HUMAN ACTIVITY RECOGNITION BASED ON WEARABLE SENSORS USING MULTISCALE DCNN ENSEMBLE

JESSICA SENA DE SOUZA

HUMAN ACTIVITY RECOGNITION BASED ON WEARABLE SENSORS USING MULTISCALE DCNN ENSEMBLE

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais - Departamento de Ciência da Computação como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: William Robson Schwartz

Belo Horizonte
Outubro de 2018

JESSICA SENA DE SOUZA

HUMAN ACTIVITY RECOGNITION BASED ON WEARABLE SENSORS USING MULTISCALE DCNN ENSEMBLE

Dissertation presented to the Graduate Program in Ciência da Computação of the Universidade Federal de Minas Gerais - Departamento de Ciência da Computação in partial fulfillment of the requirements for the degree of Master in Ciência da Computação.

Advisor: William Robson Schwartz

Belo Horizonte
October 2018

Souza, Jessica Sena de

S729h Human Activity Recognition based on Wearable Sensors using Multiscale DCNN Ensemble / Jessica Sena de Souza. — Belo Horizonte, 2018
xxii, 44 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais - Departamento de Ciência da Computação
Orientador: William Robson Schwartz

1. Computação, 2. Aprendizado de Máquina, 3. Reconhecimento de Atividades Humanas, 4. Sensores de dispositivos móveis, 5. Sensores vestíveis. I. Orientador. II. Título.

CDU 519.6*82.10(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Human Activity Recognition based on Wearable Sensors using Multiscale DCNN Ensemble

JÉSSICA SENA DE SOUZA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

William Robson Schwartz
Prof. William Robson Schwartz - Orientador
Departamento de Ciência da Computação - UFMG

Marcos André Gonçalves
Prof. Marcos André Gonçalves
Departamento de Ciência da Computação - UFMG

Leonardo A. B. Tôrres
Prof. Leonardo Antônio Borges Torres
Departamento de Engenharia Eletrônica - UFMG

Belo Horizonte, 18 de outubro de 2018.

I dedicate this thesis to the giants I stand on the shoulders. From the literature and from the Smart Sense Laboratory.

Acknowledgments

So many people have contributed to the completion of my master. It's safe to say that this thesis would not have been possible were it not for the great collaborators I've had the good fortune of working with as well as my family and friends who supported me every day.

First, and most importantly, I wish to thank my mother, Maria do Carmo, she raised me, unconditionally supported me, loved me and her good examples have taught me to work hard for the things that I aspire to achieve. To her, I dedicate this thesis. I am also very grateful to my beloved Alex who first supported me to pursue my passion for Computer Science when I was still in undergraduate school and has been with me every day since. He has been a constant source of support and encouragement during the challenges of graduate school and life. I am truly thankful for having you in my life.

I would also like to thank my advisor William Schwartz and my friend (almost a co-advisor) Artur Jordão. Artur's constant positive encouragement, enthusiasm, and mentorship over the past two years have made the research not only possible but enjoyable.

I owe so much of my success in this work now and in the future to William directly or indirectly given that he has trained so many of my close mentors and collaborators. William has been a guide to me on all possible fronts including technically and strategically in research and in all of the softer skills which are critical to doing impactful work.

I performed my entire research in the Smart Sense Laboratory. As I now know, it's a rare thing indeed to work with such a highly productive, collaborative and enjoyable group. I am indebted to all of you for providing a stimulating and fun environment in which to learn and grow. I'm grateful to the entire Smart Sense team, but for the sake of brevity, I will cite only a few: Jesimon Santos, Victor Hugo, Gabriel Gonçalves, Carlos Caetano, Cassio Elias, Antonio Carlos, Fernando Akio, Maiko Lie e Ricardo Kloss. I'm grateful to call you friends and I hope our friendship may endure for a lifetime.

To all my friends, thank you for listening, offering me advice, and supporting me through this entire process. Special thanks to Cesar Augusto, Vanessa Vale, Erico Pereira, Bhalchandra Thatte, Humberto Lopes, Kenia Santos, Marlucy Antonina, Matheus Gonçalves, Hugo Oliveira, Alan Deivite, Virginia Mota e Camila Laranjeira. Life is made of short moments that remain in the memory and I am happy to look at mine and see that my best moments have been shared with you.

I thank the members of the examination committee, professors Leonardo Torres and Marcos André Gonçalves, for the enrichment that they brought to this work.

I would like to thank the National Council for Scientific and Technological Development – CNPq (Grant 311053/2016-5), the Minas Gerais Research Foundation – FAPEMIG (Grants APQ-00567-14 and PPM-00540-17) and the Coordination for the Improvement of Higher Education Personnel – CAPES (DeepEyes Project).

"The most exciting phrase to hear in science,
the one that heralds new discoveries,
is not 'Eureka!', but 'That's funny ...' "
(Isaac Asimov)

Abstract

Sensor-based Human Activity Recognition (sensor-based HAR) provides valuable knowledge to many areas, such as medical, military and security. Recently, wearable devices have gained space as a relevant source of data due to the facility of data capture, the massive number of people who use these devices and the comfort and convenience of the device. In addition, the large number of sensors present in these devices provides complementary data as each sensor provides a different information. However, there are two issues: heterogeneity between the data from multiple sensors and the temporal nature of the sensor data. We believe that mitigating these issues might provide valuable information if we handle the data in the correct way. To handle the first issue, we propose to processes each sensor separately, learning the features of each sensor and performing the classification before fusing with the other sensors. To exploit the second issue, we use an approach to extract patterns in multiple temporal scales of the data. This is convenient since the data are already a temporal sequence and the multiple scales extracted provide meaningful information regarding the activities performed by the users. We extract multiple temporal scales using an ensemble of Deep Convolution Neural Networks (DCNN). In this ensemble, we use a convolutional kernel with a different height for each DCNN. Considering that the number of rows in the sensor data reflects the data captured over time, each kernel height reflects a temporal scale from which we can extract patterns. Consequently, our approach is able to extract both simple movement patterns such as a wrist twist when picking up a spoon and complex movements such as the human gait. This multimodal and multi-temporal approach outperforms previous state-of-the-art works in seven important datasets using two different protocols. We also demonstrate that the use of our proposed set of kernels improves sensor-based HAR in another multi-kernel approach, the widely employed inception network.

Palavras-chave: Human activity recognition, Multimodal data, CNN ensemble, Multiscale temporal data.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

The use of sensors provided by wearable devices to recognize human activities has grown every year. As discussed by Lara and Labrador [2013], there are many reasons for this growth: the increasing interest of several areas, such as medical, military, and security applications; the convenience and comfort of using such devices (it does not change or hinders the action due to their use); the feeling of privacy (as opposed to monitoring with cameras where depending on the activity performed or the location, the user feels uncomfortable); and it is already naturally inserted into people's lives, facilitating the data capture.

An interesting feature of wearable devices is the increasing number of embedded sensors. Figure 1.1 shows some of the sensors present in current smartphones. This large range of sensors provides rich and complementary information regarding the activities performed by users. For instance, the accelerometer, magnetometer, and gyroscope can bring information regarding movement, the barometer can bring altitude information, the GPS and WiFi provide location information, while heartbeat sensors can bring information regarding the emotional state. However, sensors have several dissimilarities between their signals, such as a different number of axes, scales, meanings, or data nature (e.g., angle, value, degree, frequency). Therefore, due to the heterogeneous nature of the sensors, an important line of research that has gained attention focuses on the investigation of how to combine (i.e., fuse) these distinct sensors in order to improve human activity recognition Ha et al. [2015]; Ha and Choi [2016]; Yao et al. [2017].

Besides the sensor data heterogeneity, another issue that must be considered is the multi-temporal scale nature of the data. An activity is composed of several complex movements and different durations, i.e., while some activities can only be distinguished by small and fast movements, others need to be analyzed for longer periods of time to

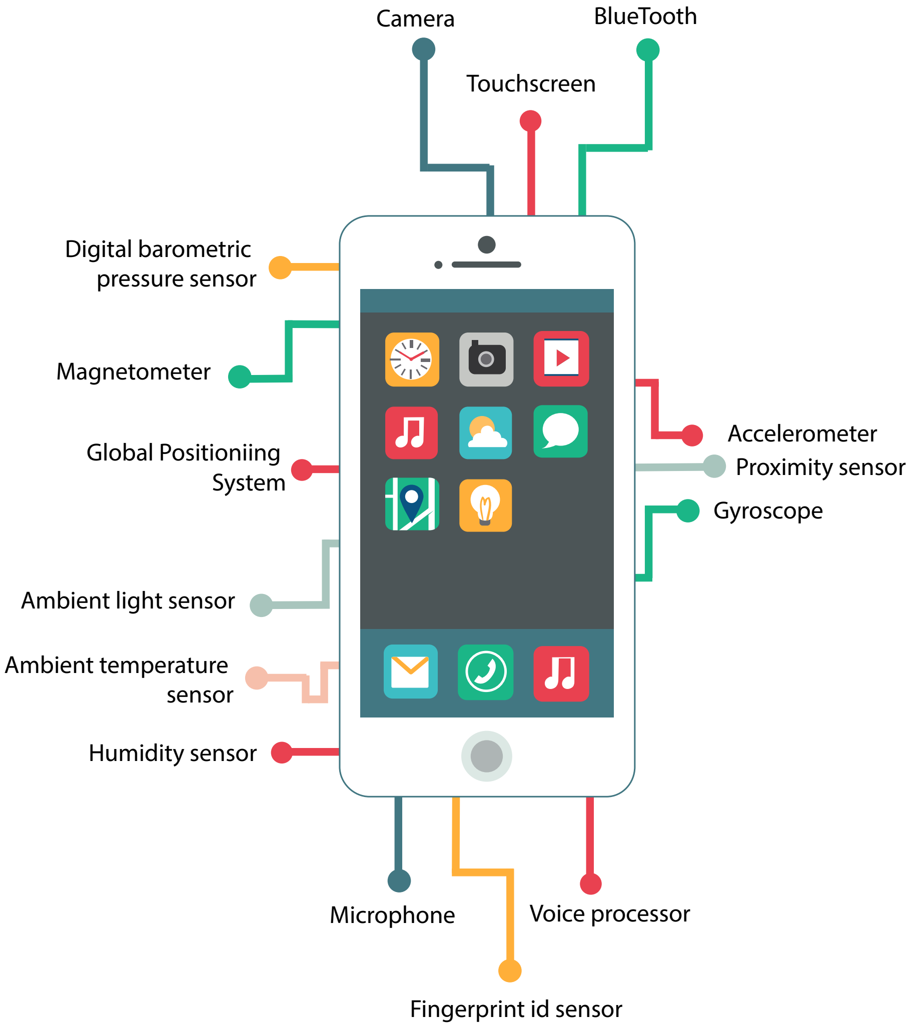

Figure 1.1: Sensors provided by the most common used smartphones.

be classified. In this way, it is hard to recognize an activity only looking at a point in the temporal space.

Figure 1.2 presents the standard pipeline of human activity recognition (HAR) based on wearables sensors. In this figure, we have two devices being wear by the user in his/her daily activities, which are able to provide different information regarding the activities throughout their sensors. First, the raw data of these two sensors are concatenated into a common matrix and then, features are extracted. Usually, there are two types of methods to extract these features, handcrafted methods (using statistic operations, such as average, minimum, maximum) or learned features (using deep learning methods). At the end of the pipeline, the classification is performed using some machine learning algorithm.

In the last years, the most used method to extract and to classify sensor data

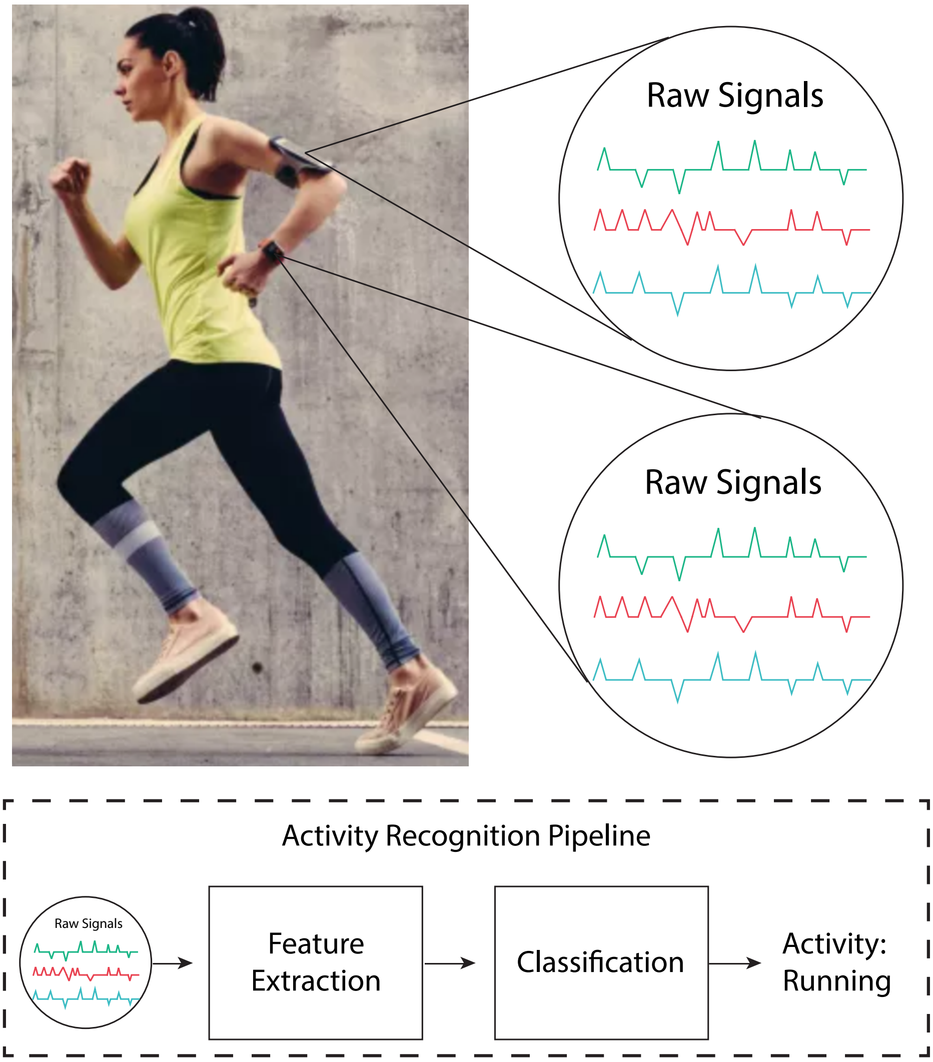

Figure 1.2: Standard human activity recognition pipeline using wearables sensors.

has been deep convolutional neural networks. In traditional DCNN methods Chen and Xue [2015]; Jiang and Yin [2015]; Ha et al. [2015]; Ha and Choi [2016], a single kernel is set for each layer, which discards all other possible temporal scales for that particular layer. In these networks, each stacked convolutional layer learns features at a larger semantic level than the previous one and, in the sensor context, a deeper convolutional neural network would learn features in multiple temporal scales due to its depth (each layer learns a higher temporal scale than the previous one). However, the convolutional maps that go to the next layer are the activations for the kernel in the previous layer. In this way, when one chooses a single kernel size for a specific layer, it might discard important information in this layer which would only be selected by another kernel size. Due to the deep convolutional neural network (DCNN) input format for sensors (where columns refer to the sensor axes and rows to data-capture over time), the height of the convolutional kernel represents the size of the temporal window used to learn activities patterns. Since there are several possibilities for the kernel height, we can see each size as a temporal scale to extract potential patterns.

Based on the above discussion, we develop two hypotheses. The first is related to learn several temporal scales simultaneously. The intuition is that employing multiple kernel sizes (implemented with multiple convolutional layers) at each level of a convolutional neural network would discriminate the human activities better than using the traditional pipeline of setting only one kernel size per level. The second hypothesis regards the processing of the sensors. We believe that processing the sensors separately is a key step in heterogeneous sensor fusion since each sensor brings different information and might be more appropriate to extract this information before fusing with other sensors. To validate our hypotheses, we propose a neural network composed of two main strategies, each one designed based on the discussed premises, as follows.

1. A novel way to extract temporal information by employing an ensemble of temporal scales implemented with multiple DCNNs. As each DCNN has a kernel size that reflects one scale of a pre-defined temporal scale range, we can extract patterns of multiple sizes, ranging from short movements, such as a gentle twist of the wrist, to large and complex motions, such as the human gait.

2. A multiple stream processing system, similar to the work of Yao et al. [2017], to individually process the sensor data. More specifically, each stream is fed with one sensor and employs a DCNN ensemble on the input to extract information before fusing at the end of the pipeline in a late fusion way.

To evaluate this two main parts, we execute an extensive evaluation in seven distinct datasets using the two most appropriate protocols for reporting results in sensor-based HAR. The multimodal premise was evaluated against the work of Yao et al. [2017] and our multiple kernels strategy was compared to an adaptation of the Inception network Szegedy et al. [2015]. The experimental results demonstrate that the separated versions of our approach surpass the baselines and also the employment of our proposed kernels is more suitable for the sensor-based HAR than the kernels originally proposed in the Inception module.

Our two strategies integrate a multimodal deep convolutional neural network ensemble to learn individually the features of each sensor before performing the fusion and to model multiple temporal scales of an action sequence. The proposed approach works directly on the raw sensor data, with no pre-processing, which makes it general and minimizes engineering bias. According to experimental results, our approach outperforms previous methods, achieving, to the best of our knowledge, the state-of-the-art performance in sensor-based human activity recognition.

We also evaluate our approach and the literature methods based on deep learning regarding the number of epochs spent until convergence in training. This is a relevant experiment since the next step to be explored in the task of recognizing human activities in sensor data, is to embed the methods on mobile devices, therefore, requiring fast and efficient methods. We show that our approach is 3× faster than the second place in this experiment and it can be trained with 10× less iterations than the average number of epochs in the literature.

1.1 Motivation

Several areas are interested in classifying the activities performed by an individual. The most common is health care. Within healthcare, an important application is the detection of falls. With the average life expectancy of the population increasing, the number of elderly living alone increases, and a fall without immediate assistance can be critical in certain circumstances. Similarly, individuals with dementia or other mental pathologies can be monitored to detect dangerous activities without invasion of their privacy (as opposed to cameras). Another application within healthcare is patient monitoring. For instance, patients with heart disease, obesity or diabetes are often required to perform predefined physical training as a part of their treatment.

Another field of application for sensor-based HAR are the military and homeland security. Accurate information on the activities performed by the soldiers, as well as their health conditions, could improve the performance and safety in tactical situations. The strategists can use this information in decision-making in both combat and training scenarios.

Due to the aforementioned importance of the sensor-based human activity recognition, the area has received many contributions in the last years. However, the problems in this area are not solved having plenty of room for new advances. One of the possibilities of contributions is extracting multiple temporal scales from the sensor data Chen and Shen [2017]. Another opening for exploration is the fusion of multiples heterogeneous sensors available in the wearable devices to extract complementary information regarding human activities.

1.2 Contributions

During the development of this work, a technical paper entitled *"Multiscale DCNN Ensemble Applied to Human Activity Recognition Based on Wearable Sensors"* containing the contributions of this thesis was published in the proceedings of the 26th European Signal Processing Conference (EUSIPCO) Sena et al. [2018]. Additionally, we contribute as co-author in the journal paper *"Human Activity Recognition based on Wearable Sensor Data - A Benchmark"*, which created a significant standardization of metrics and protocols on seven important datasets and made an extensive evaluation of several methods for human activity recognition based on wearable sensor domain. Currently, this work is under major review in the IEEE Sensors Journal and pre-printed in the arxiv.org database Jordao et al. [2018].

1.3 Work Organization

The remaining of this work is organized as follows. In Chapter 2 we introduce some background concepts we find useful to better understand the explanation of the proposed approach. We provide a review of human activity recognition based on wearable sensors in the Chapter 3. In Chapter 4 we carefully describe each component of our approach. We present, in Chapter 5, the experiments executed to validate the multiscale DCNN ensemble and we discuss the results obtained. Finally, Chapter 6 provides the conclusions and direction to future works.

Chapter 2

Background

The goal of this chapter is to provide important background to understand the proposed approach. We start by explaining deep convolutional neural networks and how some properties and parameters correlate with the sensor-based human activity recognition task. Then, we discuss two data fusion approaches as well as ensemble techniques and their main characteristics.

2.1 Deep Convolutional Neural Network

Some of the most influential innovations in the field of computer vision, Convolutional Neural Networks (CNN) [Krizhevsky et al., 2012] have also become an important tool in sensor-based human activity recognition due to the possibility of modeling the input signal as 2D matrices. The original idea, proposed by Fukushima [1980], is a neural network architecture inspired by the structure and function of the mammalian visual cortex. By proposing a mechanism invariable to shifting in the input patterns, Fukushima [1980] mitigate some problems of the pre-existent algorithms, such as, poor generalization and inability of learning patterns that might occur in different parts of the image. It is important to solve this issue since it is impossible to present to a network all pattern variations of a chair, for instance. Therefore, the network has to be able to learn from a small set of data and generalize to the real world.

The CNN name derives from the convolution operator used to extract features from the input data preserving the spatial relationship and allowing different parts of the network to specialize in high-level features like a texture or a repeated pattern. This process minimizes the number of parameters and reduces the overfitting. Established by Lecun et al. [1998], the widely used concept of multiple layers of neurons allow more complex features to be learned at deeper layers of the network. The following is a summary of the main concepts of the DCNN architecture, for further study please refer to the work of Krizhevsky et al. [2012]; Zeiler and Fergus [2014]; Simonyan and Zisserman [2014]; Szegedy et al. [2015]; He et al. [2016].

2.1.1 Architecture

The Convolutional Neural Network architecture is different than regular Multilayer Perceptron (MLP) network. As shown in the Figure 2.1, regular MLP uses a series of hidden layers to extract patterns from an input. A set of neurons builds each hidden layer and each neuron is fully connected to all neurons in the previous layer. Because of that, the hidden layers of an MLP network are commonly called fully-connected layers. At the end of the regular neural network, there is a last fully-connected layer, called the output layer, that represents the predictions. The number of neurons in the last layer is equal to the number of classes in the data and, given an input, each of these neurons indicates the likelihood of the input to belong to one of the possible classes.

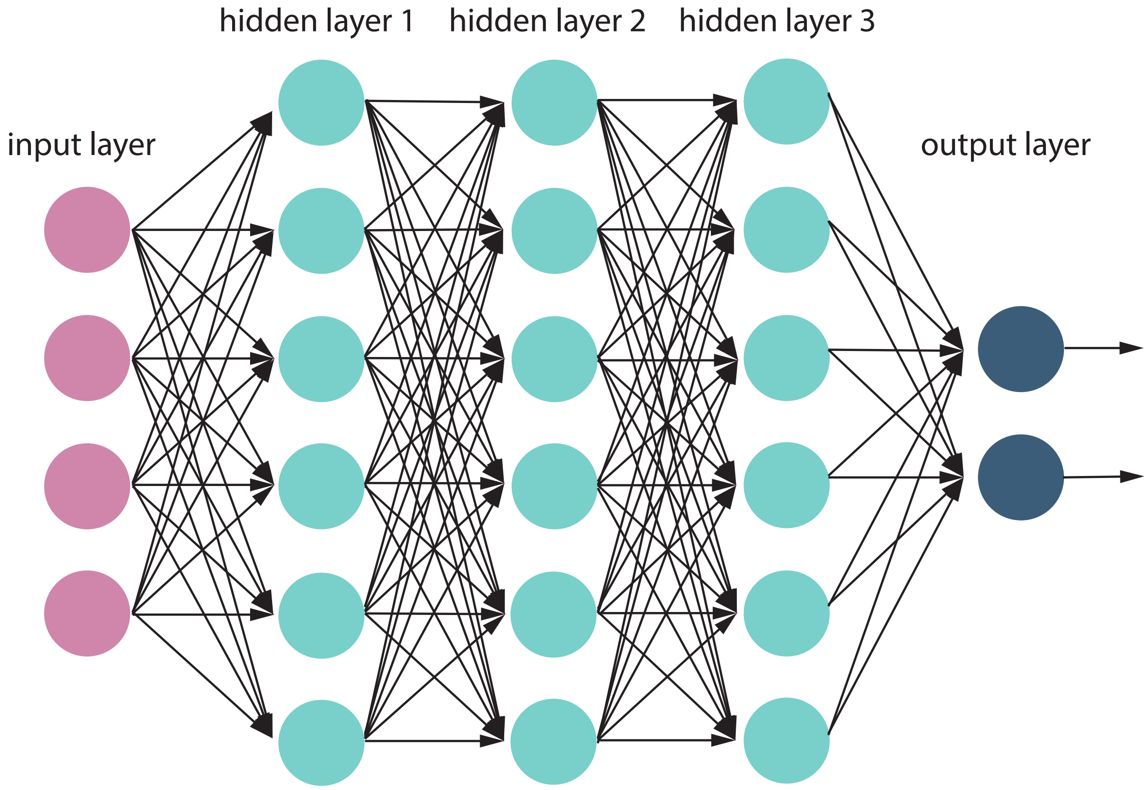

Figure 2.1: The multilayer perceptron architecture is composed of three or more layers (an input layer and an output layer with one or more hidden layers). Since an MLP is a fully connected network, each node in a layer connects to every node in the following layer.

Convolutional neural networks have different properties, as illustrated in Figure 2.2. First, the layers are arranged in three dimensions: width, height, and depth. Then, neurons in one layer only connect to a small portion of the layer ahead of it, instead of connecting to all the neurons in the following layer. In addition, the final output will be reduced to a single vector of probability scores, organized along the depth dimension. Finally, the hidden layers in Convolutional Neural Networks are a combination of convolution layers, pooling layers, normalization layers, and fully-connected layers.

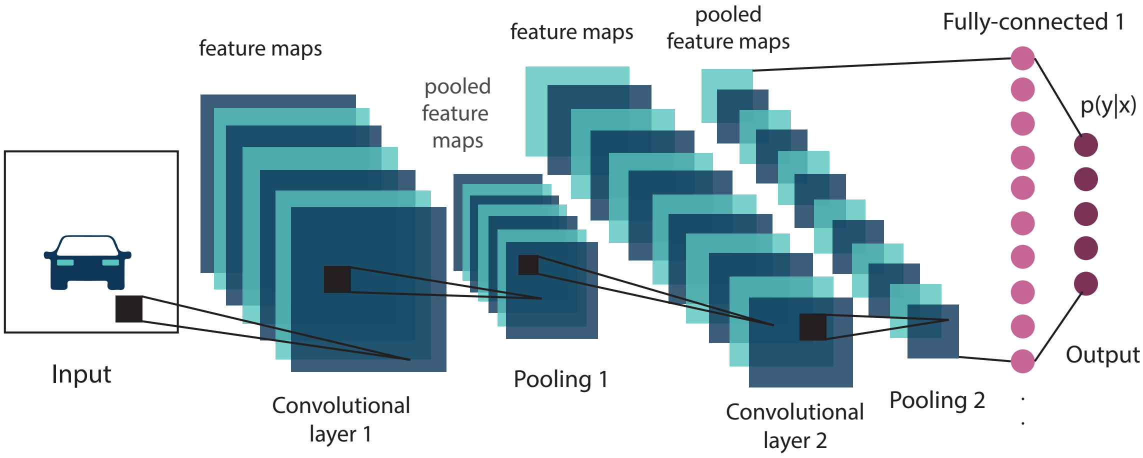

Figure 2.2: CNN architectures are feed-forward artificial neural networks and are composed of three main types of layers: convolutional layer, pooling layer, and fully connected layer. Each layer has a specific purpose and together they create a shift-invariant architecture most commonly applied to analyzing visual imagery.

The input of a Convolutional Neural Network is a $n \times m \times r$ matrix where n is the height and m is the width of the matrix and r is the number of channels, e.g. an RGB image has $r = 3$. One of the major CNN benefits is the ability of learning local patterns in the signal since it explores the spatial-local correlation by enforcing each neuron to connect to only a local region. The size of this connectivity is defined by a hyper-parameter called kernel (also known as filter) of size $h \times w \times c$ where h and w are smaller than the dimension of the image and c can either be the same as the number of channels of the input or smaller and may vary for each kernel. The kernel is used to detect what features, such as edges, are present throughout an image. A kernel is composed of a matrix of values, called weights, that are trained to detect specific features. The filter slides over the whole image enhancing patterns that it has been trained to capture. This operation (illustrated in the Figure 2.5), is called convolution, an element-wise product and sum between two matrices that provides a value indicating the presence of the information captured by the filter in each location of the data. This means that the output matrix will present high values to the parts of the image the feature is present, and low values to the parts where the feature is not.

Zeiler and Fergus [2014] propose an operation to visualize the different levels of information that are extracted by the kernels in different layers of a deep convolution architecture. The general consensus is that in an optimally trained convolution network, the filters at the first layer become sensitive to basic edges and patterns. Likewise, the filters in the deeper layers become sensitized to gradually higher orders of shapes and patterns. Figure 2.3, extracted from the work of Zeiler and Fergus [2014], summarizes the phenomenon.

As illustrated in Figure 2.4, the sensor data are measurements of the sensor axes over time. Therefore, the sensor data is commonly stored in a matrix of size $t \times a$, where a is the number of axes of the sensor and t is the temporal axis, where each row is a sample captured over time. For instance, a 5 seconds capture from a 100Hz accelerometer creates a 500×3 matrix since this sensor have three axes (x, y, z). Employing this representation allows the use of the DCNN architecture to extract patterns from the sensor data. As described above, a DCNN input is an $n \times m \times r$ matrix, in this way, a sensor data input for a DCNN architecture is a $t \times a \times 1$, since the number of channels in sensor data is 1. Regarding the convolution operation, the convolutional kernel height determines the size of the temporal pattern learned in the sensor data, since the height of the input is the temporal axis. In a similar way, since the width of the input is composed of the sensor axes, the convolutional kernel width indicates the size of the correlation between axes that will be considered. In other words, we can extract patterns of each axis separately, the width of the kernel is set to 1 $(w = 1)$, or we can extract patterns of two or more axes simultaneously $(w > 1)$.

To improve the detection capability of CNN's, pooling layers are used to add up translation invariance. Additionally, by reducing memory consumption, the pooling operation allows the employment of more convolutional layers, constructing a deeper network. Each feature map generated by the kernel convolution is sub-sampled (usually with max or average pooling) at a rate of $p \times q$ which reduces the feature maps in height (at p rate) and in width (at q rate). As an illustration, in the image context (where CNN is the most used) p and q ranges between 2 for small images (e.g. MNIST dataset [LeCun and Cortes, 2010]) and is usually not more than 5 for larger inputs (commonly p and q has the same value in this context).

In the sensor context, usually is used a rate of 2×1 to preserve the sensor axes along the convolutional layers and to sub-sample in time more smoothly since sensor samples do not have long duration (from 1 to 10 seconds, leads to small heights in the input matrix and if sub-sampled at a high rate will lead to vanishing of information

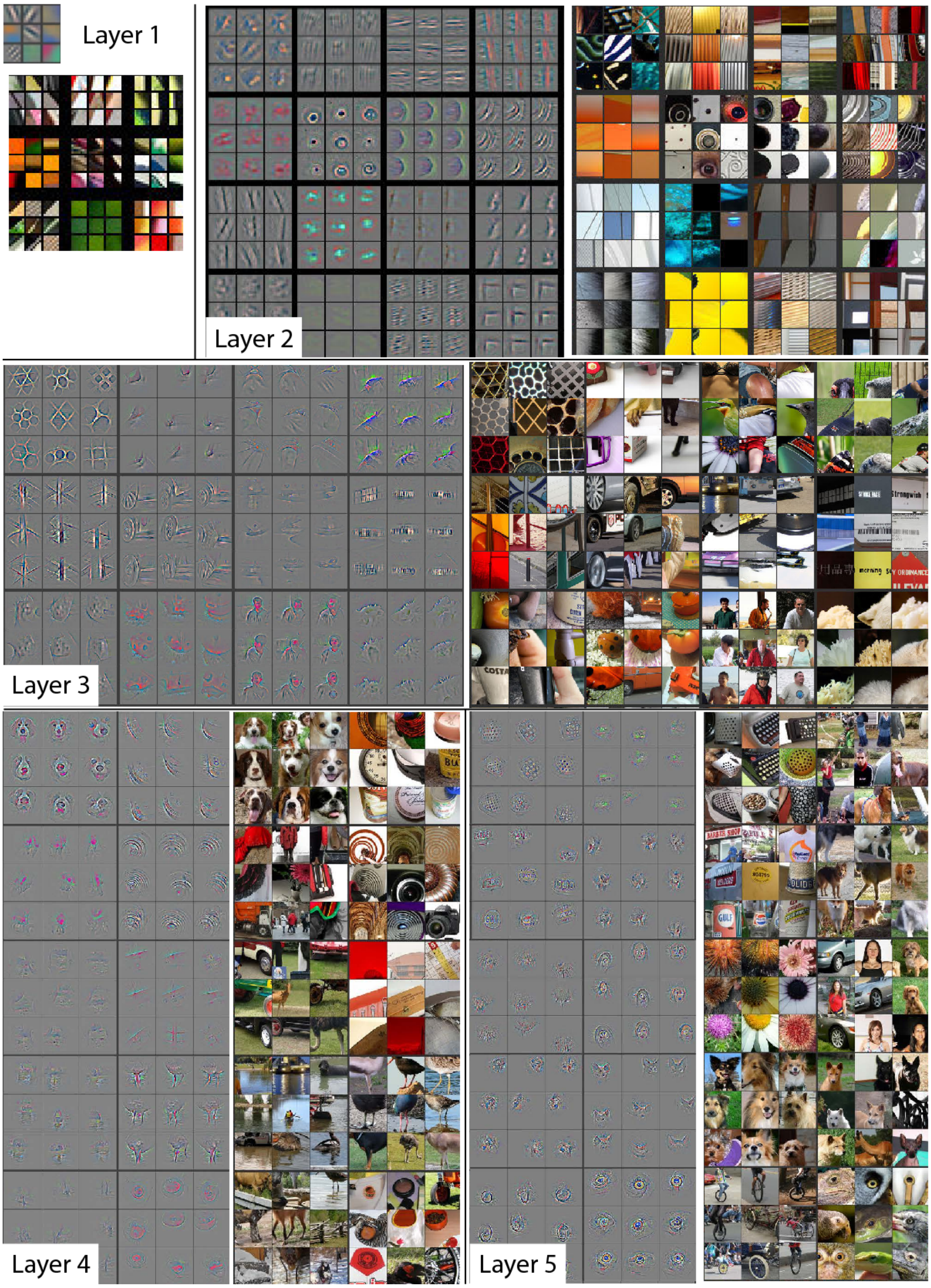

Figure 2.3: Visualization of the features extracted by DCNN kernels in a fully trained model. Figure extracted from the work of Zeiler and Fergus [2014].

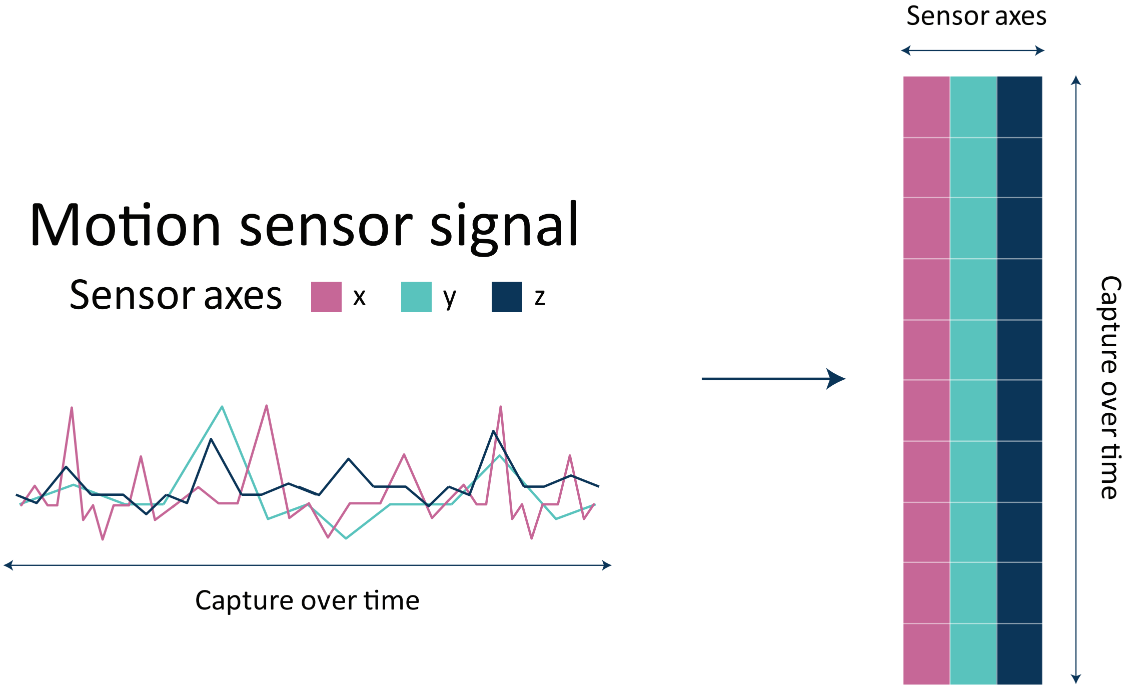

Figure 2.4: Matrix representation of the sensor data.

along the layers).

Finding a balance regarding the size of the pooling rate is very important since the larger is the rate, the more information is condensed, which leads to slim networks that fit more easily into GPU memory. However, if the pooling rate is too large, much information will be lost and predictive performance might decrease. As shown in Figure 2.6, this function progressively reduces the spatial size of the representation which minimizes the number of parameters and computational cost in the network, and also controls overfitting.

Either before or after the sub-sampling layer, an element-wise non-linearity are applied to each feature map. The most typically used is Rectified Linear Unit (ReLU) activation function, defined in the Equation 2.1.

$$relu(x) = \max(0, x) \tag{2.1}$$

ReLU is half rectified function: $relu(x)$ is zero when x is less than zero and $relu(x)$ is equal to x when x is above or equal to zero. The issue of this function is setting all the negative values to zero, which might decrease the ability of the model to fit or train from the data properly since the negative values might not be mapped appropriately. An interesting newly function being used lately, is the Scaled Exponential Linear Units

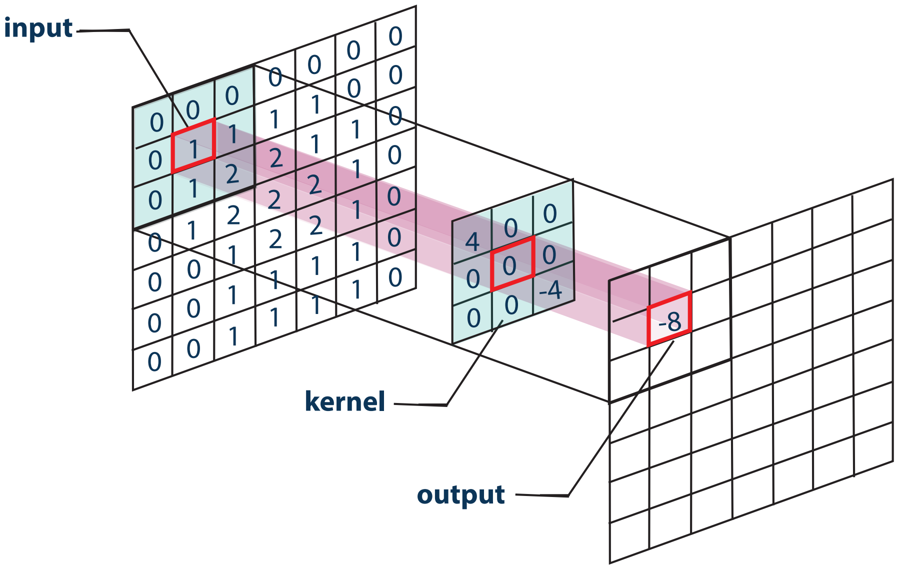

Figure 2.5: Convolution operation refers to the mathematical combination of two functions to produce a third function, merging two sets of information. In the case of a CNN, the convolution is performed on the input data using a kernel to produce a feature map as output. The convolution is performed by sliding the kernel through the input. Each location of the output, is the sum of all element-wise products between each weight of the kernel and the corresponding location value of the input.

(SELU) [Klambauer et al., 2017], defined in Equation 2.2.

$$selu(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha\epsilon^{x} - \alpha & \text{if } x \leq 0 \end{cases} \tag{2.2}$$

SELU has self-normalizing properties which make the learning highly robust and allows to train networks that have many layers. Additionally, the learning speed is faster in SELU compared to the ReLU activation function as shown in the work of Pedamonti [2018].

Regularization layers are employed to normalize the inputs to a mean activation of zero and a standard deviation of one. A commonly employed normalization is batch normalization, where the normalization is performed not just in the input of the network, but also in the inputs of each layer within the network. This regularization technique receives this name because during training the normalization is applied to activations of the previous layer for each batch. Thus, the mean and standard deviation of the activations are maintained close to 0 and 1, respectively. Batch normalization reduces the covariance shift of the hidden units which improves the generalization of the network. Another regularization technique widely used is the dropout, where randomly

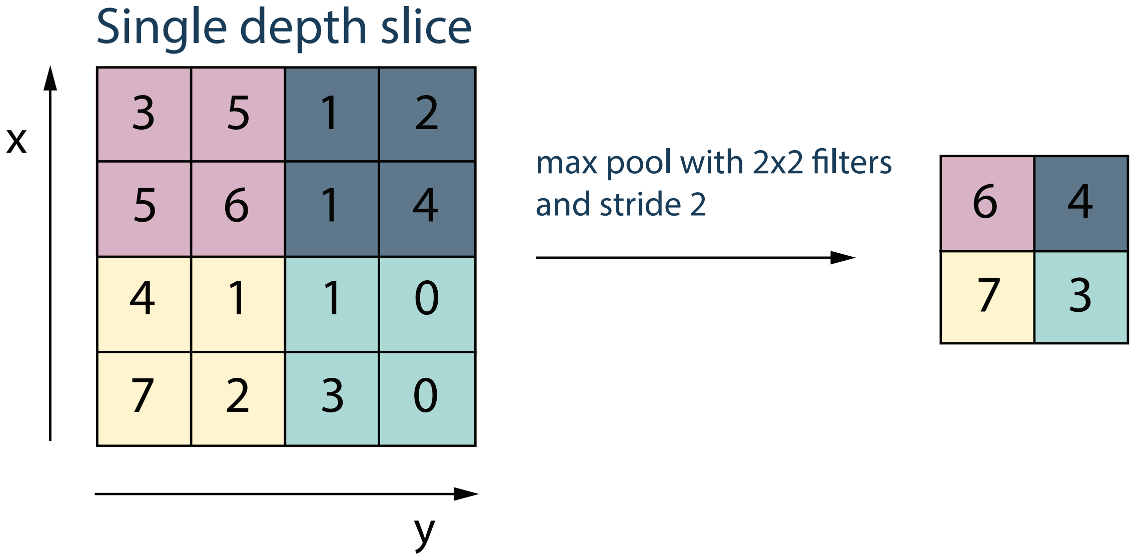

Figure 2.6: Max pooling operation.

chosen neurons are ignored during the training process. Dropout reduces overfitting in neural networks, avoiding exaggerated co-adaptations in the units of the training data.

2.2 Data Fusion

Multi-sensor data fusion is an approach used to combine the data from various sensors, allowing a more reliable and accurate output. This integrates data and knowledge from several sources to produce more consistent, accurate, and useful information than the information provided by the sources separately. Commonly, this fusion is performed in one of two stages: at the feature level, called early fusion, or at the decision level, called late fusion. Regarding fusion of learning algorithms, it is common to call the fusion as Ensemble and the literature propose many techniques to perform this fusion [Rokach, 2010]. We discuss ensemble techniques and the types of data fusion below.

2.2.1 Early and Late Fusion

As discussed in Snoek et al. [2005], early and late fusion differ in the way they integrate the features extracted from the data on the various modalities. Early fusion performs the fusion at the feature level, combining the extracted features into a single representation relying on supervised learning algorithms to classify semantic concepts. In this way, early fusion can be defined as a fusion technique that integrate the features before learning semantic concepts. A disadvantage of this method is that this type of approach has to deal with many heterogeneous features which are sometimes hard to combine. The general steps of this approach are depicted in Figure 2.7.

Late fusion addresses the problem of combining the prediction scores of multiple classifiers. In this fusion technique, the features extracted from each input are classified separately and these scores are combined afterwards to yield a final classification score in a decision level fusion. Late fusion can be defined as a fusion technique that first turns learned concepts into scores and then integrates the scores to learn complementary concepts. Late fusion focuses on the individual strength of modalities. Instead of yielding a jointly feature representation of modalities, late fusion provides a multimodal semantic representation. The major disadvantage of this approach is the high cost in terms of learning effort, since each modality requires a separate stage of supervised learning. In addition, the combined representation requires an additional stage of learning. The general steps of late fusion technique are illustrated in Figure 2.8.

2.2.2 Ensemble

Ensemble method is a well-known and widely used supervised learning algorithm that intends to improve the predictive performance by combining multiple learning algorithms. Ensemble learning has two interesting characteristics. First, results achieved by using ensemble techniques are superior to the results obtained from any of the learning algorithms elements alone [Opitz and Maclin, 1999; Polikar, 2006; Rokach, 2010].

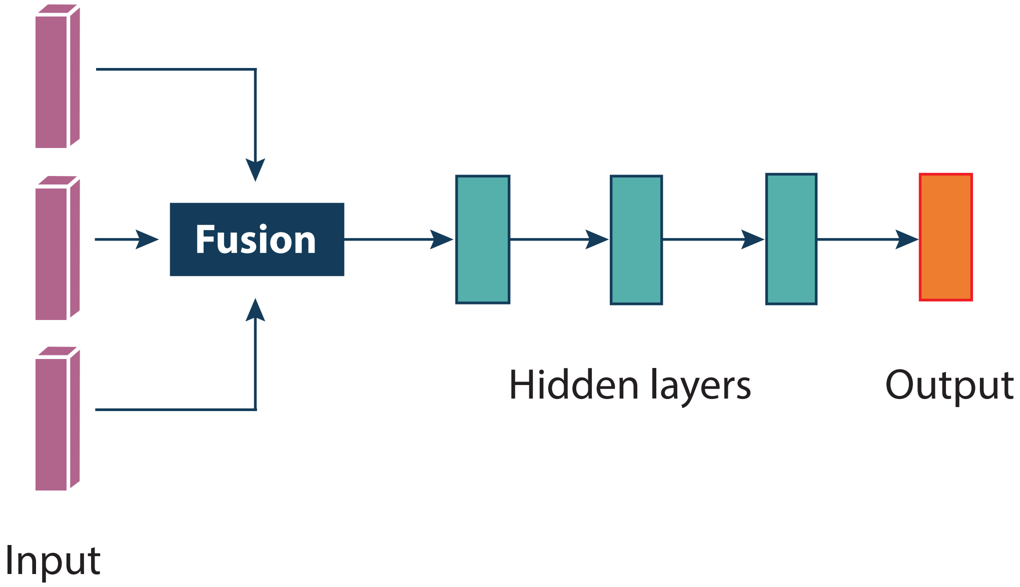

Figure 2.7: Early fusion scheme where the fusion is performed at the feature level.

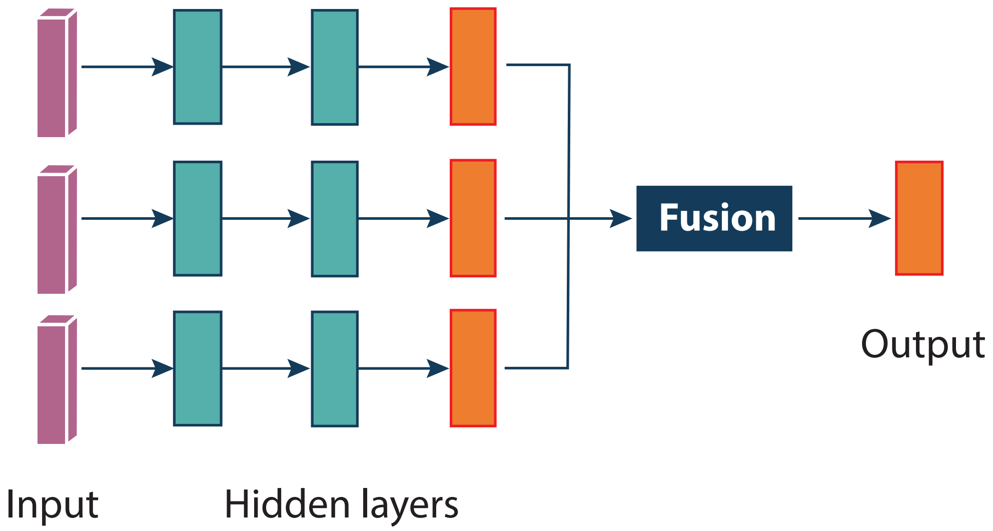

Figure 2.8: Late fusion scheme where the fusion is performed at the decision level.

Likewise, the premise of this work is that the use of multiple kernel sizes lead to better modeling of the data than to use only one size of kernel. Second, ensemble methods achieve better results when the methods to be merged show significant diversity among them [Kuncheva and Whitaker, 2003; Sollich and Krogh, 1996]. The main point of diversity in our approach is that each DCNN's of the ensemble has different kernel sizes, thus, each one learns different patterns of the data. In addition, the extended version of our approach (where we process the sensors independently) shows another point of diversity: the diversity among the data of the sensors provided by the wearable devices (as discussed in Chapter 1).

There are many ensemble techniques, among which we can cite Bootstrap aggregating (bagging), Boosting and Stacking. In bagging, the models in the ensemble vote with equal weight and each model is trained in a random subset of the training data to promote variance. Boosting builds the ensemble incrementally, by training each new model to focus on the instances misclassified by the previous models. Lastly, stacking (or stacked generalization) combines the prediction of the learning algorithms in the ensemble and use this meta-data to make the final prediction, achieving the highest generalization accuracy among the ensemble methods [Wolpert, 1992]. Stacking is usually employed to combine models built by different learning algorithms and tries to induce which are reliable and which are not [Rokach, 2010].

Chapter 3

Literature Review

In recent years, the employment of multiple sensors brought improvements in the accuracy of sensor-based activitiy recognition. However, the fusion of sensors is still a challenging task due to the heterogeneity of the data. To handle this, some works perform fusion in the raw data (i.e., early fusion), concatenating the sensors into a common matrix used as input for machine learning methods. Kwapisz et al. [2010] for instance, extracted handcrafted features (i.e., average and standard deviation), also known as engineered features, from the raw signal to represent the activities. The authors examined three classifiers, multilayer perceptron, decision tree (J48) and logistic regression, to estimate the best one able to separate the categories of activities on the features extracted. In their experiments, multilayer perceptron achieved the best classification results. Similarly, Chen and Xue [2015] used the raw data to feed a Deep Convolutional Neural Network (DCNN) with three convolutional layers and used the size of the kernel to extract temporal patterns and correlation between the axes. More specifically, they extracted correlation between neighboring pairs of signal axes in the first layer using a 12×2 convolutional kernel and extracted temporal information using a kernel of 12×1 in the remaining layers.

DCNNs are known by their power to automatically discover the best representations from raw data to perform classification. This extraction technique is immune to the bias present in the handcrafted feature where a human creates the feature based on her/his knowledge of the data. Following this intuition, Jordao et al. [2017] employed the DCNN as the feature extraction step and applied a partial least squares analysis on each max pooling layer of the network to reduce the dimensionality of the representation. Then, they used the concatenation of the latent variables as a feature to feed a multilayer perceptron classifier. On the other hand, Jiang and Yin [2015] applied a discrete Fourier Transform (DFT) to preprocess the input and used a DCNN composed by a stack of two convolutional layers, a fully connected and a softmax layer to recognize the activities.

Our proposed approach use DCNNs to extracted features from the raw data since the preprocess of the input does not show empirically improvements in the accuracy. The difference is that we use the DCNN as an element of an ensemble to extract multiples temporal scales at the same time. Each DCNN is set with a specific kernel size to extract a specific temporal scale in the data. The hypothesis is that using the standard DCNN pipeline where a single kernel is set for each layer, might discard important information. We believe that this occurs because of the feature maps that go to the next layer are the activations for the kernel in the previous layer. Since each kernel learns patterns in a specific temporal scale (based on its height), all other possible temporal scales for that particular layer will not be extracted, since this patterns would only be selected by another kernel size. The general consensus is that each level (layer) of the DCNN learns a high-level semantic pattern than the previous one. In this way, discarding information at any level of the DCNN discards information that could be useful to the next level.

Using the idea of extracting different patterns from the data at the same time, a remarkable work is the Inception network proposed by Szegedy et al. [2015]. The work employed multiple kernels to recognize objects in the context of images. The idea is that salient parts in the image can have extremely large variation in size. Consequently, choosing the right kernel size for the convolution becomes difficult. A larger kernel is preferred for patterns that are distributed more globally while a smaller kernel is preferred for patterns that are distributed more locally. Thus, their filters with multiple sizes operate on the same level, extracting different patterns at the same time. They used filter sizes of 1×1, 3×3 and 5×5.

Due to the multimodal nature of each sensor, merging the sensors in the raw data may not be appropriate since, as discussed in Chapter 1, sensors have several dissimilarities between them. To address this, some authors proposed to insert a padding between the sensors to separate the data and to be able to extract features from the sensors separately. For instance, Ha et al. [2015] preprocessed the matrix of sensors adding a zero-padding between each sensor and used a DCNN with the same layer structure as Jiang and Yin [2015]. However, this division is only effective in the first layer since, from the second layer onward, the data from different sensors are convoluted together. In fact, in another work, Ha and Choi [2016] proposed to insert zero-padding before each convolutional layer to avoid interference between sensors when a 2D convolutional kernel is applied. Despite this approach can separate in some way the data before performing fusion, it used the same DCNN to learn features from all sensors simultaneously, which might overcharge the model since the kernels have to learn patterns from different data nature.

In a recent work, Yao et al. [2017] brought a new perspective on merging multimodal data to perform sensor-based human activity recognition. They built an architecture with three different sequential blocks: an individual deep convolutional subnet for each input sensor to learn local patterns, a common deep convolutional subnet that concatenates all sensors and learns the high-level relationship among them and, at the end of the architecture, a recurrent neural network (RNN) structure to learn meaningful temporal features. Since the use of convolutional and recurrent networks are already well-established in the sensor literature, the main advance of Yao et al. [2017] is to go beyond of just placing a boundary between the sensors in the input matrix. Instead, they separated the sensors from the beginning to extract features individually and learn which patterns better classify human activities for each sensor before merging and benefiting from their complementarity. Our approach follows the intuition of Yao et al. [2017] of extracting patterns from each sensor separately before performing the fusion. However, in contrast with the fusion at the feature level performed in their work, we execute the fusion at the decision level since empirical evaluations showed it is more suitable to use the sensor data. Furthermore, we do not employ RNNs to extract temporal data, instead, we propose a novel approach to handle temporality by simultaneously processing multiples temporal scales of the data using an ensemble of DCNNs.

Another way of improving results is to employ ensemble techniques to classify wearable sensor data. Catal et al. [2015], for example, proposed to apply a majority vote technique to ensemble multilayer perceptron, decision tree (J48) and logistic regression and compose the final predictor. They achieved a more accurate classification when compared with Kwapisz et al. [2010], which used the same features and same classifiers but analyzed the classifiers separately. In a similar manner, Feng et al. [2015] employed a weighted majority voting to ensemble these classifiers. In their work, 179 handcrafted features were extracted (167 time domain features and 12 frequency domain features) from multiple sensors and seven random forest classifiers with different weights were used as base classifiers. On the other hand, Kim et al. [2012] used a bagging technique to recognize human activities. They first divided the activity into a set of action units and extracted handcrafted features (average and correlation) from each of them. Then, they classified each action units of the set using a bagging of decision trees. Finally, based on the proportion of each action unit, they predicted to which activity each action unit belonged. Similarly, Kim and Choi [2013] proposed to use another ensemble technique, the boosting compose of decision trees but with a smaller number of action units. In contrast, Min and Cho [2011] created a system to dynamically chose a subset of base-classifiers using the class probabilities of an input activity. The outputs of the chosen classifiers are subsequently combined in a fusion step.

Different from the discussed ensemble approaches, the diversity generated in our ensemble resides in the feature extraction. Each element of our ensemble is a neural network composed of convolutional layers, that extract features, and a multilayer perceptron, that is used as the classifier. Each neural network has a different kernel size in its convolutional layers allowing the extraction of patterns from multiples scales at the same time. The fusion occurs at the decision level of the network (after the classification of each DCNN) using a softmax layer in an end-to-end way, connecting the entire pipeline from the input to the output. The idea of our proposed ensemble is to learn the correct scale for each type of activity and create an ensemble for each sensor to explore the individual characteristics before fusing and making use of the complementarity between them.

Chapter 4

Methodology

In this work, we propose an approach to recognize human activities using data provided by smartphones and smartwatches. This approach is based on two hypotheses: (i) the use of multiples sensors might improve accuracy due to the complementarity between the sensors, (ii) activities are best described using multiple temporal scales to extract patterns. To test these hypotheses, our approach consists of three main steps. First, we divide the sensors into different inputs to process each one individually. Then, for each sensor, we build an ensemble of temporal scales extracted through DCNN streams that are subnets within our network. Finally, we use an approach based on late fusion to merge the multi-modal and multi-temporal information. The following sections detail each step of this process. Figure 4.1 summarizes an overview of our method.

4.1 DCNN Stream

Our approach is an end-to-end neural network composed of subnets integrated through an ensemble technique. These subnetworks, for convenience, let us call them *streams*, are Deep Convolutional Neural Networks composed of two parts, as illustrated in Figure 4.2.

The first block of the stream is a convolutional block with two convolutional layers intercalated by two max-pooling layers. Convolutional layers allow us to learn patterns in the temporal scale defined for each stream. Pooling layers control overfitting and reduce the number of parameters and the computation cost. At the end of the subnet, we have a fully-connected block consisting of a flatten layer, a fully-connected layer and a softmax layer. We use scaled exponential linear units (SELUs) [Klambauer et al., 2017] as the activation function of the fully connected block. While the convolutional block provides a meaningful and invariant feature space, the fully-connected block is

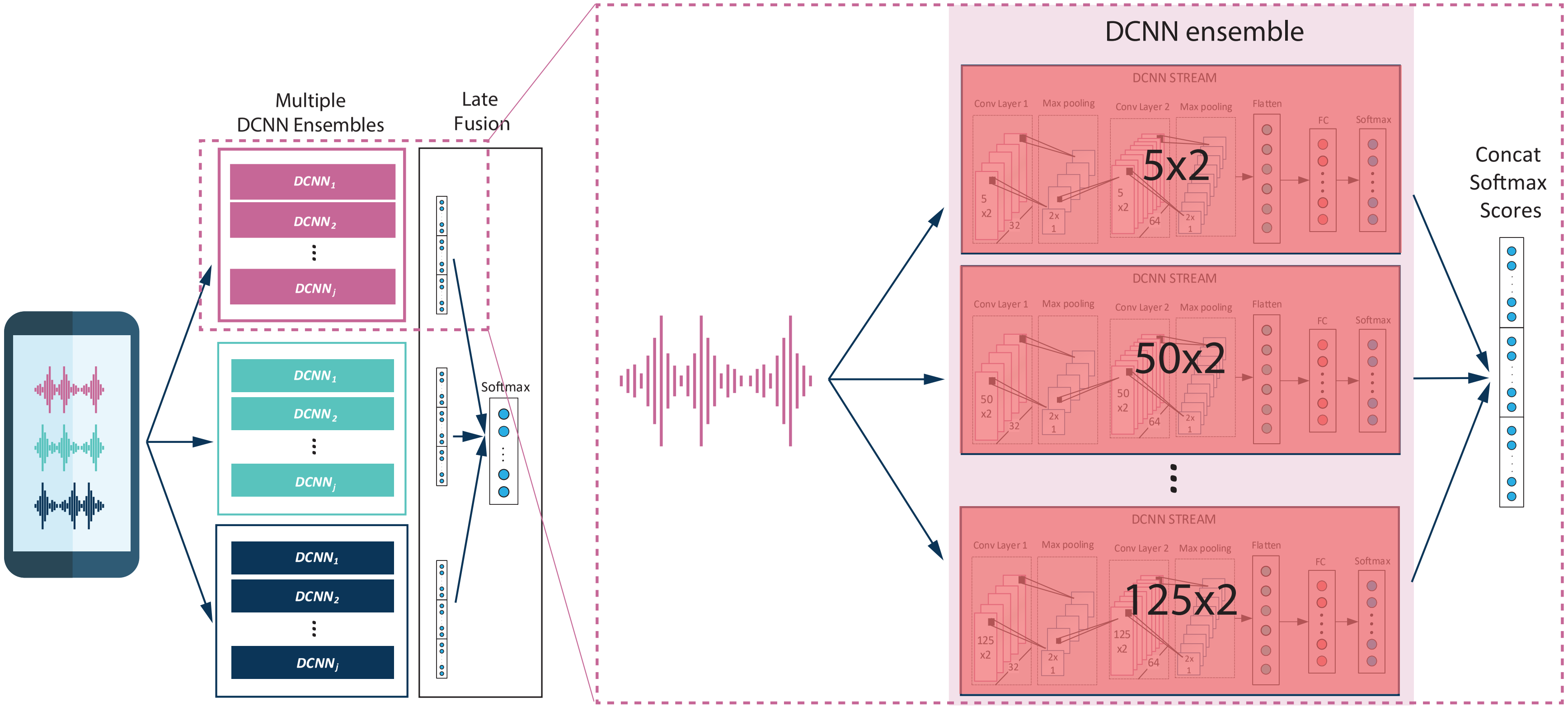

Figure 4.1: Our Multimodal DCNN Ensemble relies on two premises. The first is separately processing each sensor and the second is to extract patterns from multiple temporal scales. For each sensor, we create a DCNN ensemble that extracts multi-temporal information.

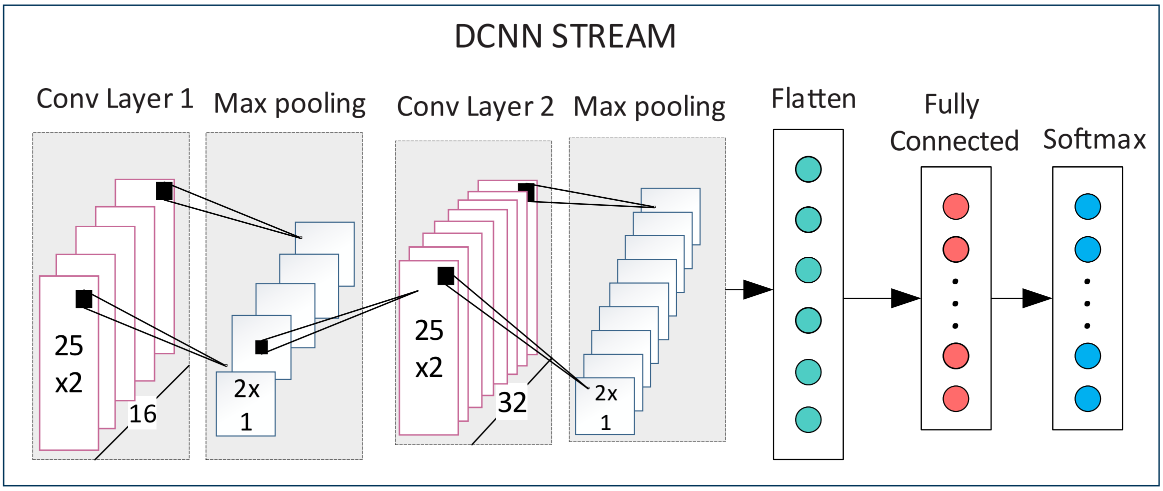

Figure 4.2: The deep convolution neural network stream.

learning a non-linear function in that space, which translates the features learned by the convolutional block to the softmax scores.

4.2 DCNNs Ensemble

The sensor data is commonly stored in a matrix of size $t \times a$, where a is the number of axes of the sensor (for instance, three axes (x, y, z) on motion sensors) and t is the temporal axis, where each row is a sensor sample at a given time instant. Therefore, given a 2D kernel (h, w), our premise is that the height of the kernel (h) is responsible for determining in which temporal scale we are learning the patterns. For instance, a h equal to 25 in a matrix of 500 rows (a sample of size 5 captured at a frequency of 100Hz) learns patterns of 0.25 seconds while a h equal to 250 learns patterns of 2.5 seconds. Thus, the larger the kernel height, the larger the temporal pattern it captures.

Based on the aforementioned premise, we employ an ensemble of deep convolutional streams with different kernel sizes each, to extract information from multiple temporal scales. The architecture of our multiscale ensemble is illustrated in Figure 4.3. The number of DCNNs in each ensemble is pre-defined as a parameter of our architecture, called *pool*. The pool is a set of kernels $K = \{K_1, K_2, ..., K_j\}$ containing a variety of kernel sizes ranging from a small to a large kernel. For each kernel in our pool, we add a DCNN in the ensemble and set its two convolutional layers with the specific kernel. For instance, in Figure 4.3, we have a pool of j kernels where three of them have their streams explicitly drawn in the figure composing a kernel pool $K = \{5 \times 2, 25 \times 2, ..., 250 \times 2\}$.

The multiscale ensemble is the most important contribution of this thesis since,

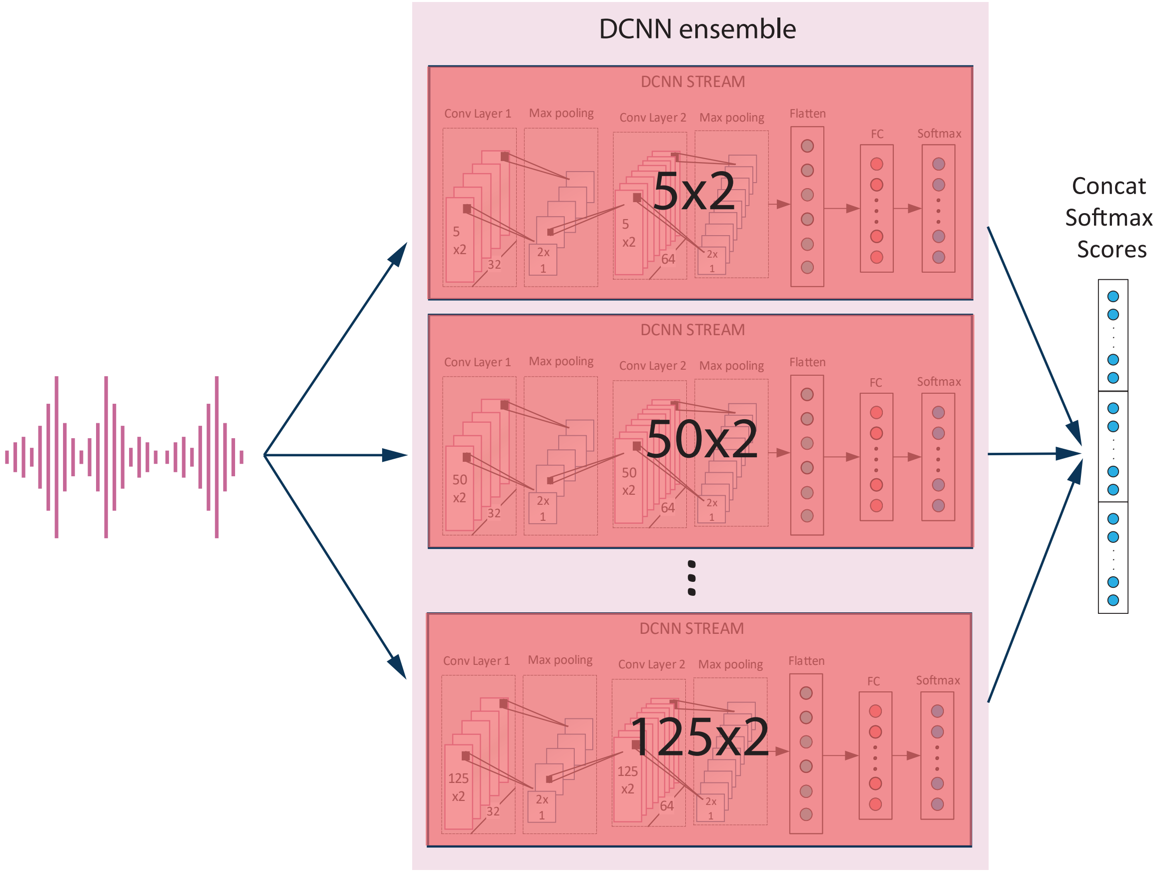

Figure 4.3: The DCNN ensemble is composed of streams so that each stream extracts patterns from a specific temporal scale and classifies the sample for that scale. We merge all scores into a late fusion approach which allows us to take advantage of the complementarity between sensors and between temporal scales.

to the best of our knowledge, we are the first to extract patterns on sensor data using multiple scales. As shown in the architecture of our main approach (Figure 4.1), an ensemble is built for each sensor, so we have several ensembles, according to the number of sensors processed (in the example illustrated in Figure 4.1, we have three sensors and consequently, three ensembles of DCNNs).

4.3 Decision Level Fusion

At the end of the DCNN ensemble stage, we have an ensemble for each sensor, and each ensemble is composed of j streams. Thus, it is necessary to merge this information to take advantage of the complementarity provided by both the multiple sensors and the multiple temporal scales. According to our experimental results, the best way to merge these streams is by using meta-learning of the scores. Therefore, we concatenate all the scores of the streams (j × number of sensors) in a single feature vector and pass it to a classification layer (softmax).

The training of our network is performed in an end-to-end way, which optimizes the weights of the entire network since it maps the input of all the modalities to a single output. Consequently, the network dynamically learns which scales and sensors are most appropriate for each activity.

Chapter 5

Experimental Results

The proposed multimodal DCNN ensemble (MDE) method is composed of two main parts, the ensemble of convolutional neural networks and the individual processing of the sensors. To evaluate the contribution of each part, we implemented two simplified versions of our proposed approach. The first, called *DCNN Ensemble*, is illustrated in Figure 5.1. In this version, we do not separate the sensors, instead, we concatenate all sensors into a single array, in the same way of the majority of works. Thus, we employ only a single ensemble of kernels since we have only one input. The goal is to measure the contribution of the multi-temporal scale approach implemented with the DCNN ensemble, in a scenario without multimodal processing of sensors. Figure 5.2 shows the second simplified version of our approach, called *Multimodal Stream* that aims at measuring the contribution of individual processing of the sensors. In this version, we create a network following the multimodal hypothesis but without using the DCNN ensemble approach. Instead, we employ only a single DCNN stream (see Figure 4.2) for each sensor. In this DCNN stream, we empirically choose the value of 25×2 to set the kernel size.

We compare our approach and its simplified versions with all methods evaluated by Jordao et al. [2018]. Thereby, in addition to the methods mentioned in Chapter 3, we also show results from three other handcrafted methods [Kwapisz et al., 2010; Catal et al., 2015; Kim and Choi, 2013] surveyed by Jordao et al. [2018]. Usually, this family of methods extracts statistical features and applies a classifier to recognize activities. We include them in our evaluation mainly because they present better results in some datasets than approaches based on deep learning. Due to the multimodal nature, we evaluate the MDE and Multimodal Stream only on datasets that contain more than one sensor.

The remaining of this chapter is organized as follows. We start describing the

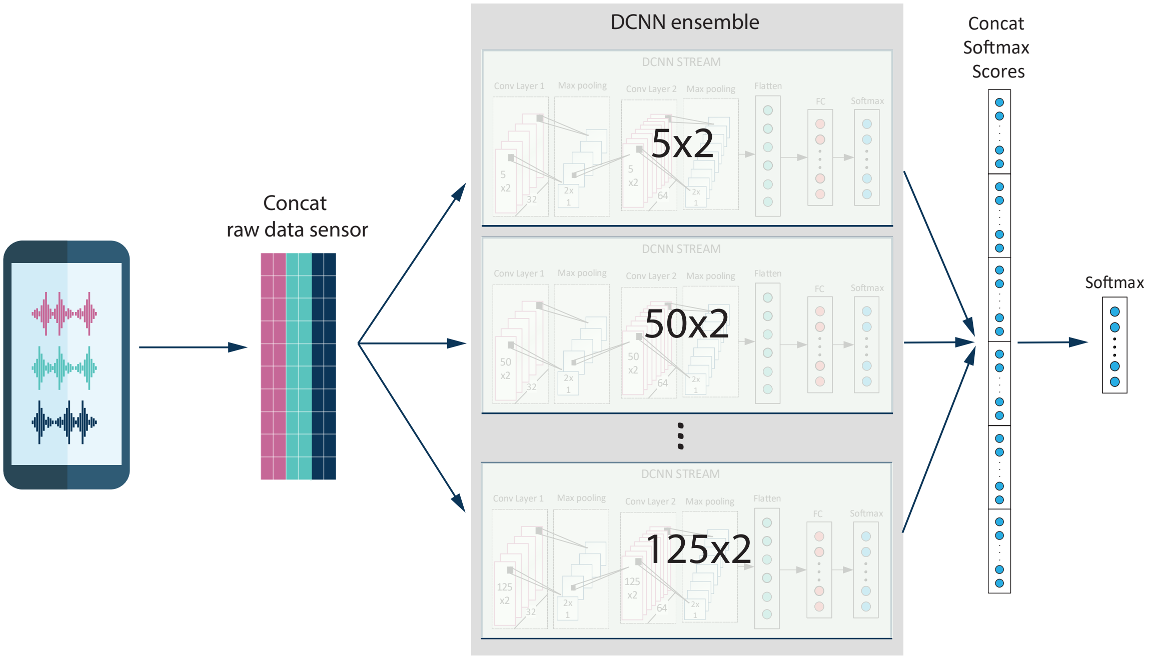

Figure 5.1: DCNN Ensemble approach on previous concatenated sensors data

datasets and protocols employed in our work. Then, we discuss the evaluation of fusion approaches that led to the employment of late fusion in our method. Finally, we present the experiments, results, and discussions regarding our MDE method and its simplifications.

5.1 Experimental Setup

One of the most latent problems in wearable sensor-based human activity recognition is the lack of standardization of metrics, evaluation protocols, and datasets, which makes it difficult to compare the methods in the literature. While some works record their own datasets to perform experiments, others use datasets from the literature but do not clarify the evaluation protocol employed, which prevents the reproducibility of the results. Recently, a work has endeavored to solve this issue by bringing the first standardization to the domain. Jordao et al. [2018] performed a thorough study and standardized seven datasets of the wearable sensor literature in four different protocols. In this section, we quickly describe the experimental setup employed in this work using the framework proposed by Jordao et al. [2018] and then, we discuss the results achieved by our proposed approach and its two simplifications.

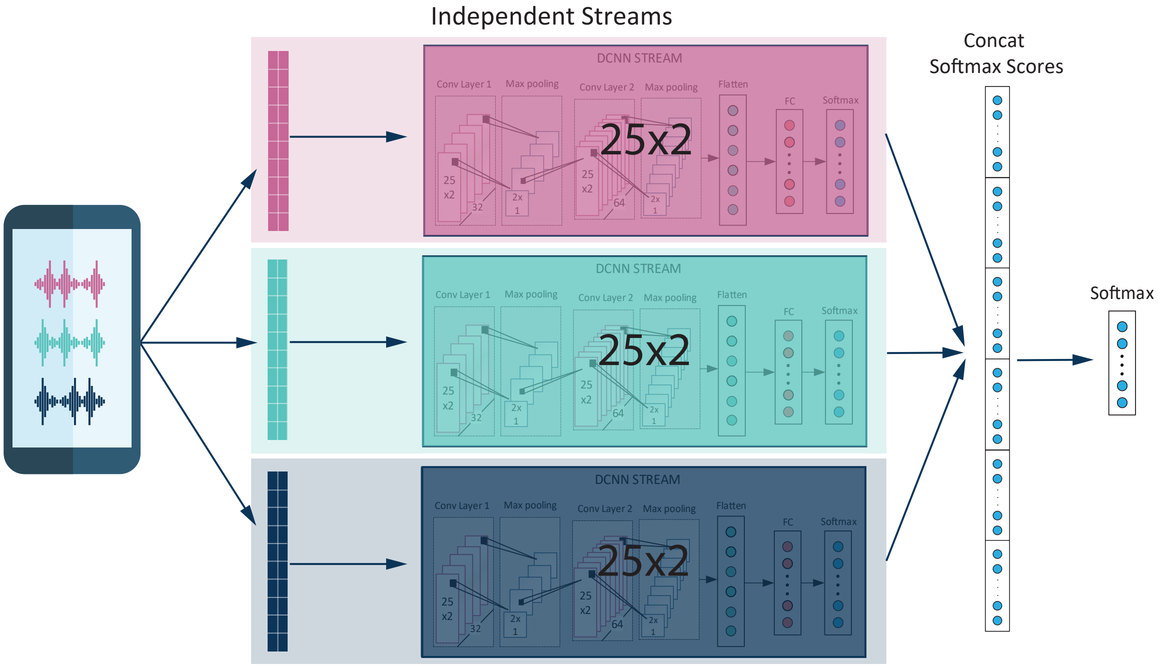

Figure 5.2: Multimodal stream approach. A simplified version to evaluate one of the main steps of our MDE approach. This version is based on the premise of processing the sensors separately to extract meaningful features before fusion. We do not use the DCNN ensemble in this version, instead, we use only one stream to process each sensor and we set the kernel size as 25 × 2.

5.1.1 Architecture Parameters

Most settings of our approach were discussed in Chapter 4 and we provide the implementation of our approach and its simplifications on GitHub[1]. However, we will add some technical details to ensure the reproducibility of our approach.

We implement our approach and its simplifications using Keras Python library running on top of the TensorFlow framework. Keras is a high-level neural networks API and provides a clean, intuitive, modular and extensible interface that enables fast experimentation with deep neural networks. TensorFlow is a highly flexible and versatile open source deep learning framework for creating artificial intelligence applications.

As pointed by Jordao et al. [2018], most of the works based on convolution networks omit some important parameters, hindering comparison between methods. To handle this issue, the protocol created by Jordao et al. [2018] sets some parameters. The maximum number of epochs was set to 200, and the method stops its training when the loss function reaches a value less or equal than 0.2. These values were set empirically by observing the trade-off between execution time and accuracy. Regarding deep

[1] https://github.com/jessicasena/mde

Table 5.1: Main features of the datasets used in this work and previous reported by Jordao et al. [2018]. Acc, Gyro, Mag and Temp indicate accelerometer, gyroscope, magnetometer and temperature, respectively.

Dataset	Frenquency (Hz)	#Sensors	#Activities	#Samples
MHEALTH	50	3 (Acc, Gyro, Mag)	12	2555
PAMAP2	100	4 (Acc, Gyro, Mag, Temp)	10	7522
USCHAD	100	2 (Acc and Gyro)	12	9824
UTD-MHAD1	50	2 (Acc, Gyro)	21	3771
UTD-MHAD2	50	2 (Acc, Gyro)	5	1137
WHARF	32	1 (Acc)	12	3871
WISDM	20	1 (Acc)	7	20846

learning implementation decisions, we use *cross-entropy* as the loss function of our network. Cross-entropy measures the performance of a classification model whose output is a probability value between 0 and 1. The loss increases as the predicted probability diverge from the actual probability. We employ the RMSprop [LeCun et al., 2012] as optimizer since it provides an efficient execution time. As dropout layer, we use alpha dropout [Klambauer et al., 2017] since it fits well to scaled exponential linear units by randomly setting activations to the negative saturation value. Alpha dropout keeps the mean and variance of inputs to their original values, to ensure the self-normalizing property even after dropout. We set the dropout rate to 0.1.

Regarding the ensemble implementation, it is important to mention that in the DCNN stream (see Figure 4.2) we use 16 filters in the first convolutional layer and 32 filters in the second. In addition, the results shown in our experiments were performed using $K = \{2 \times 2, 3 \times 3, 5 \times 2\, 12 \times 2\, 25 \times 2\}$ as our pool of kernels.

5.1.2 Datasets

Jordao et al. [2018] conducted a survey in the literature and gathered seven important datasets. This set of datasets composes an interesting diversity of number of samples, types of activities performed and number of available sensors, making it possible to evaluate the robustness of the methods in different scenarios. The datasets were processed and standardized with a sampling rate of 5 seconds, except for the datasets of the UTD-MHAD family that had to be sampled at 1-second rate. We evaluate our approach in these seven datasets following strictly the procedure defined by Jordao et al. [2018] (Refer to Jordao et al. [2018] for more details regarding the evaluation procedure). Next, we briefly discuss the datasets selected by Jordao et al. [2018] and used in our evaluation. Additionally, Table 5.1 summarizes the main features of these datasets.

WHARF [Bruno et al., 2015] This dataset consists only of accelerometer signals sampled at a frequency 32Hz. WHARF is composed of 14 activities, however, the framework of Jordao et al. [2018] only uses 12 activities because of the protocols applied.

USCHAD [Zhang and Sawchuk, 2012] A relevant dataset suitable for training neural networks due to its large number of samples. Composed of accelerometer and gyroscope data captured for 12 activities collected at 100Hz.

UTD-MHAD [Chen et al., 2015] This dataset is divided into two subsets, as recommended by the authors. The first subset, the UTD-MHAD1 consists of activities where the sensor is positioned in the subject's right wrist, containing the largest number of activities (21) among the datasets used in this work. The second subset, the UTD-MHAD2 is composed of activities performed with the sensor in the subject's right thigh and consists of 5 activities. Both have accelerometer and gyroscope data available.

WISDM [Lockhart et al., 2011] It is a challenging dataset due to the small sampling rate used to capture the data. However, this dataset provides a large number of samples and subjects (20846 samples and 36 subjects). It consists of only accelerometer data.

PAMAP2P [Reiss and Stricker, 2012] This dataset provides accelerometer, gyroscope, magnetometer, and temperature acquired at a sampling rate of 100Hz. It is divided into two subsets, PAMAP2-Protocol and PAMAP2-Optional, that differ in the number of activities. Jordao et al. [2018] use PAMAP2-Protocol, which has 12 activities, in their experiments due to the protocol applied.

MHEALTH [Baños et al., 2014] This dataset consists of electrocardiogram, accelerometer, gyroscope, and magnetometer captured data at 50Hz and contains 12 activities (Similar to Jordao et al. [2018], we do not use electrocardiogram data because a large number of samples of this data source is damaged).

5.1.3 Evaluation Protocols

According to Jordao et al. [2018], Leave-One-Subject-Out (LOSO) and Leave-One-Trial-Out (LOTO) are the most appropriate for reporting results in sensor-based HAR.

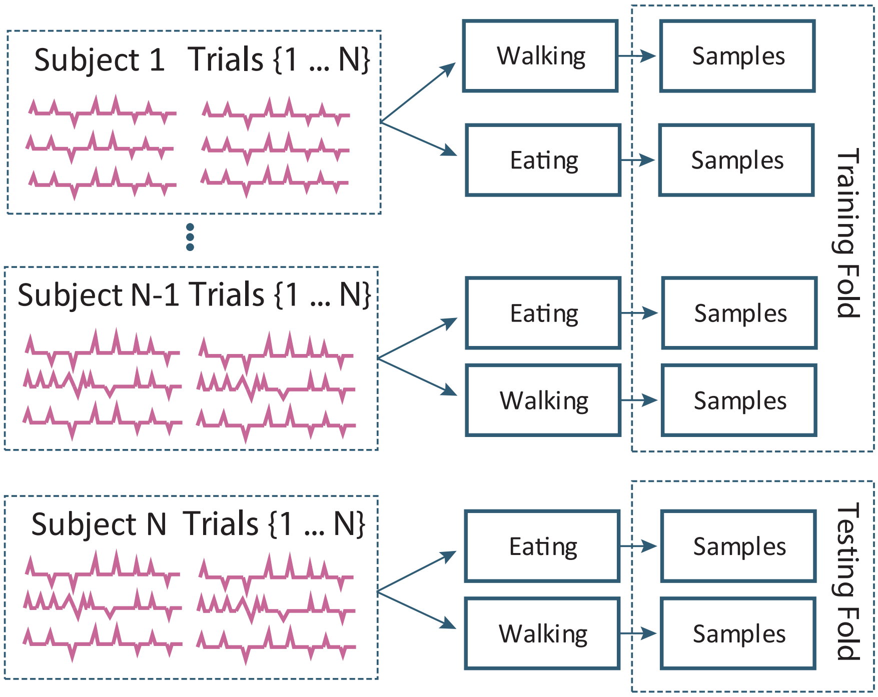

Figure 5.3: Leave One Subject Out (LOSO) protocol.

Leave-one-subject-out protocol As illustrated in Figure 5.3, in this protocol the data is separated in training and test so that the test has only one subject at a time and the training has the other subjects. LOSO represents the real scenario of applications for wearables devices, where a method is trained in known subjects and applied to new subjects later. This protocol also analyzes the generalization quality of the method since the training and test data do not have the same distribution.

Leave-one-trial-out protocol In this protocol, a trial consists of an entire execution of an activity until a transition to another activity. Thus, the data is separated into trials where each trial contains only a continuous capture of an activity. Therefore, the training is performed with all trials except one that is separated to be used during test. The LOTO protocol has the benefits of generating a large number of samples and certifying that the contents of a trial do not appear in the train and test at the same time. Therefore, different from the cross-validation protocols inappropriately used in the literature, ensuring a correct evaluation of the performance. Figure 5.4 shows a overview of this protocol.

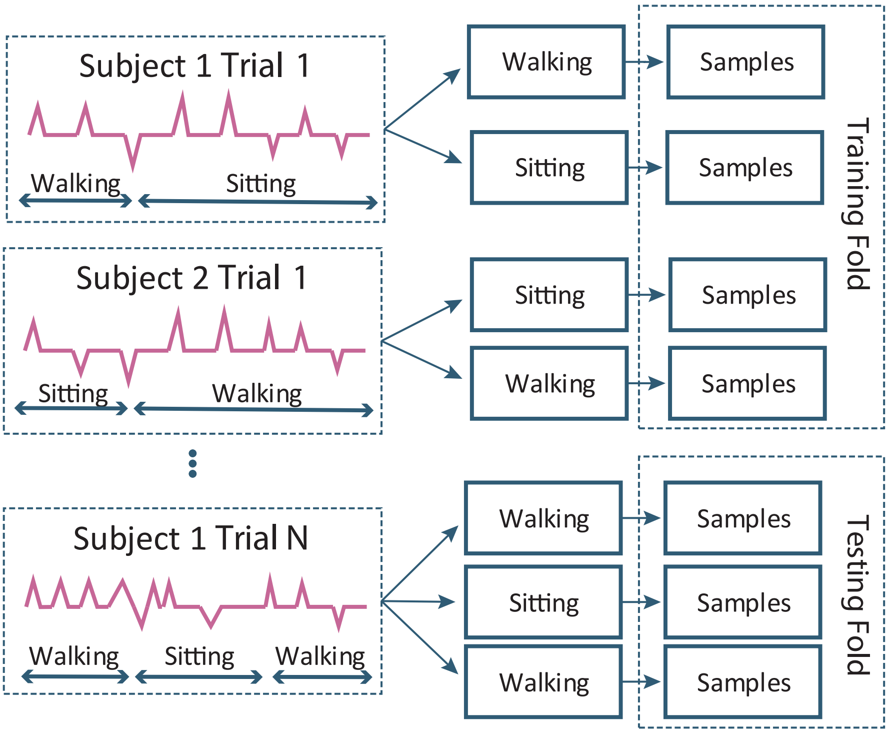

Figure 5.4: Leave One Trial Out (LOTO) protocol.

5.2 Comparison with Kernel Ensemble Baseline

To analyze the contribution of the pool of kernels and to evaluate the contribution of our DCNN ensemble, we use the Inception network module proposed by Szegedy et al. [2015] as a baseline. Although the inception was originally designed for object detection in images, it is analogous to our approach since it also applies multiple kernels to the same input to extract different pattern sizes. We could not compare our DCNN Ensemble with inception's full architecture because the available datasets to sensor-based HAR do not have enough data to train a network the size of inception (in the object detection domain the inception was trained using 1.2 million of images provided by ImageNet dataset [Deng et al., 2009], in our context the dataset with the largest number of samples used in our evaluation has 20,000 samples). One option would be to use the network pre-trained on the ImageNet and perform transfer learning. However, the pre-trained model is restricted to the use of three channels and requires a minimum array of 139x139 pixels as input. The sensor data is composed of one channel, and our largest dataset has a matrix of 500x10. Therefore, it is not possible to use the pre-trained inception network without deforming our data.

Due to the aforementioned restrictions, we performed a study of the appropriate number of inception modules that should be used for the context of wearable sensors.

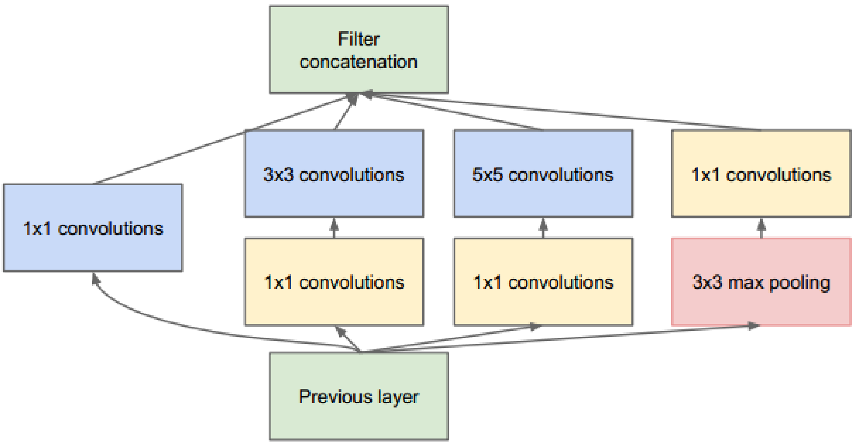

Figure 5.5: Inception module with dimension reductions. The image is from the original paper [Szegedy et al., 2015].

Our experiments showed that the addition of more than one module deteriorate the results. Therefore, all inception-based experiments in this work were executed by using only one inception module.

Another important point is that we add to the inception module the fully connected block used in our stream. This considerably increased the inception performance, since the fully connected block is capable of fusing different patterns extracted by different kernels sizes and also regularize the network since we use SELU activation function. We employed as baselines the two modules proposed by Szegedy et al. [2015]: the naïve (Figure 5.5) and the dimensionality reduction module (Figure 5.6). In addition, to evaluate our kernel pool, we adapt each type of inception module to the wearable sensors domain by using the same pool of kernels used by our approach instead of the kernels proposed in Szegedy et al. [2015].

Tables 5.2 and 5.3 shows the results obtained with the described approaches on LOTO and LOSO protocols, respectively. According to the results, the use of our pool of kernels improves the result of the inception original modules for all datasets. This support our hypothesis that extracting multiple temporal scales is appropriate for the sensor domain. Besides, our DCNN ensemble approach outperforms all four inception-based methods using LOSO and LOTO on the seven datasets, which points out that our ensemble is more suitable to employ multiple kernels to extract temporal information in the context of wearables sensors.

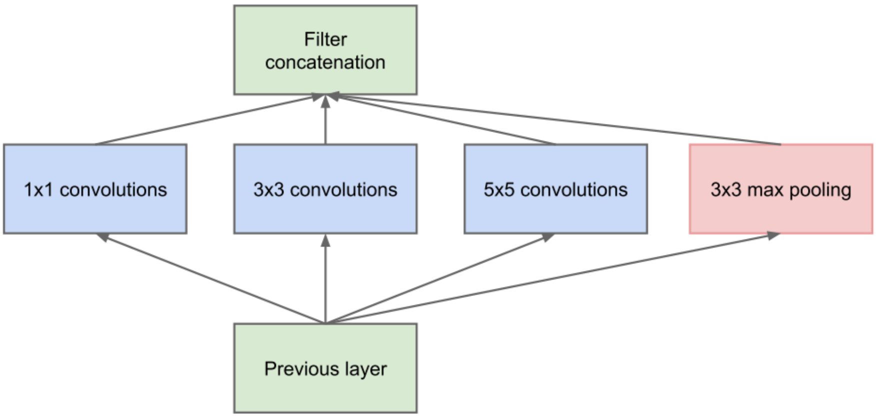

Figure 5.6: Inception module, naïve version. The image is from the original paper [Szegedy et al., 2015]

Table 5.2: Comparison of our Multimodal DCNN Ensemble (MDE) and its simplifications (DCNN Ensemble and Multimodal Stream) to the state-of-the-art architectures surveyed by Jordao et al. [2018] using LOTO on seven standardized datasets. Also, we show the results of Yao et al. [2017] and the results of two inception modules [Szegedy et al., 2015] using the original proposed kernels and our proposed pool of kernels. (A) refers to the inception naïve module and (B) refers to the inception dimensionality reduction module. Cells with the symbol ”-” denote that it is not possible to execute the method on the respective dataset, due to its architecture. Cells with the symbol ’×’ denote that we do not evaluate multimodal methods in unimodal datasets.

	LOTO (Accuracy (%))						
Methods	**WHARF**	**UTD1**	**UTD2**	**WISDM**	**USCHAD**	**MHEALTH**	**PAMA**
Kwapisz et al. [2010]	44.51	15.99	69.61	79.08	76.52	89.75	70.58
Catal et al. [2015]	64.84	47.80	81.37	80.52	87.77	91.84	81.03
Kim and Choi [2013]	61.12	50.98	75.27	56.26	85.70	91.51	78.08
Chen and Xue [2015]	72.55	-	-	86.55	84.66	89.95	82.32
Jiang and Yin [2015]	70.79	-	-	83.82	80.73	52.78	-
Ha et al. [2015]	-	-	-	-	-	85.31	80.13
Ha and Choi [2016]	-	-	-	-	-	82.75	71.19
Yao et al. [2017]	×	12.70	22.41	×	81.34	31.35	70.59
Szegedy et al. [2015] (A)	43.98	50.87	76.27	83.02	-	-	-
Szegedy et al. [2015] (A) + pool	49.86	53.06	76.71	84.89	-	-	-
Szegedy et al. [2015] (B)	51.76	52.36	74.62	79.18	-	-	-
Szegedy et al. [2015] (B) + pool	60.74	56.66	78.62	86.83	-	-	-
DCNN Ensemble (our)	**75.50**	62.03	81.63	**89.01**	88.49	**93.09**	**83.99**
Multimodal Stream (our)	×	48.90	79.82	×	85.95	83.17	79.62
MDE (our)	×	**69.61**	**83.78**	×	**90.08**	84.61	76.35

Table 5.3: Comparison of our Multimodal DCNN Ensemble (MDE) and its simplifications (DCNN Ensemble and Multimodal Stream) to the state-of-the-art architectures surveyed by Jordao et al. [2018] using LOSO on seven standardized datasets. Also, we show the results of Yao et al. [2017] and the results of two inception modules [Szegedy et al., 2015] using the original proposed kernels and our proposed pool of kernels. (A) refers to the inception naïve module and (B) refers to the inception dimensionality reduction module. Cells with the symbol "-" denote that it is not possible to execute the method on the respective dataset, due to its architecture. Cells with the symbol '×' denote that we do not evaluate multimodal methods in unimodal datasets.

	LOSO (Accuracy (%))						
Methods	**WHARF**	**UTD1**	**UTD2**	**WISDM**	**USCHAD**	**MHEALTH**	**PAMA**
Kwapisz et al. [2010]	42.19	13.04	66.67	75.31	70.15	90.41	71.27
Catal et al. [2015]	46.84	32.45	74.67	74.96	75.89	94.66	85.25
Kim and Choi [2013]	51.48	38.05	64.60	50.22	64.20	93.90	78.08
Chen and Xue [2015]	61.94	-	-	83.89	75.58	88.67	83.06
Jiang and Yin [2015]	65.35	-	-	79.97	74.88	51.46	-
Ha et al. [2015]	-	-	-	-	-	88.34	73.79
Ha and Choi [2016]	-	-	-	-	-	84.23	74.21
Yao et al. [2017]	×	11.45	22.40	×	71.52	31.88	72.61
Szegedy et al. [2015] (A)	36.64	40.71	72.55	78.64	-	-	-
Szegedy et al. [2015] (A) + pool	41.14	41.44	72.46	81.99	-	-	-
Szegedy et al. [2015] (B)	42.07	39.62	68.34	73.86	-	-	-
Szegedy et al. [2015] (B) + pool	49.97	42.23	72.96	80.99	-	-	-
DCNN Ensemble (our)	**69.79**	46.75	79.38	**86.22**	82.66	**96.27**	**87.59**
Multimodal Stream (our)	×	36.99	74.59	×	79.68	90.20	80.58
MDE (our)	×	**57.13**	**81.99**	×	**83.40**	88.97	77.70

5.3 Comparison with Multimodal Baseline

Yao et al. [2017] brought advances to sensor fusion employing multiple streams to process each sensor separately. To the best of our knowledge, that is the only multimodal method using multiple streams that have been proposed so far in the context of wearables sensors. Our multimodal stream and MDE explore this intuition. It is important to note that due to the multimodal premise of the approaches, we do not evaluate the work of Yao et al. [2017] and our multimodal approaches (MDE and Multimodal Stream) on WHARF and WISDM datasets since they consider only the accelerometer sensor.

The approach proposed by Yao et al. [2017] shows poor results both on LOTO (Table 5.2) and LOSO (Table 5.3) protocols reporting accuracy lower than very simple approaches such as handcrafted methods, in all datasets evaluated. Particularly, their approach performs poorly in UTD-MHAD family and MHEALTH datasets. We believe this is because the network proposed by Yao et al. [2017] has a very complex structure which can cause overfitting since these datasets do not have a large number of samples. In addition, in the datasets of the UTD-MHAD family, the sample size does not allow it to be divided into time-steps to fed the network, which it is essential to the approach of Yao et al. [2017] since it uses recurrent neural network (RNN).

In contrast to the approach proposed by Yao et al. [2017], our method showed superior results even using only the multimodal hypothesis through our multimodal stream approach (without DCNN ensemble as explained in the beginning of this chapter). Furthermore, using the multimodal DCNN ensemble, we solve the temporality issue in an apparently more efficient way since it does not use RNNs and still is able to surpass more sophisticated approaches such as Yao et al. [2017].

5.4 State-of-the-art Comparison

Table 5.2 (LOTO protocol) and Table 5.3 (LOSO protocol) show the results of our main approach, the multimodal DCNN ensemble (MDE), and its two simplifications, the DCNN ensemble and the multimodal stream (both explained at the beginning of this chapter). Our approaches overcome the results of our two baselines (inception module [Szegedy et al., 2015] and Yao et al. [2017]), as discussed before, and all methods of the literature surveyed by Jordao et al. [2018]. Our method, achieves to the best of our knowledge, the state-of-the-art in the seven datasets evaluated. We reiterate that many efforts have been done to achieve modest improvements in HAR based on wearable sensor data, which reinforces that the Multimodal DCNN Ensemble and the DCNN Ensemble provide notable improvements.

According to the results, in the MHEALTH and PAMAP2P datasets, the DCNN ensemble shows superior results when compared to the multimodal DCNN ensemble in both protocols tested. We believe this is occurring because we had to reduce the number of parameters in the MDE network for these two datasets due to the limited computational resources available to run our experiments. Thus, we use a smaller pool of kernels and a fully connected with fewer neurons in the stream fusion block in these datasets.

5.5 Methods Convergence

Since the sensor data to recognize human activities is mostly provided by smartphones and smartwatches, the next frontier in this task is to run the methods directly on these devices. A relevant feature regarding the satisfaction of the wearable device users is the suitability of the model to the particularities of the user since there are different ways of executing the activities. The data provided by the current datasets do not statistically

Table 5.4: Average number of epochs until the methods convergence using LOSO protocol and all the seven datasets described in this chapter.

Methods	#
Chen and Xue [2015]	70.01
Jiang and Yin [2015]	131.15
Ha and Choi [2016]	48.06
Ha et al. [2015]	28.68
Yao et al. [2017]	200.0
DCNN Ensemble (ours)	20.24
Multimodal Stream (ours)	75.83
MDE (ours)	**9.28**

represent these ways. On the contrary, the dataset with the largest number of subjects is WISDM with only 36 individuals. To better fit the model with the particularity of the user, we can retrain the network in the device through interaction with the user, where he/she does some activity and annotate (ground truth) the activity performed. This allows the system to adjust to the particularities of the user. Therefore, it is crucial that systems run fast and with high performance, since the user wants a quick and reliable response.

We evaluate the number of epochs until the convergence of the methods since the time spend at training is directly linked to the number of epochs used to train a network. It is important to empathize that the number of epochs is not an integer because the value refers to the average of the methods in all datasets using LOSO protocol. Table 5.4 shows the results of this evaluation. Also it is important to remind that the maximum number of epochs was set as 200, and the method stops its training when the loss function reaches a value less or equal to 0.2.

The worst method was Yao et al. [2017] that did not converge within the established maximum of 200 epochs and so had stopped the training. The method of the literature that performed better on this experiment was Ha et al. [2015]. They take 28.68 epochs to converge. We evaluate our main approach as well as its two simplifications. The version using only multi streams performed poorly compared to our approach and the best performance of the literature. The version using only the DCNN ensemble performed superiorly to the literature by 8.44 percentage points. Our best result (and by far the best result of the entire experiment) was for the MDE method which takes only 9.28 epochs to train its whole network. Therefore, our MDE approach is 3× faster than the best placed method of the literature, Ha et al. [2015], and 10×

faster than the average number of epochs in the previous state-of-the-art deep learning methods to sensor-based human activity recognition.

Chapter 6

Conclusions and Future Works

In this work, we proposed a multiscale ensemble-based approach of deep convolutional neural networks to address sensor-based human activity recognition (HAR). Our approach is able to learn individually the features of each sensor before performing the fusion and to model multiple temporal scales of an activity sequence. We demonstrate its suitability for HAR on wearable sensor data by performing an evaluation on seven important datasets. Our approach outperforms previous state-of-the-art results and an Inception module network adaptation used as a baseline to our convolutional kernel ensemble premise. We demonstrate that our approach works directly on the raw sensor data, with no pre-processing, which makes it general and minimizes engineering bias. As future work, we intend to study a dynamically way to choose the kernels employed in the ensemble.

Bibliography

Baños, O., García, R., Holgado-Terriza, J. A., Damas, M., Pomares, H., Ruiz, I. R., Saez, A., and Villalonga, C. (2014). mhealthdroid: A novel framework for agile development of mobile health applications. In *IWAAL*, pages 91--98.

Bruno, B., Mastrogiovanni, F., and Sgorbissa, A. (2015). Wearable Inertial Sensors: Applications, Challenges, and Public Test Benches. *in IEEE Robot. Automat. Mag.*

Catal, C., Tufekci, S., Pirmit, E., and Kocabag, G. (2015). On the use of ensemble of classifiers for accelerometer-based activity recognition. *in Applied Soft Computing.*

Chen, C., Jafari, R., and Kehtarnavaz, N. (2015). UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *ICIP*.

Chen, Y. and Shen, C. (2017). Performance analysis of smartphone-sensor behavior for human activity recognition. *Ieee Access*, 5:3095--3110.

Chen, Y. and Xue, Y. (2015). A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer. In *SMC*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Feng, Z., Mo, L., and Li, M. (2015). A random forest-based ensemble method for activity recognition. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 5074--5077. IEEE.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193--202. ISSN 1432-0770.

Ha, S. and Choi, S. (2016). Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors. In *IJCNN*.

Ha, S., Yun, J.-M., and Choi, S. (2015). Multi-modal convolutional neural networks for activity recognition. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pages 3017--3022. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770--778.

Jiang, W. and Yin, Z. (2015). Human activity recognition using wearable sensors by deep convolutional neural networks. In *ACM Multimedia Conference*.

Jordao, A., Kloss, R., and Schwartz, W. R. (2017). Latent hypernet: Exploring all layers from convolutional neural networks. *arXiv preprint arXiv:1711.02652*.

Jordao, A., Nazare Jr, A. C., Sena, J., and Schwartz, W. R. (2018). Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art.

Kim, H., Kim, M., Lee, S., and Choi, Y. S. (2012). An Analysis of Eating Activities for Automatic Food Type Recognition. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2012, Hollywood, CA, USA, December 3-6, 2012*, pages 1--5.

Kim, H.-J. and Choi, Y. S. (2013). Eating activity recognition for health and wellness: A case study on asian eating style. In *Consumer Electronics (ICCE), 2013 IEEE International Conference on*, pages 446--447. IEEE.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. *CoRR*, abs/1706.02515.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097--1105.

Kuncheva, L. I. and Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181--207.

Kwapisz, J. R., Weiss, G. M., and Moore, S. (2010). Activity recognition using cell phone accelerometers. *SIGKDD Explorations*, 12(2):74--82.

Lara, O. D. and Labrador, M. A. (2013). A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192--1209.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. ISSN 0018-9219.

LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.

LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K.-R. (2012). Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9--48. Springer.

Lockhart, J. W., Weiss, G. M., Xue, J. C., Gallagher, S. T., Grosner, A. B., and Pulickal, T. T. (2011). Design considerations for the wisdm smart phone-based sensor mining architecture. In *Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data.*

Min, J.-K. and Cho, S.-B. (2011). Activity recognition based on wearable sensors using selection/fusion hybrid ensemble. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 1319--1324. IEEE.

Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169--198.

Pedamonti, D. (2018). Comparison of non-linear activation functions for deep neural networks on mnist classification task. *arXiv preprint arXiv:1804.02763.*

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21--45.

Reiss, A. and Stricker, D. (2012). Introducing a new benchmarked dataset for activity monitoring. In *ISWC.*

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1--39.

Sena, J., Santos, J. B., and Schwartz, W. R. (2018). Multiscale dcnn ensemble applied to human activity recognition based on wearable sensors. In *Signal Processing Conference (EUSIPCO), 2018 Proceedings of the 26th European.* IEEE.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556.*

Snoek, C. G., Worring, M., and Smeulders, A. W. (2005). Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399--402. ACM.

Sollich, P. and Krogh, A. (1996). Learning with ensembles: How overfitting can be useful. In *Advances in neural information processing systems*, pages 190--196.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1--9.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5:241--259.

Yao, S., Hu, S., Zhao, Y., Zhang, A., and Abdelzaher, T. (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*, pages 351--360. International World Wide Web Conferences Steering Committee.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818--833. Springer.

Zhang, M. and Sawchuk, A. A. (2012). Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In *UbiComp*.