A Multi-View Architecture for the SHL Challenge

Massinissa Hamidi
LIPN-UMR CNRS 7030
Univ. Sorbonne Paris Nord
Villetaneuse, France
hamidi@lipn.univ-paris13.fr

Aomar Osmani
LIPN-UMR CNRS 7030
Univ. Sorbonne Paris Nord
Villetaneuse, France
ao@lipn.univ-paris13.fr

Pegah Alizadeh
Pôle Universitaire Léonard de Vinci,
Research Center
La Défense, France
pegah.alizadeh@devinci.fr

ABSTRACT

To recognize locomotion and transportation modes in a user-independent manner with an unknown target phone position, we (team *Eagles*) propose an approach based on two main steps: reduction of the impact of regular effects that stem from each phone position, followed by the recognition of the appropriate activity. The general architecture is composed of three groups of neural networks organized in the following order. The first group allows the recognition of the source, the second group allows the normalization of data to neutralize the impact of the source on the activity learning process, and the last group allows the recognition of the activity itself. We perform extensive experiments and the preliminary results encourage us to follow this direction, including the source learning to reduce the phone position's biases and activity separately.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning by classification**; • **Human-centered computing** → **Ambient intelligence**.

KEYWORDS

SHL challenge; human activity recognition; multi-view learning; neural architecture search

ACM Reference Format:
Massinissa Hamidi, Aomar Osmani, and Pegah Alizadeh. 2020. A Multi-View Architecture for the SHL Challenge. In *Adjunct Proceedings of the 2020 ACM International JointConference on Pervasive and Ubiquitous Computing and Proceedings of the2020 ACM International Symposium on Wearable Computers (UbiComp/ISWC'20 Adjunct), September 12–16, 2020, Virtual Event, Mexico.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3410530.3414351

UbiComp/ISWC '20Adjunct, September 12–16,2020, Virtual Event, Mexico

ACM ISBN 978-1-4503-8076-8/20/09…$15.00
https://doi.org/10.1145/3410530.3414351

1 INTRODCUTION

Activity recognition has been studied actively in the context of ubiquitous computing [2]. The development of wearable devices such as smartphones, smartwatches, fitness trackers, etc. carried by people all day long, allows the accumulation of sensory data for recognizing people's activities such as running, walking, cycling, being in a car, subway, bus and so on. Often, research works around activity recognition assume that the sensory data used to predict a given activity (during deployment) are generated from the same user and wearable devices placed at the same body locations as during the training phase. However, in real-world scenarios, this assumption does not necessarily hold. The main challenges that these approaches face are related to how to develop activity recognition models that are both user- and position-independent. These challenges become even harder to overcome when the target user and the body location which will serve to predict activities are unknown or unavailable *a priori*.

In this regard, the Sussex-Huawei locomotion and transportation (SHL) challenge 2020 [5, 8] has collected a dataset containing 105 days of wearable sensory data generated by smartphones place on four different body parts and collected by three users. The goal of the challenge is to recognize eight locomotion and transportation activities from sensor data of a smartphone in a position-independent manner. More precisely, the goal is to predict the user's activity from the data coming from a smartphone placed on an unknown part of the body, while the provided training data is collected from smartphones on torso, hips, bag and hand positions, and from a different user.

Assuming the locomotion-transportation mode as a concept, in this paper, we consider the phones located in different positions as multiple views of the same concept. We propose to (1) leverage these views entirely in order to learn a joint representation via position-specific convolution-based circuits. We then (2) determine the target position and (3) fine-tune the corresponding circuit so as to increase the circuit's robustness. The rationale behind this approach is that multiple sources (positions in our case) have different levels of informativeness with regard to the concept (locomotion-transportation mode) that we want to learn. Learning a joint representation, as a first step, helps the model compensate for the potential lack of informativeness of some sources,

i.e. the case where only and only a single and unknown phone position is available during the test time. We obtained a 58.29% (avg.) f1-score over all positions on the validation set and, noticeably, a 10% improvement for the Hand position after fine-tuning the baseline model while maintaining recognition performances of other positions (Torso, Hips, and Bag).

The rest of the paper is organized as follows. The SHL dataset is described in Section 2 and Section 3 details our proposed approach. Section 4 presents obtained results followed by a conclusion in Section 5.

2 SHL CHALLENGE DATASET

Three participants 1, 2 and 3 performed full-time data collection in realistic scenarios. The detectors (HUAWEI phones) have four positions on a person body: Hand, Torso, Hips and Bag. The data movements have been annotated in 8 labels: Still, Walking, Run, Bike, Car, Train and Subway. The SHL dataset is organised in recording data for three users according to Table 1.

The objective is to learn the modes of transportation and locomotion in a *user-independent* manner with an *unknown phone position*. The train, validation and test data was generated by segmenting the whole data with a non-overlap sliding window of 5 seconds. Train data contains the raw sensors data from user 1 and four phone locations during 59 days with given activity labels. The 5 seconds frames for the train data are consecutive in time i.e. the recorded data are *not shuffled*. Test user is a combination of users 2 and 3 recorded for 40 days. The phone position which is only one position is not given, and the frames are shuffled, i.e. two successive frames in the file are likely not consecutive in time. And finally the Validation data is a combination of users 2 and 3 with given phone positions and containing 6 recorded days. The frames for the train data are consecutive in time and not shuffled. Sensors modalities (channels) of 4

Table 1: Data situation for Challenge 2020. The test data come from a single and unknown body location.

2020	bag	torso	hip	hand	labels	users	days
Train	✔	✔	✔	✔	✔	1	59
Validation	✔	✔	✔	✔	✔	2, 3	6
Test	?	?	?	?	✗	2, 3	40

synchronised phones are recorded as: Accelerometer (x, y, z), Gyroscope (x, y, z), Magnetometer (x, y, z), Orientation (quaternions in the form of w, x, y, z), Gravity (x, y, z), Linear acceleration (x, y, z), Ambient pressure for all 8 classes and phone positions. In this work, we use two forms of sensor channels, the raw data and the magnitudes for channels with three axes x, y and z which are calculated from the following formula and then normalized as well:

$$m_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$$

3 PROPOSED APPROACH

Our approach is based on three major steps: (1) we construct a *joint representation* via position-specific circuits, (2) we determine the target source (test phone position) using a *source discrimination* model, and (3) we *fine-tune* the corresponding circuit by optimizing the recognition performances. Figure 1 summarizes the proposed approach. In the following, we detail the components of our suggested approach.

Joint Representation

Regarding Figure 1, we construct a joint representation from the different input sources and phone positions. The whole network is composed of 4 different circuits (related to Hand, Hips, Bag, and Torso) where each circuit processes the inputs of a specific source and each individual circuit produces a vector representation for each individual source.

Position-Specific Circuit. Figure 2 illustrates the architecture of each circuit individually related to its phone position. Each circuit neural architecture is constructed by stacking up to 3 Conv1d/ ReLU/ MaxPool/ BatchNorm blocks to processes the input channels related to the phone position [6, 9]. These blocks are followed by a Concatenate layer and a Dense layer to recognize the phone position finally. As an example *view_Hand* predicts a vector embedding for the given inputs of the Hand sensors.

Source Discrimination

The source discrimination model is based on the energy of input signals. We hypothesize that, while performing activities, different positions carry different energies. This is due to the fact that the amplitude of movements varies from one position to another. For example, the amplitude of hand movements is more significant than those of the torso (see Figure 3).

Signal energy. Assume s is a signal modality of a given position with length N. The signal energy is computed as $E = \sum_{i=0}^{N-1} s_i$ where s_i is the ith sample of the s signal. In order to see how the signal energies are different respecting various phone positions, Figure 3 shows the signal energy computed for the first 100 frames of different positions. Note that we compute these statistics for the validation dataset.

Fine Tuning

After determining the source (phone position) of the test data, we fine-tune the corresponding circuit in order to make the

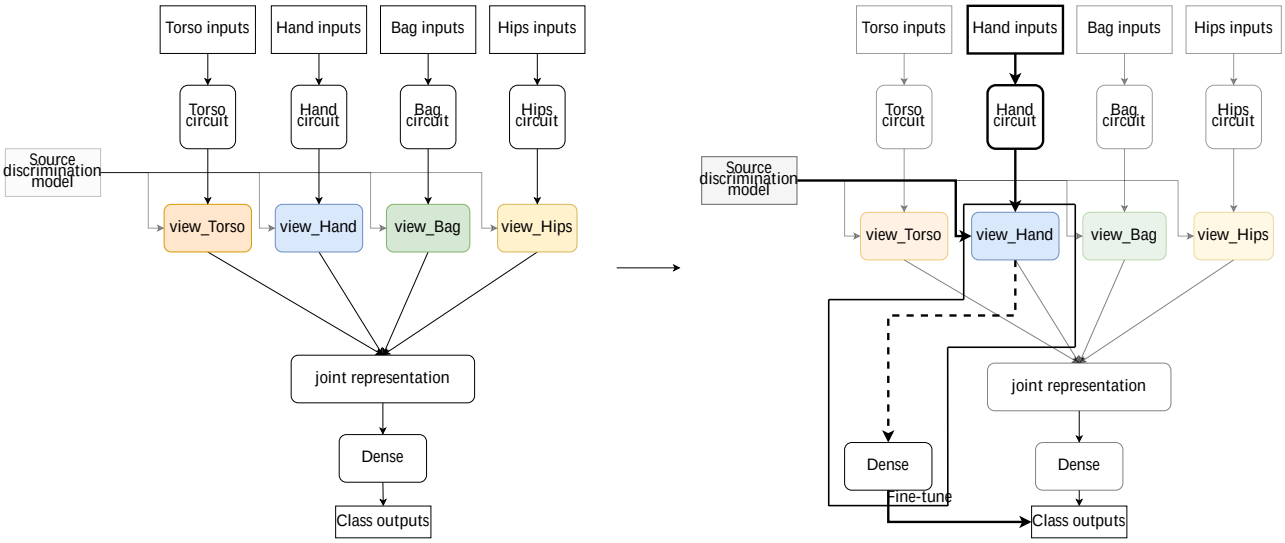


Figure 1: Diagram summarizing our proposed approach. First (left), the whole network is trained to optimize a joint representation and by the same occasion the representation of each individual phone position. Second (right), after determining the test source, the corresponding circuit (highlighted in the figure) is fine-tuned by optimizing its recognition performances.

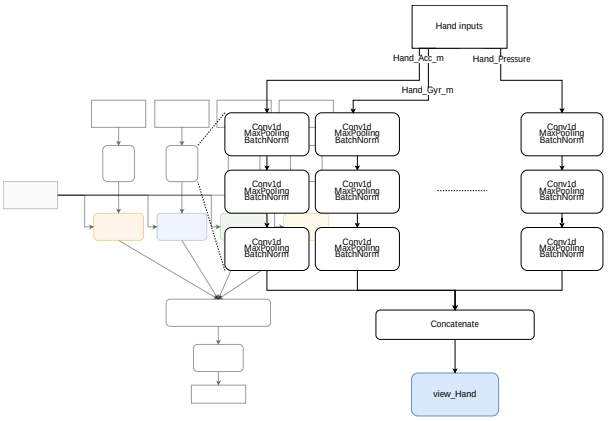


Figure 2: Each phone position architecture is constructed by stacking up to 3 Conv1d/ReLU/MaxPool/BatchNorm blocks for processing each input channel individually. These blocks are followed by a Concatenate layer and a Dense layer (referred to view_Hand for the phone positioned on Hand).

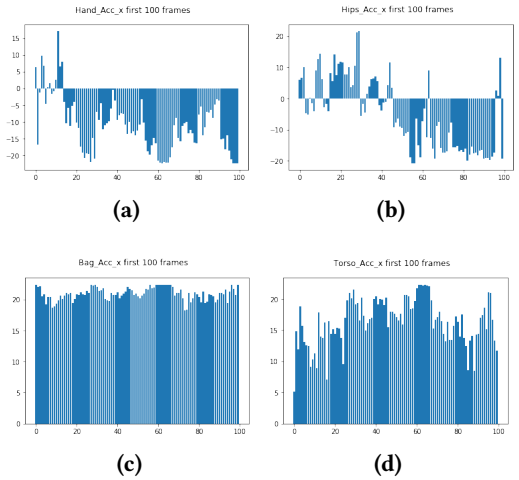


Figure 3: Signal energy of the first 100 frames of Acc_x channel for (a) Hand, (b) Hips, (c) Bag, and (d) Torso.

model more robust w.r.t this particular source. Fine-tuning for a specific position may produce what is referred to as "catastrophic forgetting". This would make our final model unreliable if presented with inputs from another position. In this initial set of experiments, to compensate for the simplicity of the discrimination model, we chose to fine-tune for the hand position and make sure that we do not lose much for the remaining positions. That is, we make sure that the final model performs equally well for all sources and alleviate catastrophic forgetting. To do this, after selecting the circuit (in our case *Hand circuit*), we fine-tune its weights using inputs of each source individually. This results in 4 different models for Hand, Hips, Torso and Bag.

Precisely, for each source, we construct a new network using the *Hand circuit* (selected by the source discrimination model) and up to 3 additional dense layers, which are added on top of the *view_Hand* (See Figure 1 right). The additional dense layers of the new network are first trained while the base network is frozen (set to inference mode). Afterward, the weights of the whole new network are trained to optimize the recognition performances [1].

Additionally, as a baseline, we populate all inputs of the network using a single source in order to assess how fine-tuning influences the recognition performances.

Hyperparameters Tuning

The hyperparameters of the proposed architectures in Figures 1 and 2 are tuned using the Tree-structured Parzen Estimator (TPE) [3] to optimize the performance of the corresponding network for recognizing mobility and transportation modes. In total, 46 different hyperparameters are tuned during this step.

Hyperparameters space. Table 2 summarizes the different hyperparameters that are tuned and their respective domains (values). According to Figure 2 each phone position architecture contains 3 blocks where each block contains a convolutional network. The three block parameters given in Table 2 are the hyperparameters for the three convolutional networks respectively. Notice that these parameters are tuned for various multimodal channels while they are the same for the four phone positions. On the other hand, each phone-position view layer has a set of hyperparameters namely view_Position and according to Figure 1 (left), the joint representation layer has its own set of hyperparameters. And finally, the global learning rates should be tuned for the system as well. The second set of hyperparameters is specific for each phone position and should be tuned according to each source separately.

Search strategy. We use the Tree-structured Parzen Estimator (TPE) in order to explore the hyperparameter space. Similar to the Bayesian optimization, TPE is part of the sequential model-based optimization approaches. These, sequentially, construct models to approximate the performance of hyperparameters using previously explored configurations in order to predict which hyperparameters instantiation to explore next. However, unlike Bayesian optimization, TPE deals naturally with situations where elements of the hyperparameter space are known to be irrelevant given particular values of other elements. In other words, it preserves specified conditional dependence over hyperparameters [3].

[1]Note that this second training phase is performed using a low learning rate so as not to lose what was jointly learned so-far from each individual position. Note also that even if the base model is set to training mode (allowing weights updates), it is kept in inference mode which means that the BatchNorm layers will not update their batch statistics.

Table 2: Summary of the hyperparameters tuned using the TPE. Hyperparameters of the blocks 1, 2, and 3 are tuned per channel and are common for all the same phone positions while those of position-specific views are tuned per each position.

hyperparam.	domain (values)
Block 1	
*channel*_numfilters_0	8, 16, 32, 64
*channel*_kernelsize_0	7, 11, 17
Block 2	
*channel*_numfilters_1	8, 16, 32, 64
*channel*_kernelsize_1	5, 7, 11, 17
Block 3	
*channel*_numfilters_2	8, 16, 32, 64
*channel*_kernelsize_2	2, 3, 5, 7, 11, 17
Position-specific views	
view_*Position*_3	10, 15, 20, 25, 30, 35, 40
Joint representation	
hiddenunits_3	128, 256, 512, 1024, 2048
dropout_3	[0.3, 0.9] (uniform distrib.)
Global	
learning rate	[1e-05, 1e-01] (uniform distrib.)

4 RESULTS

In this section, we present an empirical evaluation of our proposed approach based on the Keras [4] framework with Tensorflow [1] backend. We use the Microsoft-NNI toolkit [2] which provides a comprehensive list of exploration strategies particularly based on hyperparameter tuning.

Hyperparameters Tuning

Figure 4 shows the result of the hyperparameter tuning phase. Each curve in the figure is a different instantiation of the hyperparameters, i.e. the values that each hyperparameter takes at a given time step and the resulting recognition performance accuracy. Each individual network is trained using the raw data inputs generated by all available data sources (phone-positions). Note that training of a given network stops after 7 subsequent epochs without improvement over the median of recognition performances obtained so far. Hyperparameters tuning allows us to substantially improve the recognition performances. Noticeably, we get more than 20% improvement.

[2]https://github.com/microsoft/nni

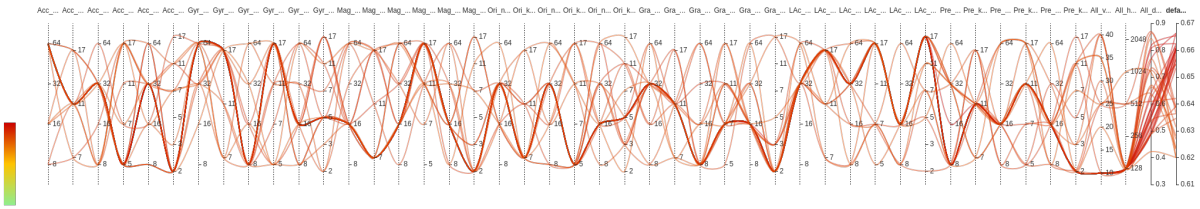
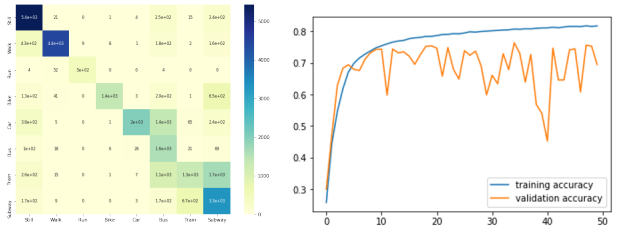


Figure 4: Most influential values of the tuned hyperparameters and their corresponding recognition performances accuracy (Top 20%). The rightmost bar corresponds to the recognition performances where red color indicates the highest performance.

Training Using All 4 Sources

Figure 5 shows recognition performances of our proposed network (see Figure 1) trained on magnitude channels inputs to construct a joint representation from the different inputs on four various phone positions related circuits. This network achieves approximately 81% and 75% accuracy on the training and validation sets respectively. Using computed magnitudes rather than raw inputs improves recognition performances by more than 10%. Although, we notice that the obtained model often confuses Car with Bus and Train with Subway activities. Note that the validation data is performed, similarly to the training phase, using all circuits related to all phone positions.



(a) (b)

Figure 5: Performances of the proposed network trained on magnitude inputs to construct a joint representation from the different phone positions. (a) confusion matrix and (b) evolution of training and validation accuracy over epochs.

Fine-tuning

To assess the behavior of the model without fine-tuning, Figures 6(a)–(d) show confusion matrices obtained by populating the network, at every turn, with inputs from a unique position. Figures 6(e)–(h) and 6(i)–(l) show the confusion matrices of the model fine-tuned using Hand and Hips inputs, respectively, and validated on all 4 positions individually.

We notice that even if the model that is fine-tuned using Hips inputs (Figure 6) substantially improves recognition performances on both Hips and Bag inputs, but Hand inputs are not handled well. In contrast, when we fine-tune the model using Hand inputs, we obtain a model that performs equally-well on each individual position.

Computational Resources

We use the computing power provided by Magi [3]. In particular, we allocated two nodes: a CPU (96 GB RAM and 56 cores@2.20 GHz) and a GPU (4 × Tesla K40M). Training a single model on the whole training set (196072 examples) takes approximately 3 hours (on avg.). This includes training the network on the whole inputs as well as fine-tuning specific circuits. Prediction both on the validation (28789 examples) and test (57573 examples) sets takes approximately 1 minute.

5 CONCLUSION

We presented in this paper our (team *Eagles*) proposed approach as part of the SHL challenge 2020. The goal was to recognize modes of locomotion and transportation in a user-independent manner with an unknown target phone location. We proposed an architecture to leverage the entire perspectives featured by the phone positions in order to learn a joint-representation via position-specific circuits and to separate the position problem from the recognition one. The rationale behind this approach is that various locations have different levels of informativeness with respect to the concept we want to learn. Learning a joint representation helps the model compensate for potential lack of informativeness of some sources. We notably obtained a 62.29% f1-score over Bag position on the validation set after fine-tuning the baseline model using Hand inputs. The recognition result for the testing dataset will be presented in the summary paper of the challenge [7]. Several experiments are done, preliminary results encourage us to follow this direction. Notably, learn firstly the position of the phone to limit its potential bias before learning the activity itself.

[3] http://magi.univ-paris13.fr/wiki/