Multimodal Deep Learning for Activity and Context Recognition

Multimodal Deep Learning for Activity and Context Recognition

VALENTIN RADU, The University of Edinburgh
CATHERINE TONG, University of Oxford
SOURAV BHATTACHARYA, Nokia Bell Labs
NICHOLAS D. LANE, University of Oxford and Nokia Bell Labs
CECILIA MASCOLO, University of Cambridge
MAHESH K. MARINA, The University of Edinburgh
FAHIM KAWSAR, Nokia Bell Labs and TU Delft

Wearables and mobile devices see the world through the lens of half a dozen low-power sensors, such as, barometers, accelerometers, microphones and proximity detectors. But differences between sensors ranging from sampling rates, discrete and continuous data or even the data type itself make principled approaches to integrating these streams challenging. How, for example, is barometric pressure best combined with an audio sample to infer if a user is in a car, plane or bike? Critically for applications, how successfully sensor devices are able to maximize the information contained across these multi-modal sensor streams often dictates the fidelity at which they can track user behaviors and context changes. This paper studies the benefits of adopting *deep learning* algorithms for interpreting user activity and context as captured by multi-sensor systems. Specifically, we focus on four variations of deep neural networks that are based either on fully-connected Deep Neural Networks (DNNs) or Convolutional Neural Networks (CNNs). Two of these architectures follow conventional deep models by performing feature representation learning from a concatenation of sensor types. This classic approach is contrasted with a promising deep model variant characterized by modality-specific partitions of the architecture to maximize intra-modality learning. Our exploration represents the first time these architectures have been evaluated for multimodal deep learning under wearable data – and for convolutional layers within this architecture, it represents a novel architecture entirely. Experiments show these generic multimodal neural network models compete well with a rich variety of conventional hand-designed shallow methods (including feature extraction and classifier construction) and task-specific modeling pipelines, across a wide-range of sensor types and inference tasks (four different datasets). Although the training and inference overhead of these multimodal deep approaches is in some cases appreciable, we also demonstrate the feasibility of on-device mobile and wearable execution is not a barrier to adoption. This study is carefully constructed to focus on multimodal aspects of wearable data modeling for deep learning by providing a wide range of empirical observations, which we expect to have considerable value in the community. We summarize our observations into a series of practitioner rules-of-thumb and lessons learned that can guide the usage of multimodal deep learning for activity and context detection.

Additional Key Words and Phrases: Mobile sensing, sensor fusion, multi-modal, deep neural networks, deep learning, context detection, activity recognition

Authors' addresses: Valentin Radu, The University of Edinburgh; Catherine Tong, University of Oxford; Sourav Bhattacharya, Nokia Bell Labs; Nicholas D. Lane, University of Oxford and Nokia Bell Labs; Cecilia Mascolo, University of Cambridge; Mahesh K. Marina, The University of Edinburgh; Fahim Kawsar, Nokia Bell Labs and TU Delft.

2474-9567/2017/12-ART157 $15.00
https://doi.org/10.1145/3161174

ACM Reference Format:
Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D. Lane, Cecilia Mascolo, Mahesh K. Marina, and Fahim Kawsar. 2017. Multimodal Deep Learning for Activity and Context Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 157 (December 2017), 27 pages. https://doi.org/10.1145/3161174

1 INTRODUCTION

The popularity of wearables, and mobile sensing devices in general, has given rise to a growing interest in complex sensing applications (e.g. user activity and context recognition), with such tasks already available on commercial wearables to track jogging [42], sleep [48] and even posture [45]. Common to these recognition tasks is their reliance on numerous low-energy small form-factor sensors (e.g., light detector, magnetometer, accelerometer, barometer, heart-rate). With each sensing modality carrying a unique perspective, combinations of multiple such sensing streams can boost detection quality and exceed their potential in isolation. Taking this approach, the Microsoft Band [48] determines when a user is asleep by combining heart rate levels with accelerometer data of wrist motion, while the MSP [13] distinguishes between walking and climbing stairs, which are relatively similar in acceleration patterns, with extra information from a barometer.

Majority of current multimodal sensing solutions rely on *shallow* classifiers, (such as Decision Tree, Random Forest, SVM) operating on independent features extracted from each sensing modality. These features are used to perform sensor fusion following two strategies: Feature Concatenation (such as in [9, 28]) that treat features uniformly irrespective of their sensing modality to produce a single feature vector for classification; and Ensemble Classifiers (applied in [22, 72]) in which outputs of classifiers operating only on features of one modality are blended together. However, an important challenge for activity and context classification is to integrate seemingly incompatible sensor types (consider fusing accelerometer data and camera frames). Because sensing modalities vastly differ by sampling rate, statistical properties and data types, standard approaches for model training struggle to merge the information available from these diverse sources. The key here is to not only extract discriminative features from individual sensors, but also to discover features that jointly use separate sensors streams to capture information neither has in isolation.

Deep learning [2, 14] presents a promising, much unexplored opportunity to combat this sensors fusion challenge. In an area of rapid innovation, deep learning algorithms have shown to be remarkably successful in unimodal applications, such as recognition of words [27], objects [35] and faces [66]. One of its defining characteristics is the ability to learn dense hierarchical networks that transform relatively raw forms of data into inferences (e.g., an activity class). These networks merge the roles of features extraction and classification stages present in shallow modeling methods (e.g. SVMs [5]) and replace the need for hand-engineered, task-specific features with layers of data representations that act as features, automatically learned directly from data. There is already building evidence suggesting deep methods could overcome current bottlenecks in learning cross-sensor features for routine detections. New training methods that leverage variation in information [63], multi-view representations [69], or modified autoencoders [52, 70] are able to fuse highly heterogeneous pairs of data types, such as text mixed with images [64] and audio linked with video [52, 60]. The resulting bi-modality deep models offer considerable accuracy gains in tasks like image captioning [63] and emotion recognition [15, 33, 43] (merging facial expressions with sound).

This paper presents a case study of adopting deep learning algorithms for multimodal human activity and context recognition, using sensor data collected with mobile devices. We investigate the sensor fusion approaches taken in both deep and shallow learning, and ask the following questions: *(i) how do the studied techniques fare with existing practices? (ii) under what circumstances (e.g. modeling task, data types) is it beneficial to apply deep learning? (iii) how the technique may be deployed?* These questions are examined under two approaches to multimodal learning.

The first, *Feature Concatenation* (FC), is attractive in terms of simplicity as a strategy though it is at risk of missing crucial intra-modality correlations. The second, a novel alternative that we term *Modality-Specific Architecture* (MA) is a deep learning specific technique that places emphasis on learning both intra-modality and cross-modality relations. Our empirical study spans four datasets representative of a wide range of activity and context recognition tasks in ubiquitous computing. For each dataset, we evaluate: (i) four deep learning techniques based on FC and MA, with Deep Neural Networks (DNN) and Convolutional Neural Networks (CNNs) as base classifiers; (ii) two shallow classifier techniques: Decision Tree, Random Forest; (iii) any available task-specific classifier. Our MA architectures, trained here on mobile sensing data, are adaptations of the architecture first proposed in [52] for speech detection integrating video and sound modalities.

Our results show that these deep network architectures exceed current machine learning solutions over a range of mobile sensing tasks (human activity recognition, gait recognition, sleep stage detection and indoor-outdoor detection). This consistently better performance demonstrates a *general-purpose* characteristic of deep neural network architectures across diverse multimodal detection tasks, even matching that of highly-engineered *purpose-built* methods designed for a specific detection task. In addition to comparing accuracies, we also investigate the computational overhead of these techniques, as well as document the various lessons learnt in evaluating our training framework. Our findings suggest wearable resource limits (such as energy) are not a barrier to the adoption of explored deep learning methods.

Key contributions of this research are:

- A systematic study of multimodal deep learning techniques applied to a broad range of activity and context recognition tasks. Our empirical results demonstrate the ability of feature representational learning to produce accurate results even across highly heterogeneous sensors under different settings.
- We study the Modality-Specific Architecture, a specific type of split-architecture deep learning never previously applied to wearable modeling tasks. Within this variety of deep learning, we are also the first to test this architecture in-conjunction with convolutional layers (relative to feed-forward fully-connected DNN layers).
- Summary of experiences into general rules of thumbs and lessons learnt for future adoption of multimodal deep learning techniques in mobile sensing research.
- A system resource feasibility study of the overhead imposed by multimodal deep architectures. Experiments with two mobile processors show that memory, battery and computational footprint of these detection algorithms are not a barrier to their adoption.
- Development of a framework [18] to support training and evaluation of these models on different multimodal sensing tasks.

2 SHALLOW LEARNING APPROACHES

Shallow methods are commonly used for multimodal activity and context recognition. The term *shallow* is used to contrast with alternate deep learning architecutres [2, 14]. We first summarize the challenges of multimodal fusion, followed by surveying the two commonly adopted multimodal modeling strategies:

(i) Feature Concatenation

(ii) Ensemble Classifiers

2.1 Challenges for Multimodal Sensor Fusion

Different sensing modalities carry information with various perspectives, which often complement each other, allowing for useful information gain. As such, leveraging a diverse set of sensors on mobile devices when available can only be beneficial to detection accuracy.

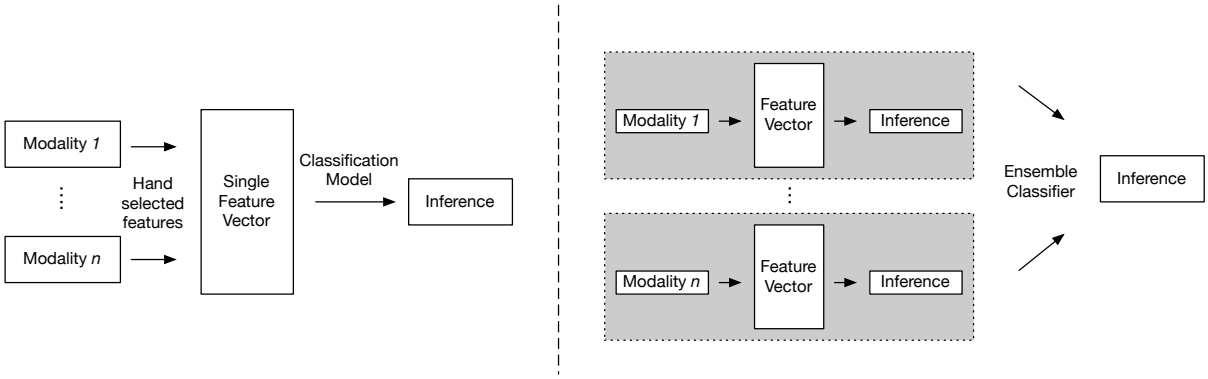
However, building appropriate models for multimodal fusion, i.e. models which fully leverage information contained within and across each sensor, is not trivial. The difficulty is mainly attributed to intrinsic differences between sensor data. Coming from different input channels with varying data types and sampling rates, each modality is characterized by distinctive statistical properties, representation and correlation structures. Typical models treat multimodal data as heterogeneous, resulting in ambiguous features and requiring special detection pipelines to deal with this multimodal data.

As a result of these differences, it is difficult to systematically recognize useful cross-modality relationships in addition to uni-modality ones. For instance, human speech can be perceived through three modalities: a stream of 2D-structured pixels (video), a time-series real-valued acoustic waveform (audio) and a sparse distribution of words (text). Cross-modality relationships which exist between low-level features across different modalities are highly non-linear and thus difficult to find. If we only consider video and audio, correlations are still highly non-linear and difficult to find from a mix of pixels and waveforms even if attempting by hand [52]. More generally, multi-modality data present even greater challenges since this difficult and time-consuming process needs to be performed for each sensor *pair*.

To illustrate this aspect in the context of multimodal sensing with mobile devices, we consider the following examples. A generic human activity recognition task utilizing the GPS, accelerometer and audio signals (e.g. indoor-outdoor detection) presents a range of challenges for modality fusion. Obviously, sampling rates differ substantially between sensor pairs: GPS signals change on time-scales of a minute, while accelerometer signal streams at sub-second rates (30Hz). This means that a GPS sample must be correlated to thousands of accelerometer readings; learning from such an imbalance distribution can easily degenerate typical models. Similarly, considering the GPS and audio signals, these are again very different. Even high sampling rate pairs like accelerometer and audio still need heavy filtering and preprocessing to align. Another example where these observations hold the same is the case of context understanding for indoor localization with smartphones. Typical sensing modalities for this task include inertial sensors (accelerometer and gyroscope) and WiFi scans, which come at different frequencies and in completely different formats (uniform data streams vs. uneven blobs of wireless environment observations). Integrating these sensing modalities is often done through heavy-engineered solutions, such as particle filters for indoor localization [58], but in most cases these are task-specific and constrained to a predetermined environment.

Several challenges exist in preprocessing sensor signals to comply with the input structure expected by a classifier. Generally, shallow classifiers operate on small size inputs which take advantage of the already distilled information presented to them in the form of features, specially hand-engineered for specific tasks – which can be an art on its own. Selecting the appropriate features may not be always intuitive to system designers even for a single modality; this difficulty is even more stringent with multimodal data as features have to complement each other across sensing modalities.

Nevertheless, shallow methods consider features as independent knobs, incapable of learning the key interdependent relations between sensing modalities, e.g. correlation in lip pose and motions between audio and visual data for speech recognition [52].



(a) Feature Concatenation (b) Ensemble Classifier

Fig. 1. Schematic of common approaches to shallow multimodal learning with (a) Feature Concatenation and (b) Ensemble Classifier models. For Feature Concatenation hand-selected features are extracted from each sensing modalities and concatenated into a single features vector as input to a classifier, while the Ensemble Classifier approach performs detections on each sensing modality independently to combine their estimations as final inference.

2.2 Strategies in Shallow Multimodal Learning

Existing sensor fusion strategies in multimodal activity recognition models can be categorized into two families, depending on whether sensor fusion occurs at a feature or at a classifier level. We call these (i) Feature Concatenation (FC), and (ii) Ensemble Classifiers (EC) respectively, schematically represented in Figure 1.

Feature Concatenation with Shallow Classifiers. With this strategy, hand-selected features from each modality are combined into a single feature vector presented to a classifier for detection across all features.

Various multimodal sensing systems have adopted this approach using different classifiers for diverse recognition tasks (SVM [9], AdaBoost [28], GMMs [44]). While some classifier types, such as ensemble learners like Random Forest [5], may do better than others at teasing out relationships between features the degree to which multimodal information is maximized is dependent on the quality of these hand-crafted uni- and cross-modal features. Often feature selection in concert with the extraction of a large number of candidate features for each sensing modality is attempted to automate this process (a technique adopted in systems like MSP [13]), though still bounded by the quality of selected features. In practice, Feature Concatenation can easily overlook inter-sensor relationships with the number of explored feature combinations limited by the curse of dimensionality [5].

Ensemble of Shallow Classifiers. On the other hand, the integration of modalities is done after separate classifiers operating on each sensor (modality) provide their estimation. These estimations provided by each sensing modality classifier are fused to yield an overall class estimation. A range of classifier fusion methods exists, including probability-based Bayesian fusion models and majority voting schemes (more details in [30]).

Like FC, variations of EC are also commonly adopted in multimodal activity models [30, 56, 72]. One key attraction is that available classifiers for each sensor type (tested and verified with other applications) can be readily adopted to undertake a new task on same sensing modality. In essence, this facilitates the creation of robust sensor-specific classifier generic to multiple tasks, while merging their results enables the evidence of each modality type to be considered before a final inference. However, a fundamental weakness of EC is that because fusion takes place so late a lot of potential information and cross-sensor relationships are already lost.

3 DEEP LEARNING TECHNIQUES

In this section, we discuss deep learning models used in multimodal activity and context recognition. We begin with an overview of existing methods, followed by a technical description of two kinds of multimodal deep learning models: *Feature Concatenation* (FC) and *Modality-Specific Architecture* (MA).

3.1 Overview

Deep learning models have been applied successfully to a growing number of detections with a single modality. Through their ability to learn feature representations directly from raw data (images, voice, text) rather than relying on domain-specific features and their hierarchical structure, deep models present a viable solution to overcome the challenges of multimodal sensing exposed above.

The structure of a generic deep learning architecture is presented in Figure 2. This consists of interconnected units that are grouped together in layers. Information propagates through layers, each performing transformations on their input as a function of internal feature representations, globally contributing to the final classification result. The first layer (input layer) accepts data in raw or lightly processed format, while the final layer (output layer) provides the class (e.g. categories of activity and context) according to the value of associated unit. Layers in between input and output layers are called hidden layers, because their values are not monitored, although essential to propagating information based on their layer parameters. Unit parameterization is automatically determined during training and depends on the layer type (e.g. whether it is convolutional or feed-forward layers) as well as training methods, pre-training and fine-tuning [14].

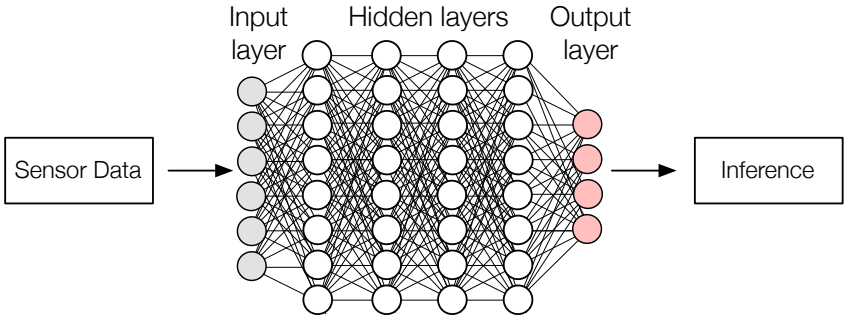


Fig. 2. Typical Deep Neural Network structure with feed-forward fully-connected layers.

Input Layer Representation. Deep architectures embody feature representational learning, with input layer taking values from raw sensor data, or lightly processed data.

Raw Data. Raw measurements, e.g. sensor readings, can be used as input directly.
Feature Selection. It is common for deep architectures to have a light preprocessing stage to change the dimensionality of the raw input signal (for example very sparse word vectors) in a process resembling features extraction in traditional classifiers, although this is a light and generic (valid across many tasks) data transformation process.

3.2 Multimodal Deep Learning

Deep learning architectures hold important properties which are advantageous to multimodal recognition tasks. As mentioned before, due to their feature representation learning, there is no need for a preprocessing phase to

extract domain-specific features from input data, this having two important consequences for multimodal data. First, custom layer representations can be trained to combine both uni-modal and cross-modal data from sensors. Second, the hierarchy in learned representations means that cross-modal relationships can be learned at both low- and high-levels of abstraction, corresponding to raw data and refined aggregated concepts respectively.

The same strategies for combining information from multimodal inputs as discussed for shallow classifiers can be applied to deep neural networks, based on the level where fusion is achieved: Feature Concatenation (FC) with concatenated inputs from multiple sensing modalities, and Modality-Specific Architecture (MA), with sensor-specific branches for each modality before fusion is achieved later in the network. We describe these two in more details below.

Essential to all machine learning classifiers is the training process. Various approaches to achieve training and tuning of multimodal architectures exist here, commonly built on the back-propagation algorithm. Back-propagation offers the flexibility of distributing gradients both on the joint section of the network, but also on the split section on each modalities branches. Flexibility of the network is not limited just to back-propagation, but also manifested in flexibility over data availability (or quality). As such, training algorithms specific to deep architectures facilitate robustness to missing modality inputs, which allows it to generate (or reconstruct) missing input modality from the available input by using the joint representation of relations between modalities. For example, auto-encoder algorithms, originally designed for uni-modal deep models to improve noise tolerance, are adapted to provide a type of tolerance to missing modalities [43, 52, 70]; this in turn, is understood to assist in discovering representations that are less prominent, but still discriminative, in power. One illustrative example of such an architecture is the image caption generation network, which can have its image representation layers improved (i.e., increased detection quality) through a training method that requires the model to generate both reasonable text and images when only an image is provided (with text related input layers set to a default state).

Many implementations of deep learning architectures with multiple modalities have been proposed in literature including Restricted Boltzmann Machine (RBM), CNN, DNN [51, 52]. We employ DNNs and CNNs under both sensor fusion strategies FC and MA, constructing 4 different architectures (FC-DNN, FC-CNN, MA-DNN, MA-CNN), which are described in the following sections.

3.3 Feature Concatenation Deep Learning

We refer to a commonly used approach in multimodal data integration using a deep classifier with concatenated modalities input as Feature Concatenation (FC).

In this approach, sensor fusion is performed right at the input layer by concatenating raw sensor streams (or lightly processed data) of multiple modalities, to achieve a single large input space. Data propagation pipeline inside the network proceeds as earlier described, performing a set of transformations on the concatenated input (Figure 2). This simple design allows easier training, since the model is less sensitive to hyper-parameter settings.

An important remark here is that feature representations inside the network have access to the whole space of sensing modalities (cross-modality information). However, previous work [52, 63] have shown that intra-sensor correlations (within the same modality) are stronger than inter-sensor ones (across multiple modalities). Since hidden layers in FC architectures are exposed to cross-modality information, it is harder to specialize them during training to extract the essential intra-sensor relations, so these get easily neglected. In addition, training an FC deep model is also problematic for an unbalanced mixture of inputs, as the units inside the network are easily dominated by those few proven modalities.

Feed-forward Deep Neural Network (FC-DNN). Feed-forward Deep Neural Networks are comprised of multiple stacked fully-connected layers, with the information passing from the input layer, starting as concatenated multimodal data and being transformed sequentially by each layer according to their internal feature representations and activations. Information flows in one direction through the network, which makes it easy to stack several hidden layers together. Class estimation is provided by the output layer as described before. In essence, modalities fusion is achieved at the input level by combining sensor streams into a joint input to propagate through the DNN.

Different activation function and regularization methods also bring their contribution to transformations propagated between hidden layers.

Convolutional Neural Network (FC-CNN). Similar to FC-DNN, sensing modalities are concatenated into a single input, which is interpreted with a Convolutional Neural Network in this case.

Convolutional Neural Networks have brought major leaps across many research areas [41]. Two layers are specific to this construction, Convolution layers and Pooling layers. Convolution layers are characterized by shared weights and biases as stacks of filters, much smaller in size than the input signal being convolved over. A stack of filters is trained to recognize different patterns (or features) no matter where these are encountered in the input space by sliding across the whole input. Although these filters are determined automatically in training, a good initialization strategy can determine non-overlapping behaviors of filters, each specializing in recognizing different patterns.

Convolution layers are typically followed by a Pooling layer, which reduces the size of the new representation constructed by filter activations. Pooling and Convolution layers can be stacked to produce more complex features, progressing in composition throughout the network. A common example comes from computer vision where first Convolution layers detect simple features like lines, points, colors, followed by other Convolution layers that activate for shapes, corners, edges and so on until by the end they recognize full body-parts or faces.

Last few layers in a CNN are typically fully-connected layers, as the ones found in DNNs (described in previous section), taking advantage of strong features generated by the previous layers to determine a class estimation.

3.4 Modality-Specific Architecture in Deep Learning

We refer to this construction comprising of two types of hidden layers – hidden layers related to a specific sensor type and hidden layers that capture unified concepts across sensor types – as Modality-Specific Architecture (MA). In this construction, separate architectures are built for each modality to first learn sensor-specific information before their generated concepts are unified through representations that bridge across all the sensors (i.e., shared modality representations) later in the network (as illustrated in Figure 3). MA is based on the architecture proposed in [52], although our formulation and experiments represent the first time this solution has been tested on mobile sensor data.

Deep Neural Networks (MA-DNNs). Formalizing the earlier high-level description of a multimodal deep learning architecture: the state ($\mathcal{A}_i^{L+1}$) of each individual DNN layer: (x_i^{L+1}) of layer ($L+1$) is dependent on the unit weights connecting the j^{th} node in layer L to the i^{th} node in layer $L+1$. The output is determined by the activation function, for example for a logistic activation function this can be formulated as:

$$\mathcal{A}_i^{L+1} = \frac{1}{1 + \exp(-\sum_j w_{ij}^{L+1} x_j^L)} \tag{1}$$

As shown in Figure 3, separate architectural branches (M_k) exist for each sensor type without any intra-branch connections between layers until later unifying cross sensor layers (U_l) in the larger multimodal architecture. While M_k layers learn representations tied to a single modality (such as the accelerometer), U_l layers seek to

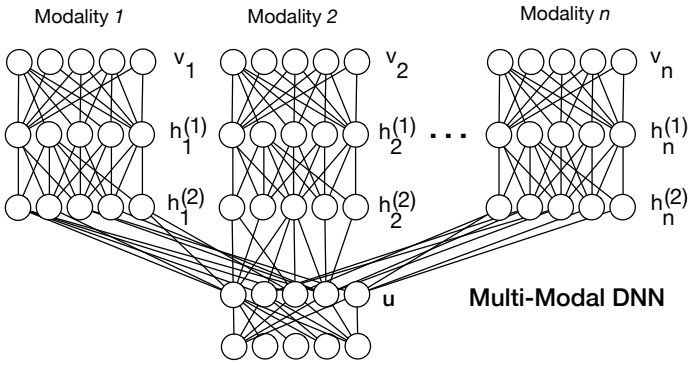


Fig. 3. Modality-specific Deep Neural Networks. The network adopts a split architecture: separate branches exist for each of n modalities, which are then joint in the unifying cross-sensor layers.

learn representations that fuse information between sensors. Collectively, all layers contribute to the learning of a joint representation of all sensor modalities; in other words, $P(\mathbf{v}_{acc.}, \mathbf{v}_{gyro.}, \mathbf{v}_{gps}, \ldots \mid \Theta)$ where Θ spans all model parameters. With respect to the architecture, a key hyper-parameter is the depth (i.e., the number of layers) of every M_k branch and U_l. This changes the complexity and richness of feature representation learned at each architectural branch. Some sensors are simple (such as a light indicator) requiring little representational power, others are complex (such as audio data) and benefit from the rich degree of processing to capture the information they contain. As a result, it is common for M_k to be of variable depth across different branches. The respective depth of each also impacts, at least conceptually, the type of semantic information that is being attempted to be fused between each sensor. Depth is also a factor for U_l in terms of controlling the opportunity for representations that fuse sensors to be discovered, therefore this is impacted by the richness, and sheer number, of the sensors that fan in. As is standard practice, such hyper-parameters are decided with cross-validation at training time.

Multimodal DNN Learning Algorithm. Conventional deep model training processes, like back-propagation with supervised training [14], is well suited for this task, due to the flexibility to split gradients onto each modality branch. In our earlier work [57], we have explored the unsupervised approach as an initial training step with auto-encoders, followed by traditional back-propagation to fine-tune the network feature representations in a construction called Restricted Boltzmann Machines. However, the impact of this extra step on accuracy is not significant enough overall to be deemed important, but the associated increase in training time is a penalty. For that reason we use the traditional back-propagation for the experiments presented in the evaluation section.

Performing Inferences. Post training, inferences with a multimodal deep model occur similarly as with regular DNNs. Sensor data of each type is provided as input to the corresponding architectural branch of the MA classifier. Each branch of the network performs its internal computations independently, same as described for FC before, producing feature signatures influenced by their internal feature representations. These are then combined in the joint part of the network, following a typical forward pass from here.

Convolutional Neural Network (MA-CNN). The multimodal construction using CNNs is analogous to that of MA-DNN. Each sensing modality has its own dedicated CNN to extract preliminary features over several layers (operating as described for FC-CNN). These produce intra-sensor features which are combined through fully-connected layers to identify the target class.

4 EVALUATION

In this section, we empirically compare different techniques for multimodal learning on wearable devices. We evaluate the performance of feature representational learning methods, with two feature concatenation (FC) deep classifiers: FC-DNN and FC-CNN, and two Modality-Specific Architecture (MA) deep classifiers: MA-DNN and MA-CNN, as detailed in Section 3. These are compared against two commonly adopted shallow methods, Random Forest (RF) or Decision Tree (DT), or any task-specific purpose-built technique where available. To offer an overview, our key findings are summaries below:

- Feature representational learning works consistently well across a wide range of activity recognition tasks (recognition of activity, gait, sleep stage and indoor-outdoor), outperforming shallow classifiers throughout while avoiding reliance on hand-tuned dedicated features.
- MA-CNN gives the best accuracy in 3 out of 4 datasets studied despite the nature of classification tasks being very different. Energy consumption measurements indicate this is also sustainable on common wearable devices. This makes MA-CNN a strong candidate as a default classifier for activity recognition and context detection with wearables and mobile devices.
- MA deep classifiers outperform FC on all four datasets, achieving accuracies that are on average 5% better. The difference in accuracies between the MA and FC approaches is most obvious in complex classification tasks, such as activity recognition, where MA outperform by up to 16%. Nevertheless, both MA and FC based deep classifiers achieve better accuracies than shallow classifiers.

4.1 Methods

This subsection details the datasets and baselines used for evaluation.

Datasets. We consider four publicly available datasets in order to represent custom setups in a wide range of activity recognition tasks performed on wearable and mobile devices. Table 1 summarises key features of the data.

STISEN Heterogeneity Activity Recognition Dataset, collected and studied by Stisen et al. [65]. This dataset contains readings of two motion sensors, Accelerometer and Gyroscope, from 9 users performing 6 activities (‘Biking’, ‘Sitting’, ‘Standing’, ‘Walking’, ‘Stair Up’, ‘Stair Down’). Great device diversity is captured in the dataset, each participant collecting data with 8 different smartphones and 4 smartwatches.

GAIT A dataset comprising of data from 460 participants, which forms a diverse sample of the population with distribution across ages (8 to 78 years old) and genders. Previously studied by Ngo et al. [53] with highly engineered signal-based solutions. Gait recognition is done on 5 classes: walking on flat surface, walking up slope, walking down slope, descending stairs and ascending stairs. The data was captured with two inertial sensors (accelerometer and gyroscope), each sampling in triaxial dimension. Remarkable to this dataset is the large population sample and diversity as mentioned earlier.

Sleep-Stage (SS) The Sleep-EDF Database [19], part of PhysioNet, contains physiological data (two EEG readers, one EOG and one EMG) collected from 20 people, annotated with 6 sleep stages (‘Awake’, ‘Stage 1’, ‘Stage 2’, ‘Stage 3’, ‘Stage 4’, ‘REM’). All participants in this dataset suffer from sleep disorders, making it substantially difficult to find simple patterns across all subjects.

Indoor-Outdoor (IO) This dataset [56] contains smartphone sensor readings (light, proximity, magnetic, microphone, cell, battery thermometer), from two different phones and annotated with ‘indoor’ or ‘outdoor’. This was collected in 3 different environments (university campus, city center, residential areas), which brings high variations in the signal patterns as previously highlighted in [56].

Dataset	No. of users	No. of classes	No. of modalities
STISEN	9	6	2
GAIT	460	5	2
Sleep-Stage	20	5	4
Indoor-Outdoor	2	2	7

Table 1. Summary of datasets used in our evaluation. Each dataset is selected based on its intrinsic complexity, such as the number of users (GAIT), diversity of sensing devices (STISEN), diversity of participants (SS) and number of sensing modalities (IO).

Baselines and tools. This section introduces the classifiers used for comparison on the four datasets summarized before (Table 1). We consider the following popular shallow classification techniques as our benchmark:

- Random Forest (RF): shallow-classifier-based feature concatenation, ensemble of decision tree classifiers;
- Decision Tree (DT): shallow-classifier-based feature concatenation; C4.5 is often used in wearable devices due to the low resource footprint and effectiveness in types of activities with few degrees of freedom (small feature space), e.g. walking or running.

For each recognition task, we compare the performance of deep classification techniques (FC-DNN, FC-CNN, MA-DNN, MA-CNN) with shallow techniques (RF and DT). In the case of the GAIT dataset, we also extend the comparison to five other purpose-built inference models, previously considered in the literature and established as best-performing for gait recognition [53].

We compare the performance of these classifiers based on the F1-score, defined as the harmonic mean of precision and recall, $\frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$. This metric is sensitive to misclassifications, and also robust to unbalanced distributions of samples across classes.

Methodology. Evaluation of shallow classifiers was performed in line with their original works, such as employing the same features as described in [65], for activity recognition or by extracting ECDF features (which shows good performance in our evaluation despite its compression effect). In cases where an earlier analysis is not available (e.g. for SS), training and testing of traditional classifiers are performed with Weka [23].

Datasets are split into training set and test set following the leave-one-outmethod, permuting which instance is left aside as test data and averaging across all iterations to get the final performance. It is never trivial to train deep neural networks optimally, so to speed up the training process, we performed a random search to identify the most appropriate hyperparameters (including depth and number of nodes) for each task, guided by the best F1-score on the test set.

Deep neural networks are particularly good at extracting their own internal features representation, performing well directly on raw data. The only intervention on these datasets was to normalize the sampling rate with a low-high pass filter, this being common practice when using Android devices due to the irregular sampling rate. Sampling frequency is chosen in alignment with previous analysis on such datasets. Appropriate time window size is another task-specific parameter as well as the overlapping between signal segments (experimenting with values between 50% and 70%), which control the amount of useful information provided to the classifier as independent inputs and increase the number of training samples respectively. We consider these the minimal preprocessing requirements for training with deep neural networks. To highlight the advantages of just minimal preprocessing of data for training deep neural networks, we perform other common preprocessing transformations like frequency domain (by

transforming the time window signal using the Fast Fourier Transform) and the Empirical Cumulative Distribution Function (ECDF) features at specific interest points, to compare with.

4.2 Comparison of Multimodal Context Recognition Techniques

Here we present the results achieved by the previously described multimodal techniques on the four context recognition tasks: activity recognition (Stisen dataset), gait recognition (Gait dataset), sleep stage detection (SS dataset) and indoor vs. outdoor detection (IO dataset).

Figure 4 shows the average F1-scores of selected classifiers. We interpret these results taking the following views:

Deep vs. Shallow Classifier. Deep learning classifiers outperform shallow classifiers across all four datasets, generalizing across diverse tasks without the pain of features identification as required when working with traditional shallow classifiers. When comparing deep solutions against common shallow classifiers (DT, RF), the average accuracy difference is substantial, 27%. CNN-based architectures dominate in all but one dataset (GAIT), where a task-specific approach [53] performs slightly better.

Deep: Feature Concatenation vs. Modality-Specific Architecture. Table 2 presents the F1 scores for best performing MA deep classifiers, FC deep classifiers and shallow classifiers across the four datasets. From this we see that MA consistently outperforms FC deep architectures in terms of accuracy. This is an indicator that early representations on each sensing modality help to discriminate between classes right from the first few layers of the network, in contrast to concatenated modalities inputs which mix data representations too early, thus missing valuable insights within each sensing modality.

Feature Concatenation: Deep vs. Shallow. Considering both shallow and deep classifiers adopting FC, FC-deep classifiers on average outperform FC-shallow classifiers, by 24% in many cases.

	MA- deep	FC- deep	Best-performing shallow
STISEN	**81.6**	70.36	74.5 (RF)
GAIT	89.5	88.6	**93.22** (NGO2014)
Sleep-Stage	**66.4**	65.1	55.04 (RF)
Indoor-Outdoor	**82.3**	80.1	58.92 (RF)

Table 2. Comparison of MA-deep classifier, FC-deep classifier and the best performing shallow classifier for each dataset. We highlight the best-performing architecture under each task.

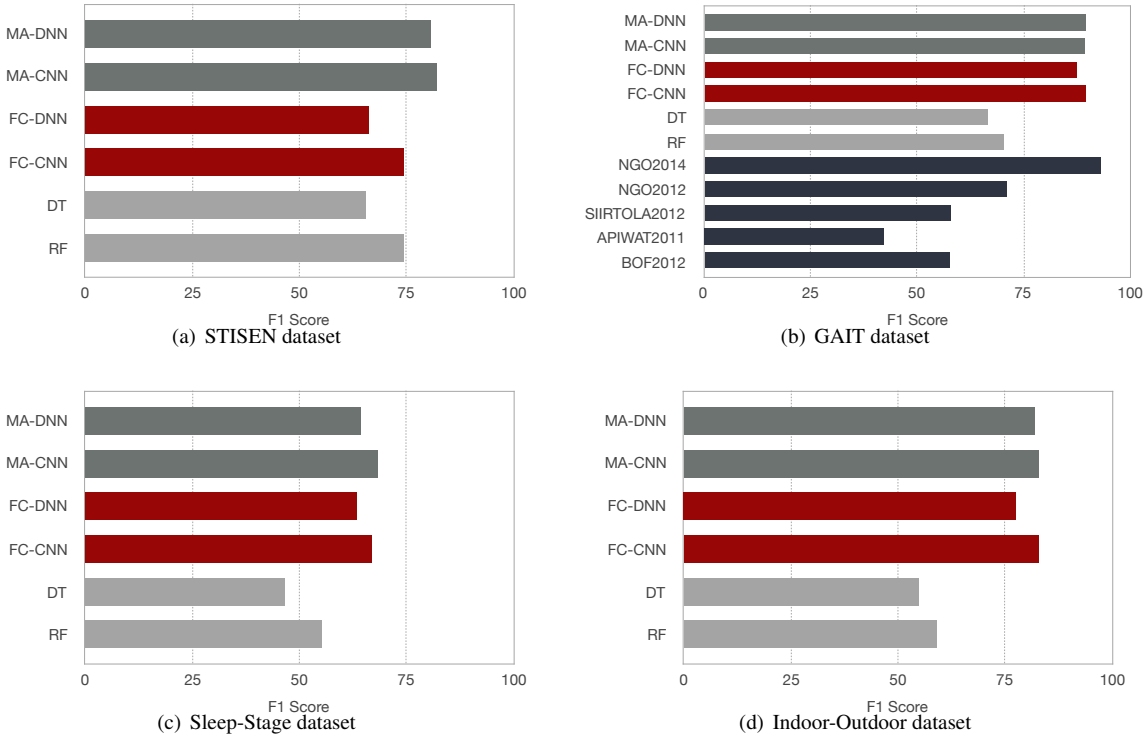


Fig. 4. F1 score achieved by different classifiers, (MA- and FC- neural networks based methods, and shallow or purpose-built methods) on the four datasets. This evaluation is performed with input samples in time domain for deep learning classifiers and features extracted specifically for each dataset to use in shallow classifiers. Despite the effort with shallow classifiers to extract the most relevant features for each domain task (based on previous studies for each dataset), deep learning classifiers still perform better without such requirements.

4.3 Activity recognition with large device diversity: STISEN dataset

The first experiment is conducted on the Stisen dataset, which features data from only 9 users but with large device diversity. To obtain reliable user-independent results, we perform training with the leave-one-out policy, so that data from each subject is used in turn once as test data, while data from all other eight subjects is contained in a training set. In the preprocessing phase, sample rate of sensor data collected with Android devices is normalized to 50Hz, and segmented in time windows of 2 seconds with overlapping of 50%. This is required to guarantee that inputs to neural networks are always the same size and capture a constant time window.

Results for this experiment averaging over 9 subjects are presented in Figure 4(a), which shows that MA deep classifiers achieve the best accuracies, with F1-score of 82%, while FC deep classifier and shallow classifiers both achieving just about 70%-75%. FC deep classifiers are trained with the same hyper-parameters determined for MA deep classifiers.

A more in-depth perspective is provided in Figure 5, where the accuracy of each deep classifier is presented per subject. This shows that MA deep classifiers are able to maintain an accuracy above their FC counterparts. It is clear

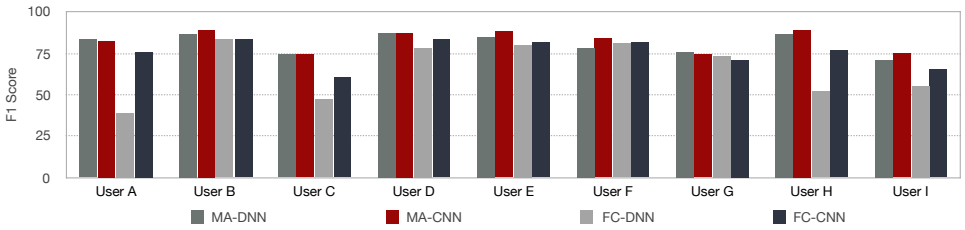


Fig. 5. Per-user comparison of the four deep learning methods MA-DNN, MA-CNN, FC-DNN, and FC-CNN on the activity recognition task using the Stisen dataset.

that FC deep classifiers perform suboptimally for some users, indicating a bad generalization due to not identifying relevant intra-modality feature representations, while the MA is uniformly better across all users.

Though deep learning methods achieve better performance than shallow classifiers, the complexity of this dataset has an impact on the accuracy magnitude, with values below 90%. To improve this, we found that incremental training with the MA-DNN architecture, where less than 5 minutes of labeled data from the same device (customizing the model to one individual-device pair) is enough to boost performance by about 18%, well into the desirable region.

Feature representation learning. Deep classifiers taking raw data (time domain) as input are found to produce better results than shallow classifiers, which must rely on adequate features selection. We consider the following alternative forms of preprocessing transformations on our input data:

- Fast Fourier Transformation (FFT), input data in frequency domain;
- Empirical Cumulative Distribution Function (ECDF) with chosen sample points.

These transformations are akin to features extraction as their role is to filter the raw data stream into a different, more compact representation.

In Figure 6 we observe that, when compared to raw data inputs, applying these transformations in the data preprocessing stage lowers the accuracy of MA deep classifiers (also trained with transformed samples), by almost 5%. This decrease in accuracy of MA classifiers after applying feature extraction (data transformations) suggests that MA deep classifiers are more capable to perform inferences directly from raw data. Access to raw data allows these classifiers to extract their own unfiltered representation of strong features in sensor signals.

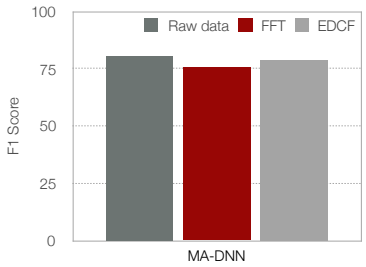


Fig. 6. Comparison of accuracies achieved by classifiers without features extraction (no data transformation) and with features extraction (data transformation with FFT and ECDF) on the STISEN dataset.

4.4 Activity recognition with large number of participants: GAIT dataset

This recognition experiment is performed on the GAIT dataset, which joins data from a very large number of participants (460), with a broad demographic distribution. We split this dataset into two sections, used for training and test, following the same distribution as presented in [53].

In this experiment, we compare deep classifiers with purpose-built solutions and shallow classifiers. As before, we evaluate four deep classifiers: FC-DNN, FC-CNN, MA-DNN and MA-CNN. For benchmarking these, we use the same shallow classifiers (DT and RF) and additionally some earlier purpose-built solutions engineered for this task and dataset alone as presented in [53]: NGO2014, NGO2012, SIIRTOLA2012, APIWAT2011, and BOF2012.

Figure 4(b) presents the performance of above mentioned models, showing that both representational learning methods (FC and MA) and the purpose-built shallow classifier, NGO2014, produce very good results. The highest accuracy was achieved by NGO2014 at 93.2%, followed by MA-DNN and FC-CNN at 89.7%. It is worth noting here that this gap in accuracy between the highly-engineered purpose-built detector, NGO2014, and the general purpose deep classifiers is surprisingly narrow. This indicates that general representational learning methods can closely match the performance of purpose-built methods, potentially limited only by data surplus and training effort.

Figure 7 presents a per action class comparison of performance across the best-considered models. It is easy to observe the consistently good performance of deep classifiers across classes. Further, Figure 8 shows a per user cumulative distribution of the F1 scores achieved by FC and MA deep classifiers. This distribution shows that even on a per-user level, MA deep classifiers outperform FC deep classifiers, with above 90% accuracy in more than 50% of the cases.

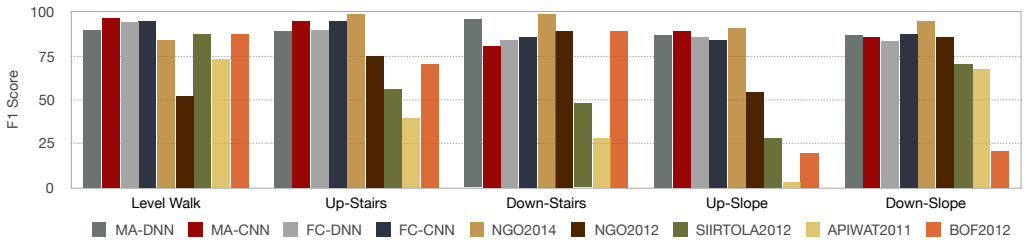


Fig. 7. Per-action class comparison between the performance of all deep learning classifiers and purpose-built solutions on the GAIT dataset.

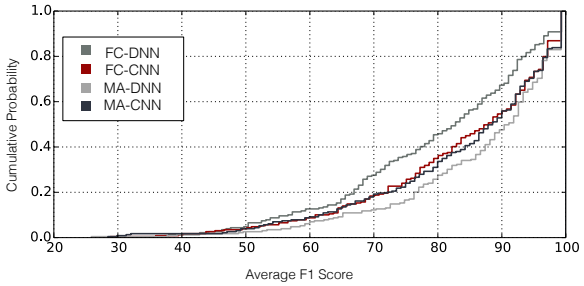


Fig. 8. Cumulative distribution of per-user F1 score of deep classifiers on the GAIT dataset.

Feature representation learning. Here we compare different preprocessing transformations for this dataset, similar to those presented for the Stisen dataset. As shown in Figure 9, data transformations significantly lower the accuracy of all deep classifiers (by 11% on average) compared to inference on raw data. This decrease in accuracy after applying feature extraction confirms earlier observations made on the Stisen dataset, suggesting that deep classifiers are capable of performing inference directly on raw data much better than when these are interpreted by various transformations.

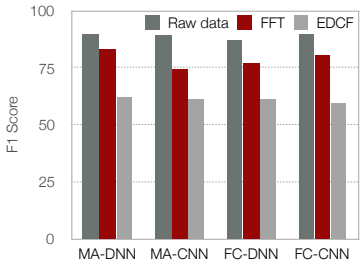


Fig. 9. Comparison of accuracies achieved by deep classifiers without feature extraction and with features extraction (data transformation with FFT and EDCF) on the GAIT dataset.

4.5 Sleep Stage Detection

This experiment considers other sensing modalities, different from the accelerometer and gyroscope available in the previous two datasets, detecting sleep stage from physiological data signals.

The SS dataset has 4 modalities with a sampling frequency of 100Hz. In the preprocessing stage, the sampling frequency is down-sampled to 10Hz to reduce the size of input to our neural networks (and thus the computational costs), while also capturing a large enough time window of 10 seconds for each classification instance. We find that sleep stages 1 and 3 are highly similar, so we group them as a single class. Features for shallow classifiers were extracted using the ECDF method presented before.

From the results presented in Figure 4(c), we can see that feature representational learning methods achieve much better accuracies than shallow methods. On average, we find that deep classifiers enable a 29% accuracy improvement over shallow methods. When looking at a break down of F1-scores per class (Table 3), we find that this improvement is mainly attributed to an almost doubling of performance on classes 'Sleep Stage 4' and 'REM', which shallow classifiers find particularly difficult to detect. MA deep classifiers are slightly more accurate than FC by about 2%, while shallow classifiers are clearly suboptimal for this task. One possible explanation is the simplicity of chosen features to train shallow classifiers, though it also highlights the difficulty of identifying relevant features in new and unexplored detection tasks.

Feature representation learning. Figure 10 presents a comparison between time domain signals input (raw data) and the same two signal transformations as before, FFT and ECDF provided as input to the four deep classifiers. While the performance of deep classifiers on the raw data is nearly 70%, operating on signal transformations reduces the accuracy to about 12% on average.

	MA-DNN	MA-CNN	FC-DNN	FC-CNN	RF	J48 (DT)
Wake	66.13	**70.94**	65.06	69.28	51.28	39.74
Stage 1 & 3	27.59	**30.03**	25.07	29.50	33.15	33.63
Stage 2	77.08	**80.18**	76.19	79.65	70.62	60.43
Stage 4	68.40	**79.59**	68.40	74.20	44.53	31.28
REM	72.48	**75.91**	70.86	73.94	47.33	37.09
Weighted average	64.50	**68.22**	63.19	67.01	55.04	46.59

Table 3. F1 scores for all classifiers in each sleep stage detection on the Sleep-Stage dataset.

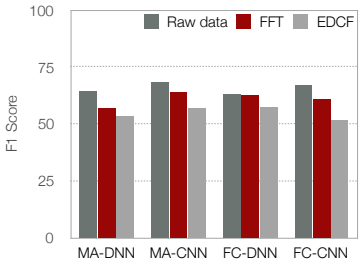


Fig. 10. Comparison of accuracies achieved by classifiers without features extraction and with features extraction (data transformation FFT and ECDF) on the Sleep-Stage dataset.

4.6 Indoor-Outdoor Detection

The IO dataset has its own unique characteristics which make this an interesting exploration – 7 independent and different sensing modalities. It is a binary classification task, though highly diverse across the three environments where data was collected from.

'Indoor' and 'outdoor' areas are sampled from three environments: Campus, City Centre, Residential area, which we labeled as 'env1', 'env2' and 'env3' respectively. We perform the training with the leave-one-out (environment) method, so that each environment takes turn in being the test dataset while the other two assist in training the classifier. Table 4 presents results of each classifier over the three iterations.

Figure 4(d) again confirms that feature representational learning methods are successful, as they achieve much better accuracy than shallow methods (by 43% on average). This observation still holds on a per-environment level, as shown in Table 4. It can be observed that deep classifiers perform consistently better across all environments. Another observation is that all classifiers perform significantly poorer when tested with 'env2', suggesting that it is the hardest environment to detect; this is also true taking a geographical perspective of the areas where data was collected from, with the university Campus and the Residential environments being in closer geographical proximity and farther away from the City Centre. We also see that within shallow classifiers, Random Forest perform better than Decision Tree by 7%.

Feature representational learning. In Figure 11 and as previously observed, classification on raw data with deep classifiers achieves the best performance, here about 81%. After preprocessing the input data with previously mentioned transformations (FFT and ECDF), the accuracy of classifiers drops by about 13% on average.

Training set	Test set	MA-DNN	MA-CNN	FC-DNN	FC-CNN	DT	RF
env2 + env3	env1	87.75	**87.87**	84.5	**87.87**	77.05	68.45
env1 + env3	env2	65.44	**67.38**	57.62	**67.38**	26.6	38.7
env1 + env2	env3	92.64	**92.97**	90.19	**92.97**	60.95	69.63

Table 4. F1 scores for cross-environment evaluation on the Indoor-Outdoor dataset.

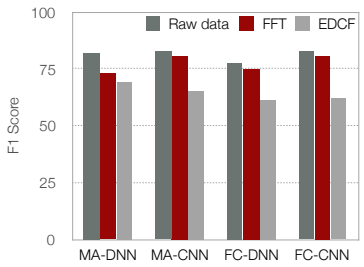


Fig. 11. Comparison of accuracies achieved by deep classifiers without feature extraction and with features extraction (data transformation with FFT and ECDF) on the Indoor-Outdoor dataset.

5 MOBILE HARDWARE FEASIBILITY

Deep architectures exert significant resource challenges on embedded platforms, mainly due to their high demands of memory, computations and energy. In the following, we present runtime experiments on two embedded platforms, namely Snapdragon 400 and Snapdragon 800 SoCs, see Figure 12. Feasibility experiments presented in this section are focused mainly on system resource-usage of the deep architectures. Deployment experiments are performed with an efficient hand-tuned implementation of all deep models presented earlier in this work.

5.1 Target Mobile Hardware

Qualcomm Snapdragon 400. While targeting wearable devices, this Qualcomm processor offers similar performance to many smartphones. It is found within a range of smartwatches, such as the LG G Watch R [42] and includes a quad-core 1.4 GHz CPU and 1 GB of RAM. Additional GPU and DSP processors are also available, but due to a lack of driver support we are forced to use the CPU only for experiments.

Qualcomm Snapdragon 800. As the second embedded platform we use the Snapdragon 800 SoC and run all the deployment experiments on this platform and measure energy consumption and overall runtimes. As in the case with Snapdragon 400, we only use the CPU on Snapdragon 800 to execute the deep models.

5.2 Model Runtime Implementation

To assess the resources demanded by the various multimodal DNNs validated in the prior section, a shared runtime is implemented that executes the inference stage (only) of each model. This prototype is realized through a mix of modules from the Torch [67] ported individually to each processor, that is supported by a set of custom C/C++ components implemented by the authors. Although Torch introduces a degree of overhead, as it acts as an interpreter for the high-level Lua language (in which we encode all models and their parameters), it also offers a number of low-level, highly optimized mathematical operation APIs, which are useful for deep model executions. That said,

(a) Snapdragon 400

(b) Snapdragon 800

Fig. 12. The development board for profiled wearable-class hardware; processors and measured energy profiles of the boards are identical to that of processors found in commercial wearables. For example, the Snapdragon 400 (a) is found within smartwatches, such as the LG G Watch R Smartwatch and (b) Snapdragon 800 is found within Samsung Galaxy 9005 and Nokia Lumia 1520.

certain native Torch operations are replaced with C/C++ extensions to exploit processor-specific opportunities for execution speedup, and better memory management. Furthermore, additional components (e.g., FFT library) are used for any conventional feature extraction as needed by the model specification.

Overall, this implementation can be characterized as adopting best practices understood by those who regularly hand-optimize deep learning models – either for scalability, or in this case operation in resource-limited environments. More obvious examples of incorporated optimizations include keeping of minimal model architectures needed by the inference stage, and the profiling of the data flow of runtime execution to understand memory and cache bottlenecks; or even changing the power profile of processor components to improve the trade-off between execution times and energy use.

5.3 System Resource Usage Experiments

For each of the four activity datasets evaluated in the prior section, we examine runtime resource demands, e.g., memory, computation and energy, of all deep models studied in this paper. Performance comparisons are drawn across all models on two embedded platforms.

The memory requirements of different types of deep model architectures trained on individual data sets are summarized in Table 5. While storing individual model parameters we use 32-bit precision. The largest model (33.1 MB) was found to be the FC-DNN, trained on the STISEN dataset and the smallest model (0.2 MB) was the MA-DNN model trained on the Indoor-Outdoor dataset.

Table 6 and 7 respectively illustrates the average running time (in milli-seconds) and energy consumption (in mJ) of the deep models observed on the Snapdragon 400 platform. We repeated each inference 1000 times and took the average time and energy to mitigate the effect of inherent variations in OS executions. Similarly in Table 8 and 9 we present the runtime results as observed on the Snapdragon 800 platform, which is heavily impacted by the Android scheduling system, resulting in poorer performance than Linux based Snapdragon 400. Despite this, results indicate that the deep models can be executed efficiently on embedded platforms.

Lastly, in Figure 13, we present a variation of CPU load and memory requirement observed on the Snapdragon 400 platform, while executing a MA-CNN model trained on the Gait dataset. The MA-CNN model begins by executing two convolutional layers in parallel and keeping the CPU load almost 100%. The convolution layers require small number of parameters and this keep the overall memory demand low. Once the convolution operations are completed, the CPU load drops to a lower value as the OS becomes occupied in loading the parameters from memory, making memory demand rise. Once all the parameters are loaded in the memory the CPU load becomes rises again to obtain the final inference result.

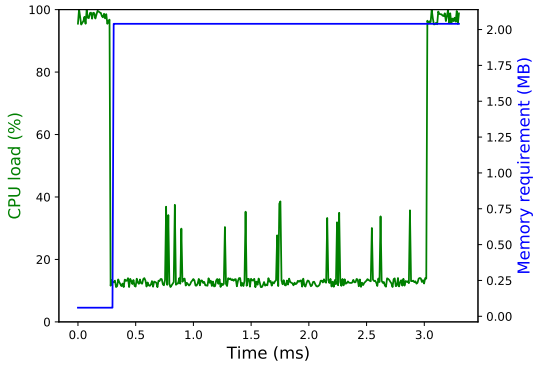


Fig. 13. CPU load and memory requirement against time.

	STISEN (MB)	GAIT (MB)	Sleep-Stage (MB)	Indoor-Outdoor (MB)
MA-DNN	22.1	12.5	1.9	0.2
MA-CNN	8.4	18.6	3.7	0.4
FC-DNN	33.1	2.1	0.3	0.6
FC-CNN	8.4	6.0	0.3	0.6

Table 5. Trained Model sizes (in MBytes) across all data sets under 32-bit precision.

	STISEN (ms)	GAIT (ms)	Sleep-Stage (ms)	Indoor-Outdoor (ms)
MA-DNN	2.8	1.9	1.5	1.7
MA-CNN	9.7	3.3	2.5	0.9
FC-DNN	3.5	2.2	0.8	0.5
FC-CNN	6.5	7.7	1.5	0.7

Table 6. Average model execution time (milli-seconds) observed on Snapdragon 400.

	STISEN (mJ)	GAIT (mJ)	Sleep-Stage (mJ)	Indoor-Outdoor (mJ)
MA-DNN	5.4	3.7	2.9	3.3
MA-CNN	18.7	6.4	4.8	1.7
FC-DNN	6.7	4.2	1.5	1.0
FC-CNN	12.5	14.8	2.9	1.3

Table 7. Average energy consumption (milli-Joule) due to individual deep models observed on Snapdragon 400.

	STISEN (ms)	GAIT (ms)	Sleep-Stage (ms)	Indoor-Outdoor (ms)
MA-DNN	38.8	26.3	20.8	23.6
MA-CNN	134.4	45.7	34.7	12.5
FC-DNN	48.5	30.5	11.0	6.9
FC-CNN	90.1	106.7	20.8	9.7

Table 8. Average model execution time (milli-seconds) observed on Snapdragon 800.

	STISEN (mJ)	GAIT (mJ)	Sleep-Stage (mJ)	Indoor-Outdoor (mJ)
MA-DNN	63.6	43.2	34.1	38.6
MA-CNN	220.5	75.0	56.8	20.5
FC-DNN	79.6	50.0	18.2	11.4
FC-CNN	90.1	106.7	34.1	15.9

Table 9. Average energy consumption (milli-Joule) due to individual deep models observed on Snapdragon 800.

6 DISCUSSION AND LIMITATIONS

This section discusses the practicality of using deep learning for general ubiquitous computing tasks, observed from our experience, describing the difficulties encountered and choices made in training deep neural networks; continuing the discussion with current limitations and the opportunity for future work.

Sensor Signal Preprocessing. From the experiments reported in Section 4, it is clear that operating directly on raw data (i.e., time domain), instead of preprocessing data with FFT or ECDF is not only more computationally efficient (avoiding data transformations) but also effective in training a more accurate model. Choosing an optimal time window size is also important and very specific to the detection task. When activity classes are very similar, a larger time window can help to capture more information – the obvious downside is that a larger input increases computation cost for the neural network. We find that it is a good practice to start with small time windows (such as 30 ms) and gradually expand to capture more information.

The Training Process. A good strategy in training is to start with a small network (such as two layers with a small number of neurons per layer) and gradually expand the size and complexity of the network driven by observations on a separate set of instances (validation set).

When calibrating the network, we consider the key factors: bias, variance, training data distribution. High bias occurs when the model is performing poorly on both training and validation datasets; this is usually addressed by increasing the size of the model or that of the training set. High variance occurs when the classifier overfits training data with good performance on training set but poor on validation set, in which case reducing the network size or introducing regularization (discussed below) can help. In situations where datasets are too small, alternative solutions have proven successful in eliminating this limitation by adopting Active Learning in Bayesian DNNs [16].

In the following we detail some assumptions and decisions made in this exploration:

Learning Rate. In the traditional stochastic gradient descent, this indicates the weights update strength in back-propagation. Careful selection of its value is important: a large value may never converge, while a value too small may take infinitely long to converge. A typical value is 1e-3, though no two training sets are the same, so variations need to be considered to achieve a good convergence. In our exploration, values between 1e-1 and 1e-5 were the

most effective. Learning rate also works in pair with momentum which encourages updates when gradients are consistently in the same direction. Adam algorithm for learning rate policy was determined to be very efficient in many cases.

Number of epochs. The training process executes forward and backward propagations over the entire dataset in one epoch. It is typical for networks to converge very slowly on a complex dataset, affected by noise in data or high similarity in classes. In this situation, a higher number of epochs is required to converge, lengthening the training time. With too many epochs the network may just produce insignificant updates, which should be detected by an early stop policy when the network has converged. In our implementation, we restricted the number of epochs to a maximum of 400 with an adaptive learning rate policy.

Network initialization. The simplest approach to initialize the weights between neurons and biases is with random small values chosen from a mean zero and one variance distribution. Auto-encoders are another solution in initializing the network before supervised learning. From unlabelled data instances an auto-encoders can extract features by reproducing the input to the output on sections of the network. However, in our case datasets are already labeled and networks have only a small number of layers so the impact of auto-encoders is minimal.

Dropout ratio. The dropout layer is a common method for regularization due to its simplicity and efficiency. This implies randomly dropping connections between neurons during training to avoid reliance on single paths through the network (dominant neurons). Its disadvantage comes from extending training time due to more combinations of network connections needing to be reinforced for the network to learn effectively. Training time is affected by dropout factor and width of the connecting layers.

Batch Normalization. This is another very efficient solution for regularization. With the presence of a Batch Normalization layer, weights are normalized again after each update, which allows for a larger value for learning rate to be used – faster training time.

We make our implementation code available as a training framework dedicated to multimodal sensing data for other researchers to use in their work [18].

Additional Deep Learning Methods. Advances in deep learning continue to proliferate and produce a growing range of potential avenues with the potential to improving modeling of wearable and mobile sensor data. It is important to note that in this investigation we concentrated only on performing inferences on static frames, discarding their temporal connection with other frames. This can be seen as one shot classification, not requiring to track sensor signals for a long period of time, ideal for applications requiring occasional sensing. However, other solutions like Recurrent Neural Networks (RNNs) [20] and Long Short-Term Memory (LSTM) can take advantage of this time correlation to improve performance even further. We leave this as open opportunity for further research.

7 RELATED WORK

Applications of Multimodal Learning. Multimodal learning has a vast application domain. Applications have been seen in audio-visual speech recognition [52], image captioning [63], machine translation [34], sentiment analysis [55] and affect recognition [30]. In the space of ubiquitous computing, example applications include human activity recognition [1], sleep detection [12] and emotion recognition [36]. Many recognition tasks were previously only primarily performed with unimodal learning, with the availability of low-energy sensors, many such tasks are recently explored using multimodal learning. For example, authentication models involve both gaze and touch recognition [32], or eating recognition might involve motion of head, wrist and audio [47].

Multimodal Sensor Fusion. Conceptually, classification models based on multimodal sensor data have a clear relationship to techniques of sensor fusion. Within sensor networks, and more broadly in fields such as robotics,

fusion methods routinely leverage different sensor types for purposes like localization [6, 10, 54, 61]. However, the fusion techniques developed are difficult to directly apply to the design of learning algorithms and the feature representations they operate on. More direct insights and methods are found in the design of classifiers of related domains, such as: computer vision, scene understanding [21] and affective computing [30] – as well as numerous examples of existing models of context and activity recognition. In short: the use of, and benefiting from, the input of multiple sensor types is nothing new. But understanding how to combine significantly different sensor inputs is still an open problem. Although multimodal models are routinely seen, the tools we have to construct them (ranging from ensembles of separate classifiers to co-training or simply collapsing data types into single feature vectors) each have their shortcomings; and even simple questions (such as, at what stage should data types be merged?) must still be addressed on a case by case basis (e.g., [62]).

Multimodal Deep Learning. A prime example in the general space of multimodal deep learning is audio-visual speech recognition [50], where much work has been done using neural networks [52]. A number of neural networks have been proposed to perform multimodal deep learning, including CNN [51], RBM [52] and RNN [46]. The choice of neural network often depends on the type of recognition involved, as there is currently no consensus on which network would work best. For instance, in tasks where sequential data is involved (e.g. image sentence description [46]), multimodal versions of recurrent neural networks have been frequently proposed to handle these tasks. While there is work comparing a small number of multimodal learning methods, such as [8] which compares decision tree classifiers with back propagation neural networks, we note that there has not been a comprehensive case study comparing a greater number of deep and shallow multimodal learning architectures. Finally, we wish to highlight an early version of this study was presented in poster form [57] – though the work presented here is of course a significant extension.

Deep Learning in Ubiquitous Computing. Only recently has the exploration into deep learning methods for mobile sensing scenarios begun (e.g., [24, 39]). But with the diversity of exploration rapidly expanding [3, 7, 16, 17, 25, 31, 40, 49, 71]. To the best of our knowledge, the work presented here is the first time that the detection of indoor/outdoor context and transportation mode has been attempted with any form of deep learning, even for single sensor modalities. There is still much to be understood in how such models should be architected, and which variety of algorithms will be most effective – our work adds to this knowledge, that is still in a nascent stage. Closely related models to those we propose in this work are found in [52]. However, we use simpler *Restricted* Boltzmann Machines rather than the *deep* version described in [52] (although, both are still forms of deep learning). Similarly, [33, 64] concern themselves with multimodal models but focus tightly on learning features. None of these papers consider mobile sensor data types nor the classification objectives we study here. Furthermore, few consider a mobile platform as the operating environment of their models. In fact, little multimodal study of this aspect of our work exists, although broad understanding of resource-limited deep learning is accelerating [4, 11, 26, 29, 37, 38, 59, 68] and we expect many existing results to extend to multimodal formulations, though this still remains to be verified.

8 CONCLUSION

In this paper, we perform a systematic study of multimodal deep learning architectures to assess how and when these new techniques satisfy the exigencies of activity and context inferences with mobile devices. We present experiments with four distinct variants of deep neural networks across very diverse and difficult context detection datasets, while comparing their performance with common shallow classifiers and hand-crafted task-specific detectors. Two of these variants are state-of-the-art in deep learning architectures for performing modalities fusion used in other scenarios (video, voice, text) – here, referring to as MA-DNN and MA-CNN – and are for the first time used with wearables and mobile sensing devices for activity recognition and context detection. Experiments that

span a wide range of sensor types, competing multimodal learning algorithms, and activity and context detection tasks, collectively show our proposed general-purpose deep approach to multimodal sensor fusion modeling is both broadly applicable and is able to exceed the performance of previous general solutions and even match task-specific sensor-tuned solutions. This innovation in sensor data modeling is complemented with a practical proof-of-concept implementation designed to measure the overhead of these techniques on two state-of-the-art mobile/wearable processors. Results show that devices that adopt the deep modeling approach, emphasized here, are able to maintain sustainable norms of size, weight and lifetime despite the increased complexity of deep learning methods.

ACKNOWLEDGMENTS

This project received funding from the European Commission's Horizon 2020 research and innovation programme under grant agreement No 687698, through a HiPEAC Collaboration Grant. We thank all the anonymous reviewers for their constructive comments, which helped us to improve the quality of this work.

REFERENCES

[1] Michael Barz, Mohammad Mehdi Moniri, Markus Weber, and Daniel Sonntag. 2016. Multimodal Multisensor Activity Annotation Tool. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct (UbiComp '16)*. ACM, New York, NY, USA, 17–20. https://doi.org/10.1145/2968219.2971459

[2] Yoshua Bengio, Ian J. Goodfellow, and Aaron Courville. 2015. Deep Learning. (2015). http://www.iro.umontreal.ca/~bengioy/dlbook Book in preparation for MIT Press.

[3] S. Bhattacharya and Nicholas D. Lane. 2016. From smart to deep: Robust activity recognition on smartwatches using deep learning. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. 1–6. https://doi.org/10.1109/PERCOMW.2016.7457169

[4] Sourav Bhattacharya and Nicholas D. Lane. 2016. Sparsification and separation of deep learning layers for constrained resource inference on wearables. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 176–189.

[5] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[6] Tatiana Bokareva, Wen Hu, Salil Kanhere, Branko Ristic, Neil Gordon, Travis Bessell, Mark Rutten, and Sanjay Jha. 2006. Wireless sensor networks for battlefield surveillance. In *Proceedings of the land warfare conference*. 1–8.

[7] Heike Brock, Yuji Ohgi, and James Lee. 2017. Learning to judge like a human: convolutional networks for classification of ski jumping errors. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM, 106–113.

[8] Donald E. Brown, Vincent Corruble, and Clarence Louis Pittard. 1993. A comparison of decision tree classifiers with backpropagation neural networks for multimodal classification problems. *Pattern Recognition* 26, 6 (1993), 953 – 961. https://doi.org/10.1016/0031-3203(93)90060-A

[9] Andreas Bulling, Jamie A. Ward, and Hans Gellersen. 2012. Multimodal recognition of reading activity in transit using body-worn sensors. *TAP* 9, 1 (2012), 2. https://doi.org/10.1145/2134203.2134205

[10] Jose A Castellanos and Juan D Tardos. 2000. *Mobile robot localization and map building: A multisensor fusion approach*. Kluwer academic publishers.

[11] Guoguo Chen, Carolina Parada, and Georg Heigold. 2014. Small-footprint Keyword Spotting Using Deep Neural Networks. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'14)*.

[12] W. Chen, A. Sano, D. L. Martinez, S. Taylor, A. W. McHill, A. J. K. Phillips, L. Barger, E. B. Klerman, and R. W. Picard. 2017. Multimodal ambulatory sleep detection. In *2017 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*. 465–468. https://doi.org/10.1109/BHI.2017.7897306

[13] Tanzeem Choudhury, Gaetano Borriello, Sunny Consolvo, Dirk Haehnel, Beverly Harrison, Bruce Hemingway, Jeffrey Hightower, Predrag "Pedja" Klasnja, Karl Koscher, Anthony LaMarca, James A. Landay, Louis LeGrand, Jonathan Lester, Ali Rahimi, Adam Rea, and Danny Wyatt. 2008. The Mobile Sensing Platform: An Embedded Activity Recognition System. *IEEE Pervasive Computing* 7, 2 (April 2008), 32–41. https://doi.org/10.1109/MPRV.2008.39

[14] Li Deng and Dong Yu. 2014. *DEEP LEARNING: Methods and Applications*. Technical Report MSR-TR-2014-21. http://research.microsoft.com/apps/pubs/default.aspx?id=209355

[15] Samira Ebrahimi Kahou, Xavier Bouthillier, Pascal Lamblin, Çağlar Gülçehre, Vincent Michalski, Kishore Reddy Konda, Sébastien Jean, Pierre Froumenty, Yann Dauphin, Nicolas Boulanger-Lewandowski, Raul Chandias Ferrari, Mehdi Mirza, David Warde-Farley, Aaron

Courville, Pascal Vincent, Roland Memisevic, Christopher Pal, and Yoshua Bengio. 2015. EmoNets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces* (2015), 1–13. https://doi.org/10.1007/s12193-015-0195-2

[16] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian Active Learning with Image Data. *CoRR* abs/1703.02910 (2017). http://arxiv.org/abs/1703.02910

[17] Petko Georgiev, Sourav Bhattacharya, Nicholas D. Lane, and Cecilia Mascolo. 2017. Low-resource Multi-task Audio Sensing for Mobile and Embedded Devices via Shared Deep Neural Network Representations. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 50 (Sept. 2017), 19 pages. https://doi.org/10.1145/3131895

[18] Github repository 2017. Multimodal Deep Learning Framework. https://github.com/vradu10/deepfusion.git. (2017).

[19] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101, 23 (2000), e215–e220. Circulation Electronic Pages: http://circ.ahajournals.org/cgi/content/full/101/23/e215 PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

[20] Alex Graves, A-R Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 6645–6649.

[21] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. 2010. Multimodal semi-supervised learning for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 902–909.

[22] Haodong Guo, Ling Chen, Liangying Peng, and Gencai Chen. 2016. Wearable sensor based multimodal human activity recognition exploiting the diversity of classifier ensemble. In *Proceedings of UbiComp*. ACM.

[23] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.* 11, 1 (Nov. 2009), 10–18. https://doi.org/10.1145/1656274.1656278

[24] Nils Hammerla, James Fisher, Peter Andras, Lynn Rochester, Richard Walker, and Thomas Plötz. 2015. PD Disease State Assessment in Naturalistic Environments using Deep Learning. In *AAAI 2015*.

[25] Nils Hammerla, Shane Halloran, and Thomas Ploetz. 2016. Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables. In *Proceedings of IJCAI*. ACM.

[26] Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* (2015).

[27] Awni Y. Hannun, Carl Case, Jared Casper, Bryan C. Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. Deep Speech: Scaling up end-to-end speech recognition. *CoRR* abs/1412.5567 (2014). http://arxiv.org/abs/1412.5567

[28] Samuli Hemminki, Petteri Nurmi, and Sasu Tarkoma. 2013. Accelerometer-based Transportation Mode Detection on Smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys '13)*. ACM, New York, NY, USA, Article 13, 14 pages. https://doi.org/10.1145/2517351.2517367

[29] Loc N Huynh, Youngki Lee, and Rajesh Krishna Balan. 2017. DeepMon: Mobile GPU-based Deep Learning Framework for Continuous Vision Applications. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 82–95.

[30] Ashish Kapoor and Rosalind W Picard. 2005. Multimodal affect recognition in learning environments. In *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 677–682.

[31] Thomas Kautz, Benjamin H Groh, Julius Hannink, Ulf Jensen, Holger Strubberg, and Bjoern M Eskofier. 2017. Activity recognition in beach volleyball using a Deep Convolutional Neural Network. *Data Mining and Knowledge Discovery* (2017), 1–28.

[32] Mohamed Khamis, Florian Alt, Mariam Hassib, Emanuel von Zezschwitz, Regina Hasholzner, and Andreas Bulling. 2016. GazeTouchPass: Multimodal Authentication Using Gaze and Touch on Mobile Devices. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 2156–2164. https://doi.org/10.1145/2851581.2892314

[33] Yelin Kim, Honglak Lee, and E.M. Provost. 2013. Deep learning for robust feature generation in audiovisual emotion recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. 3687–3691. https://doi.org/10.1109/ICASSP.2013.6638346

[34] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. 2014. Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. *CoRR* abs/1411.2539 (2014). http://arxiv.org/abs/1411.2539

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[36] Saewon Kye, Junhyung Moon, Juneil Lee, Inho Choi, Dongmi Cheon, and Kyoungwoo Lee. 2017. Multimodal Data Collection Framework for Mental Stress Monitoring. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers (UbiComp '17)*. ACM, New York, NY, USA, 822–829. https://doi.org/10.1145/3123024.3125616

[37] Nicholas D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, L. Jiao, L. Qendro, and F. Kawsar. 2016. DeepX: A Software Accelerator for Low-Power Deep Learning Inference on Mobile Devices. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 1–12. https://doi.org/10.1109/IPSN.2016.7460664

[38] Nicholas D. Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, and Fahim Kawsar. 2015. An early resource characterization of deep learning on wearables, smartphones and internet-of-things devices. In *Proceedings of the 2015 International Workshop on Internet of Things towards Applications*. ACM, 7–12.

[39] Nicholas D. Lane and Petko Georgiev. 2015. Can Deep Learning Revolutionize Mobile Sensing?. In *HotMobile 2015*.

[40] Nicholas D. Lane, Petko Georgiev, and Lorena Qendro. 2015. DeepEar: Robust Smartphone Audio Sensing in Unconstrained Acoustic Environments Using Deep Learning. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 283–294. https://doi.org/10.1145/2750858.2804262

[41] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep Learning. *Nature* (2015).

[42] LG G Watch R 2017. LG G Watch R. https://www.qualcomm.com/products/snapdragon/wearables/lg-g-watch-r. (2017).

[43] Wei Liu, Wei-Long Zheng, and Bao-Liang Lu. 2016. Multimodal Emotion Recognition Using Multimodal Deep Learning. *CoRR* abs/1602.08225 (2016). http://arxiv.org/abs/1602.08225

[44] Hong Lu, Jun Yang, Zhigang Liu, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. 2010. The Jigsaw Continuous Sensing Engine for Mobile Phone Applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*. ACM, New York, NY, USA, 71–84. https://doi.org/10.1145/1869983.1869992

[45] Lumo Lift 2017. Lumo Lift. http://www.lumobodytech.com. (2017).

[46] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. 2014. Explain Images with Multimodal Recurrent Neural Networks. *ArXiv e-prints* (Oct. 2014). arXiv:cs.CV/1410.1090

[47] Christopher Merck, Christina Maher, Mark Mirtchouk, Min Zheng, Yuxiao Huang, and Samantha Kleinberg. 2016. Multimodality Sensing for Eating Recognition. In *Proceedings of the 10th EAI International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth '16)*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 130–137. http://dl.acm.org/citation.cfm?id=3021319.3021339

[48] Microsoft Band 2017. Microsoft Band. http://www.microsoft.com/Microsoft-Band/. (2017).

[49] 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.

[50] Y. Mroueh, E. Marcheret, and V. Goel. 2015. Deep multimodal learning for Audio-Visual Speech Recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2130–2134. https://doi.org/10.1109/ICASSP.2015.7178347

[51] Sebastian Münzner, Philip Schmidt, Attila Reiss, Michael Hanselmann, Rainer Stiefelhagen, and Robert Dürichen. 2017. CNN-based Sensor Fusion Techniques for Multimodal Human Activity Recognition. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers (ISWC '17)*. ACM, New York, NY, USA, 158–165. https://doi.org/10.1145/3123021.3123046

[52] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. 2011. Multimodal Deep Learning. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, Lise Getoor and Tobias Scheffer (Eds.). Omnipress, 689–696.

[53] Trung Thanh Ngo, Yasushi Makihara, Hajime Nagahara, Yasuhiro Mukaigawa, and Yasushi Yagi. 2015. Similar gait action recognition using an inertial sensor. *Pattern Recognition* 48, 4 (2015), 1289 – 1301. https://doi.org/10.1016/j.patcog.2014.10.012

[54] Reza Olfati-Saber and Jeff S Shamma. 2005. Consensus filters for sensor networks and distributed sensor fusion. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*. IEEE, 6698–6703.

[55] Soujanya Poria, Erik Cambria, Newton Howard, Guang-Bin Huang, and Amir Hussain. 2016. Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing* 174, Part A (2016), 50 – 59. https://doi.org/10.1016/j.neucom.2015.01.095

[56] Valentin Radu, Panagiota Katsikouli, Rik Sarkar, and Mahesh K. Marina. 2014. A Semi-supervised Learning Approach for Robust Indoor-outdoor Detection with Smartphones. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys '14)*. ACM, New York, NY, USA, 280–294. https://doi.org/10.1145/2668332.2668347

[57] Valentin Radu, Nicholas D. Lane, Sourav Bhattacharya, Cecilia Mascolo, Mahesh K Marina, and Fahim Kawsar. 2016. Towards multimodal deep learning for activity recognition on mobile devices. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM, 185–188.

[58] Valentin Radu and Mahesh K. Marina. 2013. HiMLoc: Indoor Smartphone Localization via Activity Aware Pedestrian Dead Reckoning with Selective Crowdsourced WiFi Fingerprinting. In *In Proc. Indoor Positioning and Indoor Navigation (IPIN)*. IEEE. http://dx.doi.org/10.1109/IPIN.2013.6817916

[59] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*. Springer, 525–542.

[60] Devendra Singh Sachan, Umesh Tekwani, and Amit Sethi. 2013. Sports Video Classification from Multimodal Information Using Deep Neural Networks. In *2013 AAAI Fall Symposium Series*.

[61] Gyula Simon, Miklós Maróti, Ákos Lédeczi, György Balogh, Branislav Kusy, András Nádas, Gábor Pap, János Sallai, and Ken Frampton. 2004. Sensor network-based countersniper system. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 1–12.

[62] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. 2005. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 399–402.

[63] Kihyuk Sohn, Wenling Shang, and Honglak Lee. 2014. Improved Multimodal Deep Learning with Variation of Information. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 2141–2149. http://papers.nips.cc/paper/5279-improved-multimodal-deep-learning-with-variation-of-information

[64] Nitish Srivastava and Ruslan R Salakhutdinov. 2012. Multimodal Learning with Deep Boltzmann Machines. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger (Eds.). Curran Associates, Inc., 2222–2230. http://papers.nips.cc/paper/4683-multimodal-learning-with-deep-boltzmann-machines.pdf

[65] 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 Mitigating Mobile Sensing Heterogeneities for Activity Recognition. In *The 13th ACM Conference on Embedded Networked Sensor Systems*.

[66] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. 2014. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[67] Torch 2017. Torch. http://torch.ch/. (2017).

[68] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. 2014. Deep neural networks for small footprint text-dependent speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*. IEEE, 4052–4056. https://doi.org/10.1109/ICASSP.2014.6854363

[69] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. 2015. On deep multi-view representation learning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. 1083–1092.

[70] Pengcheng Wu, Steven C.H. Hoi, Hao Xia, Peilin Zhao, Dayong Wang, and Chunyan Miao. 2013. Online Multimodal Deep Similarity Learning with Application to Image Retrieval. In *Proceedings of the 21st ACM International Conference on Multimedia (MM '13)*. ACM, New York, NY, USA, 153–162. https://doi.org/10.1145/2502081.2502112

[71] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. 2017. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 351–360.

[72] Piero Zappi, Thomas Stiefmeier, Elisabetta Farella, Daniel Roggen, Luca Benini, and Gerhard Tröster. 2007. Activity Recognition from On-Body Sensors by Classifier Fusion: Sensor Scalability and Robustness. In *3rd Int. Conf. on Intelligent Sensors, Sensor Networks, and Information Processing (ISSNIP)*. 281–286. http://www2.ife.ee.ethz.ch/~droggen/publications/wear/EDAS_ISSNIP.pdf

Received February 2017; revised August 2017; accepted October 2017