FROM BODY TO BRAIN: USING ARTIFICIAL INTELLIGENCE TO IDENTIFY USER SKILLS & INTENTIONS IN INTERACTIVE SCENARIOS

by

MICHALIS PAPAKOSTAS

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington, in the context of the joint Ph.D. program with the National Center for Scientific Research "Demokritos", in Partial Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2019

To my family and my beloved childhood friends for being always by my side.

ACKNOWLEDGEMENTS

Firstly I would like to express my sincerest gratitude to my supervisors Prof. Fillia Makedon and Dr. Vangelis Karkaletsis for believing in me and for their continuous and unconditional support in my research. Their intuition and guidance have been invaluable towards completing my Ph.D.

In addition, I would like to thank my colleagues and labmates at the Heracleia Lab, UTA and SKEL Lab, NCSR, for the numerous fruitful conversations and collaborations we had over the years. They all played a special role in the successful completion of this Thesis. Special thanks go to my dear friend and colleague Dr. Konstantinos Tsiakas for being a constant inspiration in and out of the Lab and to Dr. Theodoros Giannakopoulos and Prof. Vangelis Spyrou, senior researchers at NCSR Demokritos, for their scientific mentoring. Their knowledge has been an indispensable factor towards improving my research skills and becoming a better scientist. Moreover, I would like to thank Prof. Vassilis Athitsos, committee member of this Thesis and Professor at UT Arlington, for his support and valuable research insights during my Ph.D.

Lastly, I would like to thank my parents Kostas and Eva, my uncle Demetri and my aunt Susan for being the greatest source of emotional support and encouragement I could ever ask for.

April 9, 2019

ABSTRACT

FROM BODY TO BRAIN: USING ARTIFICIAL INTELLIGENCE TO IDENTIFY USER SKILLS & INTENTIONS IN INTERACTIVE SCENARIOS

MICHALIS PAPAKOSTAS, Ph.D.

The University of Texas at Arlington, 2019

Supervising Professor: Fillia Makedon

Artificial Intelligence has probably been the most rapidly evolving field of science during the last decade. Its numerous real-life applications have radically altered the way we experience daily-living with great impact in some of the most basic aspects of human lives including but not limited to health and well-being, communication and interaction, education, driving, daily, and entertainment.

Human-Computer Interaction (HCI) is the field of Computer Science lying in the epicenter of this evolution and is responsible for transforming rudimentary research findings and theoretical principles into intuitive tools, responsible for enhancing human performance, increasing productivity and ensuring safety. Two of the core questions that HCI research tries to address relate to ***a) what does user want?*** and ***b) what can the user do?*** [1]

Multi-modal user monitoring has shown great potential towards answering those questions [2]. Modeling and tracking different parameters of user's behavior has provided groundbreaking solutions in several fields such as smart rehabilitation, smart driving, and workplace-safety [3, 4, 5].

Two of the dominant modalities that have been extensively deployed for such systems are speech and vision-based approaches with a special focus on activity and emotion recognition [6]. Despite the great amount of research that has been done in these domains, there are numerous other implicit and explicit types of user-feedback produced during an HCI scenario, that are very informative but have attracted very limited research interest. This is usually due to the great levels of inherent noise that such signals tend to carry, or due to the highly invasive equipment that is required to capture this kind of information. These factors make most real-life applications almost impossible to implement. [7]

This research aims to investigate the potentials of multi-modal user monitoring towards designing personalized scenarios and interactive interfaces that focus on two different research axis. Firstly we explore the advantages of reusing existing knowledge across different information domains, application areas, and individual users in an effort to create predictive models that can expand their functionalities between distinct HCI scenarios. Secondly, we try to enhance multi-modal interaction by accessing information that stems from more sophisticated and less explored sources such as Electroencephalogram (EEG) and Electromyogram (EMG) analysis using minimally invasive sensors. We achieve this by designing a series of end-to-end experiments (from data collection to analysis and application) and by performing an extensive evaluation on various Machine Learning (ML) and Deep-Learning (DL) approaches on their ability to model diverge signals of interaction. As an outcome of this in-depth investigation and experimentation, we propose CogBeacon. A multi-modal dataset and data-collection platform, to our knowledge the first of its kind, towards predicting events of cognitive fatigue and understanding its impact on human performance.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

TECHNOLOGY AS A TOOL TO UNDERSTAND HUMAN BEHAVIOR

1.1 Introduction

Several research communities around the world are focusing towards a deeper understanding of how and why humans behave, act and make decisions the way they do. From psychology to neuroscience and HCI research, there is an increased need to decompose complex human behaviors in an effort to understand previously unknown secrets of the human brain and mind. Human-Computer Interaction and modern medicine are two fields that can be directly benefited by the design of novel approaches towards tackling problems that seemed mysteries in the past [9, 10, 11]. Recent advances in sensory technology along with novel methods in multi-modal data acquisition and analysis have lately revealed new horizons of how researchers perceive and approach such vague and challenging questions [12, 13].

However, as very accurately highlighted by the authors in [14], the most critical obstacle in understanding human behavior, lies in our inability to systematically monitor and interpret the various disperse brain processes that support our reactions and trigger our natural, active, and flexibly changing behavior and cognition.

As humans, we can consider ourselves as active agents that are continuously interacting with their environment, producing and perceiving countless information at any given moment. A non-stop process that eventually affects drastically our bodily needs, our reactions and our mental desires (Figure-1.1) [15]. The goal of AI-powered User Modeling and Monitoring is to find ways to describe those complex processes in ways that can both capture uni-

versal behavioral patterns but at the same time are able to create personalized and adaptive system behaviors that match the specific skills of each individual, towards achieving a certain goal.

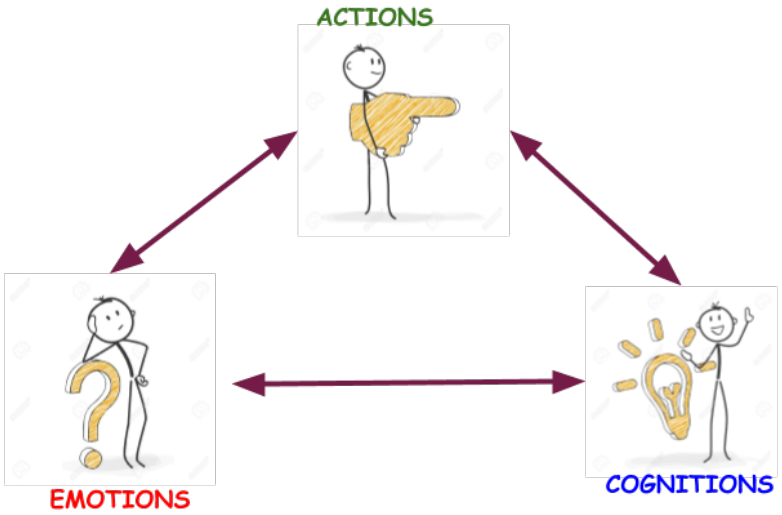


Figure 1.1. Relation between Actions, Cognition, and Emotions in Human Behavior. Actions can trigger emotions and thoughts, while at the same time they can be the result of our feelings and cognition. Their highly dependent relation and interaction are what make us able to perceive the world around us and respond to the different stimuli of our surroundings. Figure inspired by [15].

1.2 Sensor-based Analysis of Human Behavior

Understanding any kind of human behavior in-the-wild remains a far-reaching goal, despite the numerous technological advances of recent years. However, the scientific community is undoubtedly closer today to that goal than it has ever been before. Assuming a scenario with a semi-controlled environment and a clear set of possible goals, the present technology can potentially provide astonishing results on understanding human behavior, with respect to action recognition, action prediction,

emotion detection, assessment of cognitive skills and many other behavioral characteristics. The goal of such systems is to exploit this information in order to assist the user/s towards achieving their short and long term goals more efficiently than they used to do before [16]. Some example scenarios could be assisting workers in an assembly line towards improving productivity and increasing safety, monitoring patients towards a safer and more effective physical rehabilitation scenario, advancing well-being during Activities of Daily Living (ADLs), improving quality of training and education and several other similar applications that we daily face in our lives.

In the following sub-sections, we make a brief discussion of how modern sensory technology can be used as a tool to understand different aspects of human behavior towards achieving the aforementioned goals.

1.2.1 Analyzing User Actions

Capturing and analyzing user actions is probably the most scenario-dependent problem that behavioral researchers have to face, due to the vast number of possible outcomes. Narrowing down the space of target actions to be analyzed is probably the most important step of such systems. Despite the great complexity of the problem, understanding physical activities and both verbal and non-verbal interaction behaviors are two aspects that play a central role in most Intelligent Interactive Systems (IISs).

1.2.1.1 Understanding Physical Actions

Action and activity (a series of actions) recognition is probably the most popular area of behavioral modeling. Task-based action recognition [17] and activity-recognition in-the-wild [18] are both two areas that have traditionally attracted great research interest. The most dominant approaches towards recognizing human ac-

tivities have been mainly based on camera and wearable sensors (ie. gyroscope, accelerometers or EMG).

Image and video-based methods are traditionally preferred due to their non-invasive nature and their ability to track multiple targets at once. Several ML and DL approaches have been proposed over the years for camera-based activity recognition or gaze detection with the most prominent being techniques based on Conditional Random Fields (CRFs) [19, 20], SVM [21, 22] classifiers and CNN-LSTM [23, 24] combinations. While a great amount of research has been traditionally focused on action-oriented feature extractors and descriptors [25, 26, 27], lately deep learning architectures have been used as end-to-end modeling mechanisms that can automate the whole process of activity recognition and translate raw RGB or RGB-D frames to a sequence of action or activity labels [28, 29].

On the other hand, activity recognition based on wearable sensors offers a richer, more personalized and usually more accurate description about someone's motion variability, as data are captured by sensors, that are directly attached on the subject's body. However such systems ignore any kind of information that comes from the surrounding environment (ie. objects, space characteristics or other people in the room), which may eventually affect someone's behavior in terms of how or what physical actions are performed. Similarly to the case of camera-based monitoring, SVM, GMM/HMM [30, 31] and more recently CNN-LSTM[32, 33] based approaches have the lion's share when it comes to modeling information coming from wearable devices. Related research has offered over the years a great variety of different features that can be potentially applied to describe actions and activities from wearables. Despite the fact that feature extraction is a process that is usually highly dependent on the nature of the signal (ie gyroscope or EMG or accelerometer), it is very often that traditional statistical features coming from both the time and spectral domain,

are essential and in some cases enough to capture and distinguish different activities [34, 35, 36].

1.2.1.2 Understanding Actions through Audio

Speech and soundscape analysis is also a core source of information towards understating human behavior [37, 38]. Analyzing user-speech and environmental-audio patterns through microphones can provide a great variety of behavior-related information able to depict different aspects of user's intentions [39, 40]. Such intentions may be expressed either explicitly, in the form of voice commands, or implicitly by analyzing the context of the incoming audio signal for patterns that can be highly correlated to specific events (ie. a snoring sound is very likely to imply that someone is sleeping) [41, 42]. Speech and audio processing along with physical-action recognition are the two areas that have dominated the field of computational behavioral research and on many occasions they are highly correlated. Computationally wise, physical and audio based action recognition have traditionally used similar modeling and feature extraction methods, by tailoring them accordingly based on the special characteristics of each modality. However, the question of how audio-extracted features can be correlated to specific physical activities is still a highly unexplored and very challenging topic [43, 44].

1.2.2 Analyzing User Emotion

Emotion detection and recognition have become lately one of the hottest topics in the field of behavioral analytics. As discussed at the beginning of this chapter, actions, emotions, and cognitions are the three different faces of human behavior and are highly interconnected. Emotions can be responsible to trigger specific actions (ie. I started crying because I was feeling sad) but in the same time can be the result of our

own actions or actions made by others (ie. playing my guitar makes me feel happy). However, recognizing emotion accurately remains an extremely challenging problem due to the great variability that is observed across different subjects when expressing the same emotions and the subjective ground that emotions are based on by definition. Several approaches have been proposed for emotion detection, with the most popular ones stemming again from the fields of audio [45, 46, 47] and image/video processing through facial expression and body-posture analysis [48, 49]. Other proposed technologies for emotion recognition include facial electromyographic activity (fEMG) [50], monitoring arousal using ECG [51], galvanic skin response (GSR) [52], respiration sensors [53] or EEG based approaches [54]. Despite the great variability of available methods that have shown promising results, the latter ones have attracted less research attention due to their higher levels of uncertainty and because of the invasive nature that is demanded by the required sensory technology.

1.2.3 Analyzing Cognitive Skills

Cognitions describe our mental ability to acquire, process and apply new or pre-existing knowledge as well as our skill to learn through experience and senses. It comprises several intellectual functionalities such as attention, engagement, long and short-term memory, cognitive flexibility and task-switching ability, problem-solving and decision making skills and many others. Interpreting cognitive functionalities in-the-wild is probably the most challenging topic in the multidisciplinary field of behavioral sciences [55]. In addition, cognitive development and plasticity heavily depend on our bodies with which, we experience and interact with our surroundings; a term known as embodied-cognition [56]. Assessing and observing cognitive behavior demands long-term assessment and monitoring of individuals and usually takes place through the analysis of specific performance-related metrics in controlled

task-based cognitive tests, similar to those offered by the NIH toolbox or the PsyToolkit library. Some popular cognitive tests are the Wisconsin Card Sorting Test that evaluates cognitive flexibility and reasoning, the Flanker Inhibitory Control and Attention Test that examines executive function and attention, the Stroop Task that evaluates cognitive performance with respect to the Stroop-effect [57], the N-Back Task for working memory and working memory capacity and others. All such tests are very popular in clinical and experimental psychology [58, 59].

Measuring and modeling the effect of cognitive functioning through sensory technology has been usually addressed through EEG analysis and task-based performance measures [60, 61]. Other technologies include vision-based systems for embodied-cognition analytics [4, 62], GSR sensors [63] and camera or Electrooculography (EOG) based eye-tracking technologies [64] . However due to the complicated nature of the problem single based modality approaches usually are sub-optimal to create generalized behavioral profiles and multi-modal monitoring has been extensively applied for a richer and more robust description of the different cognitive phenomena that are monitored [65, 66].

1.3 Data Limitations & the Need to Learn from Others

As it is probably clear by now, the process of capturing, analyzing and eventually describing and explaining patterns of human behavior is nonetheless a complicated and non-trivial procedure. Different behaviors can be captured using various combinations of sensors and modalities, which in their turn come with their own set of advantages and disadvantages. Hence, making any different scenario a problem of deciding upon an optimal trade-off between sensor intrusiveness and information quality towards achieving the preferred results. Moreover, different target groups usually have different needs and express divergent behaviors under the same condi-

tions. In addition, most controlled experiments inevitably make assumptions that limit the applicability and generalizability of the underlying theories under real-life conditions. In-Lab data collections often tend to introduce biases that eventually hinder the efforts towards creating universally accepted systems for human behavior and biometrics monitoring [67, 68]. But even when the available data are sufficient to robustly represent the targeted problem, the need for collecting and annotating physiological and behavioral data is a very tedious, expensive and time-consuming process. A process that in many cases demands the expertise and know-how of experts from fields out of the broader area of computer science, who may not always be available.

Based on the aforementioned observations, there is an increased need towards designing computational methods that can reuse information and knowledge and apply it across different tasks, users or modalities for the purposes of building more robust and generative HCI designs. Learning behavioral patterns between different actions, emotions, and cognitions and deciphering how these patterns are expressed across individuals or target groups is a key problem of today's research in computational behavioral sciences and HCI and a central issue that this thesis tries to address.

1.4 Motivation & Thesis Outline

In this chapter, we discussed the benefits, potentials, and obstacles of using multi-sensing data to analyze and understand different aspects of human behavior. We presented an elaborate and systematic literature review on what is considered to be the most prominent technology towards capturing different behavioral patterns. Additionally, we tried to address the various parameters that need to be taken into account when designing HCI systems that depend on human behavior. Our guidelines on making such decisions are dictated by the fundamental relationship between

actions, emotions and cognitive processes and the different application domains that these technologies can be applied.

As wearable technology evolves so rapidly, sensors become less invasive and Machine Learning algorithms grow more powerful, it comes as a natural consequence to drive computational behavioral research towards designing more universally accepted designs. Architectures that can generalize, adapt and re-tailor their functionalities more intuitively are becoming essential to serve the growing demands of modern HCI systems. Methods able to be refined at a computational level to facilitate similar goals under varying environmental parameters are gradually becoming the center of attention [69, 70, 71]. Such technological advances could play a crucial role on enhancing the outcomes of interaction between humans and machines and would help us tackle more complex questions such as the effects of cognitive and physical fatigue on human performance [72].

Thus, the major research questions that rise, relate to what extent recent advances in Machine Learning and AI technology can support such generative applications. How feasible is to create robust system-behaviors that are capable of retaining high levels of personalization and keep user's active and engaged in the interaction-loop by ensuring safety and high quality in terms of performance and accuracy? Also, what are the data-demands in such systems and how possible it is to artificially generate the required information in order to meet the special needs of different machine learning algorithms? All these concerns are taken into account and examined throughout this Thesis by a set of different HCI, application-based evaluations of different problems varying from physical activity recognition to EEG modeling and analysis for Brain-Computer Interfaces (BCIs).

The rest of this Thesis is outlined as follows. In Chapter-2 we make an in-depth discussion on the principles of designing multi-modal interaction systems and

we evaluate different ML approaches on their ability to model various types of sensory information. In Chapter-3 we evaluate the capabilities of Convolutional Neural Networks to model and generalize across alternative types of modalities and tasks by transferring knowledge between information domains and we discuss how data-augmentation can potentially be the key to overcome data limitations and provide better modeling solutions. In Chapter-5 we introduce the concept of Brain-Computer Interfaces and we discuss a novel approach of how EEG signals can be exploited towards predicting user performance through implicit feedback. Chapter-4 discusses the concept of user fatigue, explains its implications in human behavior, health and performance and presents a modern approach of how subjective and objective measures can be combined towards predicting physical fatigue and designing interactive rehabilitation scenarios. In Chapter-6 we present CogBeacon; to our knowledge, the first multi-modal dataset specifically designed to address cognitive fatigue and we show an in-depth analysis of the collected data. Finally in Chapter-7 we summarize our findings, highlight the takeaways of this research and provide suggestions for future directions in the area of computational behavioral modeling.

CHAPTER 2

MULTI-MODAL USER MONITORING

2.1 Introduction

In Chapter-1 we discussed the individual components that compose human behavior, namely: actions, thoughts, and emotions. We highlighted the heavy dependence that is observed between them and reviewed the most dominant technologies in terms of sensors that are used to capture, monitor and analyze individual patterns across those three components. However, an obvious remark that can be made through this analysis, is that understanding more complex human behaviors demands our ability to track simultaneously multiple interaction signals and monitor parameters that do not always have an apparent correlation [73, 74].

Consequently, raises the need to design multi-modal interfaces that capture and track multitudinous signals and use advanced machine learning and statistical methods to extract interdependent patterns of behavior. The foundations of designing multi-modal systems for human-behavior monitoring were set almost a decade ago and are still drawing the guidelines of how to approach such problems [2, 75, 76]. Nonetheless, as a constantly evolving field of research, several novel ideas have been proposed, that aim to refine aspects of the system design process, without though deviating significantly from the traditional principles [77, 78, 79].

In the following sections, we go through the most popular theoretical models of combining multi-sensing data and we show how these models can be applied in two proof-of-concept scenarios for behavioral modeling. In the first experiment, we discuss a multi-modal architecture designed for monitoring sleeping behavior and

thereinafter we show a multi-sensing implementation for analyzing physical exercises, for fitness monitoring purposes [17, 34]. The goal of these experiments is to investigate how multi-modal monitoring can enhance the outcomes of the interaction and how flexible are such methods towards describing universal patterns of behavior across different users using minimally or non-invasive sensors.

2.2 Modeling Complex Behaviors and Fusing Multi-Modal Data

Multi-modal system design is not a trivial task, given the multiple alternative combinations of input and output channels that a system can potentially support. Depending on the nature of our application and the state on which the design process stands, there are different decisions that need to be taken with respect to what interaction modalities must be considered and how they must be combined. Two of the most popular proposals on how these decisions can be taken are given by Coutaz et al. [80] and Nigay et al. [81] with the CARE and CASE multi-modal design frameworks respectively. The CARE model aims to define how different modalities must be potentially combined from a high-level perspective that aims to capture different user behaviors as they appear on the physical world. On the other hand, CASE addresses the different possibilities of fusing individual modalities from a low-level perspective, when these modalities are seen as feature vector representations. In the following subsections, we give a more in-depth description of the design principles described by the two frameworks.

2.2.1 The CARE Model

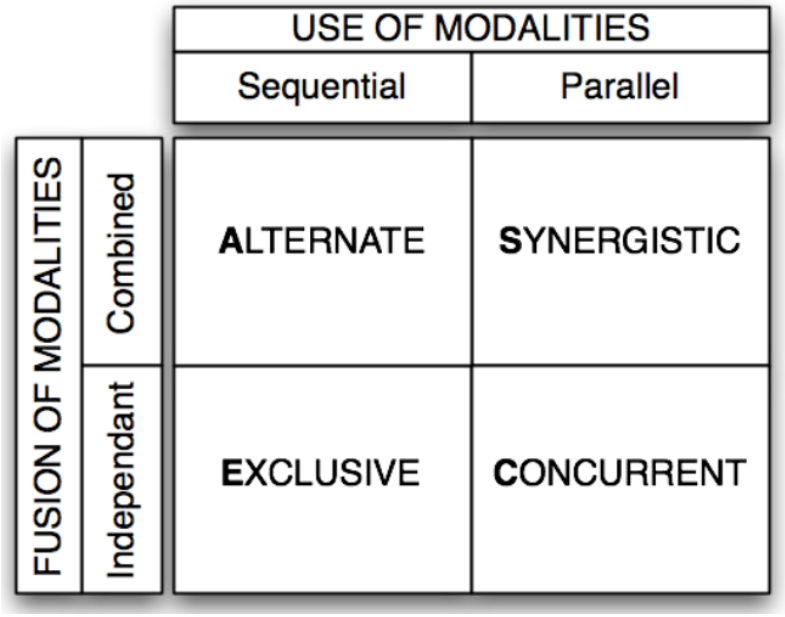
The CARE model focuses on the user-machine interaction level. The framework introduces four different properties, which are **Complementarity**, **Assignment**, **Redundancy** and **Equivalence**. *Complementarity* addresses the need for

using multiple complementary modalities in order to grasp the desired meaning. For example, understanding the meaning of the phrase "put that there" would require both pointing gestures and voice recognition technology in order to be resolved. *Assignment* indicates that only one modality can lead to the desired meaning, ie. the steering wheel of a car is the only way to direct the car. *Redundancy* implies multiple modalities which, even if used simultaneously, can be also used individually to lead to the desired meaning. For example, a user utters a "play" speech command and pushes a button labeled "play". Eventually, only one "play" command would be taken into account and the two modalities can be considered redundant. In specific HCI scenarios (ie. in applications related to assistive technologies), having redundant modalities can potentially enhance interaction outcomes as it provides a wider spectrum of options to the user. Finally, *Equivalence* entails multiple modalities that can all lead to the desired meaning, but only one would be used at a time, ie. speech or keyboard can be used to write a text and they cannot co-exist simultaneously.

2.2.2 The CASE Model

The CASE model also introduces four design properties namely: **Concurrent**, **Alternate**, **Synergistic** and **Exclusive** designs. Each of those four properties reflects a different way of scheming multi-modal systems based on two main factors: a) how the modalities are fused and b) how the modalities are captured and activated. *Synergistic Design* represents architectures that capture in parallel multiple modalities and those modalities are processed jointly using either early or late fusion modeling approaches. *Concurrent Design* on the other hand, differs in the fact that even though the different modalities are captured in parallel processing takes place independently. This approach has the advantage that different modalities can be potentially used to detect different actions or behaviors but usually increases the

computational demands of the system. *Exclusive Design*, describes systems that use different modalities in different parts of the interaction and process these modalities independently. Exclusive design assumes that the various modalities function sequentially and don't coexist as in the case of Concurrent architectures. Lastly, *Alternate Design* aims to describe systems that capture the multiple modalities sequentially as in the previous case but processing happens jointly and takes place at the end of the data collection. Figure-2.2.2, illustrates the four scenarios described above.



The CASE Models: The CASE Model. Figure by [2].

2.3 Applications based on Multi-Modal User Modeling & Monitoring

Inspired by the aforementioned analysis and design principles we proposed two AI-driven systems that take advantage of multi-modal interaction and target different application areas in the domains of health and well-being. The two frameworks

follow the Synergistic and Concurrent designs as described by the CASE model in the previous section. Our evaluation aims to explore, to what extent traditional machine learning approaches can be used towards capturing more complex interaction patterns using multiple sensors. In addition, we focus on testing the ability of such methods to maintain high levels of performance across different users when the available data are limited and how different combinations of modalities can alter the quality of the final results. Both experiments were performed in an end-to-end fashion (ie. from data collection, to analysis and application design) using the facilities offered by the Heracleia- Human-Centered Computing Lab at UTA and National Center of Scientific Research, "Demokritos".

2.3.1 Monitoring Breathing Activity and Sleep Patterns Using Multi-Modal Non-Invasive Technologies

The proposed system uses a combination of non-invasive sensors to assess and report sleep patterns and breathing activity. A contact-based pressure mattress and a non-contact regular RGB camera. To evaluate our system, we used real data collected in Heracleia Labs assistive living apartment. Our system uses Machine Learning and Computer Vision techniques to automatically analyze the collected data, recognize sleep patterns and track breathing behavior. It is non-invasive, as it does not disrupt the users usual sleeping behavior and it can be at the clinic and at home with minimal cost. Going one step beyond, we developed a mobile application for visualizing the analyzed data and monitor the patients sleep status remotely

2.3.1.1 Experimental Setup

For applying our experiments we used two different types of sensory input: a) a mattress that measures pressure and b) a regular webcam.

The FSA bed mat system produced by Vista Medical Ltd provides a 1920mm×762mm sensing area which contains an array of 64×27 pressure sensors. Each of the sensors provides a measurement in the range 0 to 100 mmHg with a scan frequency up to 5 Hz. The measurements can be recorded and manually annotated over a period of time and can be exported as a set of time-stamped vectors containing the values of each of the 1728 pressure sensors for each time stamp.

The webcam resolution was 1080x720 and it was placed on the side of the bed and a few inches above its surface.

The two sensors were capturing data in parallel but processing took place independently using different computational mechanisms.

2.3.1.2 Sleep Posture Recognition

We implemented the a sleep posture recognition method using the data stream provided by the pressure mat. The output of the pressure mat is a vector with 1728 features. Each feature represents the output value of each pressure sensor. Thus, each feature has a value between 0 and 100. Since the nature of the raw data (range of values) is similar to a gray-scale image we handle this problem as an image-processing problem. Each of the sensors can be considered as a pixel of a gray-scale image with an intensity ranging from 1 to 100. Thus each frame can be considered as a 64 x 27 pixel image.

For experimentation purposes, we collected data from 5 different individuals. All individuals were of average weight and height. Each subject lied on the mat for about 5 minutes , simulating different sleep patterns. In total five different sleeping patterns were simulated. In particular, we recognized if subjects are (i) lying on their back, (ii) on their stomach, (iii) left side or (iv) right side and (v) if they were

just sitting on the bed. In Figure-2.2 we show a visualization of the pressure values captured in one data-frame for each different posture.

For classification, we separated the data into 5 equal subsets, where each subset was related to a specific subject. Each subset contained 100 feature vectors (pressure mat scans) to represent each class (5 classes in total, each related to a different posture). Thus each subset had 500 training samples, which represented all the classes. We used 4 out of the 5 subsets to train our model (ie. 400 samples per class and 400*5=2000 training samples in total) and the 5th subset (500 test samples in total - 100 samples from each class) was used to evaluate our system. For a more efficient evaluation we performed 5-fold cross-validation. We used PCA to reduce the data dimensions from 1728 to 20 features and we tested two classification approaches using KNN-1 and 1vsAll SVM using a linear kernel. Table- 2.3.1.2 illustrates the classification accuracy percentages for each class, while Table-2.3.1.2 displays the average accuracy for each different classifier.

	Back	Stomach	Right Side	Let Side	Sitting
SVM	66.4	100	60	80	100
KNN-1	63.2	82.6	60	74.8	80

Table 2.1. Classification Accuracy per Class (%)

	Average Accuracy
SVM	81.28
KNN-1	72.12

Table 2.2. Average Classification Accuracy (%)

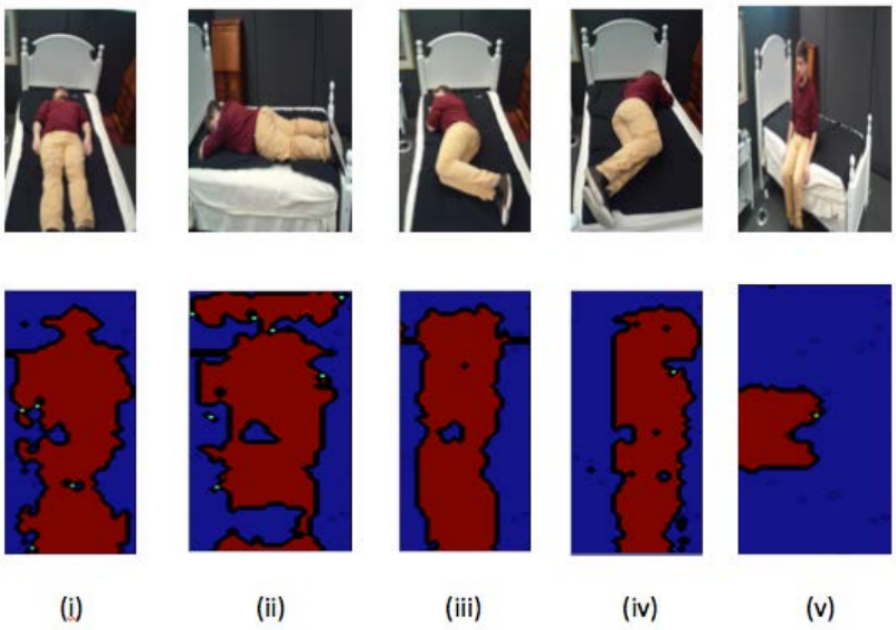


Figure 2.2. Images on the top display the actual posture while bottom images illustrate the visualization of the output from the pressure mat.

The classification results come to validate previous findings, which also claim that SVM based classification can lead to more accurate results. However, during the experimentation, we realized that efficiency is highly depended on the body proportions of each subject. Moreover, as the subject change postures, the pressure mat produces inevitably noisy measures, which complicate our problem. Thus, the use of additional sensors as proposed in [82] may lead to higher and more robust accuracy [82, 83].

2.3.1.3 Monitoring Breathing Patterns

To monitor the breathing patterns during sleep we developed a system that combines the data taken from a regular webcam and the output stream of the pressure mattress. We applied standard computer-vision techniques to monitor in real-time

the movement of the chest as the subject breaths combined with a simple offline analysis of the pressure mat data to eventually characterize breathing activity.

2.3.1.3.1 Monitoring Chest Movement

To monitor the chest movement we developed a motion tracking algorithm using the frame difference technique (equation-2.1).

$$\begin{cases} Diff_i = I_{k+2} - I_k \\ Diff_{i+1} = I_{k+4} - I_{k+2} \end{cases} \tag{2.1}$$

A webcam was placed on the left side of the bed and a few inches above its surface. The camera was placed properly to capture only the region of the subjects chest. Using this simple approach we were able to detect quite accurately the motion in the area of the chest while the subject was breathing.

Any other motion except, the movement of the chest, was considered as noise and thus, was not taken into account. To prevent our system from getting affected by such noise we applied a hysteresis threshold to cut off any other transient movement (such as a change of posture, an unexpected hand movement, etc.)(Figure-2.5). The camera captures about 15 frames per second, however, for better motion detection, only every second frame is being processed (6 to 7 FPS). Under regular breathing conditions, movement was detected every 0.3 seconds. If motion occurs constantly in 15 sequential processed frames (around 2.5 seconds), in the certain region of the frame, the system locks in this area and the motion tracking begins. Since breathing can be considered as an almost periodic event, the system apparently locks and tracks only the area of the chest where consistent movement occurs. If no motion is detected for a certain amount of time, which was set to 20 processed frames (about 4 seconds) the systems forgets the locked area, which was tracking and unlocks from the target.

In such a case we have to consider the possibility that our subject had a breathing failure.

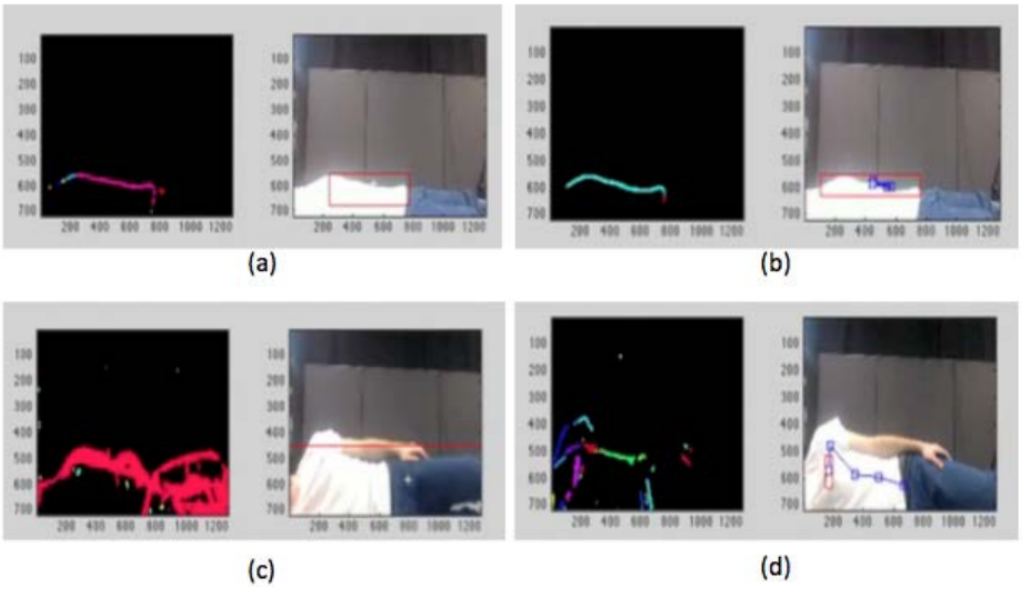


Figure 2.3. (a) Motion has been detected - system has not lock to a target yet, (b) System locked to the target after detecting movement in 15 sequential frames, (c) Unexpected motion occurred due to change of posture - motion detected but not tracked, (d) System re-locked to the chest area when breathing pattern detected.

2.3.1.4 Measuring Motion Level

Additionally to the vision-based breathing monitoring technique, we used the pressure mattress device to monitor the levels of motion over time. Our assumption was, that as the subject breathes, greater fluctuations to the output stream of the sensors that lie under the chest could be observed. The subjects were asked to lie on their back, remain still and just breath. They were also encouraged to take both regular and deeper breaths. We collected data for 4 minutes.

The pressure pad provides 1 data-frame every 0.25 seconds. Hence, we could collect 4 different measures from each pressure sensor every second. To analyze the output stream and measure the levels of motion over time, we subtracted from each data-frame the average of the 10 previous frames (equation-2.2). Then for each data-frame, we summed all the pressure values. Thus, every data-frame was represented by a single value (equation-2.3). For the 4 minutes of experimentation, we collected in total 803 such values. Each of these values can be considered as an indicator of the pressure applied to the pressure mat at each moment. Before visualizing the data we applied a Gaussian filter to smooth the curve.

$$F'_{i+10} = F_{i+10} - mean\{F_i, F_{i+1}, ..., F_{i+9}\} \tag{2.2}$$

$$F'' = \sum_{1728}^{K=1} v_k \tag{2.3}$$

In the graph of Figure-2.5 we can observe periodic-like peaks of different amplitude, which refer to different amounts of pressure applied to the pressure mat over time. Since the subjects remained motionless it is safe to assume that these variations on the pressure, indicate the motion caused by the breathing activity. Higher amplitude can be translated as deep breathing while low amplitude combined with low frequency may be an indicator for respiratory problems or even breathing inactivity.

2.3.1.5 Mobile Application

In order to relay the information that is collected by the sleep monitoring system, the data must be aggregated and then displayed to the person who is responsible for monitoring the sleep patterns. This task was achieved through a mobile application interface on the Android and iOS operating systems using various visualization

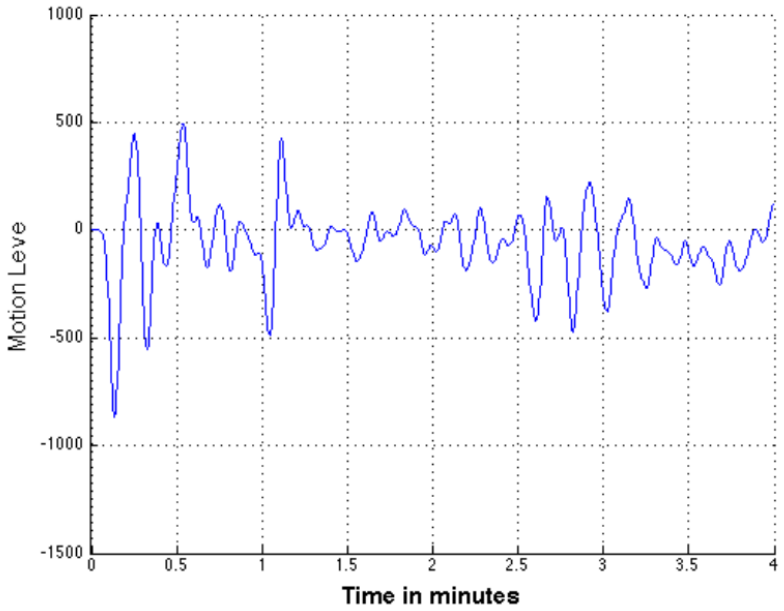


Figure 2.4. Motion level over time.

techniques such as two dimensional area charts, line charts, and pie charts in order to enhance the quick retrieval and comprehension of the patterns throughout the entire sleep cycle [84]. A donut chart was used to represent the breathing averages for each hour during sleep and then display the smallest average in the center of the donut to highlight the problem.

Alerts are known to be a useful feature for monitoring technologies in order to prevent injuries and avoid high-risk situations. Therefore, we implemented an "Alert" feature that allows the app to display any alerts provided by the system with respect to abnormal sleeping behaviors throughout the sleep cycle.

There are three main fragments inside the application: a) A dashboard including all of the data visualizations, b) an "Alerts" page showing all of the alerts sent from the monitoring system and c) a settings page to toggle the network data collection. The data is populated dynamically from a middle-ware web server that aggregates

the data and then makes it available to the mobile application through a RESTful API. The middle-ware server collects the data from the monitoring application and stores it in a database. The data aggregation is performed on the web server and it includes the following: Taking the breathing patterns and averaging them over an hour, day, and month, averaging the pressure on the mattress, and averaging the pressure averages over an hour, day, and month.

Figure 2.5. (i) Android Version, (ii) iOS Version.

2.3.1.6 Takeaways

We proposed an approach for a robust, non-invasive, multi-modal sleep monitoring system using low-cost technology. Our work comes to extend previous findings and techniques in sleep posture recognition by providing an additional functionality for breath monitoring and a user-friendly way for data visualization [82]. Our findings prove that breathing patterns can be monitored and identified using simple modeling methods and possible respiratory failures can be immediately prevented.

2.3.2 A Fitness Monitoring System based on Fusion of Visual and Sensorial Information

In this work, we present a method to recognize physical exercises in a real home environment. Towards this end, we combine sensorial information using a smartphone accelerometer, with visual information captured from a simple web camera. Low-level features inspired by the audio analysis domain are used to represent the accelerometer data, while simple frame-wise features are used in the visual channel. Extensive experiments prove that the fusion approach achieves 95% of overall performance when user calibration is adopted, which is a 4% performance boost compared to the best individual modality which is with the accelerometer sensor. The final dataset, compiled explicitly for the purposes of this project can be found online for free [1] .

2.3.2.1 Proposed Methodology

2.3.2.1.1 General

Figure-2.6 illustrates the conceptual architecture of the proposed methodology. A smartphone attached to the user's arm (using an armband) is used for accelerometer data acquisition in a high sampling rate (around 1KHz). In total, almost 3000 samples per second are acquired through the accelerometer sensor (1000 for each axis). These three temporal sequences of data are then analyzed in a short-term basis and long-term statistics are extracted on the resulting short-term feature vectors, following a rationale that is similar to feature extraction for audio classification and segmentation tasks. This process results in a feature vector of N_a elements (features) that characterizes the whole recording (exercising session).

[1] https://sites.google.com/view/michalis-papakostas/datasets

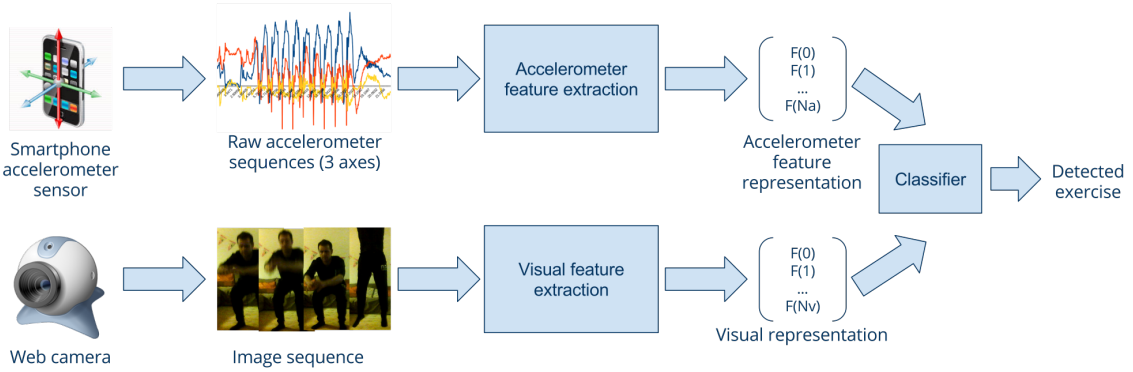


Figure 2.6. Conceptual architecture of the proposed methodology. Accelerometer data are gathered through a smartphone device attached to the user's arm. Temporal features are then computed and long-term statistics are also extracted for each recording session. This leads to a feature vector that represents the whole recording. Similarly, a feature vector of visual features computed over an aggregated image that stems from a video sequence is used to represent the visual domain. An SVM classifier is used to discriminate between the different activity classes, either based on a single-modality or on both modalities. .

In parallel, a web camera is recording the exercising session, as an alternative (or complementary) way to recognize the respective fitness activity. In that case, the raw information is a sequence of color images (video), while the feature extraction stage results in a 2D representation that visualizes the aggregated movements along the whole recording session. Again, this process leads to a feature vector of N_v values that characterizes the whole recording. A supervised classifier is then used to classify the feature vectors, either individually, or in a fusion mode, in order to extract the final classification decision.

In the following paragraphs, we describe the signal representation techniques, for both modalities, along with the respective classification approaches. Finally, the dataset adopted to evaluate the method is described, along with the experimental

results. Focus is given on the proposed feature extraction approach with regards to the accelerometer signal.

2.3.2.1.2 Signal representation

2.3.2.1.2 Accelerometer signal representation

Raw accelerometer signals are recorded using a standard Android smartphone with a sampling rate of 100 Hz. In order to extract features that achieve high discrimination ability in the particular activity recognition task, we propose the following rationale: at a first stage, the signal is split to non-overlapping short-term windows (frames) of 250 mseconds long. For each short-term frame, the following features are calculated:

1. **Maximum Sample Value**

$$max_x \leftarrow \text{maximum value in signal x} \tag{2.4}$$

2. **Minimum Sample Value**

$$min_x \leftarrow \text{minimum value in signal x} \tag{2.5}$$

3. **Maximum Absolute Value**

$$abs_{max} = |max_x| \tag{2.6}$$

4. **Minimum Absolute Value**

$$abs_{min} = |min_x| \tag{2.7}$$

5. **Average Value**

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2.8}$$

, where x_i is the value of sample i given a signal x and N is the signal length

6. **Standard Deviation**

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1} N(x_i - \mu)^2} \tag{2.9}$$

, where x_i is the value of sample i given a signal x, N is signal length and μ is the average sample value of the signal

7. **Median Value**

$$median = \frac{1}{2}(x_{\frac{N}{2}} + x_{\frac{N}{2}+1}) \tag{2.10}$$

, where x is the set of values in the signal in ascending order and N is the signal length

8. **Zero Crossing Rate**

$$ZCR = \frac{1}{N-1}\sum_{t=1}^{N-1}(sig(x_t) - sig(x_{t-1})) \tag{2.11}$$

,where $sig(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{otherwise} \end{cases}$ and N is the length of the signal. ZCR indicates the rate of sign-changes of the signal during the duration of a particular frame.

9. **Entropy of the Energy**

$$H(E) = -\sum_{i=1}^{N} p(E) log_{10} p(E) \tag{2.12}$$

,where $E = \frac{\sum x_i^2}{Total\_E}$ are the normalised sub-frame energies, $Total\_E = \sum x_i^2$ is the total signal energy and x_i is the value of a sample within a frame or a sub-frame. This feature can be interpreted as a measure of abrupt changes.

10. **Spectral Centroid**

$$C = \sum_{i=0}^{N-1} X_i p(X_i) \tag{2.13}$$

,where N is the size of the spectrum, X are the observed frequencies and p(X) is the probability to observe a value in X. C represents the center of gravity of the spectrum.

11. **Spectral Spread**

$$S = \sqrt{\sum_{n=0}^{N-1}(X - C)^2 p(X)} \tag{2.14}$$

,where N is the size of the spectrum, X are the observed frequencies, p(X) are the probability to observe value X and C is the spectral centroid. S represents second central moment of the spectrum.

12. **Spectral Entropy**

$$H(X) = -\sum_{i=1}^{N} p(PSD) log_{10} p(PSD) \tag{2.15}$$

,where $PSD = \frac{1}{N}|X|^2$ is the Power Spectral Density (PSD) of of spectrum X, $p(PS) = \frac{PS_i}{\sum_i PS_i}$ is the Probability Density Function of the PSD, N is the size of the spectrum and X are the observed frequencies. H(X) is the normalized spectral energies for a set of sub-frames.

13. **Spectral Flux**

$$FL_{i,i-1} = \sum (EN_i - EN_{i-1})^2 \tag{2.16}$$

, where $EN_i = \frac{X_i}{\sum X_i}$ is the normalized Discrete Fourier Coefficient at the $i_t h$ frame and X is the spectral of the signal. Thus, spectral flux is the squared difference between the normalized magnitudes of the spectra of two successive frames.

14. **Spectral Rolloff**

$$R = 0.9 \sum_{i=0}^{N-1} |X_i| \tag{2.17}$$

, where X is the spectrum of the signal and N is the size of the positive spectrum. Spectral rollof corresponds to the frequency below which 90% of the magnitude

distribution of the spectrum is concentrated.

Some of the aforementioned features are simple statistics, while others stem from the domain of audio signal analysis (e.g. the spectral features). More details on these particular features and their use on audio representation can be found at [85]. Figure-2.7 shows an example of five short-term feature sequences, that are extracted for each of the three raw accelerometer features. Different colors correspond to different accelerometer axes.

2.3.2.1.2 Video representation

Visual modeling of an action is performed by representing a sample video with a single binary image. To extract the binary image, we apply the well established frame-difference algorithm [86] and the final aggregated image corresponds to the summation of the intermediate difference-images. At each step, every difference image is being thresholded and smoothed using a median filter of size N, where N was set to 5 in our case. A second median-smoothing filter of the same size is applied to the resulted image after every summation. In Figure-2.9 we show three indicative images as extracted from three sample-videos.

When the video-representation image has been computed, it is segmented in a grid of size 2×2 and visual features are extracted from each segment. The final feature vector that represents the image and thus, the whole video sample, is the concatenation of all the feature vectors from each grid-segment. Grid size was decided after experimentation on several dimensions as 2×2 seemed to outperform higher-resolution grids. In this work, we chose three very well established and fast to compute visual features to work with, namely:

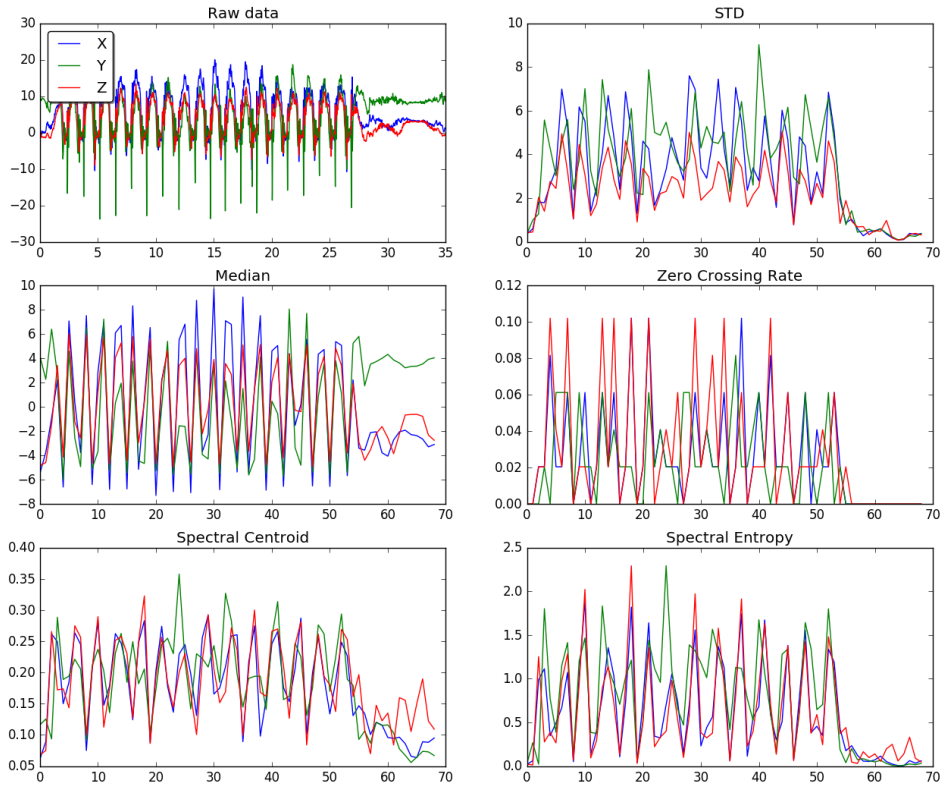


Figure 2.7. Raw accelerometer signals example along with 5 corresponding short-term feature sequences, namely the standard deviation, the median value, the zero crossing rate, the spectral centroid and the spectral entropy.

- Normalized histogram of the grayscale values
- Histogram of Oriented Gradients (HOGs)
- Local binary patterns (LPBs)

HOGs represent an object using the local distributions of intensity gradients and edge directions. They have been widely used in object and human tracking [87]. In this work, we have adopted HOGs, since they provide an efficient way to discriminate between visual objects when used in a supervised context [88]. LBPs

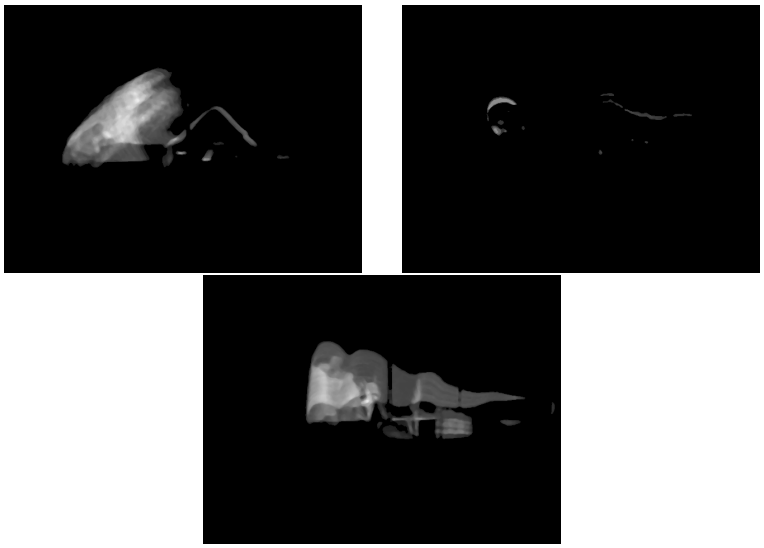


Figure 2.8. Examples of video-representation images for *"crunches"*, *"fron-plank"* and *"push-ups"* exercises respectively.

[89] form a widely used feature in modern image analysis methods. In general, LBPs encode local pixel neighborhoods using binary representations, hence their name. We have selected to adopt LBPs for their ability to represent differences in texture characteristics between images

2.3.2.2 Exercise recognition

2.3.2.2.1 Adopted classes and dataset

We have selected to adopt 8 widely-used workout exercises that can be completed in the context of a home environment. In particular, the adopted exercises are: *crunches, jumping squats, lunges, plank, push-ups, romanian squats, squats, and toe-touches.* They cover a relatively all basic body areas such as legs, abs, and chest. There are obvious confusions that are expected to occur by an automatic exercise recognition system: for example, one may expect that squats may be confused with romanian squats and/or lunges, while toe-touch is quite similar to crunches.

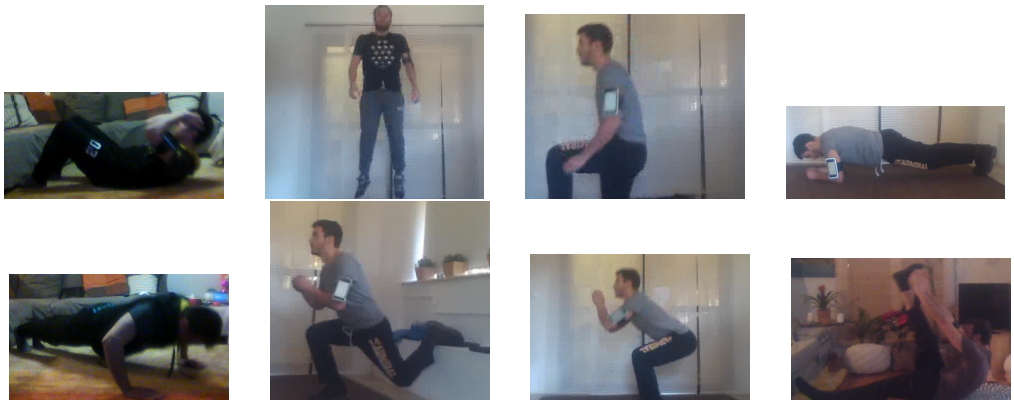


Figure 2.9. Adopted classes (row-wise): crunches, jumping squats, lunges, plank, push-ups, romanian squats, squats and toe-touches. .

The aforementioned eight exercises have been performed by *six individuals*. Each individual executed all 8 exercises from one to five times. Several recording conditions have been adopted during the data compilation. In particular:

- three different web cameras have been used and with different sampling properties (frames per second and frame resolutions).
- recordings have been carried out at five different places
- times of the recordings also varied

The above recording differentiations lead to a diversity of recording conditions, in terms of context, lighting conditions and raw signal noise. In addition, we need to emphasize on the fact that the six individuals were not provided with particular instructions on how to perform the eight exercises, leading to an additional factor of diversity with regards to the exercise execution. *The total number of recordings is equal to 130*. The final dataset can be found online for purposes of further experimentation [2].

[2]https://sites.google.com/view/michalis-papakostas/datasets

2.3.2.2.2 Classification

As described in Section 2.3.2.1.2, the feature extraction process leads to a multi-dimensional feature vector for each session recordings. Each unknown recording is therefore represented by a feature vector of either (a) sensorial information (b) visual information or (c) fused information. Each of these samples is classified using a Support Vector Machine with a probabilistic output. We have selected to use probabilistic SVMs [90] due to their ability to generalize well especially in high dimensional classification problems [91]. The model is trained using a cross-validation procedure to select the optimal SVM parameter, namely the soft margin parameter C.

2.3.2.3 Experiments

2.3.2.3.1 Performance Measures

Let CM be the confusion matrix, i.e. a $N_c \times N_c$ matrix (N_c is the total number of exercise classes), whose rows and columns refer to the true (ground truth) and predicted class labels of the dataset, respectively. Each element, $CM(i,j)$, of the confusion matrix stands for the number of samples of class i that were automatically assigned to class j. The diagonal of the confusion matrix captures the correct classification decisions ($i = j$). Then, the class recall $Re(i)$, precision $Pr(i)$ and F_1-measure are computed. As an overall performance measure, the average (among all classes) F1 measure is computed. Bellow follows the detailed explanation of the adopted evaluation metrics:

- **Recall**: The proportion of data with true class label i that were correctly assigned to class i

$$Re(i) = \frac{CM(i,i)}{\sum_{m=1}^{N_c} CM(i,m)} \tag{2.18}$$

, where $\sum_{m=1}^{N_c} CM(i,m)$ is the total number of samples that are known to belong to class i

- **Precision**: The fraction of samples that were correctly classified to class i if we take into account the total number of samples that were classified to that class

$$Pr(i) = \frac{CM(i,i)}{\sum_{m=1}^{N_c} CM(m,i)} \tag{2.19}$$

- **F1**: The harmonic mean of the precision and recall values

$$F_1(i) = \frac{2Re(i)Pr(i)}{Pr(i) + Re(i)} \tag{2.20}$$

2.3.2.3.2 Results

As explained in Section 3.5.1.1 a dataset has been compiled and manually annotated, consisting of 130 recordings of 8 exercising classes performed by 6 individuals, so on average, each exercise was performed 2.7 times by each individual to achieve high diversity in term of recording conditions. Our goal in this experimentation is to estimate the performance of each modality (sensorial and visual-based), using the performance measures described above.

In order to evaluate our method, two experimental approaches have been followed:

- With user calibration: the user provides a set of calibration recording sessions that are used to re-train the supervised models so that they are better applied on his/her moving patterns and to particular room differentiations.
- Without user calibration: no information related to the user is provided to the supervised task. The evaluation is performed using training data that contain no recording from the test user.

In both cases, repeated subsampling validation is used, i.e. in each iteration, the data is randomly split to testing and training. However, in the second scenario,

no recordings from the test user are used in the training data at all. The overall results for the two experimental scenarios are shown in Table 2.3.2.3.2. It can be seen that the accelerometer-based estimation significantly outperforms the visual-based approach, while the early fusion of both modalities leads to a performance boosting of 4% for the user calibration scenario. However, the visual domain achieves much lower performance when no user calibration is adopted, which also leads to the inability of the fused data to outperform the best individual classifier in that scenario.

Modality	With User Calibration	Without User Calibration
Accelerometer-All	91%	81%
Visual	72%	43%
Fusion	95%	80%

Table 2.3. Overall performance results for the two experimental scenarios (with and without user calibration).

2.3.2.4 Takeaways

In this work, we presented a method that combines accelerometer data and visual information in order to recognize exercising activities performed by single humans. Low-level temporal features similar to those used in audio analysis applications were implemented and applied on the accelerometer data, while simple frame-level visual features have been used to represent the visual channel. A real-world dataset of 6 humans, performing 8 different exercises has been recorded and manually annotated to train and evaluate the proposed approach. Extensive experimentation has proven that the fusion approach achieves 95% of overall performance, which means that it boosts the performance of the best individual modality (the accelerometer) by 4%. However, in the case that user calibration is missing, fusion fails to boost the per-

formance and the best F1 measure is presented by the single accelerometer modality (81%). The compiled dataset is available to the public for further experimentation [3].

2.4 Advantages, Limitations & General Observations

The applications and concepts discussed in Chapter-2 are proof that accessing multiple information sources at the same time can enhance the outcomes of most interactive scenarios. However, there are many considerations and decisions that need to be made during the design process, which eventually can affect positively or negatively the final outcome. These decisions mainly relate to, which interaction signals need to be monitored, what technology fits best to track those signals given the application scenario and how these different sources must be combined in a computational level in order to maximize system's performance. All these parameters, of course, come to complement a key requirement of any HCI design that is to minimize hardware intrusiveness.

Considering all the aforementioned observations, the two application discussed in this chapter highlighted the potentials of traditional machine learning approaches towards implementing efficient multi-modal architectures. Both evaluations showed that common classifiers such as SVMs or Random-Forests are quite capable of distinguishing between different behavioral patterns when efficient feature representations are available. Moreover, the same features and modeling techniques are able to address problems across different domains and sensors if the available data are formatted in an appropriate form. For example in Section-2.3.1 we exploited algorithms inspired from the domain of computer-vision to model data coming from a pressure-mat sensor, while in Section-2.3.2 we showed that features initially designed for audio processing were extremely efficient for modeling accelerometer data. These observations are of

[3] https://sites.google.com/view/michalis-papakostas/datasets

great importance when designing multi-modal interfaces, as they can decrease significantly the required engineering time towards building effective prototypes, able to run in real-time, especially when the data are limited.

However, it seems that working with such techniques comes with two major limitations. Firstly there is a threshold in the performance of such methods as the conditions become more ambiguous. In both scenarios, evaluation showed that outliers where very hard to identified and in most cases lead to miss-classified decisions. Secondly, as an extension to the first observation comes the fact that such methods are very hard to generalize their decisions. They are extremely dependent the feature representations and thus, in many scenarios they fail to address behaviors across new, unknown users. This effect can be smoothed out using calibration and active learning [92] techniques (ie. learning through interaction), but still limits significantly the applicability of such methods.

Towards addressing these problems, in Chapter-3 we evaluate the abilities of Deep Learning algorithms and specifically Convolutional Neural Networks (CNNs) on modeling different aspects of human behavior, in terms of performance and generalizability. We compare CNNs with the traditional ML methods presented in Chapter-2 and we discuss how data-augmentation and transfer learning can be used towards leveraging the common issue of limited data availability.

CHAPTER 3

DEEP LEARNING FOR BEHAVIORAL ANALYTICS

3.1 Introduction

Deep Learning has undoubtedly dominated the area of machine learning during recent years across almost every application domain. In particular, in vision and audio based applications, deep classifiers have offered groundbreaking results especially in problems related to object detection and speech recognition [93]. As a result, such methodologies meant to have a major impact on analyzing human and behavioral characteristics with face recognition, pose estimation and dialogue management being some of the most popular applications tackled by the technology [94, 95, 96]. CNNs and their alterations have been traditionally the most powerful tool of deep learning methods, especially in the field of computer vision [97, 98]. The most significant advantage of deep classifiers and specifically CNNs is their ability to create invariant feature representations, which are able to capture more generic patterns across the training samples and thus, produce more solid and robust classification models.

However, despite their popularity, deep learning models suffer by their demand for using tons of training data in order to build effective classifiers and avoid overfitting. A fact that comes in contrast to the more traditional modeling techniques discussed in the previous chapter.

In the following sections, we investigate the ability of CNN classifiers to tackle three diverse applications that relate to human behavioral analytics, namely: a) activity recognition [23], b) speech-music discrimination [39] and c) emotion recognition from speech [45]. We compare their performance with more traditional ML methods

and we discuss how we can exploit the concept of transfer learning in order to address the problem of limited data availability. Moreover, we explore ways of reusing pre-trained models to tackle classification scenarios stemming from different information domains and we evaluate the ability of such classifiers in terms of robustness and computational demands. Lastly, we explore ways of augmenting the available data towards meeting the training demands of deep classifiers. Our evaluation indicates that deep classifiers can significantly dominate traditional methods in terms of performance under specific scenarios. However, their architectural design and their choice of use must be very carefully orchestrated and is mainly dictated by the nature of the training data and the application domain.

3.2 CNNs & Transfer Learning

Transfer Learning is the process of modifying the parameters of a pre-trained ML model to match the feature representations that appear on a new (target) dataset and is mostly applied in cases when data are limited to train a statistical model from scratch.

The concept of transferring knowledge across domains is not new in the ML society [99]. However, before the DL era, such methods were not that popular since they usually provided results inferior compared to traditional training techniques given the existing modeling approaches. The authors in [8] were the first to demonstrate and evaluate in depth the process of transfer-learning in CNNs (Figure-3.1). Since then, transfer learning and CNNs have been extensively combined to approach similar problems - most frequently related to object segmentation [100, 101].

In this chapter, our major concern with respect to transfer learning and behavioral modeling is to discuss how such methods can be used to target problems of different nature between the original and the target datasets and how CNNs can

generalize across tasks, users and information domains when the available data are relatively limited.

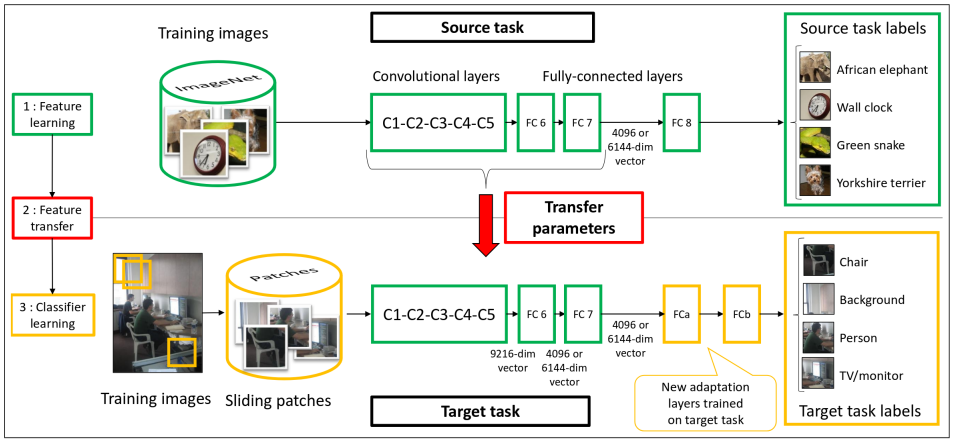


Figure 3.1. The concept of transfer-learning in CNNs as proposed initially by Oquab et al. [8].

3.3 From Spatial Analysis to Temporal Modeling

In the following sections, we discuss how to exploit CNN architectures, that are pre-trained on vast datasets, towards designing new models able to make decisions on new classification tasks. In particular, we use a CNN architecture trained on the Imagenet dataset [102], which is a dataset compiled for spacial image analysis for the tasks of object detection and recognition and we propose a new classifier, for recognizing activities in the temporal domain.

3.3.1 Recognizing Activities of Daily Living (ADLs)

In this work we show a deep learning classification method for short-term recognition of human activities, using raw color (RGB) information. In particular, we present a CNN classification approach for recognizing three basic motion activity

classes, that cover the vast majority of human activities in the context of a home monitoring environment, namely: sitting, walking and standing up. A real-world fully annotated dataset has been compiled, in the context of an assisted living home environment that is available online for further experimentation [1]. Our extensive analysis aims to highlight the benefits of deep learning architectures against traditional shallow classifiers functioning on hand-crafted features. Our experiments focus on evaluating the ability of such models to learn highly invariant representations that are robust to noisy inputs. Moreover, we emphasize the powerful potentials of transferring knowledge across tasks, using Deep Convolutional Neural Networks towards tackling problems on different information domains.

3.3.1.1 Methodology

As recent literature has shown, deep hierarchical visual feature extractors can significantly outperform shallow classifiers trained on hand-crafted features, when the amount of the available training data allows it. Deep models tend to be more robust and generalizable when countering problems that include significant levels of inherent noise. The architecture of our deep CNN was initially proposed in [24]. The model is mainly based on the Caffenet [103] reference model, which is similar to the original AlexNet [97]) and the network proposed in [104]. For our experiments, we used the *BVLC Caffe* deep-learning framework.

The network architecture consists of two convolution layers with a stride of 2 and kernel sizes equal to 7 and 5 respectively, followed by max-pooling layers. As a next step, a convolution layer with three filters of kernel size equal to 3 is applied, followed again by a max pooling layer. The next two layers of the network are fully connected layers with dropout, followed by a fully connected layer and a softmax

[1] https://sites.google.com/view/michalis-papakostas/datasets

classifier, that shapes the final probability distribution. All max-pooling layers have kernel size equal to 3 and stride equal to 2. For all the layers we used the ReLu as our activation function. The output of the network is a distribution on our three target classes, while the output vector of the semifinal fully connected layer has a size equal to 4096. We have adopted a 1000-iterations fine-tuning procedure, with an initial learning rate of 0.001, which decreases after 700 iterations by a factor of 10.

Since training a new CNN from scratch would require big loads of data and high computational demands, we used transfer learning to fine-tune the parameters of a pre-trained architecture. The original CNN was trained on the 1.2M images of the ILSVRC-2012 [105] classification training subset of the ImageNet [102] dataset. Following this approach, we manage to decrease the required training time and to avoid overfitting our classifier by ensuring a good weight initialization, given the relatively small amount of available data. Finally, the data are preprocessed by augmenting the frame dimensionality to 240x320. The input to the network corresponds to the 227x227 center crops and their mirror images.

3.3.1.2 Dataset

In order to train and evaluate our system, a dataset that consists of real-world recording sessions of RGBD data has been created. For data acquisition, an XTion sensor [2] has been used as part of the robotic sensoring infrastructure of the Radio platform. In total, 12 humans have participated in different 8 scenarios. Each recording session has been repeated on 1 to 3 different days (random number of repeats for each user). This has been done in order to ensure a certain diversity of lighting conditions. 272 recordings have been recorded and annotation in total. In each scenario the recording starts with the user sitting on a chair, after a while, he/she stands up

[2]https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/

and starts walking to the exit of the room. Figure-3.2 illustrates sample frames from the compiled dataset. The dataset can be found online and downloaded for free [3].

Figure 3.2. Examples of the compiled and annotated dataset's frames. Different rows correspond to separate users (humans). The first column corresponds to the "sitting" class, the second to the "sit to stand" class, while the third and fourth columns show walking examples. It can be seen that different walking directions, lighting conditions and obstacles have been used to increase diversity in conditions.

[3]https://sites.google.com/view/michalis-papakostas/datasets

3.3.1.3 Experiments

We have adopted the following types of experimentation based on the dataset described at Section 3.3.1.2:

1. Evaluation of the frame-wise classification method using a cross-validation procedure that splits training and testing data based on individual recordings. In other words, according to that approach, all frames of each recording are either used for training or testing. Three random subsampling cross-validation repetitions have been conducted, each repetition corresponding to a different random permutation of the videos in the dataset.
2. Evaluation using a cross-validation procedure that splits training and testing data based on subject IDs. According to that setup, the frames of all videos that belong to the same person are either used for training or testing. This has been conducted in order to evaluate the method in terms of subject-independence. Again, three random subsampling cross-validation repetitions have been conducted, each repetition corresponding to a different random permutation of the subjects (humans).
3. Evaluation against different noise ratios and comparison to a Support Vector Machine classifier using hand-crafted features. Towards this end, we have added Gaussian noise of several SNR ratios to the images before testing. In addition, an SVM classifier using typical visual features (HOGs, LBPs, color histograms) has been adopted for comparison reasons. The SVM classifier has been fine-tuned in terms of the C parameter, while a linear kernel has been adopted.

Tables 3.1, 3.2 and 3.3 present the initial confusion matrix, the row-normalized confusion matrix and the performance measures (Recall, Precision, F1, Accuracy) respectively, for the first experimental setup. Similarly, tables 3.4, 3.5 and 3.6 present

	Sitting	Standing	Walking
Sitting	18.84	0.58	0
Standing	2.89	11.2	0.98
Walking	0.2	1.17	64.54

Table 3.1. Experimental setup 1: Average Initial Confusion Matrix (normalized to sum up to 100%)

	Sitting	Standing	Walking
Sitting	97.01	2.99	0
Standing	19.18	74.32	6.50
Walking	0.3	1.78	97.92

Table 3.2. Experimental setup 1: Row-Normalized Confusion Matrix. Diagonal elements represent the respective recall rates. Note that these recall rates are not exactly equal to the recall rates presented in 3.3: this is due to the fact that these recall - precision rates are averaged *per video recording*, not per frame.

	Sitting	Standing	Walking	Average
Precision	0.86	0.86	0.99	0.91
Recall	0.97	0.76	0.98	0.90
F1	0.91	0.81	0.99	0.90
Average Accuracy		0.95		

Table 3.3. Experimental setup 1: Per class and average Recall, Precision and F1 and overall Accuracy, computed over all frames of the testing dataset.

	Sitting	Standing	Walking
Sitting	21.2	0.79	0
Standing	2.37	11.75	0.72
Walking	0.02	0.65	62.5

Table 3.4. Experimental Setup 2: Average Initial Confusion Matrix (normalized to sum up to 100%)

the confusion matrix, the row-normalized confusion matrix and the performance measures, for the second experimental setup. These results prove that the CNN classifier

	Sitting	Standing	Walking
Sitting	96.41	3.59	0
Standing	15.97	79.18	4.85
Walking	0.03	1.03	98.94

Table 3.5. Experimental setup 2: Row-Normalized Confusion Matrix. Diagonal elements represent the respective recall rates. Note that these recall rates are not exactly equal to the recall rates presented in 3.6: this is due to the fact that these recall - precision rates are averaged *per video recording*, not per frame.

	Sitting	Standing	Walking	Average
Precision	0.9	0.89	0.99	0.93
Recall	0.96	0.79	0.99	0.92
F1	0.93	0.84	0.99	0.92
Average Accuracy		0.95		

Table 3.6. Experimental setup 2: Per class and average Recall, Precision and F1 and overall Accuracy

is robust and independent to subject-specific characteristics, since the performance in the two first experimental setups is similar.

Finally, Figures-3.3 (top) and 3.3 (bottom) present respectively the mean F1 and accuracy measures for the two methods (CNN and SVM) and for different levels of signal-to-noise ratio (SNR, in dB). The CNN models illustrated in those figures are the ones trained on the first experimental scenario. We can easily infer similar behavior on the models trained on the second experimental scenario. The robustness of the CNN approach is obvious: both F1 and accuracy fall dramatically for the SVM model as the SNR ratio is reduced. On the other hand, CNN is much more robust to noise: even for 0dB SNR, the F1 measure is kept above 80%, while the respective measure for the SVM case is below 50%. In particular, the SVM classifier with hand-crafted features seems totally unstable in terms of both overall accuracy and F1 measure, for all levels of noise below 20dB SNR. Finally, we have also experimented

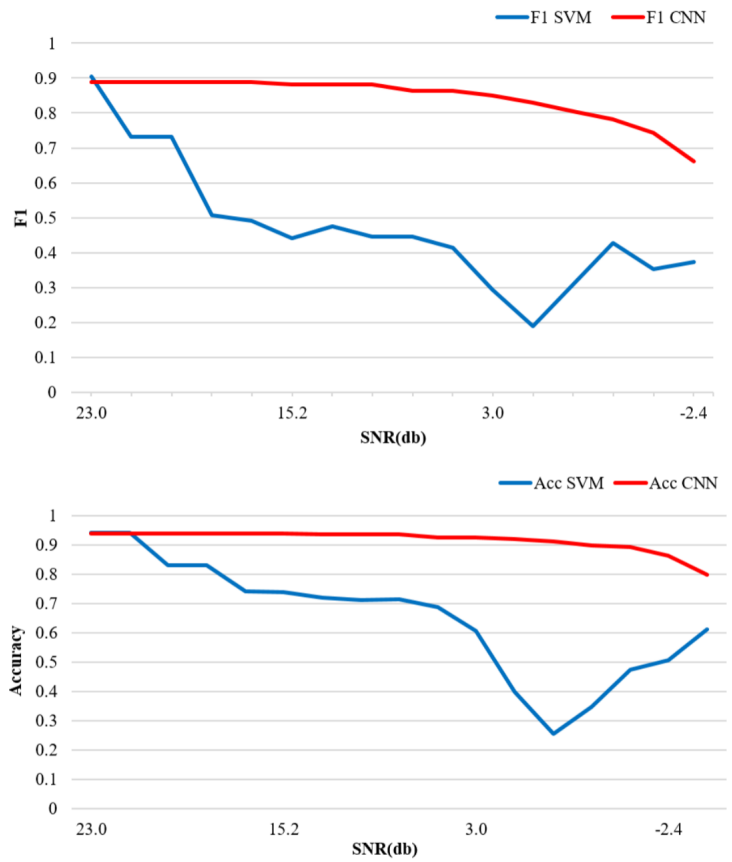


Figure 3.3. CNN and SVM comparison against different noise levels. Both CNN and SVM models are trained based on the first experimental scenario. Similar behaviours were observed in the models trained on the second experimental scenario. The robustness and consistency of the CNN classifier is obvious against traditional SVM-based methods trained on hand-crafted features..

using a temporal median filter to smooth the outputs of the frame-wise classifier using various kernel sizes. However, this did not lead to any worth noticing improvements.

3.3.1.4 Takeaways

In the previous sections showed a way of using pre-trained CNN models towards addressing the most traditional problem of behavioral modeling, that of activity recognition and specifically in the concept of ADLs, when the available data are limited.

We highlighted the advantages of using deep models against traditional classifiers and we showed how vigorous such models are against ambiguous input. Finally, a real-world dataset with varying conditions was compiled, annotated and made available to the public for the purposes of ADL recognition in a smart home environment [4].

3.4 From Computer Vision to Audio Classification

In the upcoming sections, we show a way of using transfer learning and CNNs towards training models that operate on different input modalities. Exploiting the findings of our previous analysis, we train a classifier for the traditional task of Speech-Music discrimination that expands its initial classification abilities not only in the time domain but also in a completely different information area. Moreover, we propose a way of augmenting audio data and we show the great potentials of such methods. Our implementation has been made available to the public [5] and at the time of the publication our results are considered as state-of-the-art in the task.

3.4.1 The task of Speech-Music Discrimination

Speech music discrimination is a traditional task in audio analytics, useful for a wide range of applications, such as automatic speech recognition and radio broadcast monitoring, that focuses on segmenting audio streams and classifying each segment as either speech or music. In this work, we investigate the capabilities of Convolutional Neural Networks (CNNs) with regards to the speech - music discrimination task. Instead of representing the audio content using handcrafted audio features, as traditional methods do, we use deep structures to learn visual feature dependencies as they appear on the spectrogram domain (i.e. train a CNN using audio spectro-

[4] https://sites.google.com/view/michalis-papakostas/datasets

[5] https://github.com/MikeMpapa/CNNs-Speech-Music-Discrimination

grams as input images). The main contribution of our work focuses on the potentials of using pre-trained deep architectures along with transfer-learning to train robust audio classifiers for the particular task of speech music discrimination.

We highlight the supremacy of the proposed methods, compared both to the typical audio-based and deep-learning methods that adopt handcrafted features, and we evaluate our system in terms of classification success and run-time execution. To our knowledge, this is the first work that investigates CNNs for the task of speech music discrimination and the first that exploits transfer learning across very different domains for audio modeling using deep-learning in general.

In particular, we fine-tune a deep architecture originally trained for the Imagenet classification task, using a relatively small amount of data (almost 80 mins of training audio samples) along with data augmentation. We evaluate our system through extensive experimentation against three different datasets. Firstly we experiment on a real-world dataset of more than 10h of uninterrupted radio broadcasts and secondly, for comparison purposes, we evaluate our best method on two publicly available datasets that were designed specifically for the task of speech-music discrimination. Our results indicate that CNNs can significantly outperform current state-of-the-art in terms of performance especially when transfer learning is applied, in all three test-datasets.

All the discussed methods, along with the whole experimental setup and the respective datasets, are openly provided for reproduction and further experimentation[6].

[6]https://github.com/MikeMpapa/CNNs-Speech-Music-Discrimination

3.4.1.1 CNNs for Audio Classification

Convolutional Neural Networks are probably the most popular modeling technique in computer vision related problems nowadays. Their ability to capture and represent robust and invariant features across millions of images has provided breakthrough results in some of the most traditional computer-vision problems such as activity or facial-expression recognition([106] and [107]). Despite their proven value in capturing features from multi-dimensional spaces, research that exploits CNN classifiers for non-vision problems and especially audio, has been very recently introduced and in a very limited amount of applications - mainly related to music classification or emotion recognition from speech ([108] and [45]). The aforementioned results highlight the great potentials of CNNs in modeling audio signals and indicate their potential superiority against traditional audio classifiers in several non-trivial tasks.

In our case, we use CNNs as a classification method to classify raw spectrograms, with minimum data pre-processing into Speech or Music samples. Our approach is shown below in the form of pseudo-code. In the rest of the section we discuss in depth implementation details and we show how CNNs can be designed and exploited to capture audio related features for the problem of Speech-Music discrimination. In Algorithm-1 we show the proposed computational pipeline.

3.4.1.2 Training Dataset and Augmentation

All the evaluated methods have been trained using a set of pre-segmented audio samples each one belonging to any of the two classes (speech or music). In particular, the training data consists of 750 samples containing speech and 731 samples containing music. The average duration of a music sample equals 3.2 secs while the total duration of all 750 music samples is 2428 secs (40.5 mins). On the other hand,

Algorithm 1 CNNs for Speech-Music Discrimination

target_height, target_width ← CNN input-image *height* & *width*

m_win, m_step ← length & step of *mid-term* window

s_win, s_step ← length & step of *short-term* window

median_win ← size of median filter for post-processing

for $i = 0; i < length(audio\_file); step\_i = s\_step$ **do**

    $audio\_segment \leftarrow audio\_file[i : m\_win]$

    $spectrogram \leftarrow ABS\{\ FFT\{audio\_segment,\ s\_win,\ s\_step\}\ \}$

    $resized\_spec \leftarrow LINEAR\_INTERPOLATION\{spectrogram, target\_height, target\_width\}$

    raw_prediction ← *CNN_Classifier(resized_spec)*

    filtered_prediction ← *median_filter(raw_prediction , median_win)*

the average duration of a speech sample in our training set is 3.1 secs and in total the duration of all 731 speech samples is 2237 sec (37.3 mins). All the samples from both classes were processed in a sampling frequency of 16000 Hz and where mono-channel audio samples. These speech and music segments have been gathered from several sources such as movies, youtube videos or radio-shows and have been manually annotated for the purposes of the work presented in [109].

Deep learning techniques, in most cases, require huge amounts of training data, in order to achieve satisfactory classification performance rates and avoid overfitting. In cases that the original data size is limited, data augmentation is required to overcome this data scarcity problem. Data augmentation is defined as a series of deformations applied on the annotated training samples which result in new additional training data [110]. In most computer vision applications that utilize deep learning for classification, data augmentation is achieved through image deformations such as horizontally flipping, random crops and color jittering. In our case, before extracting

the spectrogram of each training sample we add a random background sound (playing the role of noise) in three different Signal-To-Noise ratios (5, 4 and 3) and for three different crops of the original audio sample. If we also include the original (no noise) training sample, this means that this data augmentation procedure achieves a 3x3 = 9 dataset increase.

After the data augmentation, we end up with 7500 music samples (750 original samples and 6750 new samples created after data augmentation) and 7262 speech samples (731 original samples and 6531 samples created after data augmentation). In total the duration of the augmented music class equals 12587 secs (210 mins) while the average duration of a music sample is 1.7 secs. Similarly, the total duration of the augmented speech class is 11161 secs (186 mins) while the average duration of a speech sample equals 1.5 secs. Figure 3.4 presents an example of two original signals after the augmentation process.

3.4.1.3 Audio Segment Representation

Each audio stream is broken to overlapping mid-term segments of 2.4 seconds length, while a 1-second step is used (i.e. almost 60% overlap). For each segment, the spectrogram is extracted, using 20 ms short-term window size and 15 ms step (25% overlap). This spectrogram is first interpolated, using linear-mapping, to match the target input of the CNN classifier and then is fed into the network for classification. In the rest of this research, we show two different ways of how CNNs can be adopted for the task of speech-music discrimination and we thoroughly discuss the pros and cons of each different approach.

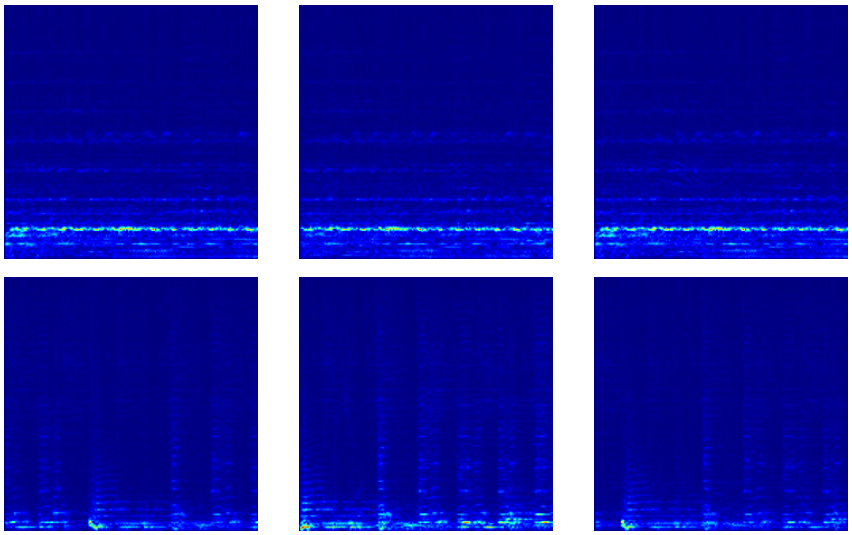


Figure 3.4. Examples of part of the augmentation process for a music (upper row) and a speech (lower row) sample. The augmentation process generates 9 new spectrograms by adding background noise at three different levels and by applying 3 different crops (9 augmentation results in overall). The first image of each row represents a spectrogram of the original audio samples while the other two are spectrograms extracted from the same wav files after augmentation..

3.4.1.4 Using CNNs to classify audio segments

As recent literature has shown, deep hierarchical visual feature extractors can significantly outperform shallow classifiers trained on hand-crafted features and are more robust and generalizable when facing problems that include significant levels of inherent noise. To classify an unknown audio segment to either speech or music, we utilize two different CNN classifiers that differ primarily in their size.

Big CNN: The first one performs upon RGB-pseudo-colored frequency images, corresponding to the spectrograms of each audio segment, as described above. The color-map matches frequency values with a different color according to their intensity. Higher frequency values are mapped with brighter colors while lower frequencies with darker ones. The reason to do so was to be able to exploit the pre-trained CNN

architecture for fine-tuning, which was originally designed to accept input images of three channels. The architecture of this deep CNN (Figure-3.5) was initially proposed in [24]. The model is mainly based on the Caffenet [103] reference model, which is similar to the original AlexNet [97]) and the network proposed in [104]. The network architecture consists of two convolution layers with a stride of 2 and kernel sizes[7] equal to 7 and 5 respectively, followed by max-pooling layers. Then a convolution layer with three filters of kernel size equal to 3 is applied, followed again by a max pooling layer. The next two layers of the network are fully connected layers with dropout, followed by a fully connected layer and a softmax classifier, that shapes the final probability distribution. All max-pooling layers have kernel size equal to 3 and stride equal to 2. For all the layers we use the ReLu as our activation function.

The output of the network is a probability distribution on our target classes, while the output vector of the semifinal fully connected layer has a size equal to 4096. The initial learning rate is set to 0.001 and decreases after 700 iterations by a factor of 10. Since training from scratch such a big CNN structure as the one proposed by [24], requires millions of images thus, having very high computational demands, we used transfer learning to fine-tune the parameters of a pre-trained model. The original CNN was trained on the 1.2M images of the ILSVRC-2012 [105] classification training subset of the ImageNet [102] dataset. Following this approach, we manage to decrease the required training time and to avoid over-fitting our classifier by ensuring a good weight initialization, given the relatively small amount of available training data. It is important to note that the network used for weight initialization was pre-trained on a dataset (*ImageNet*) completely irrelevant to our target data, proving the high invariance of CNN features and the importance of having a robust weight initialization. Finally, the input to the network corresponds to 227x227 RGB-pseudo-

[7]By kernel size we refer to the size of each dimension of the kernel. All kernels are square matrices.

colored spectrogram images and their mirrors. Table-3.7, illustrates the effect of transfer-learning on the original kernels.

Small CNN: The second CNN structure (Figure-3.6) is a smaller architecture that has been designed for the needs of the discussed problem. Despite the very good results provided by the first method, Caffenet was originally designed to tackle the Imagenet recognition task which is a problem significantly more complex than Speech-Music discrimination. In contrast to our target domain, Imagenet is a 1000-class image recognition task where many thousands of different features must be observed and captured by the classifier. However in our scenario, in all possible cases, the displayed patterns are much more simplistic compared to the Imagenet problem (Figure-**??**). Traditional computer-vision issues such as background and lighting variations do not apply in our domain thus simplifying significantly our classification task. In our scenario noise mainly occurs by background sounds -which can be considered as a visual occlusion in our case-, which in most cases (since they are in the background) do not affect much the dominant signal in the frequency-domain (mostly low frequencies are impacted). Moreover, the original Caffenet was designed to function on raw RGB images; a source of information which is redundant for any audio-based classification problem. Spectrograms have by default two information channels and thus in order to make them fit in the original Caffenet, we had to augment a third dimension using the pseudo-coloring approach described in the previous paragraph. Thus, in an effort to avoid unwanted computations and inspired by the proven value of Caffenet's architecture we decomposed the initial model to a smaller one in order to evaluate the ability of Deep Convolutional Classifiers to model the problem in general. Taking into account all the aforementioned observations and through extensive experimentation, we ended up with a smaller architecture with a reduced number of convolutional layers. In total, the new network consists of two convolution layers

less than the previous model with three consecutive pairs of convolution and pooling layers. The first convolution layer has a kernel size equal to 5 and the following two convolution layers have kernels with size equal to 3. All intermediate max-pooling layers have kernels equal to 3. Kernel stride in all layers is equal to 2. As activation, the ReLu function is deployed once again. All fully-connected layers remain also untouched as well as learning rate and learning decay rate.

We evaluated two versions of this smaller CNN; one that operates again on 227x227 pseudo-colored RGB Images and one that operates on the default gray-scale spectrogram representation with smaller size equal to 200x200. For convenience reasons, we will refer to this model for now on as CN_SM. The grayscale version of CNN_SM is designed in an effort to reduce the computational complexity of the network by getting rid of some redundant computations including the extra information channel of the input but also some of the layers included in the original Caffenet. We reduce the image in an effort to investigate if the input size affects the final outcome. In both cases, CNN_SM is trained from scratch in an end-to-end fashion. CNN_SM exploits, on the one hand, the core components of Caffenet while on the other hand has a reduced total number of parameters by a factor equal to 25% (almost 550.000 parameters less) compared to the total number of parameters that were on the initial Caffenet. As our findings indicate in Section-3.4.1.8 CNN_SM can provide results that are slightly worse compared to the first method in a significantly shorter amount of time without the need of data augmentation. However, using transfer-learning based on the pretrained architecture of Caffenet still remains the most accurate and robust method. In both scenarios though, it is obvious that CNNs can depict significant differences between the two classes even when the available data are limited showing state-of-the-art results on the task.

The input data-layer in both cases are in a batch form of 128 spectrogram images. At the end of each epoch [8] (about 16 iterations without data augmentation and 156 iterations when augmenting the initial training-set) we reshuffle the training data aiming to capture more robust feature representations.

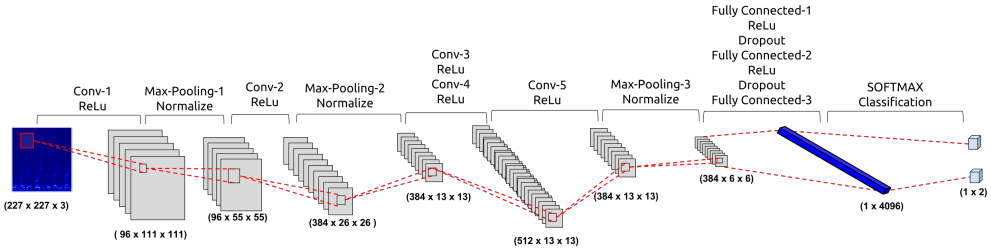


Figure 3.5. Caffenet: CNN Architecture proposed in [24].

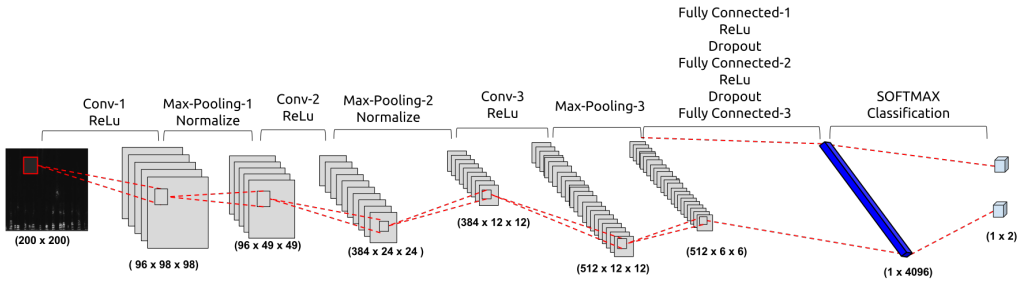


Figure 3.6. CNN_SM: Architecture proposed in this work.

[8] A single epoch consists of a forward pass and a backward pass of all the training samples. The number of iterations is the number of passes, each pass using [batch size] number of examples. A pass consists of both the forward and back-word propagation.

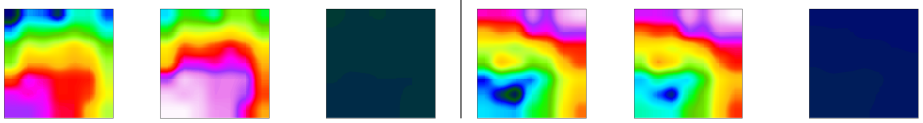
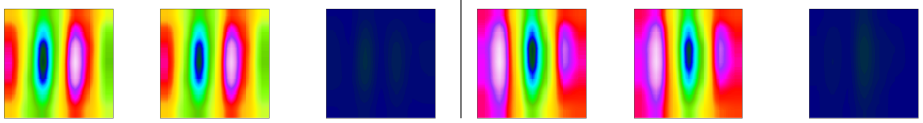
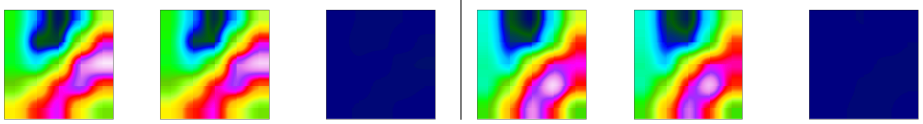
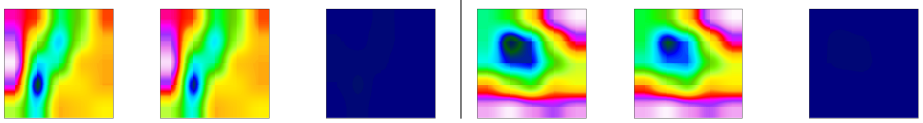
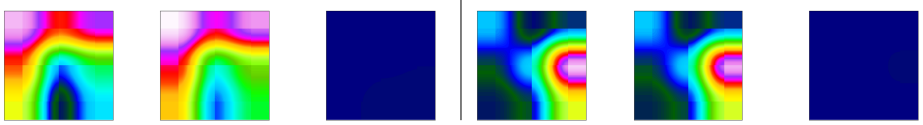
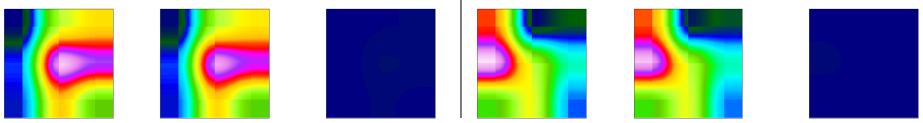
Coloromap						
Layer	Initial	Fine-tuned	Difference	Initial	Fine-tuned	Difference
1st Conv Layer						
	max=0.0529, min=-0.0386, mean_diff=0.0155			max=0.0553, min=-0.1039, mean_diff=0.0074		
						
	max=0.3058, min=-0.3580, mean_diff=0.0046			max=0.2088, min=-0.3622, mean_diff=0.0042		
2nd Conv Layer						
	max=0.0727, min=-0.06132, mean_diff=0.0013			max=0.0534, min=-0.0457, mean_diff=0.0011		
						
	max=0.0534, min=-0.0870, mean_diff=0.0010			max=-0.0068, min=-0.0465, mean_diff=0.0009		
3rd Conv Layer						
	max=0.0100, min=-0.0053, mean_diff=0.0014			max=0.0663, min=-0132.0457, mean_diff=0.0007		
						
	max=0.0962, min=-0.0161, mean_diff=0.0016			max=0.0579, min=-0.0238, mean_diff=0.0008		

Table 3.7. Convolution filters randomly picked from the first 3 layers. The first column illustrates the weights of the pre-trained Caffenet network before transfer-learning takes place. The second column displays the updated weights after fine-tuning. Lastly, third column displays how the values of each kernel have been shifted during the retraining process. *max* and *min* refer to the maximum and minimum weight values occurred in the two kernels, while *min_diff* refers to the mean shift value across all weights of each kernel.

3.4.1.5 Evaluation Datasets

In order to evaluate the performance of the proposed methodology against the traditional machine-learning methods shown in Table-3.13 a dataset (*D1*) of real recordings from several BBC radio broadcasts has been used. 33 separate uninterrupted radio streams of 10 minutes to 1-hour length each have been manually annotated originally for the purposes of the research presented in [109]. The total duration of the dataset is more than 10 hours (almost 620 minutes).

In addition, we have evaluated our method on two additional open-access datasets for comparison purposes with the work done by [111], which also used deep-learning, and specifically RBMs, on traditional audio features (MFCCs and DFT coefficients).

Dataset *D2* originally appeared in [112] and was subsequently refined in [113]. This corpus is a relatively small collection of 240 randomly chosen extracts from radio recordings. Each resulting file is 15 s long and stored in WAVE format. The original sampling frequency is 22050Hz and all samples are single channel wavs. For the purposes of this study, we reduce the sampling frequency to 16000Hz in order to match our current implementation. The dataset is partitioned by its creators into a training subset and a test subset. However, in order to have a fair comparison against the methods proposed by [111] we ignored the initial data partitioning scheme. Our final D1 test dataset consists of two classes pure music (101 files) and pure speech (80 files with male, female and conversational speech).

Dataset *D3* is available via the Marsyas website [114]. It consists of a total of 120 tracks, evenly distributed among the classes of music and speech. Each track is 30 seconds long and also stored in WAVE format. As in D2, we reduced the original sampling frequency from 22050Hz to 16000Hz. All audio samples are single channel wav files. The music class covers a wide variety of music genres and as in D2 some of

the music samples are purely instrumental. The speech class contains both male and female speakers and in some cases dialogue.

3.4.1.6 Performance Measures

Let CM be the confusion matrix, i.e. a 2×2 matrix, since 2 is the number of classes in our case. The rows and columns refer to the true (ground truth) and predicted class labels of the dataset, respectively. In other words, each element, $CM(i, j)$, stands for the number of samples of class i that were assigned to class j. The diagonal of the confusion matrix captures the correct classification decisions ($i = j$). As in Section-2.3.2 we evaluate our method using Precision (equation-2.19), Recall (equation-2.18) and F1 (equation-2.20) metrics. Note that, the confusion matrix, and therefore all adopted performance measures, have been extracted on a 1-second segment basis.

3.4.1.7 Results

The results of the proposed method, along with the compared methodologies on the D1 test dataset, are presented in Table 3.8. In general, two types of classifiers have been evaluated: (a) audio classifiers based on low-level audio features (b) classifiers applied on the spectrogram images. In particular, the following methods are evaluated (we also present the abbreviations in the following list):

1. **Audio-based classifiers**:
   (a) RF: Random Forests
   (b) GB: Gradient Boosting
   (c) ET: Extra Trees
   (d) SVM: Support vector machines
   (e) GMM+HMM: Gaussian Mixture Models + Hidden Markov Models

In order to extract hand crafted audio features he have used the pyAudioAnalysis library [85] which computes several time, spectral and cepstral domain audio features such as zero crossing rate (equation-4.1), spectral centroid and spread (equations-2.13 & 2.14), spectral flux (equation-2.16) and MFCCs. Towards this end, a short-term windowing is applied, and for each short-term window (frame) 34 features are computed. Then for each segment two feature statistics are extracted, namely the mean and standard deviation, leading to a 34 * 2 = 68 feature statistic representation for each audio segment. This final representation is used as a feature vector to classify unknown audio segments to either speech or music. More details can be found at [85].

2. **Image-based classifiers**. The following image classifiers have been directly applied to the spectrogram images for comparison reasons:
   (a) SVM: for comparison reasons we have also evaluated an image classifier applied on the spectrograms using typical visual features: *Histograms of Oriented Gradients*, *Local Binary Patterns*, *Grayscale and color histograms* (similarly to what we did in the multimodal method for fitness monitoring in Section-2.3.2). To train the SVM classifier each image was decomposed to a 4x4 grid and from each block (4x4=16 blocks in total) a feature vector was extracted. The final feature vector describing the whole image was the concatenation of each individual feature vector extracted from each of the 16 blocks.
   (b) Caffenet-S: this CNN uses the structure of the Imagenet CNN proposed by [24], but it is trained directly on the training samples of the speech - music discrimination task

(c) Caffenet-S,I: this is the same network, however, the weights of the Imagenet CNN are also used for initialization in the training phase of the speech-music classifier

(d) Caffenet-S,I,A: this is the same network (with weight initialization), but training is performed using the augmented data of speech - music

(e) CNN_SM: this is the smaller CNN proposed in Section 3.4.1.4 to discriminate speech and music segments, which is trained from scratch. For fair comparison against our Caffenet implementation, the results shown in Table-3.13 are with CNN_SM functioning on 227x227 pseudo-colored RGB images. It has to be noted that no significant differences in performance were observed when altering the type and shape of input to 200x200 grayscale (see Section-3.4.1.8).

(f) CNN_SM-A: this is the CNN of (e) trained using augmented data.

Additionally, the evaluation has been conducted with and without post-processing of the single classification decisions. We have selected a simple but effective median filtering on the extracted classification labels as a post-processing method. We have also conducted experiments with supervised smoothing approaches (e.g. HMM), but no further improvement was observed. The presented results have been produced after the application of a median filter of an 11-second window.

In Figure-3.7 we show the ROC curves of the two best methods (Caffenet -S,I,A and CNN_SM) when evaluated on D1. Judging from the presented results we argue that in general CNNs can significantly outperform traditional methods on the task. However, when transfer-learning is applied we observe a boost in performance equal to 3% when we compare the areas under the ROC curves of the two classifiers from almost 96% to 99%.

No Post-Processing

	Audio-based Classifiers					Image-based Classifiers					
	GMM	RF	GB	ET	SVM	SVM	Caffenet S	Caffenet S,I	Caffenet S,I,A	CNN_SM	CNN_SM A
Sp Rec	92.4	90.7	90.4	92.3	92.6	85.7	89.3	88.7	**93.7**	90.9	91.5
Sp Pre	79.5	82.7	82	80.9	77.4	83.6	89.4	**96.6**	93.3	91.1	90.8
Mu Rec	90.1	92.5	92.1	91.3	89.3	93.2	95.7	**98.8**	97.2	95.3	95.2
Mu Pre	95.7	96.2	96	96.8	96.8	94.2	95.7	95.7	**97.6**	96.2	95.8
Sp F1	85.5	86.5	86	86.2	84.3	84.6	89.3	92.5	**93.5**	91	91.1
Mu F1	92.8	94.3	94	94	92.9	93.7	95.7	97.2	**97.4**	95.7	95.5
Av F1	89.2	90.4	90	90.1	88.6	89.1	92.5	94.8	**95.4**	93.4	93.3

Post-Processing Segmentation

	Audio-based Classifiers					Image-based Classifiers					
	GMM	RF	GB	ET	SVM	SVM	Caffenet S	Caffenet S,I	Caffenet S,I,A	CNN_SM	CNN_SM A
Sp Rec	92.4	90.3	90.3	92.3	92.8	85.8	89.7	88.9	**93.9**	92	92
Sp Pre	81.2	83.6	83.7	82	79.2	87.7	92.2	**97.3**	94.9	95.2	95.5
Mu Rec	90.8	93	93	92	90.7	95	97	**99**	98.1	98.2	98.4
Mu Pre	96.1	96	96	96.8	96.9	94.3	95.9	95.8	**97.6**	96.5	96.8
Sp F1	86.4	86.8	86.9	86.9	85.4	86.7	90.9	92.9	**94.4**	93.6	93.7
Mu F1	93.4	94.5	94.5	94.3	93.7	94.6	96.4	97.4	**97.8**	97.3	97.6
Av F1	89.9	90.7	90.7	90.6	89.6	90.7	93.7	95.1	**96.1**	95.5	95.6

Table 3.8. Experimental results of the proposed method and comparisons to other methodologies, with and without post-processing on D1. We mainly focus on the average F1 measure as the final evaluation metric, due to its ability to be robust against unbalanced datasets, however, we report that the *overall classification accuracy* for the best two methods was: 96.6% for CNN_SM and 96.8% for the Caffenet-S,I,A method. For abbreviation purposes we define the following notations; Sp: Speech, Mu: Music, Rec: Recall, Prec: Precision, Av: Average

3.4.1.8 Comparison to other methods

One of the first efforts on speech - music discrimination is reported in [115] were the authors achieved a classification accuracy of 96%, by adopting simple time domain features, evaluated on a real-time monitoring application of a specific radio station for 2 hours of recording data. In [116], almost 20 min of audio data was used for training and testing purposes. The authors reported that on a short-term basis the overall accuracy was around 80%. When a mid-term window was used (1 s

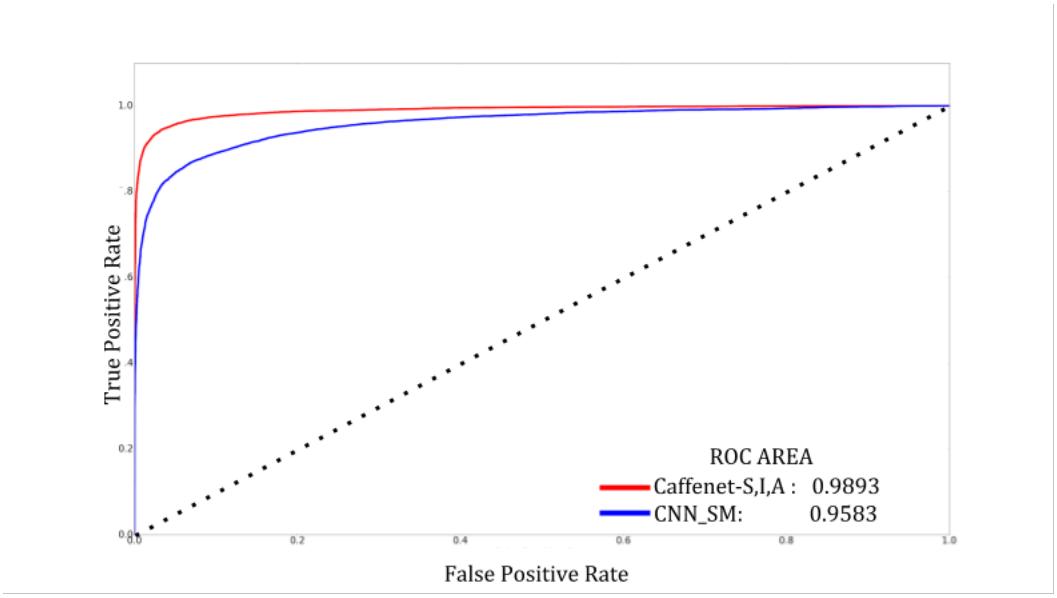


Figure 3.7. ROC curves of the two best methods (Caffenet-S,I,A and CNN_SM) when evaluated on D1.

long), the accuracy rose to approximately 95.9%. In [117], for training and testing the classifier almost 4,500 segments (10 s long each) of speech and 3,000 for music (10 s length again) were. The reported experiments showed that the error rate ranges from 1.2 to 6%, however, the assumption of homogeneous audio segments of quite a long duration (i.e., 10 s) is a simplified version of the problem. In later approaches such as [109, 118] the authors deployed more sophisticated techniques such as dynamic programming and Bayesian networks. Those two works were evaluated on the same data (D1 test dataset) as the proposed method. They report an accuracy of around 95.5%, which is comparable to current state-of-the-art approaches on such a large and diverse amount of the test data. In more detail, Table-3.9 presents the exact comparisons between the method proposed in [109] and the two dominant methods presented here, in terms of all the performance measures (speech and music recall and precision). In this case, to demonstrate the robustness of CNNs on the task the results associated with CNN_SM in Table-3.9 were estimated on the simplified image

inputs (gray-scale 200x200). As it is easily observable by comparing results in both Table-3.13 and Table-3.9, CNNs almost in all cases outperform the method presented by [109], with Caffenet-S,I,A showing again superior performance

	DP [109]	Caffenet-S,I,A	CNN_SM
Sp Rec	89.2	**93.9**	**92.1**
Sp Pre	**95.8**	94.9	95.4
Mu Rec	98.3	**98.1**	**98.4**
Mu Pre	95.6	**97.6**	**96.9**
Sp F1	92.4	**94.4**	**94.3**
Mu F1	96.9	**97.8**	**97.6**
Av F1	94.7	**96.1**	**96**

Table 3.9. Comparison between the two proposed CNN methods (Caffenet-S,I,A and CNN_SM both with postprocessing) and the work presented by [109] on audio-based features, where a Dynamic Programming (DP) approach on a Bayesian Network was deployed. Evaluation is being done on the D1 dataset. As in Table-3.13 the following notations are used; Sp: Speech, Mu: Music, Rec: Recall, Prec: Precission, Av: Average

For further experimentation we compared our dominant method (Caffenet-S,I,A) against the deep learning methods proposed by [111] when evaluated on datasets D2 and D3 (Figure-3.8). In order to have a fair comparison between the different methods, we show results in a similar manner as it was done by [111]. We present performance measurements using a confidence threshold, T_h . The goal of the threshold is to reject any classification decision if the estimated posterior probability of the winning class fails to exceed T_h. As complementary information, we also provide the percentage of patterns that have been left unclassified. [111] have experimented using different deep-learning methods based on MFCCs and DFT coefficients. The results shown on the graphs below were directly derived by that publication. It has to be

noted, that the evaluation procedure has been carried out on a segment basis, i.e. each segment of every file in the dataset has been classified separately.

As our results indicate CNNs with transfer learning significantly outperformed all methods described by [111] showing a reduction in classification error of 5% and 9% on D2 and D3 respectively compared to the best deep-learning methods with RBMs that operate in handcrafted audio features. In addition, the proposed method shows a significant improvement in the confidence levels of each decision by reducing the number of unclassified patterns by almost 5% and 9% respectively on D2 and D3.

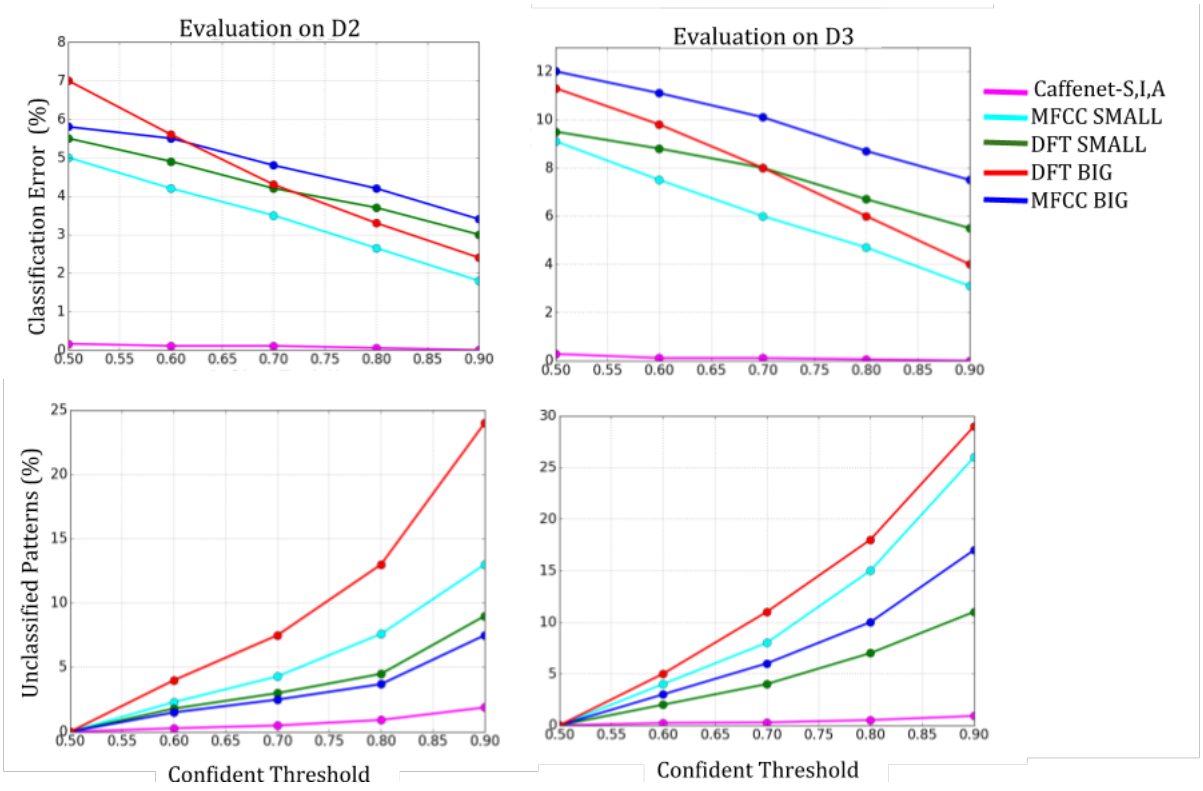


Figure 3.8. Evaluation of Caffenet-S,I,A against the RBM-based deep-learning methods proposed by [111]. Graphs on the left show evaluation on D2 test-dataset while graphs on the right refer on D3..

3.4.1.9 Computational Demands

Method	Time in mins		
	Total	Feature Ext.	CNN test
Caffenet - S, I, A	35.2	9.3	26
CNN_SM	23.5	9.3	14.3

Table 3.10. Execution Time (in minutes) on a GPU Tesla K40c of the two best methods for the whole testing dataset. Note that the size of this dataset is 620 minutes.

We have evaluated the required computational demands for the two proposed speech music classification schemes, namely the CNN_SM and the Caffenet network. As discussed above, the CNN_SM model is able to offer directly comparable results to its deeper counterpart when both methods were trained from scratch. At the same time, as Table 3.10 indicates, it requires less computational time to complete the whole evaluation process. In particular, with regards to the overall execution time required by the two models, the CNN_SM model achieves a relative computational reduction almost equal to 33%. If, however, we focus on the classification step alone, i.e. if we exclude the fixed feature extraction demand and the post-processing procedures (9.3 minutes for the whole dataset), the relative computational decrease is 45%. Note that, given the overall duration of the testing dataset (620 minutes), the CNN Caffenet-I method requires 4.2% of the true audio duration, while the CNN_SM method only 2.3% of the real audio time. This means, for example, that it takes almost a minute to classify one hour of audio data.

That is due to the decrease in the number of parameters that need to be learned and to the reduction of input's dimensionality. Based on the performance results presented in the previous subsection, it turns out that, given the simplistic visual fluctuations that spectrogram-images consist of, in terms of visual-shapes and colors, fewer parameters can be sufficient to model an audio problem with a relatively small amount of target classes, even if the available data are significantly few (around 750 samples per class).

3.4.1.10 Takeaways

To summarize, our experimentation has proven that CNNs are a very efficient method to better discriminate between speech and music, compared to typical methods that operate on the audio domain through handcrafted audio features. Utilizing transfer learning in CNNs boosts the classification performance (from 93.7% to 95.1%) on our primary evaluation dataset, and reduces the classification error on our additional evaluation datasets D2 and D3 by 5% and 9% respectively. In addition, using CNNs and transfer learning lead to higher levels of confidence in the final decision against previously published deep-learning methods on D2 and D3 that were based on traditional audio features. Moreover, data augmentation improved the results in the proposed task and proved that if performed carefully it can significantly advance the quality of our classifiers by providing the available resources towards training DL models.

Overall the experiment indicated the robustness of CNN classifiers on training behavioral patterns of variant nature but also outlines the high demands of such methods in terms of training data.

3.5 Learning Robust Representations Across Varying Input Domains

In our last evaluation, we propose a DL method towards recognizing emotion from speech by omitting any linguistic information. Goal of our approach is to address emotion, a core component of human behavior as discussed in Chapter-1, in a language-independent manner. The method aims to explore and highlight the robustness of deep classifiers against common ML methods in one of the most difficult problems of affect recognition. The final results are directly comparable to other state-of-the-art techniques and highlight the significant effect that deep models have towards building invariant feature representations even under extreme levels of ambiguity. A phenomenon that is very common in several tasks related to human behavior modeling.

3.5.1 Language-Independent Emotion Recognition from Speech

Emotion recognition from speech may play a crucial role in many applications related to human-computer interaction or understanding the affective state of users in certain tasks, where other modalities such as video or physiological parameters are unavailable. In general, a human's emotions may be recognized using several modalities such as analyzing facial expressions, speech, physiological parameters (e.g., electroencephalograms, electrocardiograms), etc. However, measuring of these modalities may be difficult, obtrusive or require expensive hardware (see Section-1.2.2). In that context, speech may be the best alternative modality in many practical applications. In this work, we present an approach that uses a CNN functioning as a visual feature extractor and trained using raw speech information. In contrast to traditional machine learning approaches, CNNs are responsible for identifying the important features of the input thus, making the need for hand-crafted feature engineering optional in many

tasks. The proposed method requires no extra features other than the spectrogram representations. Hand-crafted features were only extracted for validation purposes. Moreover, it does not require any linguistic model and is not specific to any particular language. We compare the proposed approach using cross-language datasets and demonstrate that it is able to provide superior results vs. traditional ones that use hand-crafted features.

3.5.1.1 Training Dataset and Augmentation

For our experiments, we used four different audio datasets. Three of the datasets are publicly available (Emovo [119], Savee [120], German [121]) and the last one is a custom made dataset, which includes audio samples gathered from movies. For the custom made dataset the samples were annotated manually by several researchers in NCSR Demokritos. All the movies used for the creation of our Movies-Dataset were in English except one that was in Portuguese. Statistics of the aforementioned datasets are reported in Table 3.11.

	Anger	**Disgust**	**Fear**	**Happiness**	**Neutral**	**Sadness**	**Surprise**	**Boredom**
Emovo	84	84	84	84	84	84	84	-
Savee	60	60	60	60	120	60	60	-
German	127	46	69	71	79	62	-	81
Movies	367	-	80	63	413	117	-	-

Table 3.11. Number of Audio of the Original Audio Databases for each class.

Since not all the datasets included samples for all the classes that are shown in Table 3.11, we decided to work only on their union. Thus, our final dataset consists only of the five common classes, namely anger, fear, happiness, neutral and sadness.

Our CNN architecture, described in Section-3.5.1.2, has been trained using a set of pre-segmented audio samples, randomly cropped from the original audio signal, each one belonging to any of the 5 classes (happiness, fear, sadness, anger, neutral) and with a fixed duration equal to 2s.

More specifically, we trained four different models, each time using samples from a single dataset. Each model was trained using 80% of the samples from each class of the dataset. For evaluation purposes, we performed four different experiments for each trained model (i.e., 16 experiments in total). We tested each trained model on the remaining 20% of samples from each class of the training dataset (note that those samples were used only for testing). Then we performed three additional experiments using each time all samples of each one of the other datasets.

As discussed at the beginning of this chapter, deep learning techniques require huge amounts of training data, in order to achieve satisfactory classification performance rates and avoid over-fitting. In cases that the original data size is limited, as in our scenario, data augmentation is required to overcome this data scarcity problem. Data augmentation is defined as a series of deformations applied on the annotated training samples which result in new additional training data [110]. In most computer vision applications that utilize deep learning for classification, data augmentation is achieved through image reformations such as horizontally flipping, random crops and color jittering. In our case, before extracting the spectrogram of each training sample we add a background sound (playing the role of noise) in three different Signal-To-Noise ratios (5, 4 and 3) for the crop of the original audio sample. If we also include the original (no noise) training sample, this means that this data augmentation procedure achieves a 3× dataset increase. Figure 3.9 presents some examples of the resulted spectrograms from each class after augmentation .

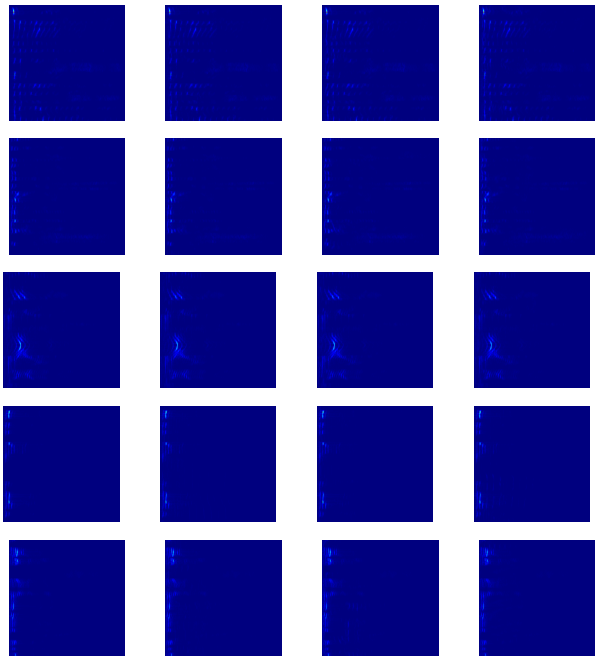


Figure 3.9. Spectrogram samples after the augmentation process for anger (first row), fear (second row), happiness (third row), sadness (fourth row) and a neutral (fifth row) classes. Augmentation the data generated 3 new spectrorams from each original sample (images in first column) by adding background noise at three different levels. Figure is best viewed in color..

From each audio stream, a single randomly cropped segment of 2 s length is extracted. For each segment, its spectrogram is extracted, using 40 ms short-term window size and 20 ms step. This spectrogram is the adopted representation for each 2-second segment of the audio stream, which is fed as input to the CNN, described in the next section.

3.5.1.2 Method

For recognizing the five-target emotion labels, we utilized a CNN classifier (*CNN_EM*) that operates upon the pseudocolored spectrogram images. As recent literature has shown, deep hierarchical visual feature extractors can significantly out-

perform shallow classifiers trained on hand-crafted features and are more robust and generalizable when countering problems that include significant levels of inherent noise. The architecture of our deep CNN structure was finalized after a very extensive experimentation process on different layer combinations and parameter tuning. Our goal was to build a model, that could depict robust feature representations for recognizing speech-emotion across all the datasets, in a language-independent manner. For our experiments, we used the BVLC Caffe deep-learning framework [103]. All Caffe trained models and necessary code to reproduce our experiments is available online [9].

The network architecture consists of four convolution layers in total, all of them with a stride of 2. The kernel sizes of the convolutional layers are of size 7, 5, 5 and 3 respectively. After every convolution and before the application of the non-linearity function we normalize the input batch using the Batch-Normalization transformation. In addition, in-between the initial three convolutional layers and after the last one, a pooling layer followed by a normalization layer is interposed. In this work, normalization layers adopt the LRN (Local Response Normalization) normalization method and all max-pooling layers have a kernel with size equal to 3 and a stride of 2. The last two layers of the network are fully connected layers with dropout, followed by a softmax classifier, that shapes the final probability distribution. For all the layers we used the ReLu as our activation function and weights are always initialized using the *xavier* [122] initialization. For the learning algorithm, we decided to use the standard SGD, as it led to superior results compared to other learning algorithms. The output of the network is a distribution on the five target classes, while the output vector of the final fully connected layer has a size equal to 4096. We have adopted a 5000-iterations fine-tuning procedure, with an initial learning rate of

[9] https://github.com/MikeMpapa/CNNs-Audio-Emotion-Recognition

0.001, which decreases after 600 iterations by a factor of 10. The input to the network corresponds to images of size 250×250 and organized in batches of 64 samples.

Given the very limited amount of available data, Batch-Normalization and the application of the xavier weight initialization boosted significantly the performance of the network by avoiding the learning process to get stuck in local minimums. In Figure 3.10 we illustrate the overall network architecture.

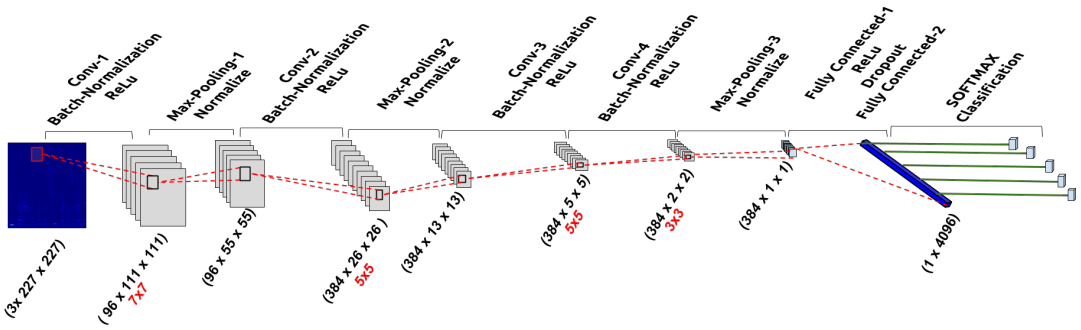


Figure 3.10. Proposed Convolutional Neural Network CNN_EM for recognising emotion..

3.5.1.3 Results

For comparison purposes we have evaluated the following two methods:

- *audio-based classification*: The pyAudioAnalysis [85] has been used to extract mid-term audio feature statistics. Classification has been achieved using the same library and through the SVM classifier. This method is used to demonstrate the ability of the SVM classifier to discriminate between emotional states directly on the audio domain. The audio features used to train the aforementioned SVM classifier are shown in Table 3.12.

Feature ID	Feature Name	Description
1	Zero Crossing Rate	The rate of sign-changes of the signal during the duration of a particular frame.
2	Energy	The sum of squares of the signal values, normalized by the respective frame length.
3	Entropy of Energy	The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes.
4	Spectral Centroid	The center of gravity of the spectrum.
5	Spectral Spread	The second central moment of the spectrum.
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames.
7	Spectral Flux	The squared difference between the normalized magnitudes of the spectra of the two successive frames.
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
9–21	MFCCs	Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the mel-scale.
22–33	Chroma Vector	A 12-element representation of the spectral energy where the bins, represent the 12 equal-tempered pitch classes of western-type music (semitone spacing).
34	Chroma Deviation	The standard deviation of the 12 chroma coefficients.

Table 3.12. Audio-based handcrafted features used to train an SVM classifier with pyAudioAnalysis.

- *image-based SVM*: an SVM classifier applied on hand-crafted image features has also been evaluated. In particular, the following visual features have been used to represent the spectrogram images: histograms of oriented gradients, local binary patterns and color histograms. The training images used to build the SVM model were exactly the same as the ones used to for our CNN approach.

The goal and the major contribution of this work with regards to its experimental evaluation is to estimate the performance of the proposed approach in the task of emotion recognition when training and testing datasets come from different domains and/or languages. Towards this end, the average F1 measure is used as an evaluation

metric, due to its ability to be unbiased to unbalanced datasets. Table 3.13 presents the experimental results in terms of the achieved F1 score within the testing data of the proposed emotion classification approach, compared to the audio-based classification and the image-based classification with hand-crafted features as explained above. The conclusions directly drawn from these results are the following:

- CNN_EM is the best method with respect to the average cross-dataset F1 measure. Audio-based classification is 1% lower, while the SVM classifier on hand-crafted visual features achieves almost 5% average F1 measure.
- CNN_EM is the best method for 9 out of 16 in total classification tasks, while audio-based classification is the best method in 5 of the classification tasks.
- CNN_EM, which operates directly in the raw data, is more robust across different domains and languages and can be used as an initialization point and/or knowledge transferring mechanism to train more sophisticated models.

In Figures 3.11 and 3.12 we illustrate how the filters of the first convolutional layer were shaped after the learning process. Feature extraction is then based on the final weight values of those filters. Darker regions correspond to the most important learned weights while brighter ones have a lower impact on the convolution outcome.

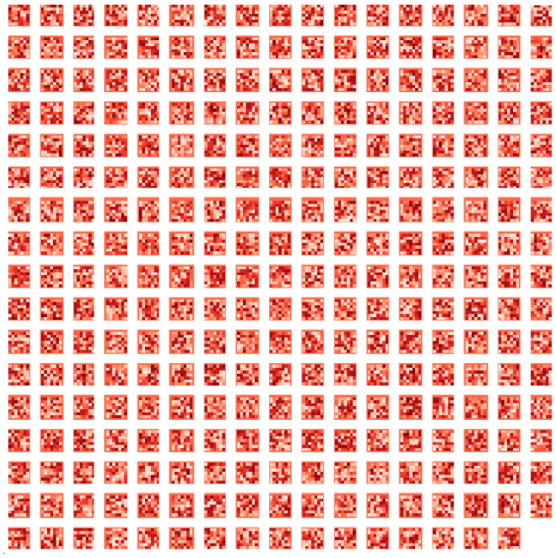


Figure 3.11. All learned filters of the first convolutional layer..

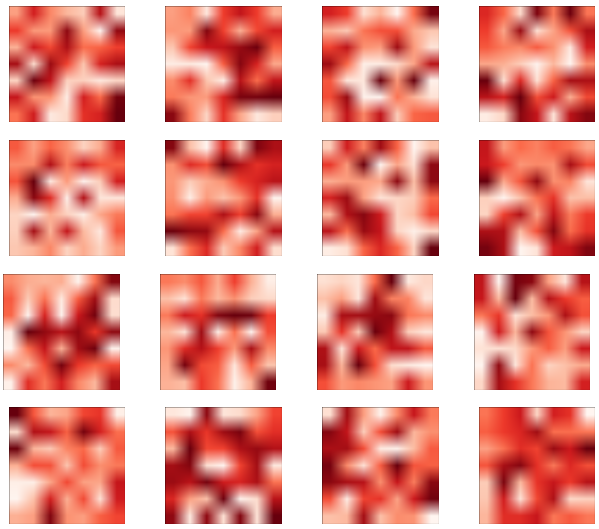


Figure 3.12. Randomly selected filters from the first convolutional layer as configured after the learning process. Darker regions correspond to the most important learned weights while brighter ones have a lower impact on the convolution outcome..

		Audio-Based SVM				Image-Based SVM				CNN_EM			
		Test Dataset											
		Emovo	Savee	German	Movies	Emovo	Savee	German	Movies	Emovo	Savee	German	Movies
Training Dataset	Emovo	0.48	**0.22**	**0.49**	**0.28**	0.42	0.14	0.42	0.20	**0.57**	0.16	0.42	0.27
	Savee	0.29	0.57	**0.34**	0.26	0.21	0.32	0.28	0.26	**0.30**	**0.60**	0.33	**0.31**
	German	0.41	**0.26**	0.64	0.32	**0.43**	0.25	0.68	0.29	0.41	0.24	**0.67**	**0.35**
	Movies	0.26	0.22	0.33	0.29	0.27	0.23	0.33	**0.30**	**0.29**	**0.24**	**0.42**	0.23
Average F1		0.35				0.31				0.36			

Table 3.13. Experimental results indicating the testing error of the proposed method and comparisons to other methodologies. Each row indicates the training and each column the testing set used. We mainly focus on the average F1 measure as the final evaluation metric, due to its ability to be robust against unbalanced datasets. Numbers in bold indicate which method achieved highest performance in each experiment.

To highlight the superiority of the proposed CNN architecture against other deep-learning-based approaches we conducted two additional experiments where we compare our method against current state-of-the-art methods.

In Table 3.14 we show how CNN_EM compares against the work in [123] when RAW spectrograms are used *without performing semi-supervised feature selection or any other kind of post or pre-possessing.* As in [123] we performed 3 different experimental setups.

1. *Single-Speaker* : where training and testing sets correspond to a single speaker
2. *Speaker-Dependent* : where samples from multiple speakers are used for training and testing takes place on different samples which belong to the same set of speakers
3. *Speaker-Independent* : where samples from multiple speakers are used for training and testing takes place on samples which belong to a different set of speakers

We evaluate on the two datasets that are common ground between the two works, namely the Savee and German (Emo-DB as referenced in [123]). For comparison purposes, we evaluate our method on the original versions of the respective datasets, i.e., without data augmentation. As the results indicate, CNN_EM significantly outperforms their approach in all cases when RAW spectrograms are used as an input to the structure.

	Savee		**German**	
	Huang et al.	**CNN_EM**	**Huang et al.**	**CNN_EM**
Single Speaker	0.31	**0.45**	0.41	**0.58**
Speaker-dep	0.29	**0.55**	0.37	**0.67**
Speaker-ind	0.27	**0.44**	0.36	**0.69**

Table 3.14. Comparison of our scores against the results reported in [123], when evaluated on the RAW spectrograms. Numbers in bold indicate which method achieved highest performance in each experiment.

In Table 3.15 we compare CNN_EM's scores against the results reported in [124], on the IEMOCAP Database [125]. We evaluated CNN_EM on the same set of target classes as in [124] (*excitement*, *happiness*, *frustration*, *neutral* and *surprise*) and by splitting the data into training and testing sets in a 80% to 20% ratio per class respectively, as reported in their work. CNN_EM outperforms their best results by 2% without any kind of additional pre-processing in contrast to [124].

Approach	Test Accuracy
Zheng et al.	0.40
CNN_EM	**0.42**

Table 3.15. Comparison of our scores against the results reported in [124], when evaluated on the IEMOCAP Database [125]. For comparison purposes we evaluated our method on the same set of target classes as in [124], which are : *excitement*, *happiness*, *frustration*, *neutral* and *surprise*. We follow a similar evaluation process as reported by Zheng et al, by choosing randomly 80% utterances of each emotion classification to construct the training dataset, the other 20% utterances for test.

3.5.1.4 Takeaways

In this work, we presented an approach that does not require any low-level features. Instead, it uses a Convolutional Neural Network, trained using raw speech information encoded as a spectrogram image. We compare the proposed approach using cross-language datasets and demonstrate that it is able to provide superior results vs. traditional methods that use either audio-based or image-based hand-crafted features.

Our evaluation showed that modern deep learning approaches and especially CNNs, which have been traditionally used for image retrieval problems, have the potential to produce breakthrough results in cross-modality problems. A viewing that can have a massive impact in behavioral modeling applications, given the complexity of the observed patterns across the various interaction signal. Language-Independent emotion recognition is an extremely complex problem even for humans. The final results highlight the great potentials of deep classifiers towards modeling ambiguous problems related to human behavior but also indicate once again their great need of vast amounts of training data.

3.6 Advantages, Limitations & General Observations

As confirmed by all the experiments discussed in Chapter-3, CNNs have an apriori advantage against traditional ML methods and that is due to their computational structure. Moreover, the produced models proved to be very robust across environmental and inherent channel noise but also across fundamental differences in the nature of the input data.

However it was made clear in all cases that the availability of training data played a central role towards using deep models and when this requirement is not satisfied, exploiting the potentials of such architectures might be unattainable or worthless. We showed that transfer-learning and data augmentation are two solutions that can potentially smoothen that effect. Although applying those techniques and especially the process of data augmentation, is not a trivial task. The nature and the native characteristics of the data must be taken into account and very carefully examined under a specific experimentation scenario in order to create accurate and useful simulations.

As discussed in Chapter-1 and presented through experimentation in Chapter-2, collecting such amounts of data in most real-life scenarios that relate to behavioral modeling using sensors and wearables, is many cases is not feasible. Lack of sufficient resources and the time required to design an experiment and eventually collect and annotate the data turns out to be a very demanding and costly process. Especially as the experimental conditions become less controlled. Hence, the design of systems that can combine diverse resources and computational algorithms remains the best solution towards designing efficient HCI systems and no end-to-end architecture can provide universal solutions to any such problem yet.

This observation becomes even more clear in the next three chapters where we address the problem of modeling cognitive and physical fatigue and their effects on user performance. As it will become clear, representing very open-ended problems restricts our ability to collect vast amounts of data that can be modeled by a single classifier and more flexible solutions like the ones presented in Chapter-2 need to be deployed.

CHAPTER 4

FATIGUE DETECTION FOR SMART REHABILITATION AND SAFER INTERACTION

4.1 Introduction

In all past chapters, we investigated how technology and ML, in particular, can be used to understand various aspects of human behavior, primarily related to action detection and emotion recognition. An important observation that can be made is that a common characteristic of all the applications discussed so far was the fact that they were targeting a clearly defined and intuitive goal, that would be very hard to get miss-interpreted by a human observer. However, ML technology can potentially provide insights for problems that we haven't completely comprehend and decipher yet. Such a problem is the concept of fatigue and how its effect on the human body can influence our performance.

Fatigue can have several consequences and may be expressed both in a physical level, in the form of muscle exhaustion or body pain, and in a mental level, as the inability to perform well in specific cognitive functions [126]. Despite the fact that is probably the most popular symptom across a variety of chronic diseases and a very common phenomenon in our daily living, science is still unable to quantify fatigue [127]. Moreover, it is very difficult to distinguish between effects caused by physical or cognitive fatigue in real-life scenarios and also to understand at which point a user under-performs due to the presence of fatigue [128].

Addressing these questions can help us design smarter interaction systems, able to prevent accidents caused by fatigue under high-risk situations [129]. Moreover un-

derstanding user fatigue is a parameter of major priority towards implementing more effective and user-centric smart-rehabilitation scenarios that can tailor their parameters based on user's physiological state [130]. In the rest of this dissertation, we will focus on ML methods designed to detect human fatigue and predict user performance. Starting from the following sections, and inspired by the findings presented during the previous chapters, we discuss how modern technology can be used to detect signs of both physical and cognitive fatigue and how such systems can be applied in smart rehabilitation and cognitive assessment scenarios.

4.2 Recognizing Fatigue - Collection & Analysis of Multi-Sensing Data

In the rest of Chapter-4 we present a framework for predicting physical and cognitive fatigue trough task-based evaluations [131] and we show how we can exploit machine learning towards detecting signs on physical fatigue [132]. In particular, we evaluate an upper-limb rehabilitation scenario using a robotic arm aiming to assess muscle fatigue. Our implementation takes into account both subjective user reports and objective physical measurements captured through wearable technology. Our initial experimentation, which is based on the principles discussed in the framework presented in Section-4.2.1, shows that despite the vague and subjective nature of fatigue, ML can be potentially used as a tool towards identifying universal patterns of fatigue across users long before the participants reach their endurance limits.

4.2.1 Towards a task-driven framework for multi-modal fatigue analysis during physical and cognitive tasks

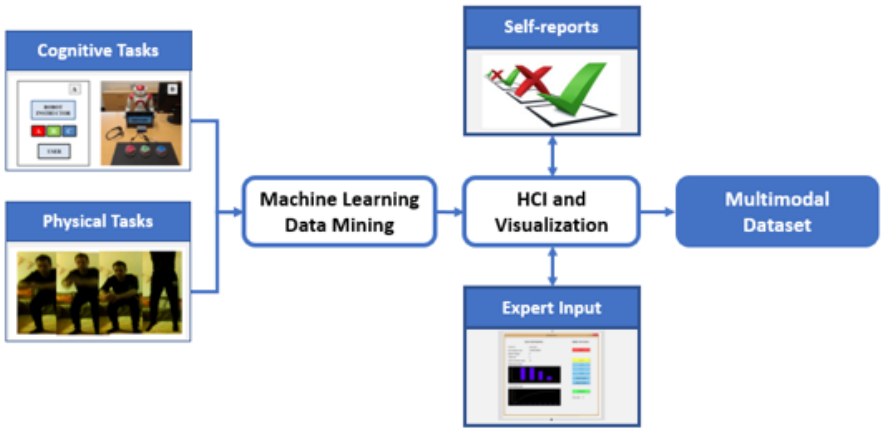
We present a multi-modal framework for data collection towards assessing and analyzing fatigue (Figure-4.1). Our goal is to combine both objective and subjective

reporting mechanisms, emphasizing on implicit self-reporting mechanisms through the use of sensors. In particular, we propose a task-driven approach that aims to extract both cognitive and physical behavioral patterns that may signal physical and/or mental fatigue, while the user is involved in a set of different tasks. Goal of the framework is to combine different measures extracted through non-invasive sensors and associating these data with various types of self-reporting mechanisms, such as post-task questionnaires or real-time feedback.

The main goal is to propose methods and models for multi-modal fatigue analysis, detection and prediction during physical and cognitive tasks. For the development of the framework, we follow a task-driven approach; data will be collected during both physical and cognitive tasks, specifically designed to extract behavioral patterns related to fatigue, as well as its effects on human performance. Task-related metrics (e.g. difficulty level and task duration) will be combined with behavioral, physiological and performance related data. This integration will lead to a multi-modal dataset/s that can be eventually used to better understand correlations between user behavior, performance, and fatigue.

4.2.2 Predicting Physical Fatigue Using EMG wearables and Subjective User Reports - A Machine Learning Approach Towards Adaptive Rehabilitation

In this work, we propose a novel method towards predicting physical fatigue. We design our approach based on objective EMG measurements and we aim to identify the presence of physical fatigue based on subjective user-reports. Based on our analysis we highlight the significance of our findings and we discuss how machine learning based modeling can become useful towards understanding fatigue and designing adaptive rehabilitation scenarios.



Task Parameters	User Behavior	Physiological Measures	Subjective Measures
Task Name	*Task Performance*	*EEG*	*Self-Report (Difficulty)*
Task Type	*Reaction Time*	*EMG*	*Self-Report (Fatigue)*
Task Difficulty	*Completion Time*	*Heart Rate*	*Self-Report (Performance)*
Task Duration	*Errors*	*ECG*	*Expert Recommendations*

Figure 4.1. The proposed framework for multi-modal data collection during cognitive and physical tasks for fatigue analysis. Below, an example of possible data captured in the context of our proposed framework. The framework combines data related to (a) Task parameters, (b) User behavior, (c) Physiological measures, and (d) Subjective measures, for multi-modal fatigue analysis..

Physical fatigue is one of the most common symptoms across a great variety of medical conditions, ranging from stroke and multiple sclerosis to chronic insomnia and myoskeletal injuries [133]. However, understanding, quantifying and predicting events of fatigue is a topic that remains vastly unexplored, primarily due to the subjective nature of the term. Each individual experiences fatigue in a very personal way that varies on its intensity and is affected not only by someone's physiological state but also by subjective factors such as emotion, which are very difficult to detect with certainty [134]. Our inability to capture and predict such events efficiently can lead to negative outcomes when it comes to physical rehabilitation since it increases the chance of causing unwanted injuries and muscular exhaustion. This realization becomes even more important when it comes to autonomous rehabilitation systems and the need

to design adaptive systems that match user's skills. Under that scope, understanding physical fatigue has become an area that attracts great research interest due to its importance towards achieving effective rehabilitation [135]. During the last twenty years, numerous works have been published that proposed modeling-methods and features to capture meaningful information from EMG[136, 36]. However, there is still no general truth on what is the most efficient way to model such signals [137].

In the following sections, we present an extensive analysis and evaluation of different machine learning algorithms towards predicting physical fatigue, on a human-robot rehabilitation scenario. We exploit statistical features that have been traditionally used in EMG and/or audio analysis and we propose a post-processing method that significantly improves the results provided by the original models. We use Delsys, a non-intrusive wearable EMG sensor and we build ML models targeting user-reported labels on physical fatigue. Our analysis focuses on evaluating the robustness and generalizability of such models across different users and exercises. Due to its computational simplicity, our method is ideal for running in real-time scenarios. The code and the data used for this work can be downloaded for free for purposes of reproducibility and further experimentation[1].

4.2.2.1 Data Collection & Experimental Setup

A study was conducted that involved 10 male and female subjects with a mean age of 26.3 years old. The subjects were asked to perform 3 exercises; shoulder flexion (SF), shoulder abduction (SA) and elbow extension (EE) Figure-4.2. These exercises were performed using the Barrett WAM arm, which is capable of applying feedback forces to the subject. The subjects were asked to hold the end-effector of the arm while performing each exercise. Two positions were important in each exercise; start

[1]https://github.com/MikeMpapa/EMG_Fatigue_Monitoring

position and the end position. For each exercise, the subjects would start from the start position and move the end-effector to the end position. The subjects were asked to hold the end-effector at that location so as to induce isometric contraction in the muscle. During this process, the robotic arm would provide resistive forces to the subject. EMG data were collected from the major muscles responsible for the movement. In SF and SA, EMG data were recorded from the deltoid and in EE from the triceps. The subjects were asked to hold the end-effector until they start to feel fatigued. When that occurred, they would inform the researcher conducting the experiment who would mark the time point. After the pass of almost 10 sec of the time the subject reported fatigue, the researcher would ask them to go back to the start position to complete the exercise. Subjects were asked to perform 3 repetitions of each exercise. A short period of rest was provided between each exercise to mitigate the cascading effect of fatigue. In total we collected 10 users $\times$ 3 exercises $\times$ 3 repetitions = 90 EMG recordings.

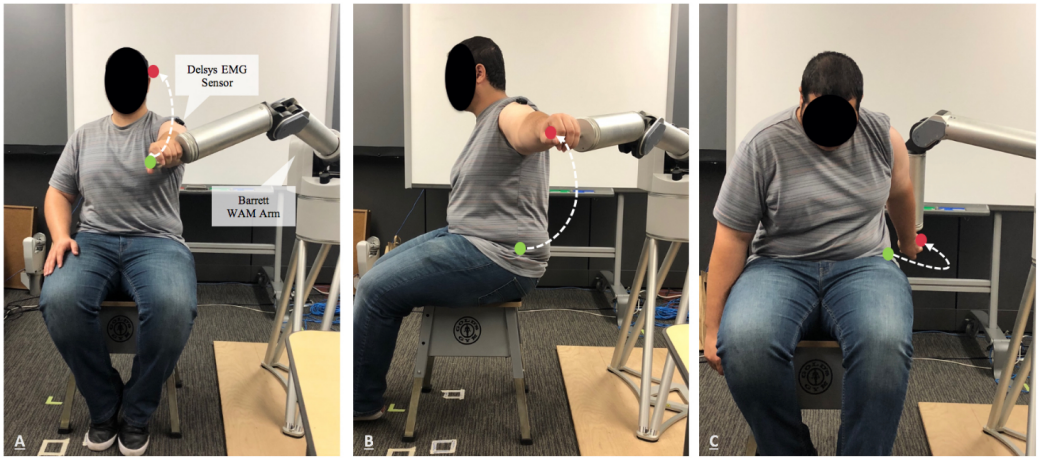


Figure 4.2. A- Shoulder Flexion B- Shoulder Abduction C- Elbow Extension. The green circles indicate the start positions and the red circles indicate the end positions.

4.2.2.2 Methodology

The Delsys EMG wearable sensors provided a sampling frequency of 1926HZ. As a first step and in order to reduce the inherent noise of the EMG recordings we filtered the signal using the median filtering technique with a window size of 11 samples. Using those filtered signals as input to our algorithm, short-term features were extracted from the time and spectral domain, which are then re-modeled in a mid-term fashion. The final feature vectors extracted from the mid-term windows were used as input samples to the classification algorithms. Targeted labels were the user-reported, binary indications of fatigue (0 meaning no-fatigue and 1 meaning fatigue). Thus, a valid set of labels as provided by the user would have the following form: [0,0,0,0,0,0,...,1,1,1,1,1,1,1,1], where each label corresponds in a sample captured by the EMG sensors (ie. 1926 labels per second).

4.2.2.2.1 Signal-Prepossessing

For splitting the EMG signals into short and mid-term windows, empirical window sizes were used, based on the fact that muscle fatigue changes are observed relatively slow. Short-term non-overlapping windows were extracted with a length of 0.25 sec, while overlapping mid-term windows were extracted capturing 2 sec of EMG information with a window step of 1s.

4.2.2.2.2 Feature Extraction

As explained in Section-3.5.1.2, a two-step feature extraction process was held in order to model the raw EMG information. Firstly a descriptive set of short-term features was extracted from each short-term window and then based on those features a set of statistical mid-term features was extracted to create the final feature vectors (FVs).

Based on extensive literature review on handcrafted feature extraction for effective EMG signal representation [136, 137], for every 0.25 sec short-term window we extracted the following list of features:

1. **Spectral Entropy**: Entropy of the normalized spectral energies for a set of sub-frames (equation-2.15)
2. **Spectral Flux**: The squared difference between the normalized magnitudes of the spectra of the two successive frames (equation-2.16)
3. **Spectral Minimum, Maximum, Standard Deviation & Mean**
4. **Minimum, Maximum, Standard Deviation & Mean** values of the time domain in a specific frame
5. **Zero Crossing Rate** : The rate of sign-changes of the signal during the duration of a particular frame. (equation-4.1)
6. **Energy Entropy**: The entropy of sub-frames normalized energies. It can be interpreted as a measure of abrupt changes (equation-2.12)
7. **Willson Amplitude (WAMP)**: The number of times that the difference between two consecutive amplitudes in a time segment becomes more than a threshold. WAMP can be seen as an indication of muscle contraction levels.

$$WAMP = \sum_{t=1}^{N-1} f(x_i - x_{i+1}) \tag{4.1}$$

, where $f(x) = \begin{cases} 1 & \text{if } x > threshold \\ 0 & \text{otherwise} \end{cases}$ and N is the length of the signal. This feature is an indicator of firing motor unit action potentials (MUAP) and therefore an indicator of the muscle contraction level.

These features are in general known for their ability to describe core characteristics of 1-D signals such as Accelerometer axis-based analysis or audio modeling and

have been proven quite informative in the past for the specific purposes of EMG classification. Especially features stemming from the time domain such as Zero-Crossing Rate, Energy Entropy and WAMP amplitude have shown great potentials for capturing EMG based patterns. However, an in-depth analysis of EMG feature selection is out of the scope of this work and feature selection was mainly inspired based on the related literature and our experimental analysis.

At every step, in addition to the features extracted from the current short-term window, we compute the deltas between the present set of features and the set of features extracted from its preceding window. Thus, describing each short-term frame with a set of 26 values (13 features from the current window plus 13 deltas).

For the mid-term window extraction, each set of 8 successive short-term FVs is described using the minimum, maximum, standard deviation and mean information extracted for each short-term feature. Hence, producing a final feature vector of 4*26 = 104 values.

During our experimentation, other features were also evaluated like signal energy, spectral spread (ie. the second central moment of the spectrum), spectral rolloff (ie. the frequency below which 90% of the magnitude distribution of the spectrum is concentrated) and spectral centroid (the center of gravity of the spectrum). However, they were omitted from our final evaluation since they didn't seem to have a significant effect on the final outcome.

4.2.2.2.3 Classification

For classification purposes, we experimented with a set of five traditional ML algorithms that have been extensively used for signal processing and EMG modeling in particular [136, 36, 138]. More specifically we evaluated the performance of the fol-

lowing methods: *Linear SVM, SVM with an RBF Kernel, Gradient-Boosting (GB), Extra-Trees (ET) and Random Forests (RF).*

4.2.2.2.4 Post-Processing

Our initial experimentation indicated that the original methods were usually failing to correlate the EMG information to the actual labels provided by the users, as they were often achieving an Average F1 lower than 70% and in many cases just slightly higher than 50% (ie. very close to random choice). Keeping in mind that classification takes place in a mid-term window basis, this made it impossible to consistently track fatigue in a long-term sequence as the algorithm would produce labels that were very hard to interpret. For example assuming again that 0 indicates 'NO-FATIGUE' and 1 indicates 'Fatigue' a possible output sequence would look like ['0,1,1,0,0,1,0,...,0,1,1,1,0,1,0]. Thus, we developed our own post-processing method that builds upon the decisions of the initial classifiers and re-evaluates their decisions by keeping track of the N past mid-term labels assigned by the model.

In particular, as a first step, we apply a median-filter of size K to the original predictions made by the classifier. Then the method gathers the successively assigned labels into groups of M. If in the N past groups, the total number of samples that have been identified as 'FATIGUE' exceeds a specific threshold, then and only then the method decides that the subject has shown signs of fatigue. Otherwise, it assumes that the classification algorithm found a set of false positives and the process continues as if the subject has not been fatigued. Using this kind of post-processing the final output of our method has the following form ['0,0,0,0,0,0,0,...,1,1,1,1,1,1,1'] and provides significantly higher classification performance in all cases, as we will discuss in Section-4.2.2.3.

In Algorithm-2 we show the pseudo-code of the proposed post-processing technique and in Algorithm-3 we show the whole fatigue detection framework again in the form of pseudo-code. Figure-4.3 illustrates the overall system architecture.

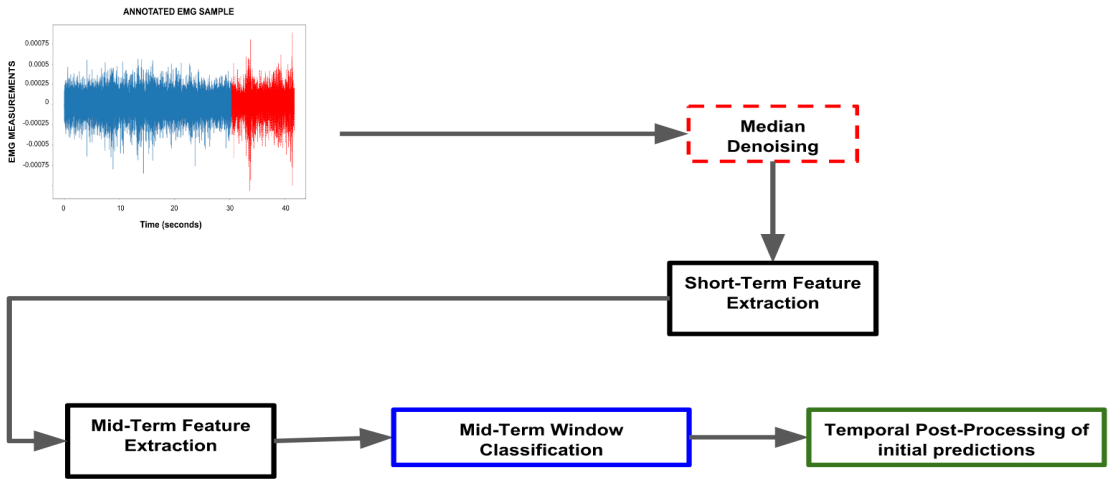


Figure 4.3. The overall system architecture. Blue and red EMG values correspond to NO-FATIGUE and FATIGUE ground truth labels respectively..

In our implementation, hyper-parameters were set to $K1 = 3$, $M = 3$, $STEP = 1$, $N = 2$, $THRESH\_VAL = 0.6$ and $K2 = 11$. For reproducibility and reusability purposes our code along with the data that used for the purposes of this study can be found and downloaded for free at github[2]. The hyper-parameters of each classifier were tuned using an exhaustive grid-search approach. It has to be noted that in terms of time delay's the algorithm makes a decision equal to the step-size of the mid-term frame (1s in our implementation), with only exception its first decision that takes place 2 sec after the recording has started.

[2]https://github.com/MikeMpapa/EMG_Fatigue_Monitoring

Algorithm 2 EMG Post-Processing Algorithm

$filtered\_labels = median\_filter(original\_predictions, K1)$

$group\_size = M$

$group\_step = STEP$

$thresh = THRESH\_VAL$

$prev\_window\_1, ..., prev\_window\_N = None$

$x1 = 0$

$x2 = group\_size$

while true **do**

    $current\_window = filtered\_labels[x1 : x2]$

    $t1 \leftarrow \frac{(current\_window='FATIGUE')}{group\_size} >= thresh$

    ...

    $tn \leftarrow \frac{(prev\_window\_N='FATIGUE')}{group\_size} >= thresh$

    if $(t1 \ \& \ ... \ \& \ tn) == TRUE$ **then**

        $state =' FATIGUE'$

        return $state$

    $prev\_window\_1 = current\_window$

    ...

    $prev\_window\_N = prev\_window\_(N - 1)$

    $x1 = x1 + group\_step$

    $x2 = x2 + group\_step$

Algorithm 3 Fatigue Detection Framework

1: $filtered\_signal = median\_filter(original\_signal, K2)$

2: $st\_features \leftarrow st\_feature\_extraction(RAW\_EMG)$

3: $mt\_FV \leftarrow mt\_feature\_extraction(st\_features)$

4: $Prediction \leftarrow Classifier(mt\_FV)$

5: $Fatigue\_Prediction \leftarrow Algortithm1(Prediction)$

4.2.2.3 Experimental Results

To examine the the robustness of the proposed method in capturing subjective-fatigue, we perform 4 different types experiments. In all our experiments we evaluate our method in terms of Precision (Pr), Recall (Rec) and F1 measures (equations-2.19, 2.18, 2.20). In all the following results 'NF' indicates the 'NO-FATIGUE' label and' F' corresponds to 'FATIGUE'.

4.2.2.3.1 Cross-User Evaluation

In the first experimentation, we perform a *Cross-User Evaluation*. A leave-one-out cross-validation technique was applied with respect to different users. At each step, all recordings from 9 users were used for training and all recordings of the remaining user were used for testing. We performed this process 10 times, each time using a different user for testing (Table-4.1). Using the proposed post-processing method, significantly improved the final results in terms of Average F1 in all cases. *Original Classification results in terms of Average F1 were: SVM = 60.6, SVM_RBF=55.4, GB=57.5, ET=54 and RF=53.7.* GB provides the best results in this evaluation, which are however directly comparable to the results provided by the SVM classifier.

	SVM	SVM-RBF	GB	ET	RF
Pr NF	70.2	75.6	64.4	62.3	58.4
Pr F	70.8	66.1	76.6	66	62.8
Rec NF	56.8	40.7	73	48.7	40.8
Rec F	81.3	89.8	68.7	77.1	77.5
F1 NF	62.8	52.9	68.4	54.6	48
F1 F	75.7	76.2	72.4	71.1	69.3
AVG F1	69.2	64.5	**70.4**	62.9	58.7

Table 4.1. Average Performance Results on Cross-User Evaluation

4.2.2.3.2 Cross-Exercise Evaluation

In the second experimentation, we aimed to evaluate the robustness of the proposed method across different exercises. Similarly as before a leave-one-out cross-validation was performed, but now in terms of different exercises. Thus, all samples, from all users that belong to single exercise were used for testing and all the rest were used for training. We repeated this process 3 times and we averaged the final results (Table-4.2). *As in the case of cross-user evaluation, post-processing significantly improved initial classification results, where the Average F1 was: SVM = 58.6, SVM_RBF=54.2, GB=54.4, ET=53 and RF=53.7.* In this scenario, SVM provides by far the best classification results.

4.2.2.3.3 Single User Evaluation

In this scenario we perform 10 different evaluations, each time using only the recordings that belong to a single user (ie. 3 exercises $\times$ 3 repetitions = 9 recordings). For each user, the evaluation process was the same as before, where we ran the classification algorithm 9 times, each time using 8 recordings as training and the remaining as testing. Table-4.3 shows the averaged results across all 10, user-based evalua-

	SVM	SVM-RBF	GB	ET	RF
					59.1
Pr NF	63.4	56.9	63.4	60.5	59.1
Pr F	74.3	67	69.4	68.2	67.4
Rec NF	65.5	58.3	57.8	57.4	56.5
Rec F	77.6	65.8	74.1	71	69.7
F1 NF	67.4	57.6	60.5	58.9	57.8
F1 F	75.4	66.4	71.6	69.6	68.5
AVG F1	**71.7**	62	66.1	64.2	63.1

Table 4.2. Average Performance Results on Cross-Exercise Evaluation

tions. Again after post-processing the initial prediction, results were significantly improved in terms of Average F1. *Initial classification performance was: SVM = 65.2, SVM_RBF=68.5, GB=70.1, ET=72.4 and RF=71.5.*. Here in contrast to previous evaluations ET and SVM_RBF provide slightly better results than the other three classification methods.

	SVM	SVM-RBF	GB	ET	RF
Pr NF	75.4	77.5	72.3	73.5	71.2
Pr F	76.2	78.5	77.9	82	79.6
Rec NF	66.6	70.3	71.1	77.8	74.7
Rec F	83.2	84.2	78.9	78.2	76.7
F1 NF	70.7	73.7	71.7	75.6	72.9
F1 F	79.6	82.3	78.4	80	78.1
AVG F1	75.1	77.5	75.1	**77.8**	75.5

Table 4.3. Average Performance Results on Single-User Evaluation

4.2.2.3.4 Single Exercise Evaluation

Our final evaluation targets to recognize fatigue based only on samples that belong to a single exercise. Similarly, as in the previous case, we performed 3 different evaluations, one for each exercise. In each evaluation we used all recordings from all users that belong to the specific exercise (ie. 10 users $\times$ 3 repetitions = 30 recordings). For each evaluation, we followed again a leave-one-out approach where we used 29 recordings as training and 1 as testing and we repeated the process 30 different times for each exercise (each time using a different recording for testing). Table-4.4 shows the averaged results across all 3 exercise-based evaluations. Post-processing results are again superior compared to the initial classifications in terms of Average F1. *Original classifier outputs without post-processing where: SVM = 61.5, SVM_RBF=65, GB=67.2, ET=68.5 and RF=66.8.*. Here all methods provide comparable results, but GB slightly outperforms the rest.

	SVM	SVM-RBF	GB	ET	RF
Pr NF	86.2	76.1	78.4	75.3	76.6
Pr F	71.2	76.3	76.2	74.7	76.3
Rec NF	56.5	66.5	68.2	63.4	66.1
Rec F	92.3	83.8	85.7	83.9	84.3
F1 NF	68.3	71	72.2	68.9	71
F1 F	80.4	79.9	81.1	79	80.1
AVG F1	74.3	75.4	**76.6**	74	75.5

Table 4.4. Average Performance Results on Single-Exercise Evaluation

4.2.3 Overall Classification Improvement

In Figure-4.4 we show the % improvement of the classification results for each evaluation scenario after applying the post-processing temporal modeling method described by Algorithm-2. As the results indicate after the application of Algorithm-2, classification results showed significant improvements for all tested classifiers. Maximum improvement with a magnitude of 13.1% is shown for the SVM classifier for the E2 evaluation scenario while minimum improvement was 4% for the RF classifier for the E3 evaluation.

Initial Classification Results (AVG F1)

	SVM	SVM_RBF	GB	ET	RF
E1	60.6	55.4	57.5	54	53.7
E2	58.6	54.2	54.4	53	53.7
E3	65.2	68.5	70.1	72.4	71.5
E4	61.5	65	67.2	68.5	66.8

Classification Results After Temporal Post-Processing (AVG F1)

	SVM	SVM_RBF	GB	ET	RF
E1	69.2	64.5	70.4	62.9	58.4
E2	71.7	62	66.1	64.2	63.1
E3	75.1	77.5	75.1	77.8	75.5
E4	74.3	75.4	76.6	74	75.5

% Improvement of Classification after Temporal Post-Processing of Algorithm-1 (AVG F1)

	SVM	SVM_RBF	GB	ET	RF
E1	8.6↑	9.1↑	12.9↑	8.9↑	4.7↑
E2	13.1↑	7.8↑	11.6↑	11.2↑	9.4↑
E3	9.9↑	9↑	5↑	5.4↑	4↑
E4	12.8↑	10.4↑	9.4↑	5.5↑	8.7↑

Figure 4.4. % Classification Improvement in terms of Average F1 after applying the temporal post-processing method described in Algorithm-2.

4.2.3.0.1 Temporal Evaluation

To evaluate the efficiency of the proposed method in terms of temporal accuracy we performed a Paired Two One-Sided (TOST) equivalence test. Equivalence statistical

tests aim to validate the fact that a difference between two sets lies within a given interval. TOST is based on the classical t-test used to test the hypothesis of equality between two means. In particular, TOST performs two types of t-tests; one to verify if the difference is below a higher threshold and a second one that evaluates if the difference is higher than a lower threshold. In this work, the difference was calculated as $T_{reported\_fatigue} - T_{detected\_fatigue}$. The threshold, which was calculated through trial and error, signifies the maximum time difference between the subject feeling fatigued and the system recognizing the event. There were two thresholds considered; the lower threshold indicates delayed detection while the higher signifies early detection. Table-4.5 illustrates the TOST results.

Despite the fact that temporal evaluation for each exercise has been reported individually, it has to be noted that no safe results can be drawn due to the limited number of samples available. Only the last column (*Avg Performance*) can be considered for reliable evaluation purposes. The rest of the results are mainly reported for informational purposes since they can help draw useful insights. For the temporal evaluation of the method, we used only the models that provided the best performance in terms of average F1 for each of the four evaluations (E1-4).

According to the results of Table-4.5 building models based on multiple users (E1) or on samples related to a single exercise (E4) were the most effective ones both in terms of *Success Rate* (ie. percentage of exercise sessions that the system successfully detected fatigue) but also in terms of temporal accuracy. In other words, these models were able to generalize their results better compared to the rest.

In the case of E1, the algorithm must have been able to depict the most generic of patterns that can eventually apply in the majority of users. However, judging from the classification results provided in Section-4.2.2.3.1 in most cases the model must have been making the wrong predictions almost 30% of the times, which indicates

that in most scenarios the algorithm was at the limits of its temporal boundaries (ie. predictions where usually ±5 sec off).

On the other hand, in the case of E4, the model must have been able to capture similar behaviors across different users, when performing the same exercise. Along with the good performance reported in Section-4.2.2.3.4, such exercise-based models seem to be the best choices towards modeling physical fatigue using subjective reports, especially on users with similar physical characteristics.

In the other two training scenarios (E2 and E3) even-though temporal boundaries were relatively low (±6 sec), Success Rate was comparably low in both cases. For models based on E2, this comes along with the findings of Section-4.2.2.3.2 and indicates that such methods are in general unable to generalize across different exercises. According to these findings integrating exercise characteristics is very critical towards designing robust models since they remain constant and must be followed by all users in the same way. For models designed based on E3, the low Success Rate is an observation that comes to our surprise and further analysis needs to be done in the future. A possible reason for the contradictory findings between the high average F1 reported in Section-4.2.2.3.3 and the low Success Rate observed in Table-4.5 might be due to the fact that evaluation of Section-4.2.2.3.3 is averaged firstly across all the sessions performed by the same user and secondly across all the users in order to get the final aggregated results. Moreover, performance is reported as a total metric across all exercises. Hence, it is very possible that in most cases of E3 the algorithm provided high performance for some users and relatively low for others. Those kinds of differences cannot be sufficiently represented by the aggregated results, but are easily observable through the analysis provided by Table-4.5. Hence models built based on E3 cannot be considered trustworthy by default since they seem to be very depended by within user variations.

MODEL		EXERCISE SF	SA	EE	Avg Per.
Multi-User	#Failure Samples	6	4	13	**23**
	#Valid Samples	24	26	17	**67**
	Success Rate	80%	87%	57%	**74%**
	p_upper	*.019*	.024	<.001	**.015**
	p_lower	<.001	<.001	.018	**<.001**
	bounds (sec)	+-10	+-7	+-6	**+-5**
Multi-Exercise	#Failure Samples	7	9	16	**32**
	#Valid Samples	23	21	14	**58**
	Success Rate	77%	70%	47%	**64%**
	p_upper	.031	.034	.025	**.038**
	p_lower	.019	<.001	<.001	**<.001**
	bounds (sec)	+-4	+-13	+-8	**+- 6**
Single User	#Failure Samples	15	11	9	**35**
	#Valid Samples	15	19	21	**55**
	Success Rate	50%	63%	70%	**61%**
	p_upper	.013	.037	.026	**.013**
	p_lower	<.001	.025	<.001	**<.001**
	bounds (sec)	+-7	+-7	+-8	**+-6**
Single Exercise	#Failure Samples	11	7	6	**24**
	#Valid Samples	19	23	24	**66**
	Success Rate	63%	77%	80%	**73%**
	p_upper	.029	.019	.029	**.014**
	p_lower	<.001	<.001	<.001	**<.001**
	bounds (sec)	+-9	+-5	+-6	**+-5**

Table 4.5. Temporal Evaluation of the computed models. Multi-User model corresponds to the models built for E1, Multi-Exercise to E2, Single-User to E3 and Single Exercise to E4. Ex#1-3 correspond to the exercise SF, SA and EE respectively.*#Failure Samples* indicates cases (exercise sessions) where the proposed method failed to detect fatigue even though it was present according to user reports. *#Valid Samples* corresponds to the total number of exercise sessions that fatigue was successfully detected by the system. TOST was performed on the amount of samples indicated by #Valid Samples. *Success Rate* indicates the percentage of #Valid Samples over the total number of available sessions (30 sessions per exercise and 90 sessions in total); *p_upper* and *p_lower* correspond to the values of statistical significance that the system would detect fatigue within $\pm$*bounds* sec from the time user reported fatigue. Even-though per-exercise evaluation is also reported for each model, only useful insights can be drawn but no safe and generalizable results due to the limited number of samples used for statistical analysis. *Only Average Performance evaluation (bold) can provide representative and trustworthy evaluation for the temporal performance of the proposed methodology.*

4.2.3.1 Takeaways

In this work, we tried to tackle the very complex problem of recognizing physical fatigue based on subjective user reports and EMG information. We proposed a post-processing method that can be applied in real-time and seems to significantly improve classification results of traditional ML-based techniques in terms of fatigue-detection. Modeling subjective fatigue can be an extremely challenging problem due to the great variability between self-reports and actual EMG measurements across different users and scenarios. In addition, creating an objective measure of fatigue is still very premature as it is a factor that depends highly on each individual and his/her mental and physical state. Despite the aforementioned challenges, it seems that a combination of carefully selected features and classifiers can provide promising results towards targeting self-reported fatigue and can be very useful towards understanding shared behaviors across users. Such observations are crucial to unraveling the effects of physical fatigue in human performance and can help us draw valuable assumptions between the vague correlation of physical and cognitive fatigue [60].

4.3 Advantages, Limitations & General Observations

In this chapter, we introduced the concept of fatigue and we highlighted its subjective nature and how it can be affected by individual differences. Towards understanding patterns of fatigue across different subjects but also within the same persons we proposed a general, task-based framework that aims to capture correlations between performance and different types of fatigue (ie. physical and cognitive). In our initial experimentation, we showed that traditional machine learning methods can be used to extract and describe physiological patterns that can be used to detect physical fatigue. Our evaluation showed that these patterns, despite their simplicity

were in many cases highly correlated with user-reports about their personal perception of physical fatigue during physical exercising. In the next two chapters, we try to build upon those findings and extend our methods and experimentation by targeting factors and behaviors related to cognitive fatigue.

CHAPTER 5

BRAIN-COMPUTER INTERFACES & COGNITIVE ASSESSMENT

5.1 Introduction

Brain-Computer Interfaces (BCI) are interactive systems that can provide communication and environmental control between a person and an external device. With the rapid growth of AI and the ground-braking improvements provided in terms of hardware during the last decades, researchers have focused their efforts on advancing those systems into intelligent architectures able to assist populations suffering from various physical or mental impairments [139]. Figure-5.1 illustrates the general computational architecture of a BCI system. Despite the fact that BCI research has traditionally targeted the augmentation of mobility-related functionalities (Figure-5.2) [140, 141], latest efforts aim to expand the advantages of BCI systems in the area of behavioral and cognitive assessment [142, 143]. This latter domain has its roots in the context of neuro-feedback therapy and addresses patients with neurological developmental disorders such as autism spectrum disorder, attention-deficit/hyperactivity disorder, stroke patients or elderly subjects [144]. However, such systems do not limit their capabilities on disabled populations as they can act as cognitive state indicators for fully-functional subjects as well. Neuro-feedback signals can provide unique information for the purposes of adaptive interaction scenarios and can be exploited towards enhancing human performance and user engagement [61]. In this chapter, we present a ML method able to predict user performance in a cognitive task. Assessing cognitive performance and user skills through BCI and ML is a completely new research domain with great potential across a large spectrum of applications that

involve high-risk decision making or training and learning and are characterized by increased cognitive demands [145].

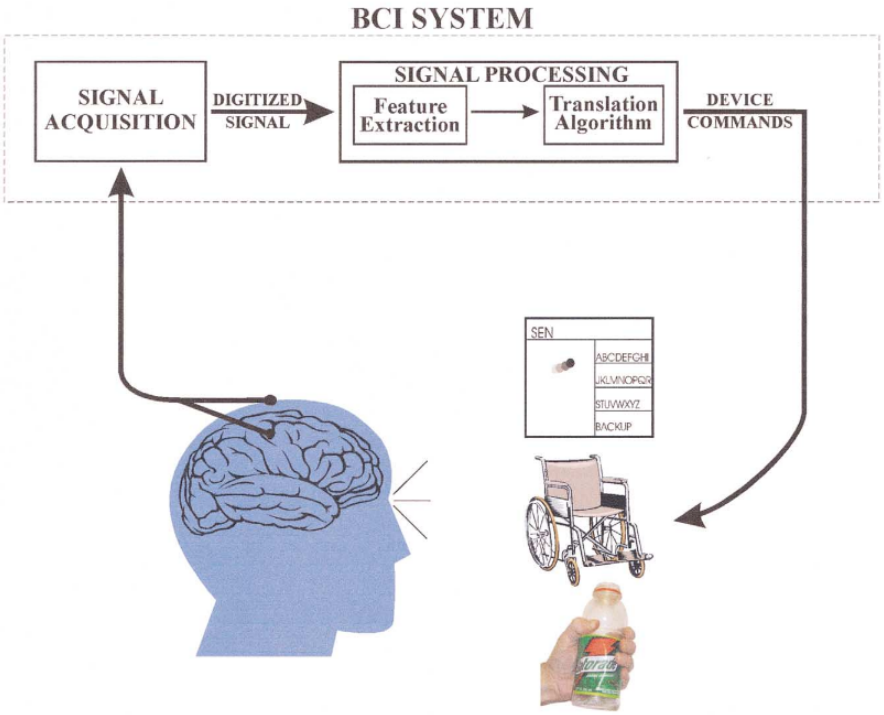


Figure 5.1. The general architecture of any BCI interface. Initially proposed by Wolpaw et al. in 2002 [139].

5.2 Predicting Cognitive Performance

As explained in the previous section, BCI-based prediction and analysis of cognitive performance powered by AI methods is still a very unexplored field of research with massive potentials. Cognitive performance assessment in real-time can have a tremendous impact on numerous application including but not limited to driver and machine operator assessment, medical doctor monitoring or cognitive rehabilitation for brain injuries such as stroke or Traumatic Brain Injury (TBI). [146, 147, 148].

Figure 5.2. Applications focused on augmenting human mobility skills. In **a** one of the most pioneering works in the domain proposed by Hochberg et al. in 2012 [140] shows a tetraplegic woman controlling a robotic arm through a BCI. In **b** [141] a man controls a wheelchair through hand gestures, recognised by a BCI system.

Moreover, such technologies may have a drastic effect on the way that current training and educational methods are performed, as they can provide unique fingerprints of user's cognitive-state as feedback towards designing adaptive environments that can better fit user skills and abilities [149]. Despite the fact that several methods have been proposed to asses cognitive aspects that can have an immediate effect on user performance, such as cognitive load or attention [150, 151], very few have tried to directly tackle the problem of performance prediction using wearables. That is partly because performance is by definition highly dependent on the evaluated task but also because the trade-off between signal-quality and invasiveness introduced by wearable sensors wouldn't allow application of such methods in real-life systems. However, in recent years off-the-shelf technology has drastically narrowed the gap between sensor-size and data-quality thus, allowing novel HCI methods to emerge.

5.2.1 Towards predicting task performance from EEG signals

Under the context discussed above, we propose a passive BCI system, that uses a wireless non-intrusive EEG sensor under a robot-assisted training task designed for cognitive assessment. As part of this work, we demonstrate our results on predicting user's task performance, from the EEG signals, before task completion. Our findings highlight the potentials of our hypotheses as we achieve a maximum accuracy rate equal to 74% when evaluated on 69 real subjects.

In particular, we test our system in a cognitive task designed to asses working memory. The task was initially proposed by Tsiakas et al. and was designed as a method to asses user engagement in an HRI training-scenario with a social robot [152]. The task selection was made to evaluate one of the most common cognitive skills of the human brain (working memory) which, is present in the vast majority of tasks someone would perform during a daily routine. Our experiments focus on predicting the user's task performance from the EEG data before the user completes the task. According to our knowledge, this is the first effort that aims to directly predict the user's performance from EEG in such a task. Initial results indicate that there is a clear correlation between the EEG measurements and the final outcome of the memory game and that there are potential patterns able to capture certain cognitive behaviors across different users.

5.2.1.1 The Sequence Learning Task

Sequencing is the ability to arrange language, thoughts, information, and actions in an effective order [153]. Extended research on the field of cognitive sciences has shown that sequence-learning tasks can be applied to evaluate human behaviors related to learning ability, short term memory, and attention [154, 155].

Towards this direction, we developed the Sequence Learning (SL) task; a working memory task that evaluates the ability of a human to remember and repeat a sequence of items (e.g., letters, numbers, actions) [152]. For our experimental setup, we deploy the NAO[1] robot as a socially assistive robot that instructs, monitors and evaluates user's performance during the task. While performing the SL task, users have three buttons in front of them ("A", "B", "C") and the robot asks the user to repeat a given sequence of these letters by pressing the corresponding buttons. The game consists of four difficulty levels where each level corresponds to a combination of 3,5,7 and 9 letters respectively. A complete session (human-robot interaction) consists of 25 turns/sequences. The level of each turn/sequence is decided randomly and all levels are equally distributed within a session. For the purposes of this research we considered a binary score at each turn, *success* or *fail*

5.2.1.2 Data Collection

For the data collection, 69 CSE undergraduate and graduate students from the University of Texas at Arlington were recruited. Each user completed a single session of the SL task (25 turns/sequences). During the task, EEG signals were recorded using the Muse EEG headset[2], a low-cost and non-invasive EEG wearable device which, has been used previously for similar research purposes [156]. The Muse provides 4 channels of data; two coming from the forehead and two from behind the ears. The EEG signals were generated at a sampling rate of 220Hz. The device provides direct access to raw EEG signals as well as to a set of specific EEG wavelengths that are known to describe specific information regarding brains activity. The frequency bands provided by the device are δ (1-4 Hz), θ (5-8 Hz), α (9-13 Hz), β (12-30 Hz)

[1]https://www.ald.softbankrobotics.com/en/cool-robots/nao

[2]http://www.choosemuse.com/

and γ (30-50 Hz). Extensive details regarding the available data can be found at [152, 157]. According to related literature δ waves provide information related to deep dreamless sleep when there is a lack of body awareness, θ waves are useful to describe deep mental states such as dreaming or deep meditation where subjects have reduced consciousness, α waves describe physically and mentally relaxed states of mind while β and γ can be used to describe awake and alert states of consciousness with heightened perception and are related to active thinking, excitement, learning, and increased cognitive processing[158].

At each turn of every session, we store separately user's EEG captured during the *listening process* (robot pronounces a new sequence) from the EEG collected during the *acting-process* (user repeats the sequence by pressing the buttons). In the following Paragraph, we describe our classification results on the task of predicting the user's final performance (fail/success) at a single turn, using only the EEG from the listening process. In Figure-5.3, we illustrate the experimental setup.

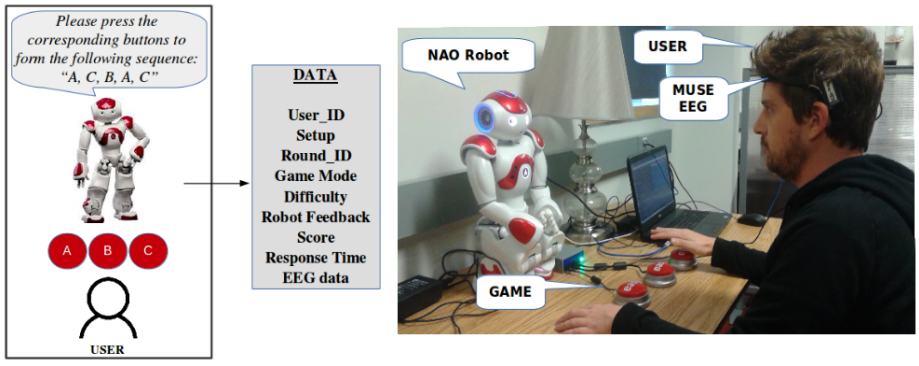


Figure 5.3. The Sequence Learning setup.

The original data and details of the SL task along with the processed data and the code for the proposed work are available online[3][4].

5.2.1.3 Results & Analysis

As explained in the previous Section, we exploit the EEG signals captured during the *listening process*, to predict the final outcome of a single turn of the SL task. For validation purposes, we perform a 10-fold cross-validation across all users. At each fold, 80% of the users (55 subjects) were randomly picked for training, and the rest were used for testing. From each user, 25 interaction results were available, equal to the total number of turns/sequences played within a session. In total, we had 1375 training samples and 350 testing samples available at each fold. The distribution of the samples across the two classes always depended on the personal performance of the users picked each time for training. Across the 10 folds, the average prior-probabilities for success and failure in a single turn/sequence were 60% and 40% respectively.

5.2.1.3.1 Feature Extraction

As discussed in Section-5.2.1.2, the Muse provides a set of frequency bands, extracted from the raw EEG in real-time through a digital signal processing component embedded in the device. For every frequency band, Muse estimates the absolute and relative band powers along with a band-power session score. According to Muse's documentation, the band session score is computed by comparing the current value of a band power to its history. Detailed information regarding the exact metrics and how they are estimated can be found at [157] and at Section-6.4.2.1. In total, for

[3]https://github.com/TsiakasK/sequence-learning

[4]https://github.com/MikeMpapa/EEG-Sequence-Learning

our experiments we exploited 15 different data streams, each coming from 4 different channels thus, ending up with an initial feature representation of size equal to $4 \times 15 = 60$. More specifically, from every channel the following data streams were analyzed; δ, θ, α, β and γ relative band powers, their respected absolute band powers and their session-score signals. From each of the 60 EEG feature-streams captured during the *listening process*, we extract the following statistical features:

- *Standard Deviation*
- *Mean Value*
- *Maximum Value*
- *Minimum Value*
- *Spectral Centroid (equation-2.13)*

  The center of gravity of the spectrum after applying FFT on the original signals.
- *Spectral Rolloff (equation-2.17)*

  The frequency below which 90% of the magnitude distribution of the spectrum is concentrated after applying FFT on the original signals.

The final feature vector representation consists of $60*6 = 360$ features, extracted from the EEG signals of a single subject and captured during the listening process, of a single turn/sequence of the SL task.

5.2.1.3.2 Classification

For classification, we experimented with 5 different classification methods; SVMs, SVMS with an RBF kernel, Random Forests (RF), Extra Trees (ET) and Gradient Boosting (GB). For tuning, the c parameter of each classifier and for training each classification method the implementation described at [85] was applied. Before feeding the training data into the classifier, features are normalized to have $mean = 0$ and $std = 1$. In Table-5.1, we show the classification results. Since the two versions of

SVM provided very similar results, we show only the linear-SVM evaluation as it was slightly superior. In all the cases, the estimated time required for a single prediction was on the scale of *milliseconds*.

	SVM		GB		RF		ET	
	S	F	S	F	S	F	S	F
Prec	0.75	0.48	0.81	0.56	0.89	0.24	0.91	0.2
Rec	0.69	0.55	0.78	0.6	0.69	0.54	0.69	0.54
F1	0.72	0.51	0.79	0.58	0.78	0.33	0.78	0.29
Acc	0.65		**0.74**		0.67		0.67	
AVG F1	0.62		**0.69**		0.56		0.54	

Table 5.1. EEG Classification Results

It is clear from the results that there is a significant statistical correlation between the EEG features and user's final performance. Despite the simplicity of the final features, the amount of captured information seems sufficient to provide a rough estimate with an average accuracy of 74% for the outcome of the task, when using a Gradient Boosting classifier.

5.2.1.4 Takeaways

We proposed a passive Brain-Computer Interface (BCI), using the Muse, a wireless non-intrusive EEG sensor under the scenario of the Sequence-learning task; a robot-assisted training task designed for cognitive assessment. Our preliminary results highlight a clear correlation between the user's brain activation and the actual outcome of the task, significantly before task completion. We evaluated our system on 69 real subjects following a user-independent modeling approach, ie. each user was considered either for training or for testing. Each interaction between the user

and the system was represented by a feature vector of 360 statistical features extracted from the 60 available data streams captured at each timestamp by the Muse. Gradient Boosting classification provided the best classification results achieving a maximum accuracy of 74with an average F1 of 69%. Our experiments highlight the great potentials of ML methods to model human performance in cognitive tasks.

5.3 Advantages, Limitations & General Observations

It is well known that cognitive performance is highly affected by fatigue. Even though the relationship between cognitive and physical fatigue is not clear yet, several research efforts have concluded through experimentation that intense physical activity can have a major impact on cognitive performance. In Chapter-4 we showed that machine learning based approaches are able to effectively incorporate and correlate objective and subjective information towards depicting robust signs of physical fatigue across users. Moreover in Chapter-5 our experiments indicated that similar modeling, when applied on EEG signals, are able not only to detect but also to predict behaviors of cognitive performance. Hence, it comes as a natural consequence our interest to explore the potentials of machine learning towards identifying patterns of relation between human's fatigue and cognitive performance.

During our experimentation on the aforementioned topics, it came to our attention that a major problem for targeting such problems was the lack of available data for research purposes. Thus, a direct problem that surfaced was our inability to effectively reproduce the conditions described by other research teams for data collection and analysis of such behavioral and physiological data for the purposes of prediction and recognition of fatigue and performance. Taking this very crucial fact into account in Chapter-6 we propose CogBeacon. The first multi-modal dataset specifically designed to address aspects of cognitive fatigue. In contrast to most pub-

lic datasets, CogBeacon comes along with an easy to install software that enables other researchers to reproduce the data collection process using minimal hardware equipment. Moreover, CogBeacon's software aims to enable researchers to expand the available data offered by the current version of the dataset hence, creating a more robust and complete collection. Finally, as an open-access software CogBeacon allows researchers to incorporate their own sensors towards investigating additional behavioral and physiological sources for cognitive fatigue analysis. This is very important as it eventually creates a data collection framework that is independent of specific hardware dependencies. A fact that is pivotal in a field such as behavioral monitoring using wearables, where technology evolves so rapidly and hardware devices are being updated and replaced by new ones in an extremely intense rate.

CHAPTER 6

COGBEACON: A MULTI-MODAL DATASET & DATA COLLECTION PLATFORM FOR MODELING COGNITIVE FATIGUE

6.1 Introduction

Cognitive fatigue (CF), which is different from but related to physical fatigue, is an ubiquitous symptom found in numerous real-world applications such as, healthcare, transportation safety, and in the industrial workplace. It is considered an "invisible" safety risk [159], often going undetected and untreated, and can cause impaired judgment and other symptoms. For example, consider a school bus driver who is so fatigued that he misses a stop sign; or an airport security officer who fails to recognize a gun inside a passing bag; or a nurse or doctor who administers the wrong medication; or a lecturer who makes mistakes, impacting the quality of education. In medicine, physical and CF are the most common symptoms across many physical and mental diseases such as Multiple-Sclerosis (MS), Lupus [160], Parkinson [161], Chronic Insomnia or bad sleep quality [162], Traumatic Brain Injury (TBI) [163], and others.

There are a variety of reasons why CF is important; it affects many clinical populations who have sustained brain injury or disease as well as individuals who have undergone chemotherapy intervention for cancer, individuals with Chronic Fatigue Syndrome, Veterans with Gulf War Illness, and even otherwise healthy aged individuals. That is, CF is pervasive. It is frequently rated as the worst, or among the worst, consequences following brain injury or disease. It degrades the quality of life because affected individuals refrain from doing activities that lead to CF, with the

result that they leave the workforce, limit social interactions, and cease to perform mental work that we know is essential for the maintenance of cognitive abilities. CF can have a direct impact on the quality of life, it can impact productivity and the efficiency of completing every-day tasks, and it can significantly increase the possibility of unwanted accidents with critical effects. CF, rather than muscular work-associated fatigue, has a greater impact on individuals with TBI than any other factor and is rated as one of the most distressing symptoms in at least half of TBI patients. CF occurs in 45-73% of the TBI population (both civilian and military)[164], with 73% of participants reporting significant levels of fatigue even five years post-injury [165]. In MS populations, fatigue is the single most commonly reported symptom [166], and one of the "most troubling symptoms" [167] because of its negative impact on the quality of life. CF, specifically, is cited as being a significant barrier to employment, educational attainment, and everyday functioning, [168, 169]. According to the Occupational Safety and Health Administration (OSHA) [170], employees suffering from fatigue are 2.9 times more likely to be involved in job-related accidents such as slips, falls, and even death. Though human error cannot be eliminated completely, accidents can be reduced and prevented by applying intelligence to identifying the root causes of fatigue, based on analysis of longitudinal behavioral data.

Motivated by the aforementioned observations, we propose CogBeacon; a dataset designed to identify signs of CF across individuals while performing a cognitive task. CogBeacon, offers access to *multi-sensing data* along with *user-reports* and *task-based performance metrics* towards identifying events of CF. Thus, allowing researchers to investigate complex correlations across these three diverse but highly correlated groups of behavioral characteristics. Moreover, along with the collected dataset, CogBeacon comes with an open-access software that provides the required back-end

computational framework needed for data collection [1]. Our goal is to motivate other researchers to extend the functionalities of the system by integrating their own cognitive tasks and sensors and enrich the available dataset by conducting their own experiments using the CogBeacon Platform.

The rest of the Chapter is structured as follows. In Section-6.2 we discuss computational methods that have been proposed in the past for CF analysis. In Section-6.3 we explain the WCST cognitive task and we present our implementation which follows the same principles and which we used for our data collection. Section-6.4.3 describes the experimental setup and provides an in-depth description of the compiled dataset. Section-6.5 and Section-6.6 show our initial findings after conducting the user study and a preliminary machine-learning analysis on the collected multi-modal data. Finally Section-6.7 summarises our findings and highlights future directions.

6.2 Background - Computational Modeling of Cognitive Fatigue

Detecting and predicting CF is not a new problem in the area of behavioral modeling. Several research efforts have tried to tackle the problem in the past by adopting various approaches under different experimental assumptions. However, it remains a wide-open problem due to its high levels of ambiguity and despite its importance in many applications, there are very few (if any) available datasets designed to tackle the problem. In 2004 Hursh et al. [171] was one of the first research groups that tried to predict CF using methods of computational modeling. In particular, they proposed FAST a tool for fatigue forecasting designed to assist operators in the transportation sector. FAST functioned based on the SAFTE model; a computational

[1] https://github.com/MikeMpapa/CogBeacon-WCST_interface

architecture for modeling fatigue based on signal analysis related to sleep activity and task effectiveness of the operator. In 2007 Donovan et al. [172] highlighted once again the potentials of using cognitive-modeling methods to predict fatigue by conducting a user study on 256 women that were under treatment for early-stage breast cancer. A few years later, Gonzalez et al. [173] used the ACT-R [174] cognitive architecture to predict user fatigue in a data entry task. Their method takes advantage of the principles described by the ACT-R architecture and estimates how specific performance parameters such as task accuracy and response time are being affected by fatigue using a rule-based decision-making approach. ACT-R has motivated other recent approaches as well, related to fatigue and performance monitoring with applications in smart driving and vocational safety [175, 176]. In 2018 Golan et al. [177] focused on the major importance of subjective reporting with respect to CF and its impact on cognitive functioning on patients suffering from MS.

Taking all the aforementioned findings into account, CogBeacon aims to provide a robust dataset and a computational platform able to serve multiple modeling approaches and research purposes. CogBeacon is designed based on the principles described by Tsiakas et al. [131] on how to design multi-sensing interaction scenarios towards assessing cognitive and physical fatigue. In contrast to most of the referenced works, we discuss a machine learning based analysis of EEG signals towards identifying CF. Our method comes as an extension of our previous findings, originally presented in [60, 178], where similar modeling solutions were deployed to predict cognitive performance on a short-term memory cognitive task. Long term scope of this work is to develop advanced computational methods for fatigue prediction and modeling able to enhance the efficiency of current approaches in assistive technologies related to medical conditions such as MS [130] or workplace training [179].

6.3 The Wisconsin Card Sorting Test

WCST is a neuropsychological test of "set-shifting", i.e. displaying flexibility in the face of changing schedules of reinforcement [180]. A number of stimulus cards are presented to the user. The user is told to match the cards, but not how to match them; instead, the system provides feedback on whether a particular match is right or wrong. There are 3 different rules that a subject can adopt (based on the color, shape or number of the symbols), and the only feedback is whether the classification is correct or not. At each turn, only one of the three rules applies and based on that rule the user has to make a choice (out of 4 possible choices). The user's goal is to derive the rule based on the feedback provided by the system. Once the user correctly identifies the rule (operationalized as several consecutive correct responses [e.g., six]), the rule changes and the user must identify the new rule. The task generates a number of psychometric scores, including categories achieved, trials, errors, and perseverative errors (i.e., when the user is unable to switch rules, despite repeated errors). WCST has been extensively used to assess dysfunction of the prefrontal cortex of the human brain. Previous brain imaging studies have focused on identifying activity related to the set-shifting requirement of the WCST [181]. Figure-6.1 shows a screenshot of the original WCST provided by the PsyToolkit Library [59].

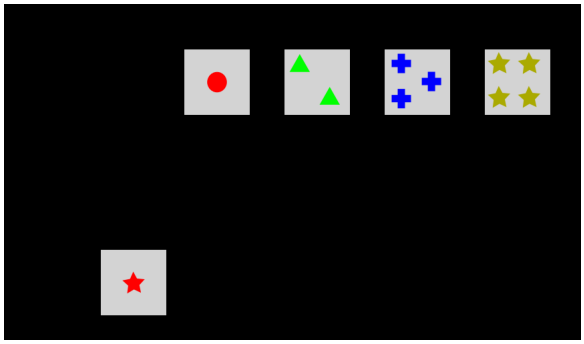


Figure 6.1. The computerized version of the WCST as offered by PshyToolkit [59].

Inspired by the principles of the original WCST we developed our own computerized cognitive game. Our task shares a relatively similar graphical environment as the game offered by [59] and provides access to the same metrics offered by the traditional WCST task. Our goal was to create a cognitive game that challenges the same cognitive functionalities as the original WCST but in the form of a computer game that has different difficulty modes and variations so it can become more engaging for the users through the introduction of alternative scenarios. As we explain in the upcoming paragraph our implementation has a mode that simulates the exact rules followed by the original task but also provides two variation modes where the number of available choices varieties throughout the game. Thus, creating an additional level of difficulty for the users.

6.3.1 Inducing Cognitive Fatigue by Increasing Complexity

To induce fatigue to the users we developed two alternative versions of the original task that aimed to increase the overall complexity and game demands with respect to user engagement and attention. In the first version (V1), the game started offering just two possible choices to the subject (against the standard four choices offered by the original task). As the game progressed the number of possible choices was increasing gradually by one until a total number of five possible choices was reached. In the second version (V2) the number of possible choices was randomly changing when the decision rule changed. As in V1, for V2 the minimum number of choices was two and the maximum five. In both modified versions the total number of rounds was almost doubled compared to the original WCST (from 60 rounds to 128), the decision rule was changing more often (every 4 rounds in V1 and V2 compared to every 6 rounds in the original WCST) and the maximum available response time was decreased by 2 seconds (from 6 sec in the original WCST to 4 sec in V1 and V2).

In Figure-6.2 we show four possible states of the V1 and V2 modified versions of the original WCST.

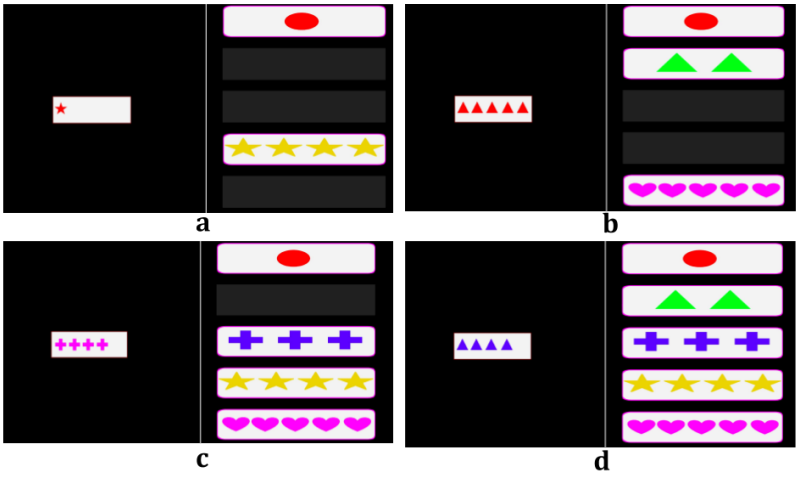


Figure 6.2. Our implementation of WCST. During a complete game the user has to play all the different cases (ie. figures a,b,c and d). In V1 the game starts with two possible choices (figure a) and the choices increase gradually by one until a total number of 5 choices (figure d) has been reached. In V2 options a,b,c,d are changing randomly after every 4 rounds under the same decision rule. At the end of V1 and V2 each user has played around 32 rounds of each a,b,c and d cases. .

To validate the hypothesis that our modified versions of the WCST were able to induce some CF to the participants, we asked them to fill out a questionnaire after the completion of the session. According to their responses, out of the 38 data collection sessions (see Section-6.4.1) that were conducted, in 28 of them (˜74% of the times) users reported being more tired at the end of the process compared to how they were feeling right before starting the experiment. Moreover, most participants suggested that they had to put more effort to adapt to the varying number of choices offered by the modified versions of the game. Based on the same post-completion questionnaires, *from a scale between 1 (No Fatigue) to 5 (Very Fatigued) an average*

increase in fatigue of 1.05 points was recorded with a standard deviation of 3.54 across all 38 data collection sessions.

The aforementioned analysis indicates that our modifications in the original task were indeed able to create a demanding environment in terms of cognitive effort for the participants that could potentially introduce signs of CF. These findings are in line with the subjective reports provided by the users in real-time while taking the task (see Section-6.5). In the following paragraphs, we describe in detail the experimental setup and we present a more in-depth analysis of the data captured during the data collection.

6.4 The CogBeacon Dataset

The CogBeacon Dataset consists of 76 WCST tasks performed by 19 individuals. During each task, we collected a great range of diverse data capturing, physiological, behavioral and performance characteristics. In addition, we recorded user-reports provided in real-time with respect to the levels of CF experienced by each participant. In Figure-6.3 we illustrate the experimental setup. The dataset along with the code for the preliminary analysis provided in Section-6.6 can be found online and are available for further experimentation [2].

6.4.1 Data Collection Process

We have collected data from 19 participants between the ages of 19 and 33 years old. All participants were either faculty or students (undergraduates and graduates) of the CSE Department at UTA. Data collection took place at the "Heracleia- Human-Centered Computing" Lab. We divided our data collection process into two sessions. Each participant had to participate in both sessions and each session took place on

[2] https://github.com/MikeMpapa/CogBeacon-MultiModal_Dataset_for_Cognitive_Fatigue

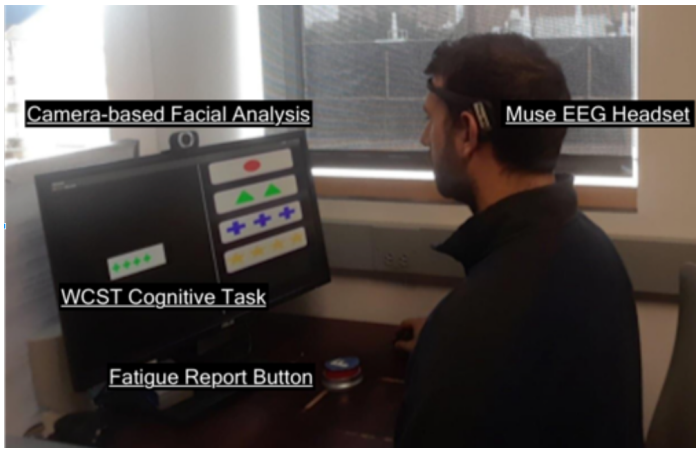


Figure 6.3. The Data Collection Experimental Setup.

a different day. Both sessions consisted of two main parts. In the first part, which was the same in both sessions, participants were asked to play the cognitive game designed by our team which follows the same rules as described by the original WCST. Our implementation follows similar guidelines as the ones provided by the PshyToolkit with 60 turns in total, 4 stimulus cards on each turn, with the matching rule changing every 6 turns. The second part, which took place right after the completion of the first test, was to play one of the modified versions, V1 or V2, with an increased number of rounds. The main difference between the two sessions was in the second part of the task. During the second part of the first session, users were asked to play the V1 version of the WCST while in the second session they had to play V2.

Our goal with the introduction of V1 and V2 as a second part was a) to expose the user to something similar to what s/he had already experienced but not the same, so that s/he must pay attention in order to adapt to the changes, b) to induce CF in the users and c) to create a rich dataset of similar but not identical tasks towards understanding CF. Table-6.1 summarises the details of the data collection process.

Game Type	#Participants	Times Played	#Total
Simulation of Original WCST	19	2	38
V1-WCST	19	1	19
V2-WCST	19	1	19
#Total Tests in Dataset			76

Table 6.1. Total number of WCST tasks included in the CogBeacon dataset

6.4.2 Sensors and Data Stored

6.4.2.1 Physiological & Behavioral Data:

We recorded the user's EEG data during task performance, using the Muse EEG headset, a non-invasive wearable device, widely used for BCI systems [182]. Muse has 4 electrodes, 2 over the prefrontal lobe and 2 behind the ears. We recorded raw EEG activation in a sampling rate of 220 Hz and using the digital signal processing unit embedded in the device we also stored information and features extracted from the individual EEG frequency bands namely: gamma 32-100 Hz (γ), beta 13-32 Hz (β), alpha 8-13 Hz (α), theta 4-8 Hz (θ) and delta 0.5-4 Hz (δ) in a sampling rate of 10 Hz. As explained in Section:5.2.1.2 *delta* waves provide information related to deep dreamless sleep when there is a lack of body awareness, *theta* waves are useful to describe deep mental states such as dreaming or deep meditation where subjects have reduced consciousness, *alpha* waves describe physically and mentally relaxed states of mind while *beta* and *gamma* can be used to describe awake and alert states of consciousness with heightened perception and are related to active thinking, excitement, learning, and increased cognitive processing [158]. Thus, for each of the four MUSE sensors the following EEG data-streams have been logged:

- **Raw EEG :**, in a sampling frequency of 220 Hz

- **Absolute Frequency Bands (A):** γ, β, α, θ and δ in sampling frequency of 10 Hz. The absolute band power for a given frequency range is the logarithm of the sum of the Power Spectral Density of the EEG data over that frequency range.

$$xA = \log \sum_{i=f\_low}^{f\_high} |G(f_i)|^2 \tag{6.1}$$

,where f_low and f_high are the minimum and maximum frequencies of frequency band x and G is the FFT of the EEG signal g

- **Relative Frequency Bands (R):** γ, β, α, θ and δ in sampling frequency of 10 Hz. The relative band powers are calculated by dividing the absolute linear-scale power in one band over the sum of the absolute linear-scale powers in all bands.

$$xR = \frac{10^{xA}}{10^{\alpha A} + 10^{\beta A} + 10^{\delta A} + 10^{\gamma A} + 10^{\theta A}} \tag{6.2}$$

, where x is one of the five frequency bands.

- **Session Score for each Frequency Band (s):** A value computed by comparing the current value of a band power to its history in sampling frequency of 10 Hz. This value is mapped to a score between 0 and 1 using a linear function that returns 0 if the current value is equal to or below the 10th percentile of the distribution of band powers, and returns 1 if it's equal to or above the 90th percentile. Linear scoring between 0 and 1 is done for any value between these two percentiles.
- **Signal Quality Indicator:** An integer value from 1 (optimal quality) to 3 (very bad quality).

To capture behavioral changes during the task, we also recorded variations in the movement of the face, capturing a set of 68 facial keypoints with a webcam placed on top of the screen. To identify facial keypoints, we deployed the method presented

by [183] that uses a Regression Tree approach and can be applied in a real-time manner. Figure-6.4 illustrates the output of the algorithm from two different users in two random frames.

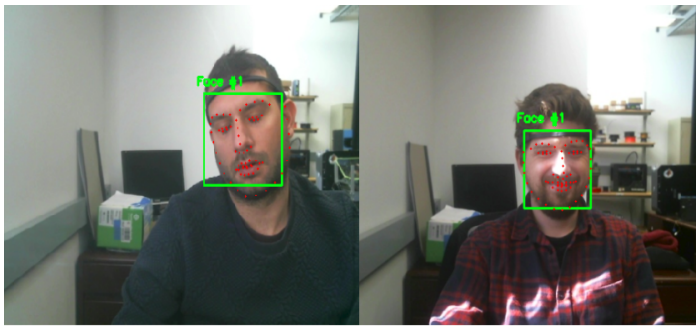


Figure 6.4. Facial-Keypoints Detection and Tracking based on [183].

6.4.2.2 Real-Time User Reports on Cognitive Fatigue:

During each test, participants were told to report when they were having trouble to keep up with the task by pressing a button that was placed in front of them. The button could be pressed at any time during a game as many times as the participants felt appropriate. Thus, a button press would act as an indicator that the user is feeling overwhelmed by the game and could be the result of someone's inability to pay attention, boredom, difficulty to remember or resolve the correct decision rule or any other reason/condition that could potentially affect task performance according to the subjective opinion of the participant. *For the purposes of this data collection all the aforementioned reasons/conditions were considered as indicators of cognitive fatigue.*

6.4.2.3 Task-based Performance Metrics:

For every round of every test, the system logs a set of metrics and scores related to user performance with respect to the task. These metrics are:

- A binary flag that indicates if user response was correct in a given round
- The cumulative number of perseverative errors until the current round. Perseverative errors are when the user continues to apply the wrong rule despite the informative feedback provided by the system
- The cumulative number of non-perseverative errors until the current round. Non-perseverative errors are the errors recorded when the user tries to figure out the new rule after a rule change. Given that there are 3 possible decision rules in total (base on color or shape or number), a user is supposed to figure out the correct rule no later than the third round after a rule change. Any error occurred before the third round is considered as non-perseverative error. All other errors are considered as perseverative errors.
- The total number of correct answers
- User's response time at every round
- An indicative round-based user score computed as:

$$score = \frac{\#available\_choices}{response\_time \times \#round\_under\_same\_rule} \tag{6.3}$$

Score was computed only if user's answer was correct. Otherwise score = 0

In addition for every round the system logs the following task characteristics:

- The number of possible choices offered by the system: 2,3,4 or 5
- The type of the correct stimuli: color, shape or number
- The value of the correct stimuli:
  - If color: green, yellow, blue, red or magenta

– If shape: triangle, star, cross, circle or heart

– if number one, two, three, four or five

6.4.3 The CogBeacon Data Collection Platform

As mentioned before the CogBeacon data collection software can be found online and downloaded for free [3]. The software is easy to install and execute and can be used to extend the current dataset and the analysis provided here. Moreover, the software can be easily modified to run across different platforms as it is mainly written in Kivy; a Python-based API that can run on Windows, Linux, iOS and Android operating systems [184]. The CogBeacon data-collection platform aims to support the integration of additional and more advanced sensors for monitoring human behavior. In addition, our future goal is to extend the functionalities of the library by incorporating more cognitive and problem-solving tasks such as a version of the SL task described in Section-5.2.1.1 towards modeling different aspects of CF and understanding its effects on human behavior and performance [3]. The current implementation provided online offers extensive functionalities compared to the ones used for the purposes of this analysis. In particular, textual and auditory-based stimuli are available online as extra features/options of our cognitive task. These functionalities are in contrast to the traditional design of the WCST which is based specifically on visual stimuli. In the case of textual stimuli, the card that is given to the user is described through text (ie. one red circle) while in its auditory version the system describes the card through audio. These functionalities are designed to evaluate user's ability to adapt to different types of stimuli, however, this kind of analysis is out of our current scope and thus, no related analysis is presented here. In Figure-6.5 we visualize the textual and auditory-based versions of our cognitive task and in Figure-6.6 we show

[3] https://github.com/MikeMpapa/CogBeacon-WCST_interface

two screenshots of the audiovisual feedback provided at each round to the user by the interface.

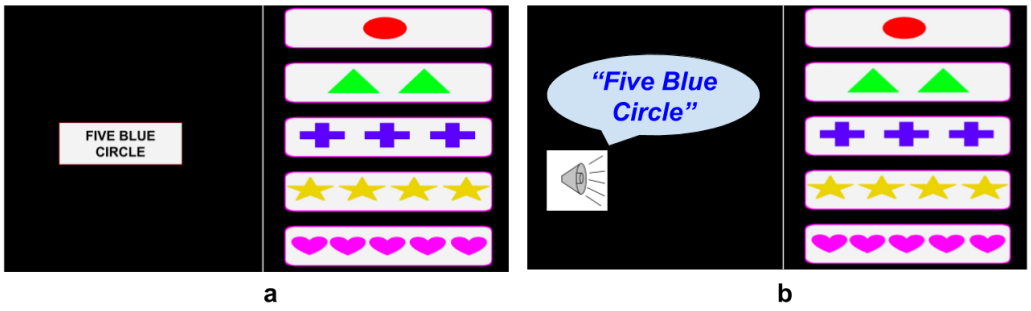


Figure 6.5. Textual Stimuli Version shown in (a) and Auditory Stimuli in (b).

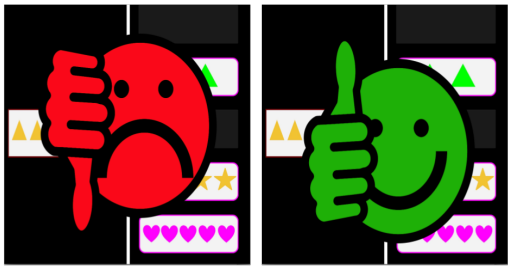


Figure 6.6. Feedback provided by the system after each user choice (Left: Negative - Right: Positive). Visual feedback is accompanied by an appropriate sound that makes the overall interaction richer and more appealing to the user, while at the same time eliminates the possibility of miss-understanding the outcome of his/her choice.

6.5 User Study - Preliminary Analysis

Figures 6.7, 6.8 and 6.9 show a cumulative analysis of CF and task-performance from versions V1 and V2 of the WCST task. Figure-6.7 illustrates the levels of CF as indicated by the users when pressing the "FATIGUE" button during the task. The X-axis shows the rounds of the game (128 total rounds) and the Y-axis shows the

levels of CF (total number of button presses). The thicker and denser the line is, the larger the group of users that it represents. At the beginning of the game, no CF was reported. As the game progressed, more and more users reported experiencing CF. By the end of the game, the vast majority of users had pressed the "FATIGUE" button at least once, while the maximum number of times the button was pressed by a user was 6.

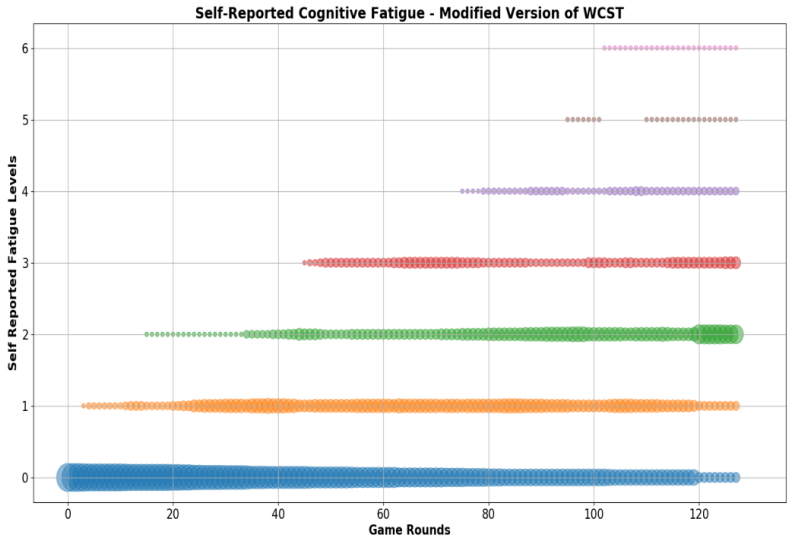


Figure 6.7. Self reported levels of cognitive fatigue during the game. The thicker and denser the line is, the larger the group of users that it represents..

The first two graphs of Figure-6.8 (top 2 graphs) illustrate how the total number of non-fatigued users decreased in comparison to the total number of fatigued users as the game progressed, while the graph of the third row shows how the percentage of fatigued users increased during the game. According to our data analysis, in 35 out of the 38 different tests of V1 and V2 combined, users reported experiencing at least some levels of CF by the end of the game. This percentage corresponds to almost

93% of the sessions, while the average number of times a user reported fatigue was 2.2 as shown in the last graph of Figure-6.8).

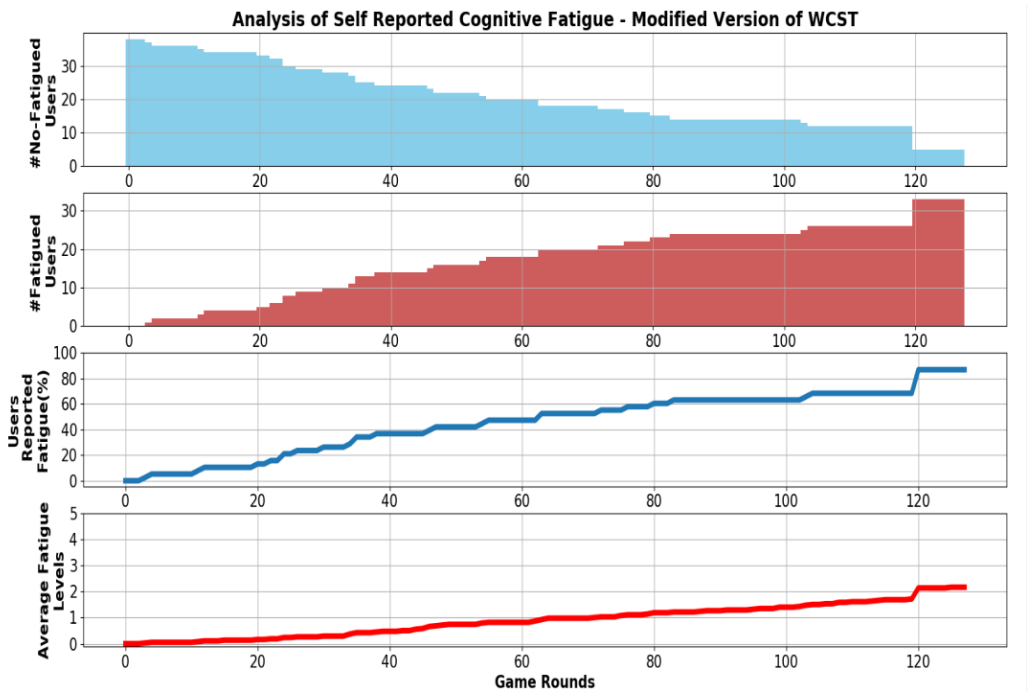


Figure 6.8. Analysis of Self-Reported Cognitive Fatigue during V1 and V2 versions of WCST..

Figure-6.9 shows how the average number of perseverative errors increased during a session across all users. On average, each user made 9.3 perseverative errors (with a standard deviation of 2.65). Perseverative errors in WCST can be considered as the "unwanted" kind of errors. While errors are unavoidable in the game since the users are supposed to learn the correct rule through the feedback, perseverative errors indicate that the user has failed to adjust to the change and keeps making decisions based on the wrong stimuli despite the negative responses provided by the system. An increasing number of perseverative errors in a healthy individual can be considered as a clear indicator of cognitive fatigue.

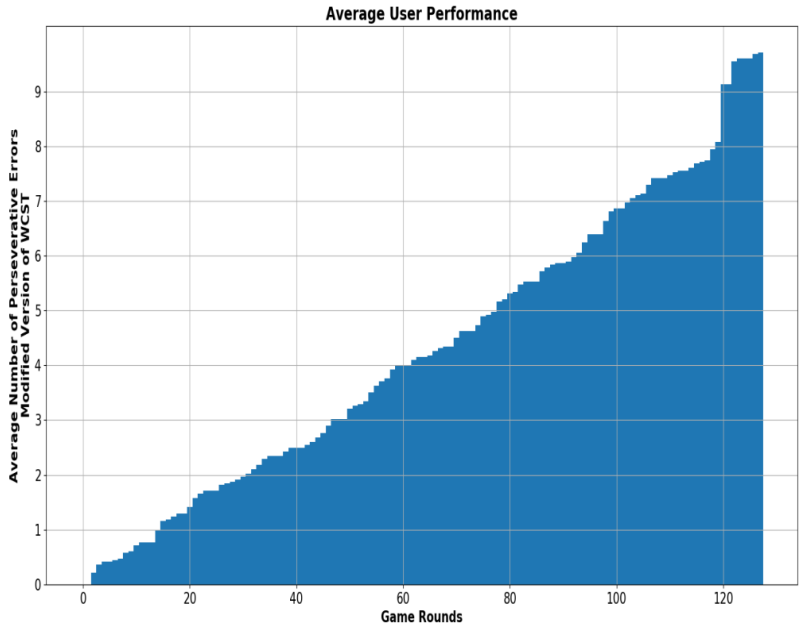


Figure 6.9. Average number of perseverative errors when playing V1 and V2 versions of WCST..

The user-study indicates that the experiment was successful in introducing CF in this group of healthy subjects which could potentially have an effect on user performance. Our initial findings showed that response time did not play an important role in the quality of decision making. Our future analysis will focus on how user responses on CF are correlated with the actual performance in the task. However, based on the small number of subjects provided by the current version of the CogBeacon dataset no safe generalizations can be drawn for this relation.

6.6 Predicting Cognitive Fatigue based on Subjective Reports and EEG signals

Our initial experimentation towards predicting cognitive fatigue was performed based on the EEG signals and the subjective user reports provided during the data collection (by pressing the button). Specifically, we used an approach similar to the

one presented in [60] and we focused on identifying the presence of fatigue in a single round of our implementation of the WCST game.

For the purposes of this experimentation, all rounds from all three variations (original WCST, V1, and V2) were combined to form a single dataset for our analysis. All the rounds that were not associated with a "button press" were considered as NO-FATIGUE samples while all the rest were used to represent the FATIGUE class. No temporal relation across consecutive rounds was considered for these initial experiments.

For modeling the EEG signals we chose to do an exhaustive grid search analysis across all the available feature streams that we were capturing during the data collection in order to choose the best signal representation (see Subsection-6.4.2.1). According to our analysis, the most promising indicators were the feature streams related to the beta 13-32 Hz (b) and gamma 32-100 Hz (g) wavelengths and in particular their absolute (A) and relative (r) values. This finding is in line with the related literature that suggests that beta and gamma waves are highly related to mental states such as alert, normal alert consciousness, active thinking, and problem-solving [158]. More specifically beta waves can be good indicators when someone is active in a conversation or when decision making and problem-solving takes place while gamma waves can be used as identifiers of heightened perception, or a 'peak mental state' when there is simultaneous processing of information from different parts of the brain.

6.6.1 Round Representation & Feature Extraction

In order to represent the EEG signals within a round in the form of a feature vector, we extracted a set of time and spectral features. In particular, the following six features were extracted for a given sequence of EEG measurements within a round of the WCST:

1. Mean Value
2. Standard Deviation
3. Maximum Value
4. Minimum Value
5. Spectral Centroid (equation-2.13)
6. Spectral Rollof (equation-2.17)

Considering that the MUSE has 4 electrodes in total the final representation for each round was a feature vector of size $4_{electrodes} \times 6_{\frac{features}{electrode}} = 24$ features.

6.6.2 Classification Results & Analysis

For classification purposes, we experimented with a set of traditional ML classifiers that have been extensively used for modeling problems of similar nature. More specifically we tried SVMs, SVM with an RBF kernel (SVMr), Random-Forests (RF), Extra-Trees (ET) and Gradient-Boosting (GB) [185, 60].

In order to evaluate our models, we performed a 10-Fold cross-validation across all the available data provided by the 76 tests available in the CogBeacon dataset (20% of the sessions for testing and the rest for training). The distribution of samples across the two classes in training and testing varied in each fold based on the total number of times users reported fatigue in the specific sessions that were used for training or testing respectively. However, in all cases the two classes were highly unbalanced towards the "NO-FATIGUE" class. Hence, in order to efficiently train our classifiers and avoid over-fitting, we omitted most of the "NO-FATIGUE" samples to avoid extreme biases and we trained the classifiers on balanced classes, with the total number of samples for each class being equal to the available "FATIGUE" samples in each fold. *For testing, we kept the original sample ratio so to have a realistic*

representation of the targeted problem. For the rest of this chapter "NO-FATIGUE" class will be represented as *NF* while "FATIGUE" class as *F*.

Table-6.2 depicts the details of the data used for experimentation while Table-6.3 shows the best results obtained by the aforementioned classifiers.

	Fold																	
#Sps	F1		F2		F3		F4		F5		F6		F7		F8		F9		F10	
	NF	F	NF	F	NF	F	NF	F	NF	F	NF	F	NF	F	NF	F	NF	F	NF	F
Tst	938	550	1034	438	959	481	1160	300	1135	305	698	596	1037	431	955	525	942	514	1325	155
Tst (%)	0.63	0.37	0.7	0.3	0.67	0.33	0.79	0.21	0.79	0.21	0.54	0.46	0.71	0.29	0.65	0.35	0.65	0.35	0.9	0.1
Tr / C	1610		1722		1679		1860		1855		1564		1729		1635		1645		2005	
Total	4708		4916		4798		5180		5150		4422		4926		4750		4746		5490	

Table 6.2. The final distribution of train and test data across all folds. As it is easy to observe the NF class dominates F in all testing cases making detection of fatigue much more challenging and highlighting the overall difficulty of the target problem. The abbreviations of Table-6.2 are the following: Sps: Smaples, Tst: Test, Tr: Train, C: Class

Looking deeper into the final results presented in Table-6.3 it is observed that despite the simplistic modeling of the problem this preliminary ML analysis can provide very promising results and great insights towards identifying robust CF patterns for the specific task. As highlighted in the previous paragraph, derivatives of gamma and beta wavelengths seem to be the most informal towards identifying intense cognitive effort. Moreover, the relatively high Precision of the NF class achieved by all classifiers indicates that when an algorithm characterized a user as not-fatigued there was a big chance (>70%) that the prediction was correct. On the other hand, the comparatively low Precision for the F class (best is 51% for the RF classifier) indicates that only in 50% of the cases that the algorithm detected fatigue the prediction was in line with the user responses. Judging now according to the Recall scores, it seems that in the cases of RF classifier for the NF class and for SVMs for the F class the algorithms were very likely to capture efficiently most samples belonging in

each corresponding class (>=70%). Based on these preliminary results we perform a post-classification by combining the predictions of all 5 methods by averaging their assigned probabilities for each label. Combining all methods provided an improvement of 2% in terms of average F1 compared to the best F1 reported by the individual classifiers.

		Rc		Pr		F1		Avg F1	Ac
Cl	S	NF	F	NF	F	NF	F		
SVM	gA	0.6	0.7	0.83	0.43	0.7	0.53	0.61	0.63
SVMr	gA	0.58	0.65	0.8	0.40	0.67	0.49	0.58	0.6
RF	bA	0.75	0.46	0.7	0.51	0.72	0.48	0.60	0.64
ET	dS	0.58	0.62	0.72	0.47	0.64	0.53	0.59	0.6
GB	bR	0.59	0.64	0.74	0.40	0.66	0.54	0.60	0.61
combined		0.72	0.56	0.79	0.46	0.75	0.51	**0.63**	**0.67**

Table 6.3. Average Classification results across all Folds for different classifiers. The *S* column indicates the EEG feature stream that provided the best results after the exhaustive grid search analysis on all the collected EEG signals (see Section-**??**). In last row we show the best results achieved by combining the predictions of all the trained models. Values in bold correspond to the methods that provided the best and more stable results. The abbreviations of Table-6.3 are the following: Cl: Classifier, S: Signal, Pr: Precision, Rc: Recall and Ac: accuracy.

The graphs of Figure-6.10 show the ROC curves of the combinatory classifier as estimated for each individual fold. ROC curve is a performance measurement for a classification problem at various thresholds settings. ROC is a probability curve and AUC represents the degree or measure of separability. It indicates how much a model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting Fs as Fs and NFs as NFs. According to Figure-6.10, in 8 out of 10 cases the combinatory classifier was able to successfully distinguish between the two class in a rate equal or higher to 66% which, is very promising given the difficulty and the

ambiguity of the problem. In two cases, Fold-3 and Fold-6 the classifier performed very poorly and failed to provide sufficient separability between 'FATIGUE' and 'NO-FATIGUE'. That indicates that in the sessions used for testing at these Folds, users had very divergent behaviors when reporting cognitive fatigue thus, confusing the predictive model. This observation highlights the fact of individual differences and provides a very useful insight for future directions.

These results are very informative about how different traditional ML techniques may behave towards modeling the targeted problem of CF detection and will guide our future directions. In addition, they come to complement our prior findings on predicting user performance through EEG, where we used a similar modeling process but for a completely irrelevant task related to short-term memory assessment [60]. Based on these observations, we could speculate that incorporating more user-specific information to our models could be proven very beneficial for targeting CF and that is where we plan to draw our attention during the next steps.

6.7 Takeaways

In this chapter we presented CogBeacon. The first publicly available multimodal dataset designed for the analysis and prediction of cognitive fatigue. Towards tackling the major problem of reproducibility and limited data availability, along with the dataset we provided free access to the data collection software thus, allowing other researches to expand the current version of CogBeacon and also integrate more sensors in an intuitive way to the back-end of the system. These contributions are crucial towards capturing additional sources of information towards understanding how CF affects specific aspects of cognitive performance across different users. Current analysis based on the conducted user study and the preliminary results on CF

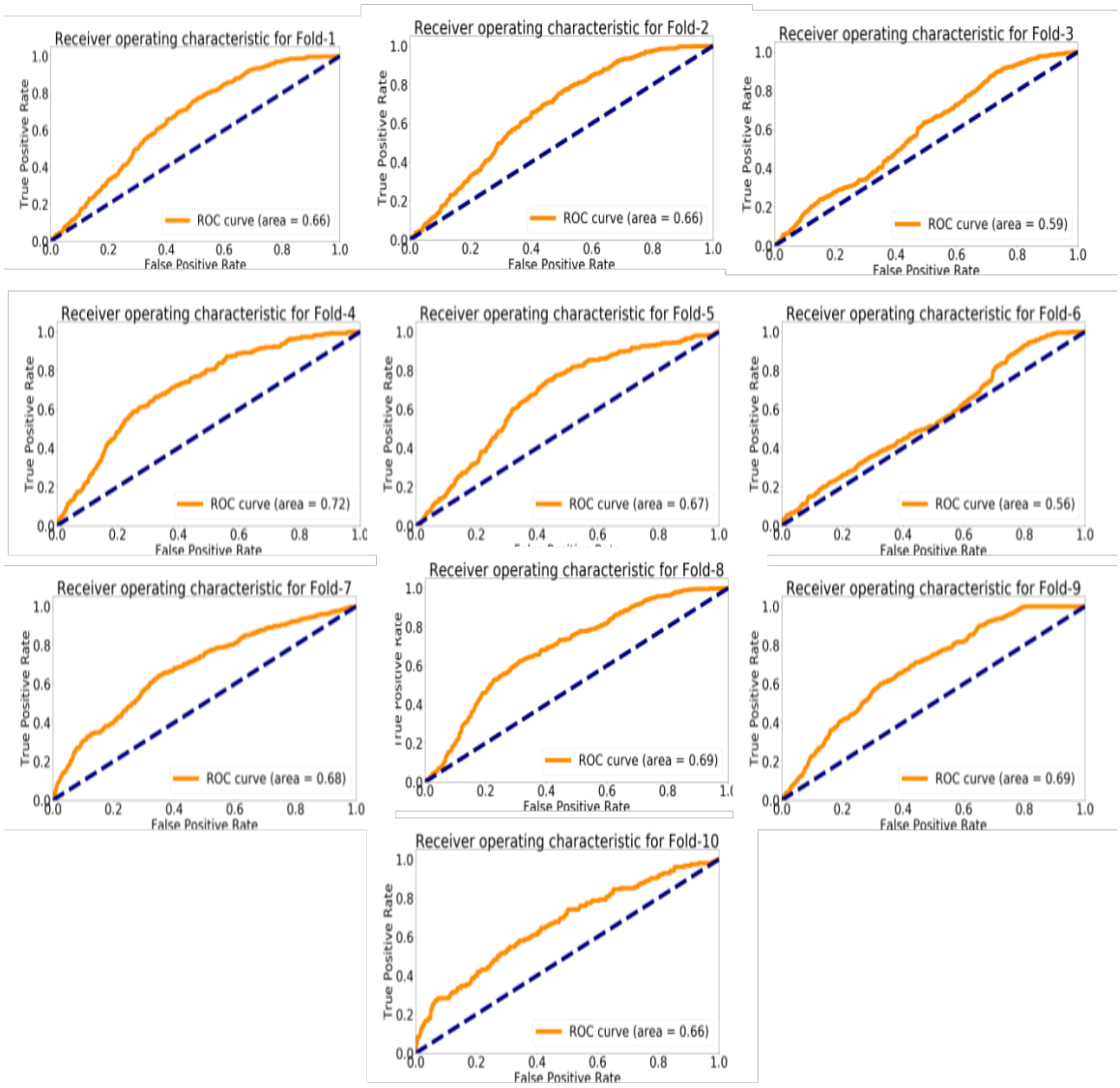


Figure 6.10. Roc Curve Estimated for each Fold after applying the combinatory classifier.

detection indicate the meaningfulness of the dataset and pave the way towards future exploration of CF detection and prediction using machine learning based techniques.

Our initial findings indicate that user reports are critical towards identifying robust patterns of CF across different subjects. However, it seems clear that person-

alized behaviors must be taken into account in the future towards improving cognitive assessment and creating more personalized and user-centric interaction scenarios.

CogBeacon is still an evolving platform. Our future steps will be focused on three main axes. Firstly enrich the current dataset with more subjects. This step would help us depict more generalized patterns of CF and will also help draw results with respect to the relation of CF to actual task performance. Secondly, we plan to incorporate more cognitive games that challenge different cognitive functionalities in the data collection software, such as the SL task presented in Section-5.2.1.1. This step is critical towards understanding how CF affects variant aspects of cognitive behavior and will help us correlate various aspects of cognitive performance. Finally, we plan to experiment with more sophisticated modeling techniques able to incorporate additional personal characteristics of the user during interaction. Such characteristics could be either information extracted from the camera sensor such as levels of motion or emotion or metrics captured directly from the user's performance in the cognitive task (such as reaction time). Figure-6.11 shows an overall illustration of the data-collection platform and highlights its various potentials for future exploration.

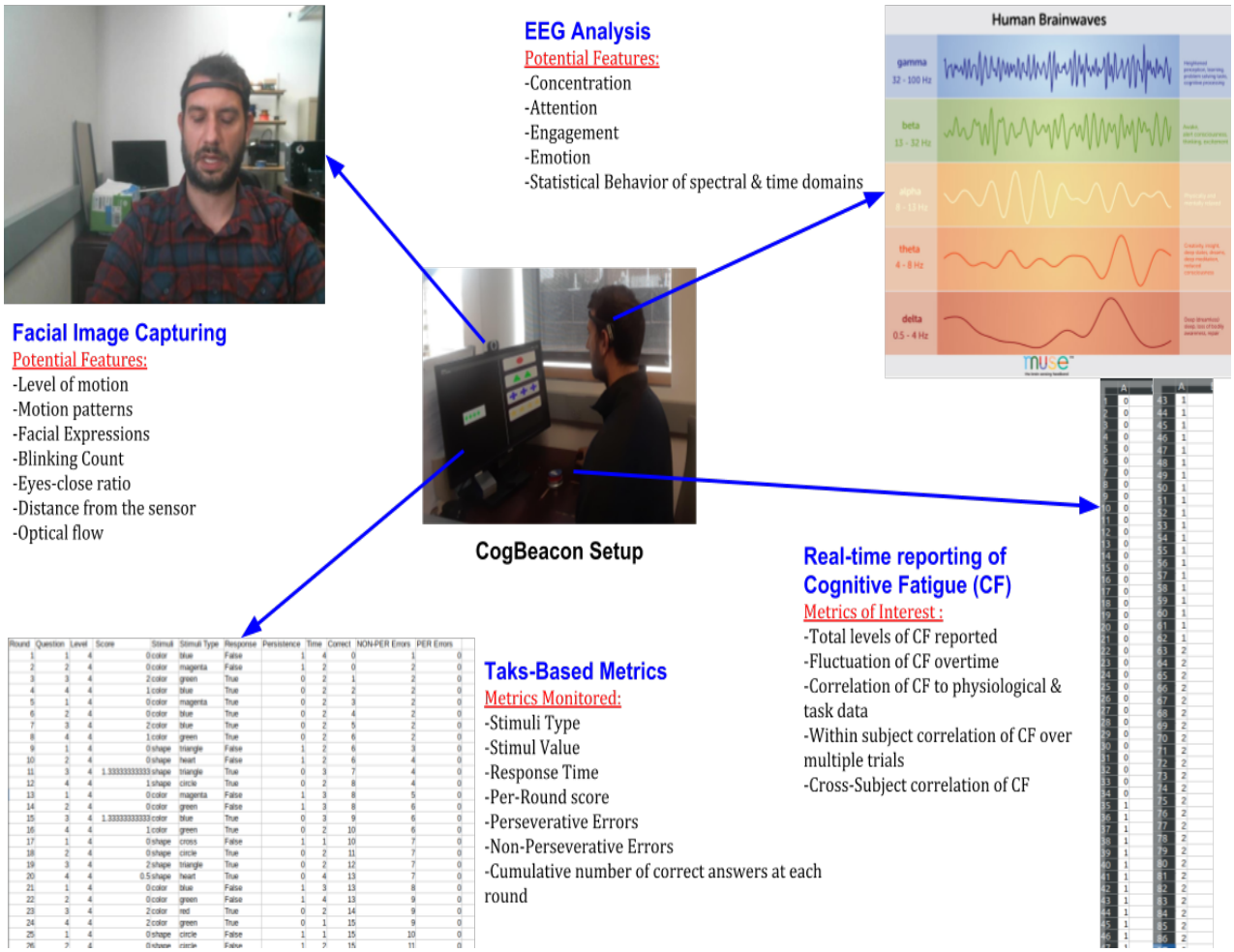


Figure 6.11. The CogBeacon Data Collection Framework. CogBeacon is an open-access software[5]. It currently supports multiple variations of the WCST cognitive task and in future versions will be enriched with additional cognitive games. It is easy to download and install almost in any platform and can easily support the integration of different sensoring devices. Currently, data have been collected using a single RGB web-camera and the MUSE EEG headset. However, integrating additional sensor is very intuitive as long as they have an API that listens to Python programming language. .

CHAPTER 7

CONCLUDING REMARKS & FUTURE DIRECTIONS

7.1 Summary

This Thesis focuses on the potentials of using machine learning methodologies towards understanding ambiguous aspects of human behavior. Through a set of end to end experiments, ie. from data collection to modeling, analysis, and application, we evaluated how modern and more traditional machine learning approaches can be exploited towards describing different human-generated signals to enhance the effectiveness and quality of various HCI scenarios. Our analysis concluded that ML-based modeling can potentially achieve state-of-the-art results in several detection and prediction problems related to human behavior analysis and that such techniques have the potential to eventually light unknown and unexplored areas related to how humans act, think and react. Thus, providing unprecedented ways towards accessing user skills, performance and intentions in a diverse spectrum of interaction scenarios.

Throughout this extensive research, we discussed the advantages and drawbacks of different techniques. Moreover, we highlighted the number one problem that arises in this field of research which has to do with the limited data availability when it comes to multi-modal monitoring and the difficulty to reproduce the required experimental conditions. This fact unavoidably leads to a lack of standardized datasets to be used for benchmarking thus, refraining researchers from making safe comparisons and improving current methodologies in challenging problems such as fatigue detection and analysis.

Motivated by this observation we proposed CogBeacon. A multi-modal dataset and data-collection software that can be used to improve AI research around modeling and understanding aspects of cognitive fatigue. Our platform which is still an evolving framework aims to help researchers draw correlations of how different cognitive functionalities are affected by cognitive fatigue and can be used to derive robust behaviors across various human subjects but also design methods towards personalized and user-centric cognitive assessment. In our next sections, we summarise the main contributions of this Thesis.

7.2 Traditional Machine Learning VS Deep Learning for Human Behavior Modeling & Monitoring

A research question that this Thesis is trying to address is to what extend deep learning approaches can provide flexible solutions in problems related to behavioral monitoring and when such solutions must be chosen against traditional machine learning algorithms. As we extensively discussed in Chapter-3, deep learning methods tend to produce better results in terms of accuracy and noise invariance when sufficient training data are available. However, collecting data from human subjects can often become a very overwhelming and expensive process. Moreover, when it comes to user monitoring it is quite common to have underrepresented behaviors in the final dataset which usually are also the behaviors of interest. Thus, collecting a sufficient amount of data points to train deep classifiers can be very difficult. In applications such as activity recognition, speaker diarisation or speech-music discrimination, deep learning methods can produce state-of-the-art results because the problems are much easier to represent through data. Thus, making the data collection and augmentation processes significantly simpler. However, when it comes to applications where the discrimination of the different classes is less intuitive, such as in the problem of

detecting and predicting fatigue, it is very common that the available data for the underrepresented classes is insufficient to train deep classifiers. In such cases traditional machine learning methods can be proven more valuable. The main reason is that in contrast to deep-classifiers which usually act as a black box, algorithms like SVMs or Random-Forests are much more intuitive and their learning process is much easier to interpret. Additionally, since they can operate on a small number of handcrafted features they allow the designer to create a more controlled environment of variables that are easier to observe over-time thus, making the feature selection process easier. Finally even though given an application there are always unique features that can be extracted; our experimentation showed that in most cases statistical and entropy-based features extracted from the time and the spectral domain can provide very valuable information about a signal's behavior. Therefore such features must always be considered towards building prototypes for human monitoring applications where most generated signals can be decomposed in the form of 1-D information streams.

7.3 Unimodal VS Multi-Modal Systems for Human Behavior Monitoring

Even though multi-modal data acquisition can, in general, provide significantly greater insights about most applications related to behavioral modeling it should not be assumed as a trivia process for multiple reasons. The first bottleneck is dictated by the nature of the application itself since integrating multiple sensors often translates to increased levels of system-invasiveness. Thus, it is a decision that the system architect has to make in order to optimize that trade-off. Secondly, the choice of the modalities themselves is a challenge that also needs to be investigated based on the behaviors of interest. As discussed in Section-2.2 there are different ways of combining different modalities and it is very crucial towards exploiting the most out of them. Overlapping modalities can potentially increase the computational cost of a

system while failing to justify a significant improvement in terms of performance. Hence, a safer approach would be to approach a problem in a divide and conquer approach where multiple single-modality based models are combined towards achieving a greater and more complex goal. This approach gives greater control to the designer towards optimizing the performance and outcomes of each individual component and can also provide flexibility towards deciding the best modeling method for each modality. For example, some modalities can be analyzed using deep learning methodologies if the available data are present and others can be addressed through more conventional ML methods. Lastly, since human behaviors are always characterized by great levels of subjectivity, incorporating user-centric reactions and opinions in the overall architectures has been proven very beneficial in most applications (see Chapter-4 and Chapter-6). That is especially true for problems that cannot be intuitively interpreted by our current knowledge of human behavior analysis, such as human physical and cognitive fatigue. Incorporating user perspective in a system can be achieved in various ways through a computational model and is a crucial step towards developing more effective interaction scenarios that can keep the users engaged and motivated.

7.4 Published Datasets

One of the main outcomes of this research is a set of various datasets related to human behavior modeling and monitoring. These datasets are either a direct product of the experiments discussed in this Thesis or are modified versions of pre-existing datasets in order to serve as ML-based benchmarks. In particular, the following six datasets have become available to the public:

1. EEG dataset for Predicting Cognitive Performance on the Sequence Learning cognitive game for short-term memory assessment [60]. Dataset available at:

https://github.com/MikeMpapa/EEG-Dataset--Sequence-Learning. Originally proposed by Tsiakas et al. [152]

2. Cognilearn - A Video-Based dataset with cognitive exercises related to cognitive assessment [4, 62, 186]. Dataset Available at: http://vlm1.uta.edu/~srujana/HTKS/CogniLearn_HTKS_Dataset.html. Dataset was built in collaboration with Dr. Srujana Gattupalli, a past colleague at the University of Texas at Arlington, for the purposes of the NSF funded project "CHS: Large: Collaborative Research: Computational Science for Improving Assessment of Executive Function in Children".
3. EMG dataset for Detection and Prediction of Physical Fatigue [132]. Dataset Available at: https://github.com/MikeMpapa/MLEmg_Monitoring_Physical_Fatigue. Originally proposed by Kanal et al. [187]
4. Video-based Activity Recognition for detecting Activities of Daily Living [23]. Dataset Available at: https://www.dropbox.com/s/919kt3dtgasu1n8/RadioData_3Classes_small.zip?dl=0. Dataset was developed for the purposes of the EU funded project " Robots in Assisted Living Environments- Unobtrusive, Efficient, Reliable and Modular Solutions for Independent Ageing" "
5. RGB and Accelerometer dataset towards Recognising Physical Activities related to Fitness Monitoring [34]. Datasaet Available at: https://www.dropbox.com/s/iu4jcyubxor2q2r/fitRecognition.zip?dl=0
6. CogBeacon - A Multi-Modal dataset for Cognitive Fatigue Analysis. Dataset Available at: https://github.com/MikeMpapa/CogBeacon-MultiModal_Dataset_for_Cognitive_Fatigue

7.5 Published Implementations

A second major outcome of this thesis is a collection of various programming implementations towards analyzing different aspects of human behavior. Throughout this research the following systems have become available to the public towards assisting the reproducibility of our results:

1. The CogBeacon Data Collection platform for analyzing cognitive fatigue. This first version of the platform currently supports three variations of the WCST cognitive game in terms of rules (see Chapter-6) and also three variations in terms of input stimuli (visual, textual and auditory). It will soon be updated to include more cognitive games and an easier to handle API. The current version can be found at: `https://github.com/MikeMpapa/CogBeacon-WCST_interface`
2. A machine learning implementation towards predicting task performance and cognitive fatigue through EEG [60]. The implementation for predicting task performance can be found online here:
   `https://github.com/MikeMpapa/EEG-Sequence-Learning`
   and its variation that focuses on analyzing cognitive fatigue can be found here as part of the CogBeacon Dataset:
   `https://github.com/MikeMpapa/CogBeacon-MultiModal_Dataset_for_Cognitive_Fatigue`
3. A machine learning methodology towards detection of physical fatigue [132, 187]. Code available at: `https://github.com/MikeMpapa/MLEmg_Monitoring_Physical_Fatigue`
4. A robust python based deep learning framework for Speech-Music Discrimination [39]. The system to our knowledge provides state-of-the-art results on the

task. Along with the methodology itself, the trained models have also become available and can be used in a plug-and-play manner. The code can be found available online at:

`https://github.com/MikeMpapa/CNNs-Speech-Music-Discrimination`

5. A deep learning and framework for recognizing Emotion from speech in a Language independent manner [45, 47]. This method provides results directly comparable to other state-of-the-art methods. As before the trained models have also become available to the public. The method can be found online at: `https://github.com/MikeMpapa/CNNs-Audio-Emotion-Recognition`
6. A machine learning approach towards multi-modal activity recognition for fitness monitoring [34]. The code can be found online at `https://github.com/MikeMpapa/recognizeFitExercise`
7. A computer vision system to track human motion and monitor activity [17]. Code available at: `https://github.com/MikeMpapa/Motion-Tracker`

7.6 Future Directions

The future directions of the research proposed in this Thesis can be summarised into five major points that describe immediate and far-reaching goals:

1. **Expansion of the CogBeacon Data-Collection Platform**

   We envision the future of CogBeacon as a standardized platform for data-collection of multi-sensing data towards the analysis of cognitive fatigue. One of our immediate goals is to expand the software's functionalities and create a more intuitive API for other researchers to use. The upcoming versions of the platform will aim to incorporate additional cognitive games, tasks, and popular tests as well as an easy way to bind new hardware and wearables with the system's back-end infrastructure. Thus, assuring a stable multi-modal synchro-

nization and accurate time-stamping of the different invents monitored by the CogBeacon platform in a device independent way. Given the rapid evolution of sensing technologies

2. **Expand Research on Fatigue Analysis**

   The concept of human fatigue remains a very vague area of research from all perspectives. In particular, questions related to how different cognitive functionalities affect and are being affected by cognitive fatigue is in the center of our future goals. Moreover, the relation between physical and cognitive fatigue is a major topic of our immediate research interests. Our goal is to develop more advanced AI-based computational methodologies towards revealing those relations and design intelligent systems able to track multiple human-generated signals and their evolution over time. Our far-reaching objective is to build technologies able to assess user skills with respect to fatigue in a universal and task-independent manner. Technologies that will be responsible to assist behavioral scientists towards understanding how these concepts affect human performance but also tools to enhance human-computer interaction and improve the outcomes of interactive scenarios both for the human-subject but also for a faster and more efficient adaptation of the system itself.

3. **Explore Novel Ways Towards Integrating Personalised Feedback**

   A common observation that emerged throughout the various experiments conducted for the purposes of this Thesis, was that integrating user feedback and personalized characteristics into the analysis could potentially lead to significant improvements over the overall system performance. Even-though generic modeling machine learning methods were in all cases able to depict and describe universal patterns of behavior across users and/or tasks in many cases it was not sufficiently enough. In the future, we plan to investigate computational

ways of integrating different types of implicit or explicit user feedback towards creating more personalized scenarios. Reinforcement Learning and Interactive Machine Learning methods could potentially pave the way towards evolving and better exploiting the prior knowledge incorporated into pre-trained models by tailoring their decision-making behavior on the fly based on the specific skills of each individual.

4. **Towards Models of General Human Intelligence**

   The ultimate future objective of intelligent behavioral modeling and monitoring is in line with the more general goal of today's AI, which is nothing else but our ability to built generic computational architectures able to learn and interpret rules on new, unknown environments and domains and make decisions based on experience; just like humans do. Having this perspective as our guideline, the future scope of this research will be to exploit machine learning for the creation of systems that can autonomously adapt on different applications, users and modalities by exploiting their prior knowledge towards providing all-around solutions. Although we are still far away from achieving this goal, as the results of this Thesis indicate, modern AI can render this path with promising answers for the creation of methods that can capture meaningful features across modalities and generalize their decision making between problems. Hence, providing encouraging results for future experimentation.

REFERENCES

[1] S. K. Card, *The psychology of human-computer interaction*. CRC Press, 2017.

[2] B. Dumas, D. Lalanne, and S. Oviatt, "Multimodal interfaces: A survey of principles, models and frameworks," in *Human machine interaction*. Springer, 2009, pp. 3–26.

[3] M. Papakostas, K. Tsiakas, M. Abujelala, M. Bell, and F. Makedon, "v-cat: A cyberlearning framework for personalized cognitive skill assessment and training," in *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*. ACM, 2018, pp. 570–574.

[4] S. Gattupalli, D. Ebert, M. Papakostas, F. Makedon, and V. Athitsos, "Cognilearn: A deep learning-based interface for cognitive behavior assessment," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. ACM, 2017, pp. 577–587.

[5] A. Rajavenkatanarayanan, Y. V. Surathi, A. R. Babu, and M. Papakostas, "Myodrive: A new way of interacting with mobile devices," in *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2016, p. 18.

[6] S. Oviatt, B. Schuller, P. Cohen, D. Sonntag, and G. Potamianos, *The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations*. Morgan & Claypool, 2017.

[7] M. A. Hanson, H. C. Powell Jr, A. T. Barth, K. Ringgenberg, B. H. Calhoun, J. H. Aylor, and J. Lach, "Body area sensor networks: Challenges and opportunities," *Computer*, vol. 42, no. 1, 2009.

[8] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on.* IEEE, 2014, pp. 1717–1724.

[9] A. Adler, *Understanding Human Nature (Psychology Revivals).* Routledge, 2013.

[10] G. Gunzelmann, B. Z. Veksler, M. M. Walsh, and K. A. Gluck, "Understanding and predicting the cognitive effects of sleep loss through simulation." *Translational Issues in Psychological Science*, vol. 1, no. 1, p. 106, 2015.

[11] J. M. Entwistle, M. K. Rasmussen, N. Verdezoto, R. S. Brewer, and M. S. Andersen, "Beyond the individual: The contextual wheel of practice as a research framework for sustainable hci," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems.* ACM, 2015, pp. 1125–1134.

[12] H. Pham, P. P. Liang, T. Manzini, L.-P. Morency, and B. Poczos, "Found in translation: Learning robust joint representations by cyclic translations between modalities," *arXiv preprint arXiv:1812.07809*, 2018.

[13] I. Broch-Due, H. L. Kjærstad, L. V. Kessing, and K. Miskowiak, "Subtle behavioural responses during negative emotion reactivity and down-regulation in bipolar disorder: A facial expression and eye-tracking study," *Psychiatry research*, vol. 266, pp. 152–159, 2018.

[14] S. Makeig, K. Gramann, T.-P. Jung, T. J. Sejnowski, and H. Poizner, "Linking brain, mind and behavior," *International Journal of Psychophysiology*, vol. 73, no. 2, pp. 95–100, 2009.

[15] G. Kielhofner, *A model of human occupation: Theory and application.* Lippincott Williams & Wilkins, 2002.

[16] A. Lioulemes, M. Papakostas, S. N. Gieser, T. Toutountzi, M. Abujelala, S. Gupta, C. Collander, C. D. Mcmurrough, and F. Makedon, "A survey of sensing modalities for human activity, behavior, and physiological monitoring," in *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2016, p. 16.

[17] M. Papakostas, J. Staud, F. Makedon, and V. Metsis, "Monitoring breathing activity and sleep patterns using multimodal non-invasive technologies," in *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2015, p. 78.

[18] S. Consolvo, D. W. McDonald, T. Toscos, M. Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, *et al.*, "Activity sensing in the wild: a field trial of ubifit garden," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2008, pp. 1797–1806.

[19] K. Zhan, S. Faux, and F. Ramos, "Multi-scale conditional random fields for first-person activity recognition on elders and disabled patients," *Pervasive and Mobile Computing*, vol. 16, pp. 251–267, 2015.

[20] A. Arnab, S. Jayasumana, S. Zheng, and P. H. Torr, "Higher order conditional random fields in deep neural networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 524–540.

[21] H. Qian, Y. Mao, W. Xiang, and Z. Wang, "Recognition of human activities using svm multi-class classifier," *Pattern Recognition Letters*, vol. 31, no. 2, pp. 100–111, 2010.

[22] T.-H.-C. Nguyen, J.-C. Nebel, F. Florez-Revuelta, *et al.*, "Recognition of activities of daily living with egocentric vision: A review," *Sensors*, vol. 16, no. 1, p. 72, 2016.

[23] M. Papakostas, T. Giannakopoulos, F. Makedon, and V. Karkaletsis, "Short-term recognition of human activities using convolutional neural networks," in *Signal-Image Technology & Internet-Based Systems (SITIS), 2016 12th International Conference on*. IEEE, 2016, pp. 302–307.

[24] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[25] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 716–723.

[26] L. Xia and J. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2834–2841.

[27] J. Waleed, T. M. Hasan, and Q. K. Abed, "Eye-gaze estimation systems for multi-applications: An implementation of approach based on laptop webcam," *Diyala Journal For Pure Science*, vol. 14, no. 02, pp. 153–173, 2018.

[28] X. Li and M. C. Chuah, "Rehar: Robust and efficient human activity recognition," *arXiv preprint arXiv:1802.09745*, 2018.

[29] G. Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with r* cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1080–1088.

[30] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical human activity recognition using wearable sensors," *Sensors*, vol. 15, no. 12, pp. 31 314–31 338, 2015.

[31] J. Ziegier, H. Gattringer, and A. Mueller, "Classification of gait phases based on bilateral emg data using support vector machines," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*. IEEE, 2018, pp. 978–983.

[32] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[33] N. Y. Hammerla, S. Halloran, and T. Ploetz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *arXiv preprint arXiv:1604.08880*, 2016.

[34] M. Papakostas, T. Giannakopoulos, and V. Karkaletsis, "A fitness monitoring system based on fusion of visual and sensorial information," in *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2017, pp. 280–285.

[35] R. J. Kate, A. M. Swartz, W. A. Welch, and S. J. Strath, "Comparative evaluation of features and techniques for identifying activity type and estimating energy cost from accelerometer data," *Physiological measurement*, vol. 37, no. 3, p. 360, 2016.

[36] A. Phinyomark, F. Quaine, S. Charbonnier, C. Serviere, F. Tarpin-Bernard, and Y. Laurillau, "Emg feature evaluation for improving myoelectric pattern recognition robustness," *Expert Systems with applications*, vol. 40, no. 12, pp. 4832–4840, 2013.

[37] K. L. Pike, *Language in relation to a unified theory of the structure of human behavior*. Walter de Gruyter GmbH & co KG, 2015, vol. 24.

[38] E. C. Blake, I. Cross, M. Simões de Abreu, J. Kang, M. Zhang, S. Mills, C. Scarre, E. B. Zubrow, T. C. Lindstrøm, E. C. Blake, *et al.*, "The acous-

tic and auditory contexts of human behavior," *Current Anthropology*, vol. 56, no. 1, pp. 000–000, 2015.

[39] M. Papakostas and T. Giannakopoulos, "Speech-music discrimination using deep visual feature extractors," *Expert Systems with Applications*, 2018.

[40] T. Giannakopoulos and S. Konstantopoulos, "Daily activity recognition based on meta-classification of low-level audio events." in *ICT4AgeingWell*, 2017, pp. 220–227.

[41] R. Navigli, "Natural language understanding: Instructions for (present and future) use." in *IJCAI*, 2018, pp. 5697–5702.

[42] T. Giannakopoulos and S. Perantonis, "Recognition of urban sound events using deep context-aware feature extractors and handcrafted features," 2018.

[43] S. A. Love, K. Petrini, M. Latinus, C. R. Pernet, and F. Pollick, "Overlapping but divergent neural correlates underpinning audiovisual synchrony and temporal order judgments," *Frontiers in human neuroscience*, vol. 12, p. 274, 2018.

[44] P. Koutras, A. Zlatinsi, and P. Maragos, "Exploring cnn-based architectures for multimodal salient event detection in videos," in *2018 IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*. IEEE, 2018, pp. 1–5.

[45] M. Papakostas, E. Spyrou, T. Giannakopoulos, G. Siantikos, D. Sgouropoulos, P. Mylonas, and F. Makedon, "Deep visual attributes vs. hand-crafted audio features on multidomain speech emotion recognition," *Computation*, vol. 5, no. 2, p. 26, 2017.

[46] E. Spyrou, T. Giannakopoulos, D. Sgouropoulos, and M. Papakostas, "Extracting emotions from speech using a bag-of-visual-words approach," in *Semantic*

and Social Media Adaptation and Personalization (SMAP), 2017 12th International Workshop on. IEEE, 2017, pp. 80–83.

[47] M. Papakostas, G. Siantikos, T. Giannakopoulos, E. Spyrou, and D. Sgouropoulos, "Recognizing emotional states using speech information," in *GeNeDis 2016*. Springer, 2017, pp. 155–164.

[48] H. Gunes, C. Shan, S. Chen, and Y. Tian, "Bodily expression for automatic affect recognition," *Emotion recognition: A pattern analysis approach*, pp. 343–377, 2015.

[49] A. Majumder, L. Behera, and V. K. Subramanian, "Automatic facial expression recognition system using deep network-based data fusion," *IEEE transactions on cybernetics*, vol. 48, no. 1, pp. 103–114, 2018.

[50] M. R. Wrobel, "Applicability of emotion recognition and induction methods to study the behavior of programmers," *Applied Sciences*, vol. 8, no. 3, p. 323, 2018.

[51] S. Brás, J. H. Ferreira, S. C. Soares, and A. J. Pinho, "Biometric and emotion identification: An ecg compression based method," *Frontiers in psychology*, vol. 9, p. 467, 2018.

[52] A. Verma, A. Dogra, K. Malik, and M. Talwar, "Emotion recognition system for patients with behavioral disorders," in *Intelligent Communication, Control and Devices*. Springer, 2018, pp. 139–145.

[53] Q. Zhang, X. Chen, Q. Zhan, T. Yang, and S. Xia, "Respiration-based emotion recognition with deep learning," *Computers in Industry*, vol. 92, pp. 84–90, 2017.

[54] W.-L. Zheng and B.-L. Lu, "Investigating critical frequency bands and channels for eeg-based emotion recognition with deep neural networks," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 3, pp. 162–175, 2015.

[55] P. Thagard, *Mind: Introduction to cognitive science.* MIT press Cambridge, MA, 1996, vol. 4.

[56] M. L. Anderson, "Embodied cognition: A field guide," *Artificial intelligence*, vol. 149, no. 1, pp. 91–130, 2003.

[57] E. Kalanthroff, A. Henik, N. Derakshan, and M. Usher, "Anxiety, emotional distraction, and attentional control in the stroop task." *Emotion*, vol. 16, no. 3, p. 293, 2016.

[58] N. Akshoomoff, T. T. Brown, R. Bakeman, and D. J. Hagler Jr, "Developmental differentiation of executive functions on the nih toolbox cognition battery." *Neuropsychology*, vol. 32, no. 7, p. 777, 2018.

[59] G. Stoet, "Psytoolkit: A novel web-based method for running online questionnaires and reaction-time experiments," *Teaching of Psychology*, vol. 44, no. 1, pp. 24–31, 2017.

[60] M. Papakostas, K. Tsiakas, T. Giannakopoulos, and F. Makedon, "Towards predicting task performance from eeg signals," in *Big Data (Big Data), 2017 IEEE International Conference on.* IEEE, 2017, pp. 4423–4425.

[61] K. Tsiakas, M. Abujelala, and F. Makedon, "Task engagement as personalization feedback for socially-assistive robots and cognitive training," *Technologies*, vol. 6, no. 2, p. 49, 2018.

[62] S. Gattupalli, "Artificial intelligence for cognitive behavior assessment in children," Ph.D. dissertation, UNIVERSITY OF TEXAS AT ARLINGTON, 2018.

[63] A. Khawaji, J. Zhou, F. Chen, and N. Marcus, "Using galvanic skin response (gsr) to measure trust and cognitive load in the text-chat environment," in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems.* ACM, 2015, pp. 1989–1994.

[64] M. K. Eckstein, B. Guerra-Carrillo, A. T. M. Singley, and S. A. Bunge, "Beyond eye gaze: What else can eyetracking reveal about cognition and cognitive development?" *Developmental cognitive neuroscience*, vol. 25, pp. 69–91, 2017.

[65] A. Rajavenkatanarayanan, A. R. Babu, K. Tsiakas, and F. Makedon, "Monitoring task engagement using facial expressions and body postures," in *Proceedings of the 3rd International Workshop on Interactive and Spatial Computing*. ACM, 2018, pp. 103–108.

[66] H. Banville, M. Parent, S. Tremblay, and T. H. Falk, "Toward mental workload measurement using multimodal eeg–fnirs monitoring," in *Neuroergonomics*. Elsevier, 2018, pp. 245–246.

[67] T. J. Hess, A. L. McNab, and K. A. Basoglu, "Reliability generalization of perceived ease of use, perceived usefulness, and behavioral intentions." *Mis Quarterly*, vol. 38, no. 1, 2014.

[68] K. Crawford *et al.*, "Six provocations for big data," 2011.

[69] A. Holzinger, "Interactive machine learning for health informatics: when do we need the human-in-the-loop?" *Brain Informatics*, vol. 3, no. 2, pp. 119–131, 2016.

[70] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in *Proceedings of the 2015 ACM on international conference on multimodal interaction*. ACM, 2015, pp. 443–449.

[71] B. R. Gaines and A. F. Monk, *Cognitive Ergonomics: Understanding, Learning, and Designing Human-Computer Interaction*. Academic Press, 2015.

[72] B. R. MacIntosh and D. E. Rassier, "What is fatigue?" *Canadian journal of applied physiology*, vol. 27, no. 1, pp. 42–55, 2002.

[73] J. Rasmussen, "Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models," *IEEE transactions on systems, man, and cybernetics*, no. 3, pp. 257–266, 1983.

[74] M. Pantic, A. Pentland, A. Nijholt, and T. S. Huang, "Human computing and machine understanding of human behavior: a survey," in *Artifical Intelligence for Human Computing*. Springer, 2007, pp. 47–71.

[75] A. Jaimes and N. Sebe, "Multimodal human–computer interaction: A survey," *Computer vision and image understanding*, vol. 108, no. 1-2, pp. 116–134, 2007.

[76] S. Oviatt, "Multimodal interfaces," *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications*, vol. 14, pp. 286–304, 2003.

[77] ——, "Theoretical foundations of multimodal interfaces and systems," in *The Handbook of Multimodal-Multisensor Interfaces*. Association for Computing Machinery and Morgan & Claypool, 2017, pp. 19–50.

[78] S. Oviatt, J. Grafsgaard, L. Chen, and X. Ochoa, "Multimodal learning analytics: Assessing learners' mental state during the process of learning," in *The Handbook of Multimodal-Multisensor Interfaces*. Association for Computing Machinery and Morgan & Claypool, 2018, pp. 331–374.

[79] B. Schuller, "Multimodal user state and trait recognition: An overview," in *The Handbook of Multimodal-Multisensor Interfaces*. Association for Computing Machinery and Morgan & Claypool, 2018, pp. 129–165.

[80] J. Coutaz, L. Nigay, D. Salber, A. Blandford, J. May, and R. M. Young, "Four easy pieces for assessing the usability of multimodal interaction: the care properties," in *HumanComputer Interaction*. Springer, 1995, pp. 115–120.

[81] L. Nigay and J. Coutaz, "A design space for multimodal systems: concurrent processing and data fusion," in *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. ACM, 1993, pp. 172–178.

[82] V. Metsis, D. Kosmopoulos, V. Athitsos, and F. Makedon, "Non-invasive analysis of sleep patterns via multimodal sensor input," *Personal and ubiquitous computing*, vol. 18, no. 1, pp. 19–26, 2014.

[83] V. Metsis, G. Galatas, A. Papangelis, D. Kosmopoulos, and F. Makedon, "Recognition of sleep patterns using a bed pressure mat," in *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2011, p. 9.

[84] D. E. Huber and C. G. Healey, "Visualizing data with motion," in *Visualization, 2005. VIS 05. IEEE*. IEEE, 2005, pp. 527–534.

[85] T. Giannakopoulos, "pyaudioanalysis: An open-source python library for audio signal analysis," *PloS one*, vol. 10, no. 12, 2015.

[86] N. Singla, "Motion detection based on frame difference method," *International Journal of Information & Computation Technology*, vol. 4, no. 15, pp. 1559–1565, 2014.

[87] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[88] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao, "Histogram of oriented normal vectors for object recognition with a depth sensor," in *Asian conference on computer vision*. Springer, 2012, pp. 525–538.

[89] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans-*

actions on pattern analysis and machine intelligence, vol. 24, no. 7, pp. 971–987, 2002.

[90] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in large margin classifiers*. Citeseer, 1999.

[91] O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification," *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 1055–1064, 1999.

[92] V. Cantoni, M. Cellario, and M. Porta, "Perspectives and challenges in e-learning: towards natural interaction paradigms," *Journal of Visual Languages & Computing*, vol. 15, no. 5, pp. 333–345, 2004.

[93] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[94] O. M. Parkhi, A. Vedaldi, A. Zisserman, *et al.*, "Deep face recognition." in *BMVC*, vol. 1, no. 3, 2015, p. 6.

[95] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299.

[96] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models." in *AAAI*, vol. 16, 2016, pp. 3776–3784.

[97] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[98] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[99] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[100] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[101] B. Q. Huynh, H. Li, and M. L. Giger, "Digital mammographic tumor classification using transfer learning from deep convolutional neural networks," *Journal of Medical Imaging*, vol. 3, no. 3, p. 034501, 2016.

[102] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[103] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[104] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[105] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[106] A. T. Lopes, E. de Aguiar, A. F. De Souza, and T. Oliveira-Santos, "Facial expression recognition with convolutional neural networks: coping with few data and the training sample order," *Pattern Recognition*, vol. 61, pp. 610–628, 2017.

[107] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition." in *IJCAI*, 2015, pp. 3995–4001.

[108] J. Lee, J. Park, K. L. Kim, and J. Nam, "Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification," *Applied Sciences*, vol. 8, no. 1, p. 150, 2018.

[109] A. Pikrakis, T. Giannakopoulos, and S. Theodoridis, "A dynamic programming approach to speech/music discrimination of radio recordings," in *Signal Processing Conference, 2007 15th European*. IEEE, 2007, pp. 1226–1230.

[110] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis." in *ICDAR*, vol. 3, 2003, pp. 958–962.

[111] A. Pikrakis and S. Theodoridis, "Speech-music discrimination: A deep learning perspective," in *2014 22nd European Signal Processing Conference (EUSIPCO)*. IEEE, 2014, pp. 616–620.

[112] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2. IEEE, 1997, pp. 1331–1334.

[113] G. Williams and D. P. Ellis, "Speech/music discrimination based on posterior probability features," in *Sixth European Conference on Speech Communication and Technology*, 1999.

[114] G. Tzanetakis and P. Cook, "Marsyas: A framework for audio analysis," *Organised sound*, vol. 4, no. 3, pp. 169–175, 2000.

[115] J. Saunders, "Real-time discrimination of broadcast speech/music," in *icassp*, vol. 96, 1996, pp. 993–996.

[116] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal, "Speech/music discrimination for multimedia applications," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 6. IEEE, 2000, pp. 2445–2448.

[117] M. J. Carey, E. S. Parris, and H. Lloyd-Thomas, "A comparison of features for speech, music discrimination," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 149–152.

[118] T. Giannakopoulos, A. Pikrakis, and S. Theodoridis, "A speech/music discriminator for radio recordings using bayesian networks," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 5. IEEE, 2006, pp. V–V.

[119] G. Costantini, I. Iaderola, A. Paoloni, and M. Todisco, "Emovo corpus: an italian emotional speech database," in *International Conference on Language Resources and Evaluation (LREC 2014)*. European Language Resources Association (ELRA), 2014, pp. 3501–3504.

[120] P. Jackson and S. Haq, "Surrey audio-visual expressed emotion(savee) database," *University of Surrey: Guildford, UK*, 2014.

[121] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of german emotional speech," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[122] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[123] Z. Huang, M. Dong, Q. Mao, and Y. Zhan, "Speech emotion recognition using cnn," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 801–804.

[124] W. Zheng, J. Yu, and Y. Zou, "An experimental study of speech emotion recognition based on deep convolutional neural networks," in *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*. IEEE, 2015, pp. 827–831.

[125] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, p. 335, 2008.

[126] R. H. Paul, W. W. Beatty, R. Schneider, C. R. Blanco, and K. A. Hames, "Cognitive and physical fatigue in multiple sclerosis: relations between self-report and objective performance," *Applied Neuropsychology*, vol. 5, no. 3, pp. 143–148, 1998.

[127] J. W. Cochran, "Effect of modafinil on fatigue associated with neurological illnesses," *Journal of Chronic Fatigue Syndrome*, vol. 8, no. 2, pp. 65–70, 2000.

[128] J. S. Kalkman, M. J. Zwarts, M. L. Schillings, B. G. van Engelen, and G. Bleijenberg, "Different types of fatigue in patients with facioscapulohumeral dystrophy, myotonic dystrophy and hmsn-i. experienced fatigue and physiological fatigue," *Neurological Sciences*, vol. 29, no. 2, pp. 238–240, 2008.

[129] G. Zhang, K. K. Yau, X. Zhang, and Y. Li, "Traffic accidents involving fatigue driving and their extent of casualties," *Accident Analysis & Prevention*, vol. 87, pp. 34–42, 2016.

[130] A. Rajavenkatanarayanan, V. Kanal, K. Tsiakas, D. Calderon, M. Papakostas, M. Abujelala, M. Galib, J. C. Ford, G. Wylie, and F. Makedon, "A survey of assistive technologies for assessment and rehabilitation of motor impairments in multiple sclerosis," *Multimodal Technologies and Interaction*, vol. 3, no. 1, p. 6, 2019.

[131] K. Tsiakas, M. Papakostas, J. C. Ford, and F. Makedon, "Towards a task-driven framework for multimodal fatigue analysis during physical and cognitive tasks," in *Proceedings of the 5th international Workshop on Sensor-based Activity Recognition and Interaction*. ACM, 2018, p. 18.

[132] M. Papakostas, V. Kanal, M. Abujelala, and F. Makedon, "Physical fatigue detection through emg wearables and subjective user reports - a machine learning approach towards adaptive rehabilitation," in *Proceedings of the 12th PErvasive Technologies Related to Assistive Environments Conference*. ACM, 2019.

[133] E. Dobryakova, H. M. Genova, J. DeLuca, and G. R. Wylie, "The dopamine imbalance hypothesis of fatigue in multiple sclerosis and other neurological disorders," *Frontiers in neurology*, vol. 6, p. 52, 2015.

[134] J. Van Cutsem, S. Marcora, K. De Pauw, S. Bailey, R. Meeusen, and B. Roelands, "The effects of mental fatigue on physical performance: a systematic review," *Sports medicine*, vol. 47, no. 8, pp. 1569–1588, 2017.

[135] P. Karthick, D. M. Ghosh, and S. Ramakrishnan, "Surface electromyography based muscle fatigue detection using high-resolution time-frequency methods and machine learning algorithms," *Computer methods and programs in biomedicine*, vol. 154, pp. 45–56, 2018.

[136] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for emg signal classification," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7420–7431, 2012.

[137] A. Phinyomark and E. Scheme, "Emg pattern recognition in the era of big data and deep learning," *Big Data and Cognitive Computing*, vol. 2, no. 3, p. 21, 2018.

[138] ——, "A feature extraction issue for myoelectric control based on wearable emg sensors," in *Sensors Applications Symposium (SAS), 2018 IEEE*. IEEE, 2018, pp. 1–6.

[139] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.

[140] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. Van Der Smagt, *et al.*, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, vol. 485, no. 7398, p. 372, 2012.

[141] L. D. Jiménez, A. Velásquez, and H. Trefftz, "Evaluation of various strategies to improve the training of a brain computer interface system," *ENERGIA*, 2013.

[142] H. Wang, A. Song, B. Li, B. Xu, and Y. Li, "Psychophysiological classification and experiment study for spontaneous eeg based on two novel mental tasks," *Technology and Health Care*, vol. 23, no. s2, pp. S249–S262, 2015.

[143] J. Atkinson and D. Campos, "Improving bci-based emotion recognition by combining eeg feature selection and kernel classifiers," *Expert Systems with Applications*, vol. 47, pp. 35–41, 2016.

[144] L. Carelli, F. Solca, A. Faini, P. Meriggi, D. Sangalli, P. Cipresso, G. Riva, N. Ticozzi, A. Ciammola, V. Silani, *et al.*, "Brain-computer interface for clinical

purposes: cognitive assessment and rehabilitation," *BioMed research international*, vol. 2017, 2017.

[145] A. De, A. Konar, A. Samanta, S. Biswas, A. L. Ralescu, and A. K. Nagar, "Cognitive load classification in learning tasks from hemodynamic responses using type-2 fuzzy sets," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2017, pp. 1–6.

[146] D. E. Kieras, "A summary of the epic cognitive architecture," *The Oxford handbook of cognitive science*, vol. 1, p. 24, 2016.

[147] W. I. Hamilton and T. Clarke, "Driver performance modelling and its practical application to railway safety," in *Rail Human Factors*. Routledge, 2017, pp. 25–39.

[148] B. Zoltan, D. Zeitlin, D. Gupta, G. Fiorenza, G. Seliger, J. Wolpaw, L. Tenteromano, N. J. Hill, and T. Vaughan, "Objective extraction of eeg features to predict recovery and determine awareness/unawareness after brain injury," *Archives of Physical Medicine and Rehabilitation*, vol. 99, no. 11, p. e143, 2018.

[149] M. J. Frank, C. Gagne, E. Nyhus, S. Masters, T. V. Wiecki, J. F. Cavanagh, and D. Badre, "fmri and eeg predictors of dynamic decision parameters during human reinforcement learning," *Journal of Neuroscience*, vol. 35, no. 2, pp. 485–494, 2015.

[150] K. Muldner, W. Burleson, and K. VanLehn, "yes!: using tutor and sensor data to predict moments of delight during instructional activities," in *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 2010, pp. 159–170.

[151] B. S. Goldberg, R. A. Sottilare, K. W. Brawner, and H. K. Holden, "Predicting learner engagement during well-defined and ill-defined computer-based inter-

cultural interactions," in *International Conference on Affective Computing and Intelligent Interaction*. Springer, 2011, pp. 538–547.

[152] K. Tsiakas, C. Abellanoza, M. Abujelala, M. Papakostas, T. Makada, and F. Makedon, "Towards designing a socially assistive robot for adaptive and personalized cognitive training."

[153] B. A. Clegg, G. J. DiGirolamo, and S. W. Keele, "Sequence learning," *Trends in cognitive sciences*, vol. 2, no. 8, pp. 275–281, 1998.

[154] S. E. Gathercole and A. D. Baddeley, *Working memory and language*. Psychology Press, 2014.

[155] K. M. Thomas, R. H. Hunt, N. Vizueta, T. Sommer, S. Durston, Y. Yang, and M. S. Worden, "Evidence of developmental differences in implicit sequence learning: an fmri study of children and adults," *Journal of cognitive neuroscience*, vol. 16, no. 8, pp. 1339–1351, 2004.

[156] X. Liu, P.-N. Tan, L. Liu, and S. J. Simske, "Automated classification of eeg signals for predicting students cognitive state during learning," in *Proceedings of the International Conference on Web Intelligence*. ACM, 2017, pp. 442–450.

[157] "The muse," http://developer.choosemuse.com/research-tools/available-data.

[158] M. Teplan *et al.*, "Fundamentals of eeg measurement," *Measurement science review*, vol. 2, no. 2, pp. 1–11, 2002.

[159] Sarah, Trotto, Safety & Health magazine, published by the National Safety Council, "Fatigue and worker safety — Experts say employers play a role in tackling the issue," https://www.safetyandhealthmagazine.com/articles/15271-fatigue-and-worker-safety, 2017, online; posted 26-February-2017.

[160] L. B. Krupp, N. G. LaRocca, J. Muir-Nash, and A. D. Steinberg, "The fatigue severity scale: application to patients with multiple sclerosis and systemic lupus erythematosus," *Archives of neurology*, vol. 46, no. 10, pp. 1121–1123, 1989.

[161] K. R. Chaudhuri, D. G. Healy, and A. H. Schapira, "Non-motor symptoms of parkinson's disease: diagnosis and management," *The Lancet Neurology*, vol. 5, no. 3, pp. 235–245, 2006.

[162] A. Qaseem, D. Kansagara, M. A. Forciea, M. Cooke, and T. D. Denberg, "Management of chronic insomnia disorder in adults: a clinical practice guideline from the american college of physicians," *Annals of internal medicine*, vol. 165, no. 2, pp. 125–133, 2016.

[163] P. Vos, Y. Alekseenko, L. Battistin, E. Ehler, F. Gerstenbrand, D. Muresanu, A. Potapov, C. Stepan, P. Traubner, L. Vécsei, *et al.*, "Mild traumatic brain injury," *European journal of neurology*, vol. 19, no. 2, pp. 191–198, 2012.

[164] C. W. Hoge, D. McGurk, J. L. Thomas, A. L. Cox, C. C. Engel, and C. A. Castro, "Mild traumatic brain injury in us soldiers returning from iraq," *New England Journal of Medicine*, vol. 358, no. 5, pp. 453–463, 2008.

[165] J. H. Olver, J. L. Ponsford, and C. A. Curran, "Outcome following traumatic brain injury: a comparison between 2 and 5 years after injury," *Brain injury*, vol. 10, no. 11, pp. 841–848, 1996.

[166] C. Tur, "Fatigue management in multiple sclerosis," *Current treatment options in neurology*, vol. 18, no. 6, p. 26, 2016.

[167] L. B. Krupp, L. A. Alvarez, N. G. LaRocca, and L. C. Scheinberg, "Fatigue in multiple sclerosis," *Archives of neurology*, vol. 45, no. 4, pp. 435–437, 1988.

[168] K. van der Hiele, D. van Gorp, R. Ruimschotel, N. Kamminga, L. Visser, and H. Middelkoop, "Work participation and executive abilities in patients with relapsing-remitting multiple sclerosis," *PloS one*, vol. 10, no. 6, p. e0129228, 2015.

[169] H. K. Kang, B. H. Natelson, C. M. Mahan, K. Y. Lee, and F. M. Murphy, "Post-traumatic stress disorder and chronic fatigue syndrome-like illness among gulf

war veterans: a population-based survey of 30,000 veterans," *American journal of epidemiology*, vol. 157, no. 2, pp. 141–148, 2003.

[170] Occupational Safety and Health Administration, US Department of Labor, "Long Work Hours, Extended or Irregular Shifts, and Worker Fatigue," https://www.osha.gov/SLTC/workerfatigue/hazards.html, online; accessed 19-February-2019.

[171] S. R. Hursh, T. J. Balkin, J. C. Miller, and D. R. Eddy, "The fatigue avoidance scheduling tool: Modeling to minimize the effects of fatigue on cognitive performance," *SAE transactions*, pp. 111–119, 2004.

[172] K. A. Donovan, B. J. Small, M. A. Andrykowski, P. Munster, and P. B. Jacobsen, "Utility of a cognitive-behavioral model to predict fatigue following breast cancer treatment." *Health Psychology*, vol. 26, no. 4, p. 464, 2007.

[173] C. Gonzalez, B. Best, A. F. Healy, J. A. Kole, and L. E. Bourne Jr, "A cognitive modeling account of simultaneous learning and fatigue effects," *Cognitive Systems Research*, vol. 12, no. 1, pp. 19–32, 2011.

[174] J. R. Anderson, C. Lebiere, M. Lovett, and L. Reder, "Act-r: A higher-level account of processing capacity," *Behavioral and Brain Sciences*, vol. 21, no. 6, pp. 831–832, 1998.

[175] L. M. Blaha, C. R. Fisher, M. M. Walsh, B. Z. Veksler, and G. Gunzelmann, "Real-time fatigue monitoring with computational cognitive models," in *International Conference on Augmented Cognition*. Springer, 2016, pp. 299–310.

[176] E. B. Khosroshahi, D. D. Salvucci, B. Z. Veksler, and G. Gunzelmann, "Capturing the effects of moderate fatigue on driver performance," in *Proceedings of the 14th International Conference on Cognitive Modeling*, 2016, pp. 163–168.

[177] D. Golan, G. M. Doniger, K. Wissemann, M. Zarif, B. Bumstead, M. Buhse, L. Fafard, I. Lavi, J. Wilken, and M. Gudesblatt, "The impact of subjective

cognitive fatigue and depression on cognitive function in patients with multiple sclerosis," *Multiple Sclerosis Journal*, vol. 24, no. 2, pp. 196–204, 2018.

[178] A. R. Babu, A. Rajavenkatanarayanan, J. R. Brady, and F. Makedon, "Multimodal approach for cognitive task performance prediction from body postures, facial expressions and eeg signal," in *Proceedings of the Workshop on Modeling Cognitive Processes from Multimodal Data*. ACM, 2018, p. 2.

[179] A. R. Babu, A. Rajavenkatanarayanan, M. Abujelala, and F. Makedon, "Votre: A vocational training and evaluation system to compare training approaches for the workplace," in *International Conference on Virtual, Augmented and Mixed Reality*. Springer, 2017, pp. 203–214.

[180] F. Lange, C. Brückner, A. Knebel, C. Seer, and B. Kopp, "Executive dysfunction in parkinsons disease: a meta-analysis on the wisconsin card sorting test literature," *Neuroscience & Biobehavioral Reviews*, 2018.

[181] N. S. Dias, D. Ferreira, J. Reis, L. R. Jacinto, L. Fernandes, F. Pinho, J. Festa, M. Pereira, N. Afonso, N. C. Santos, *et al.*, "Age effects on eeg correlates of the wisconsin card sorting test," *Physiological reports*, vol. 3, no. 7, 2015.

[182] P. Bashivan, I. Rish, and S. Heisig, "Mental state recognition via wearable eeg," *arXiv preprint arXiv:1602.00985*, 2016.

[183] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1867–1874.

[184] Kivy, " Cross-platform Python Framework for NUI Development," https://kivy.org/#home, online; accessed 25-February-2019.

[185] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for eeg-based brain–computer interfaces," *Journal of neural engineering*, vol. 4, no. 2, p. R1, 2007.

[186] A. Ghaderi, S. Gattupalli, D. Ebert, A. Sharifara, V. Athitsos, and F. Makedon, "Improving the accuracy of the cognilearn system for cognitive behavior assessment," in *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2017, pp. 177–180.

[187] V. Kanal, M. Abujelala, and F. Makedon, "Adaptive robotic rehabilitation using muscle fatigue as a trigger," 2017.

BIOGRAPHICAL STATEMENT

Michalis Papakostas was born in Athens, Greece in 1989. In 2013, he received his Diploma of Engineering (Dipl. Eng.) from Electronic and Computer Engineering Department of Technical University of Crete, Greece. In his Diploma Thesis he conducted research on Continuous-Space Language Model Adaptation using Gaussian Mixture Models, supervised by Prof. Vasilis Digalakis.

In September 2014 he joined the HERACLEIA - Human Centered Computing Lab at the University of Texas at Arlington as a PhD student, where he participated as a graduate research assistant in several NSF-funded projects, under the supervision of Prof. Fillia Makedon. Moreover, during his studies he worked as a graduate teaching assistant for the "Advanced Topics in Human-Computer Interaction" class. During the summer of 2018 Michalis worked as a research engineer intern at Toshiba - Cambridge Research Laboratories. He defended his PhD in April 2019. His PhD was in collaboration with the National Center for Scientific Research - Demokritos, under the co-supervision of Dr.Vangelis Karkaletsis, where he also worked as a Research Associate during his studies.

His research interests revolve around Artificial Intelligence, focusing on Machine Learning approaches for Human Behavior Analysis and Monitoring with applications in the broader spectrum of Human-Computer and Human-Robot Interaction. From June 2019 Michalis is expected to join the Artificial Intelligence Laboratory at the University of Michigan, Ann Arbor as a Post-Doctoral Research Fellow under the supervision of Prof. Rada Michalcea and Prof. Mihai Burzo, where he will be involved in projects related to multi-modal user modeling and monitoring.