vector G, and replaces the final feature vector with an attention-weighted local features [11].

To calculate the compatibility score, G and l_s^i are concatenated using an addition operation (additive attention [2]), followed by a dot product with a trainable weight vector u that can be expressed as [11]:

$$c_i^s = \langle u, l_i^s + G \rangle, i \in \{1, n\} \tag{1}$$

These learned compatibility scores c_i^s encourage the model to learn discriminative features tailored to different contexts. In order to utilize these learned compatibility scores $C(L^s, G) = \{c_1^s, c2^s, \ldots, c_n^s\}$ to produce a 1-dimensional vector $A^s = \{a_1^s, a2^s, \ldots, a_s^n\}$, a down-sampling convolutional layer is first applied, then the compatibility scores are normalized using a softmax function:

$$a_i^s = \frac{\exp\left(c_i^s\right)}{\sum_j^n \exp\left(c_j^s\right)} \tag{2}$$

The last step involves producing the final attention estimation g^s, replacing G, by taking the element-wise weighted average of the corresponding normalized compatibility scores in A^s with each node in L^s.

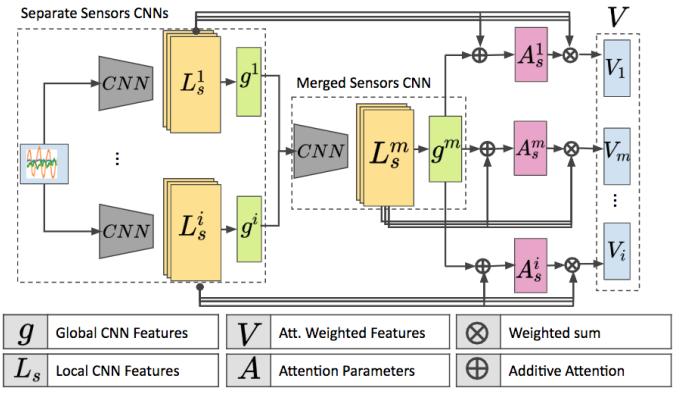


Fig. 5: Applying the attention mechanism on the separate-n-merge CNN architecture, where we use a separate CNN for each sensor modality, concatenating the resulting CNN outputs that are finally passed to the merged-sensors CNN. Only attention-weighted features are used for subsequent classification layers.

In Fig (5) we show the CNN architecture used.

$$g^s = \sum_{i=1}^{n} a_i^s \cdot l_i^s \tag{3}$$

C. Joint-learning Fusion

Taking advantage of the joint-learning fusion strategy, we can accommodate various modalities that cannot be fed to a CNN directly. By learning a shared representation between handcrafted features and CNN-generated features, our model increases its ability to learn cross-sensor representations that are more discriminating for prediction tasks [23]. This shared representation can act as a regularization technique and discover additional task-specific correlations between the handcrafted and CNN-generated features. To generate this shared representation, we first forward handcrafted features to a multi-layer-perceptron neural network, which consists of two layers, 16 hidden nodes in each layer, and uses Rectified Linear Units (ReLU) as its activation function. Then, we concatenate the resulting vector along with CNN-generated features after they are mapped to the same dimension. A sample list of handcrafted features [26] extracted from our smartphone sensor data is provided in Table III.

Feature	**Formulation**
Arithmetic mean	$\bar{s} = \frac{1}{N}\sum_{i=1}^{N} s_i$
Standard deviation	$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_i - \bar{s})^2}$
Median absolute deviation	$\text{median}_i\left(\lvert s_i - \text{median}_j\left(s_j\right)\rvert\right)$
Frequency signal Skewness	$\text{E}\left[\frac{(s-\bar{s})^3}{\sigma}\right]$
Frequency signal Kurtosis	$\mathbb{E}\left[(s-\bar{s})^4\right] / \text{E}\left[(s-\bar{s})^2\right]^2$
Interquartile range	$Q3(s) - Q1(s)$
Signal Entropy	$\sum_{i=1}^{N}\left(c_i \log\left(c_i\right)\right), c_i = s_i / \sum_{i=1}^{N} s_j$
Pearson Correlation coefficient	$C_{1,2}/\sqrt{C_{1,1}C_{2,2}}, C = \text{cov}\left(s_1, s_2\right)$
Spectral energy of a frequency band [a, b]	$\frac{1}{a-b+1}\sum_{i=a}^{b} s_i^2$

s: signal vector, N: signal vector length Q: quartile

TABLE III: A sample list of handcrafted features used for our sensor data, applied on accelerometer, gyroscope and magnetometer sensors [26].

V. Evaluation

We conducted experiments to evaluate *DeepContext's* performance for various segmentation window sizes (in seconds). First, we present and an overview of the WPI-WASH dataset utilized in our evaluations. Secondly, we describe the evaluation protocol and metrics used to assess the model's performance, given the imbalanced nature of the dataset. Finally, we assess the effectiveness of different components of *DeepContext* and discuss our empirical findings.

A. Dataset

We evaluated *DeepContext's* performance on our human smartphone context recognition dataset collected from 100 participants as part of our WASH project. In a scripted fashion, subjects were asked to visit 30 contexts while a smartphone data gathering app continuously gathered sensor data. The entire data gathering session lasted approximately 1 hour per subject. When expanded, the 30 contexts expanded to the 25 binary labels listed in Table (IV). Our experiments only studied *DeepContext's* performance for recognizing labels corresponding to "Physical State" and "Phone Placement" subcategories of the contexts defined in Table (I) The dataset was manually annotated by proctors who oversaw the study.

Phone Placement	
Phone in Bag	Phone in Hand
Phone in Table Facing Down	Phone in Table Facing Up
Phone in Pocket	
Long activity	
Walking	Sitting
Jumping	Jogging
Lying Down	Running
Standing	Sleeping
Stairs - Going Up	Stairs - Going Down
Talking On Phone	Trembling
Typing	In Bathroom
Short activity	
Coughing	Sneezing
Standing up (transition)	Laying Down (transition)
Sitting Down (transition)	Sitting Up (transition)

TABLE IV: Contexts for which data was gathered in our WASH Study Collected Contexts - Expanded into 25 binary labels

Coarse-grained labeling: The labels they assigned are however coarse-grained, not fine-grained labels. Formally, in our training data set $D = (X_1, y_1), \ldots, (X_m, y_m)$ where $X_i = \{x_{i1}, \ldots, x_{i,m}\} \subseteq \mathcal{X}$ is a bag, $y_i \subseteq \mathcal{Y} = \{0, 1\}$, X_i is a positive bag, i.e. $y_i = 1$, if there exist(s) one or more positive x_{ip}, while $p \in \{1, \ldots, m_i\}$ is unknown. In other words, within each training sensor segment, only certain sub-segment(s) are truly representative of the assigned label. However, their exact duration and location within the segment are unknown (corresponding to inexact supervision) [38].

B. Implementation

Our separate-n-merge CNN is has three layers per single-sensor CNN, and three additional layers for the merged-sensors' CNN. The number of feature maps generated in each CNN layer is 64. We also found that using larger filter sizes at the beginning of the pipeline produced better results, so we selected 8, 6, and 4 respectively as our filter sizes. We utilized Rectified Linear Units (ReLU) as our non-linear activation function. Our input batch size was 128 and we utilized dropout regularization with a probability of 20%, batch normalization, as well as L1/L2 normalization with a coefficient of $1e{-}5$. The model was trained for 100 epochs with early stopping if the validation loss stopped improving, to decrease the chance of over-fitting. For visualizing compatibility scores, we followed the same procedure used in [32].

C. Evaluation protocol

To ensure that our model generalized well when utilized on data from new subjects, previously unseen subjects during the training process, we adopted a user-level cross-validation approach (5 folds). Similar to the user-level splitting approach utilized by Vaizman *et al* [30], all of a subject's data may appear in either the training or test set, but not in both. Our final output is a multi-label output vector, where each label produced is a binary output (E.g walking vs not walking). To address the class-imbalanced nature of our WASH study dataset, we utilized *Balanced Accuracy* (BA), as our metric for evaluating our model's performance [5].

$$BA(\mathcal{D}) = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$$

which is also:

$$BA(\mathcal{D}) = \frac{1}{2}\left(Sensitivity + Specificity\right)$$

Also, in order to compute the BA of the context tuple after recomposition from the binary labels, we adopted macro-averaging that treats all binary labels with equal importance. That is, we calculate the BA score for each binary label separately and report the average across all binary labels (macro BA).

$$BA^{macro}(\mathcal{D}) = \sum_{c_i \in C} \frac{BA(\mathcal{D}, c_i)}{|C|}$$

.

When there are no annotated examples for c_i, then $BA(\mathcal{D}, c_i)$ is excluded from BA^{macro} calculation.

We compared our model performance against state of the art deep learning context (ExtraSensory MLP [30]) and Human Activity Recognition (HAR) (DeepSense CNN-GRU [35]) models. To ensure that our comparison was fair, we only utilized handcrafted features extracted from data from three sensors accelerometer, gyroscope and magnetometer. *DeepContext* and the other models compared against are all implemented in PyTorch [20], based on the authors' published source codes. Each model was then fine-tuned on our dataset and the same highly tuned number of layers and feature maps hyper-parameters for CNN were used in the DeepSense architecture to illustrate the efficiency of our attention mechanism. We generated results for a variety of window segmentation sizes to check that our models' performance was consistent.

D. Results

The overall performance of all evaluated models on our WPI-WASH dataset can be observed in Fig (6), where we compare *DeepContext* to state-of-the-art methods. Additionally, results for each label are reported in Table (V).

In order to demonstrate the effectiveness of *DeepContext*, in Figs (7) and (8), we evaluate the improvement that can be attributed to each component separately. The two components are 1) Parameterized compatibility-based attention and 2) Joint-learning fusion to incorporate handcrafted features.

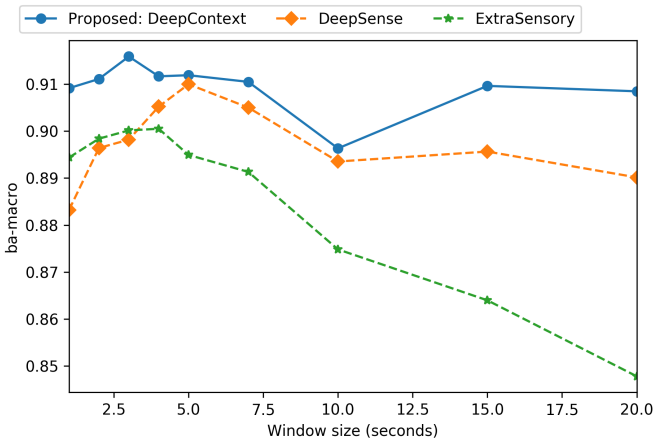


Fig. 6: *DeepSense* performance compared to other state-of-the-art deep learning methods

Label	DeepSense [35]	ExtraSensory [30]	***DeepContext***
Phone in Bag	**0.8940 ± 0.020**	0.7635 ± 0.045	0.8730 ± 0.036
Phone in Hand	0.8751 ± 0.028	0.7292 ± 0.037	**0.8862 ± 0.002**
Phone in Table, Facing Down	**0.9406 ± 0.019**	0.8720 ± 0.043	0.9370 ± 0.042
Phone in Table, Facing Up	**0.9529 ± 0.012**	0.8909 ± 0.042	0.9502 ± 0.024
Phone in Pocket	0.8201 ± 0.057	0.6838 ± 0.011	**0.8409 ± 0.036**
Walking	0.9074 ± 0.027	0.8936 ± 0.022	**0.9191 ± 0.026**
Sitting	0.9101 ± 0.037	0.8718 ± 0.032	**0.9143 ± 0.025**
Jumping	0.9250 ± 0.025	0.9004 ± 0.039	**0.9396 ± 0.004**
Jogging	0.9686 ± 0.004	0.9549 ± 0.006	**0.9739 ± 0.004**
Lying Down	**0.9276 ± 0.017**	0.8879 ± 0.011	0.9040 ± 0.022
Running	0.9267 ± 0.024	0.9193 ± 0.022	**0.9586 ± 0.013**
Standing	0.8224 ± 0.011	0.8266 ± 0.022	**0.8520 ± 0.034**
Sleeping	**0.9370 ± 0.022**	0.8732 ± 0.035	0.9175 ± 0.027
Stairs - Going Up	0.7997 ± 0.048	0.8160 ± 0.010	**0.8944 ± 0.041**
Stairs - Going Down	0.7408 ± 0.072	0.7860 ± 0.035	**0.8455 ± 0.041**
Talking On Phone	**0.9499 ± 0.003**	0.8581 ± 0.022	0.9152 ± 0.001
Trembling	0.8851 ± 0.114	0.8657 ± 0.065	**0.9414 ± 0.004**
Typing	**0.9727 ± 0.020**	0.9008 ± 0.045	0.9719 ± 0.017
Bathroom	**0.9072 ± 0.035**	0.8488 ± 0.038	0.8929 ± 0.004
Average	0.89804	0.84961	**0.91197**

TABLE V: Comparison of our Results with state-of-the-art methods, for window size = 20 seconds

Those components were placed on top of our core separate-n-merge CNN architecture. A magnified view of the two proposed components can be seen in Fig (7) to clearly show the usefulness of each one. The two proposed components are compared against our core Separate-n-merge CNN, using the same number of layers and fine hyper-parameters. We also experimented with various ways to increases the model's performance including 1) adding an LSTM layer after the final extracted features, and 2) increasing the complexity of the model by placing residual skip links on the merged-sensors CNN (Fig. 9)

VI. Discussion

We can observe that *DeepContext* consistently outperforms the state-of-the-art approaches compared against especially for larger window sizes when the data captures more background noise, and the user-provided ground-truth labeling becomes more coarse-grained and less accurately associated with the

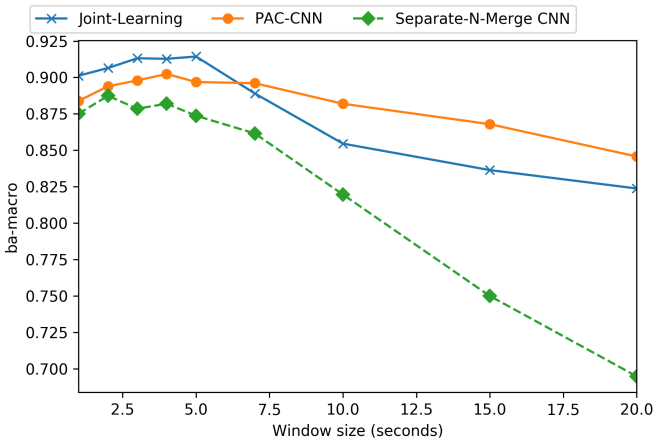


Fig. 7: Evaluating the effectiveness of *DeepContext* components separately

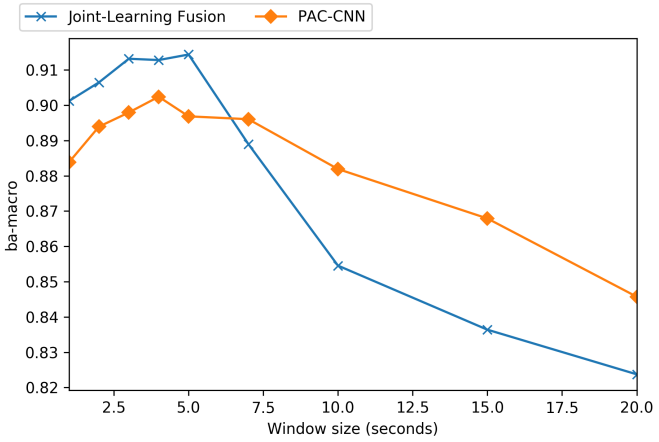


Fig. 8: Evaluating the effectiveness of *DeepContext* components separately - magnified view

entire training example. Additionally, from an application perspective, accurate predictions for larger segments are more useful, which indicates that more discriminative features have been learned regardless of the window size utilized. Intuitively, as we increase the window size, there is a greater chance for the attention mechanism to learn context-specific salient features, and more effectively suppress background noise occurring in the sensor data.

We speculate that the performance drop when using only handcrafted features with the core Separate-N-Merge CNN classifier might be because of the difficulty of capturing useful context-specific features when the window size gets larger. In Fig (9), there was a slight improvement when we tweaked the *DeepContext* architecture, by adding residual skip links [9], which demonstrates the potential for achieving even better performance by using such mechanisms on sensor data. We will explore residual skip links in future work. Figure (7) shows the significant improvements that our proposed submodules achieves on top of the Separate-n-merge architecture.

By looking at the detailed reported results per label, where

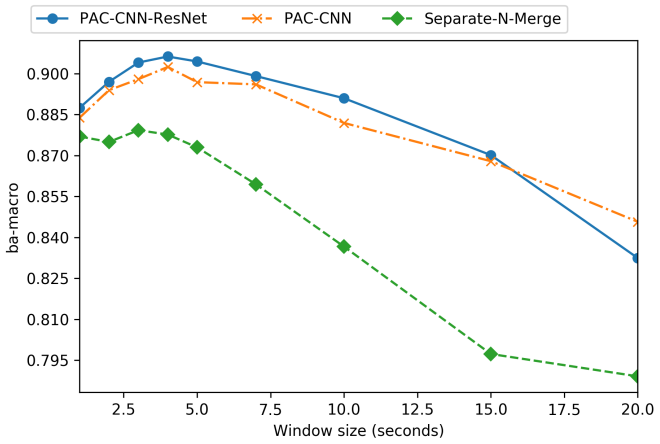


Fig. 9: Various ways to increase *DeepContext's* performance

we evaluated *DeepContext's* performance in comparison to state-of-the-art methods. *DeepContext* outperforms the other models for more than half of the labels. Even for labels where another model outperforms *DeepContext*, it's performance is very close score to that of the winning model. We speculate that this is due to the utilization of both deep learning based generated features and the domain-specific handcrafted features. One of the most challenging activities to detect, *Stairs - Going Up* and *Stairs - Going Down*, *DeepContext* is able significantly outperform the other state-of-the-arts methods. Detecting whether the subject is avoiding using stairs might provide useful insights about their mobility levels, which could facilitate the identification of potential ailments [13].

VII. Conclusion

We demonstrated the applicability of *DeepContext*, a deep learning based architecture for detecting a smartphone user's current context. Using a Convolutional Neural Network (CNN) with parameterized compatibility-based attention, *DeepContext* is able extract salient discriminative features under weakly labeled scenarios. Utilizing an attention mechanism, *DeepContext* can autonomously learn context-specific salient features, while suppressing potentially irrelevant parts of the input, tackling the issue of coarse-grained labeling that usually exists in smartphone sensor data. We have experimentally demonstrated the effectiveness of jointly learning from a combination of handcrafted features and CNN-generated features extracted from raw smartphone inertial sensor data. *DeepContext* consistently outperforms state-of-the-art methods on smartphone context sensor data gathered from 100 study participants. As future work, we aim to leverage additional contextual information gathered from subjects in-the-wild, such as semantic location, wireless connectivity, and phone state. *DeepContext* could also benefit from utilizing a large amount of data that subjects did not label in our study, using methods such as semi-supervised learning and context-aware semantic reasoning [7]

ACKNOWLEDGMENT

This work is supported by the Computer Science Dept. at Worcester Polytechnic Institute and the DARPA WASH grant HR00111780032-WASH-FP-031

REFERENCES

[1] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a Better Understanding of Context and Context-Awareness. In *Handheld and Ubiquitous Computing*, volume 1707, pages 304–307. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*, May 2016. arXiv: 1409.0473.

[3] Christina Baker-Sparr, Tessa Hart, Thomas Bergquist, Jennifer Bogner, Laura Dreer, Shannon Juengst, David Mellick, Therese M. ONeil-Pirozzi, Angelle M. Sander, and Gale G. Whiteneck. Internet and Social Media Use After Traumatic Brain Injury: A Traumatic Brain Injury Model Systems Study. *J. Head Trauma Rehab.*, page 1, April 2017.

[4] Jeffrey Basford, Li-Shan Chou, Kenton Kaufman, Robert Brey, Ann Walker, James Malec, Anne Moessner, and Allen Brown. An assessment of gait and balance deficits after traumatic brain injury. *Archives of physical medicine and rehabilitation*, 84:343–9, 03 2003.

[5] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. *Int'l Conf. on Pattern Recognition*, pages 3121–3124, 2010.

[6] Michelle Nicole Burns, Mark Begale, Jennifer Duffecy, Darren Gergle, Chris J Karr, Emily Giangrande, and David C Mohr. Harnessing Context Sensing to Develop a Mobile Intervention for Depression. *Journal of Medical Internet Research*, 13(3):e55, August 2011.

[7] Gabriele Civitarese, Riccardo Presotto, and Claudio Bettini. Context-driven Active and Incremental Activity Recognition. *arXiv:1906.03033 [cs]*, June 2019. arXiv: 1906.03033.

[8] Nils Y. Hammerla, Shane Halloran, and Thomas Ploetz. Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables. *arXiv:1604.08880 [cs, stat]*, April 2016. arXiv: 1604.08880.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. arXiv: 1512.03385.

[10] Seyed Amir Hoseini-Tabatabaei, Alexander Gluhak, and Rahim Tafazolli. A survey on smartphone-based systems for opportunistic user context recognition. *ACM Computing Surveys*, 45(3):1–51, June 2013.

[11] Saumya Jetley, Nicholas A. Lord, Namhoon Lee, and Philip H. S. Torr. Learn To Pay Attention. *arXiv:1804.02391 [cs]*, April 2018. arXiv: 1804.02391.

[12] Kostas Konsolakis, Hermie Hermens, Claudia Villalonga, Miriam Vollenbroek-Hutten, and Oresti Banos. Human Behaviour Analysis through Smartphones. *Proceedings*, 2(19):1243, October 2018.

[13] Oscar D. Lara and Miguel A. Labrador. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2013.

[14] Youngki Lee, Junehwa Song, Chulhong Min, Chanyou Hwang, Jaeung Lee, Inseok Hwang, Younghyun Ju, Chungkuk Yoo, Miri Moon, and Uichin Lee. SocioPhone: everyday face-to-face interaction monitoring platform using multi-phone sensor fusion. In *Proc. ACM MobiSys '13*, page 375, Taipei, Taiwan, 2013. ACM Press.

[15] Frédéric Li, Kimiaki Shirahama, Muhammad Nisar, Lukas Köping, and Marcin Grzegorzek. Comparison of Feature Learning Methods for Human Activity Recognition Using Wearable Sensors. *Sensors*, 18(3):679, February 2018.

[16] Emiliano Miluzzo, Nicholas D. Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proc. ACM SenSys '08*, page 337, Raleigh, NC, USA, 2008. ACM Press.

[17] Mashfiqui Rabbi Xiaochao Yang Hong Lu Giuseppe Cardone Shahid Ali Afsaneh Doryab Ethan Berke Andrew Campbell Tanzeem Choudhury. Mu Lin, Nicholas D. Lane. Bewell+: Multi-dimensional wellbeing monitoring with community-guided user feedback and energy optimization. *Wireless Health 2012*, October 2012.

[18] National Institute of Neurological Disorders and Stroke. *Tremor Fact Sheet*, 2019 (accessed October 3, 2019).

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

[20] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[21] Alfredo J. Perez, Miguel A. Labrador, and Sean J. Barbeau. G-sense: a scalable architecture for global sensing and monitoring. *IEEE Network*, 24, 2010.

[22] Mashfiqui Rabbi, Predrag Klasnja, Maureen Walton, Susan Murphy, Meredith Philyaw-Kotov, Jinseok Lee, Anthony Mansour, Laura Dent, Xiaolei Wang, Rebecca Cunningham, Erin Bonar, and Inbal Nahum-Shani. SARA: a mobile app to engage users in health data collection. In *Proc. ACM UbiComp '17*, pages 781–789, Maui, Hawaii, 2017.

[23] Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D. Lane, Cecilia Mascolo, Mahesh K. Marina, and Fahim Kawsar. Multimodal Deep Learning for Activity and Context Recognition. *ACM J. Interactive, Mobile, Wearable and Ubiq. Tech.*, 1(4):1–27, January 2018.

[24] P. Rashidi and D.J. Cook. Keeping the Resident in the Loop: Adapting the Smart Home to the User. *IEEE Trans.Sys., Man, and Cybernetics - Part A: Systems and Humans*, 39(5):949–959, September 2009.

[25] Parisa Rashidi and Alex Mihailidis. A Survey on Ambient-Assisted Living Tools for Older Adults. *IEEE Journal of Biomedical and Health Informatics*, 17(3):579–590, May 2013.

[26] Jorge-L. Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing*, 171:754–767, January 2016.

[27] Aaqib Saeed, Tanir Ozcelebi, Stojan Trajanovski, and Johan J. Lukkien. Learning behavioral context recognition with multi-stream temporal convolutional networks. *ArXiv*, abs/1808.08766, 2018.

[28] Bruno M.C. Silva, Joel J.P.C. Rodrigues, Isabel de la Torre Díez, Miguel López-Coronado, and Kashif Saleem. Mobile-health: A review of current state in 2015. *J. Biomed. Inf.*, 56:265–272, August 2015.

[29] Statista. Statista:smartphone users worldwide 2016-2021. https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, May 2013.

[30] Yonatan Vaizman, Nadir Weibel, and Gert R. G. Lanckriet. Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification. *IMWUT*, 1:168:1–168:22, 2017.

[31] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119:3–11, March 2019.

[32] Kun Wang, Jun He, and Lei Zhang. Attention-based Convolutional Neural Network for Weakly Labeled Human Activities Recognition with Wearable Sensors. *IEEE Sensors Journal*, 19(17):7598–7604, September 2019. arXiv: 1903.10909.

[33] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *arXiv:1502.03044 [cs]*, April 2016. arXiv: 1502.03044.

[34] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, and Shonali Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *IJCAI*, 2015.

[35] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek F. Abdelzaher. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *WWW*, 2016.

[36] Piero Zappi, Clemens Lombriser, Thomas Stiefmeier, Elisabetta Farella, Daniel Roggen, Luca Benini, and Gerhard Tröster. Activity Recognition from On-Body Sensors: Accuracy-Power Trade-Off by Dynamic Sensor Selection. In Roberto Verdone, editor, *Wireless Sensor Networks*, volume 4913, pages 17–33. Springer Berlin Heidelberg, 2008.

[37] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. *arXiv:1512.04150 [cs]*, December 2015. arXiv: 1512.04150.

[38] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, January 2018.