

Cairo University

Institute of Statistical Studies and Research

Department of Computer Science

Egypt

Improving Emotion Recognition using Brain Signals

A thesis submitted in partial fulfillment of the requirements for the M.Sc. degree in
Computer Science

Submitted by

Mohammed Ahmed AbdelAal Ahmed

Supervised by

Prof. Dr. Hesham Hefny

Department of Computer Science
Institute of Statistical Studies and Research
Cairo University

Dr. Assem Alsawy

Alexandria Academy for Management and
Accounting
Ministry of Higher Education

ABSTRACT

Emotion recognition has become an important factor for easier and effective interaction between human and computer. Despite the importance of emotions in people communications, most of currently human-computer interaction systems lack the ability to recognize and interpret user emotions.

Emotions can be recognized by monitoring external phenomena of human body, such as facial expression, voice intonation and body movement. Emotions can also be recognized by monitoring internal physiological signals, such as heart rate, respiration and brain signals. Physiological signals are considered more reliable in emotion recognition specially brain signals. During the last few years, many researchers and companies became interested in interpreting user intentions by monitoring brain signals.

Electroencephalography (EEG) is the most used modality to monitor brain signals. EEG measures the electrical activities of the brain through a set of electrodes placed on the scalp. It has a high temporal resolution with no risks, and it is relatively cheap. During the last decades, many commercial EEG devices were produced, and these devices are even easier to setup and use than those devices used in laboratories.

Recognizing emotions based on brain signals is not a trivial task, since emotion representation involves some issues. Also, brain signals are very hard to be interpreted. Most previous studies have a low emotion recognition accuracy, and some of them built a model for each user separately or for a special kind of users.

The objective of this research is to improve emotion recognition using the recorded brain signals by EEG devices. The proposed model can handle different kinds of users without the need to be retrained or reconfigured.

EEG-based emotion recognition involves few steps to be accomplished. EEG signal need to be preprocessed to reduce noise and its underlying components need to be separated. A huge number of features can be extracted from the preprocessed EEG

signals which need to be reduced by a feature reduction method. Finally, a classifier is used to recognize the user emotion that associated with the extracted feature vector.

This research adopts extracting spectral power, oscillation and entropy features from EEG signals, reducing the extracted feature vector by using recursive feature elimination and classifying emotions by using support vector machine.

The experimental results showed the superiority and robustness of the proposed model compared with other studies that used the same dataset. The proposed model achieved a higher accuracy and F₁-score compared to results of the other studies.

LIST OF PUBLICATIONS

1. Mohammed A. AbdelAal, Assem A. Alsawy, and Hesham A. Hefny. “On Emotion Recognition using EEG.” In *The 50th Annual Conference on Statistics, Computer Sciences and Operations Research (ISSR), Cairo University, Egypt*, pp. 35-49. 2015.
2. Mohammed A. AbdelAal, Assem A. Alsawy, and Hesham A. Hefny. “EEG-Based Emotion Recognition Using a Wrapper-Based Feature Selection Method.” In *The 3rd International Conference on Advanced Intelligent Systems and Informatics, Ain Shams University, Cairo, Egypt*, pp. 247-256. Springer, 2017.

ACKNOWLEDGMENT

I would like to take this opportunity to express my deep gratitude to my supervisors, Prof. Dr. Hesham Hefny and Dr. Assem Alsawy, for their efforts to accomplish this work. They gave me all the time and knowledge they have and helped me to get the experience and skills that I need. It was hard to finish this work without their help.

I also would like to thank my family and friends for their support and encouragement.

CERTIFICATION

I certify that this work has not been accepted in substance for an academic degree and is not being concurrently submitted in candidature for any other degree.

Name: Mohammed Ahmed AbdelAal Ahmed

Signature:

TABLE OF CONTENTS

Chapter 1 Introduction.....	1
1.1. Problem Definition	2
1.2. Research Objective	2
1.3. Thesis Organization	3
Chapter 2 Emotion Recognition	4
2.1. Introduction.....	4
2.2. Basic Definitions.....	4
2.2.1. Emotion Definition.....	5
2.2.2. Affective Computing Definition	5
2.3. Emotion Representation (Measurement)	5
2.3.1. Discrete Categories Approach	5
2.3.2. Multi-dimensions Space Approach	6
2.4. Emotion Observation	8
2.5. Summary	9
Chapter 3 Monitoring Brain Signals.....	10
3.1. Introduction.....	10
3.2. Brain Anatomy.....	11
3.3. Brain-Computer Interface (BCI).....	12
3.3.1. Neuroimaging Approaches.....	13
3.4. Electroencephalography (EEG)	15
3.4.1. 10-20 System.....	15
3.4.2. Rhythms of EEG signals	17
3.4.3. First study to analysis EEG and Emotions	18

3.5. Summary	18
Chapter 4 Background Knowledge for EEG-based Emotion Recognition	20
4.1. Introduction	20
4.2. Signal Processing	21
4.2.1. Fourier Transform	21
4.2.2. Digital Filters	22
4.2.3. Wavelet Transform.....	24
4.3. Feature Extraction.....	25
4.3.1. Energy	26
4.3.2. Spectral Power	26
4.3.3. Entropy	26
4.3.4. Oscillation	27
4.3.5. Peak-to-peak amplitude.....	27
4.3.6. Statistical Features	28
4.3.7. Compare left and right hemispheres	29
4.4. Feature Reduction	29
4.4.1. Principal Component Analysis (PCA)	30
4.4.2. Correlation coefficients.....	31
4.4.3. Analysis of variance (ANOVA).....	32
4.4.4. Recursive Feature Elimination (RFE).....	33
4.5. Emotion Classification.....	33
4.5.1. Support Vector Machine (SVM)	34
4.5.2. k-Nearest Neighbors (k-NN).....	34
4.5.3. Linear Discriminant Analysis (LDA)	35
4.5.4. Multi-Layer Perceptron (MLP)	36

4.5.5. Naïve Bayes	37
4.5.6. Decision Trees.....	37
4.6. Validation Scheme.....	38
4.6.1. K-fold cross-validation scheme	39
4.6.2. Leave-p-out cross-validation scheme.....	39
4.7. Measuring Model Performance	40
4.7.1. Confusion Matrix	40
4.7.2. Accuracy	41
4.7.3. Precision.....	41
4.7.4. Recall.....	42
4.7.5. F-Score	42
4.8. Summary	42
Chapter 5 DEAP Dataset and Previous Work	44
5.1. Introduction.....	44
5.2. DEAP Dataset	44
5.3. Previous Work	46
5.3.1. Koelstra et al. (2012).....	46
5.3.2. Matiko et al. (2014).....	47
5.3.3. Jirayucharoensak et al. (2014)	47
5.3.4. Daimi et al. (2014)	48
5.3.5. Chen et al. (2015)	49
5.3.6. Gao et al. (2015).....	49
5.3.7. Li et al. (2016).....	50
5.3.8. Tripathi et al. (2017)	51
5.3.9. Chao et al. (2018)	51

5.4. Discussion of Previous Works	53
5.5. Summary	58
Chapter 6 Proposed Model for Emotion Recognition using Brain Signals.....	59
6.1. Introduction.....	59
6.2. The Proposed Model	59
6.3. EEG Signals Preprocessing	59
6.4. Feature Extraction.....	62
6.4.1. Spectral Power features.....	62
6.4.2. Oscillation features.....	63
6.4.3. Shannon entropy.....	65
6.5. Feature Reduction	65
6.6. Emotions Classification	67
6.7. Tools	67
6.8. Summary	67
Chapter 7 Experimental Results and Discussion.....	68
7.1. Introduction	68
7.2. EEG signals preprocessing	68
7.3. Feature Extraction.....	72
7.4. Features correlation with emotional scales.....	72
7.5. Verifying the preprocessing and feature extraction steps.....	77
7.6. Checking the worthiness of partitioning the Dataset.....	78
7.7. Evaluating the Proposed Model	79
7.8. Discussion	83
7.8.1. Classical Machine Learning Studies	83
7.8.2. Deep Learning Studies	83

7.9. Summary	84
Chapter 8 Conclusion and Future Work	85
8.1. Thesis Contribution.....	85
8.2. Future Work	86
References	87
Appendix A: DEAP Dataset	99
Appendix B: MATLAB Scripts.....	111
Appendix C: Scikit-Learn Toolkit	117

LIST OF FIGURES

Figure 2.1: Valence-Arousal and Valence-Potency spaces [Fontaine et al., 2007]	7
Figure 2.2: Four quadrants of valence-arousal space	8
Figure 2.3: Different ways to observe emotion [Christy et al., 2012].....	9
Figure 3.1: Parts of a human brain [Inside The Brain, 2013].....	11
Figure 3.2: Stages of a BCI system [Mühl et al., 2014]	13
Figure 3.3: Some EEG devices.....	15
Figure 3.4: 10-20 system [Nicolas-Alonso et al., 2012]	16
Figure 3.5: 10-10 system [ACNS, 2006].....	16
Figure 4.1: Time domain and frequency domain of a signal.....	22
Figure 4.2: Common filter types based of frequency response [Smith, 2003].....	24
Figure 4.3: The process of extract features of EEG signals	25
Figure 4.4: Peak-to-peak amplitude [Learning about Electronics, 2017]	27
Figure 4.5: Example of the mapping process by PCA [Alpaydin, 2014].....	31
Figure 4.6: The optimal hyperplane obtained by SVM [Lotte et al., 2007]	34
Figure 4.7: The obtained hyperplane by LDA [Alpaydin, 2014]	35
Figure 4.8: A neural network structure of MLP [Cazala, 2017]	36
Figure 4.9: A dataset and its corresponding decision tree [Alpaydin, 2014]	38
Figure 5.1: A participant watching a music video.....	45
Figure 5.2: 10–20 system with 32 channels	45
Figure 5.3: Accuracies of valence and arousal for different features.....	57
Figure 5.4: Accuracy ratios of different classifiers relative to SVM	58
Figure 6.1: Proposed Model for EEG-based Emotion Recognition	60
Figure 6.2: Digital filter for Delta waves	61
Figure 6.3: Digital filter for Theta waves	61
Figure 6.4: Digital filter for Alpha waves	61
Figure 6.5: Digital filter for Beta waves.....	62
Figure 6.6: Digital filter for Gamma waves	62
Figure 7.1: EEG signals of one electrode	69

Figure 7.2: Time domain & Frequency domain of EEG signal	70
Figure 7.3: Time domain & Frequency domain of Delta signal	70
Figure 7.4: Time domain & Frequency domain of Theta signal	70
Figure 7.5: Time domain & Frequency domain of Alpha signal	71
Figure 7.6: Time domain & Frequency domain of Beta signal.....	71
Figure 7.7: Time domain & Frequency domain of Gamma signal	71
Figure 7.8: Accuracy of different classifiers for each emotional scale	80
Figure 7.9: F ₁ -score of different classifiers for each emotional scale	81
Figure 7.10: Comparing accuracies with the previous studies	82
Figure 7.11: Comparing F ₁ -scores with the previous studies.....	82

LIST OF TABLES

Table 3-1: Approaches of BCI systems [Nicolas-Alonso et al., 2012]	14
Table 4-1: Confusion matrix	40
Table 5-1: The summary of the presented studies.....	52
Table 5-2: Summary of the used frequency bands in the presented studies	55
Table 5-3: Comparing the previous studies in terms of dataset partitioning and validation scheme	56
Table 7-1: Part of EEG data of one trial.....	69
Table 7-2: Summary of the extracted features	72
Table 7-3: Significant feature correlations with Valence scale.....	73
Table 7-4: Significant feature correlations with Arousal scale	73
Table 7-5: Significant feature correlations with Dominance scale	74
Table 7-6: Significant feature correlations with Liking scale	75
Table 7-7: Number of electrodes with significant correlation ($p<0.01$) for each feature and frequency band.....	76
Table 7-8: Results of reproduction Koelstra et al. work	77
Table 7-9: Results of dataset partitioning experiment.....	78
Table 7-10: Results of evaluating different classifiers	80
Table 7-11: Comparing the proposed model with previous studies	81

LIST OF ALGORITHMS

Algorithm 6.1: Calculating the oscillation feature [Matiko et al., 2014]	63
Algorithm 6.2: Calculating the Shannon entropy feature based on [Shannon, 1948]..	64
Algorithm 6.3: SVM-RFE algorithm [Guyon et al., 2002]	66

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
ANOVA	Analysis of Variance
BCI	Brain-Computer Interface
CLIS	Complete Locked-In State
CNN	Convolutional Neural Network
CNS	Central Nerves System
CSA	Covariate Shift Adaptation
CWT	Complex Wavelet Transform
DBN	Deep Belief Network
DEAP	a Database for Emotion Analysis using Physiological signals
DFT	Discrete Fourier Transform
DLN	Deep Learning Network
DNN	Deep Neural Network
DSP	Digital Signal Processing
DWT	Discrete Wavelet Transform
DWPT	Discrete Wavelet Packet Transform
DT-CWT	Dual-Tree Complex Wavelet Transform
DT-CWPT	Dual-Tree Complex Wavelet Packet Transform
ECG	Electrocardiogram
ECoG	Electrocorticography
EEG	Electroencephalography
EMG	Electromyography
EOG	Electrooculography
FA	Factor Analysis
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
fMRI	Functional Magnetic Resonance Imaging
GSR	Galvanic Skin Response

HBN	Hierarchical Bayesian Network
HCI	Human-Computer Interaction
IIR	Infinite Impulse Response
k-NN	k-Nearest Neighbors
LDA	Linear Discriminant Analysis
LOO	Leave-One-Out
LSTM	Long-Short Term Memory
MAE	Mean Absolute Error
MAP	Maximum A Posteriori rule
MEG	Magnetoencephalography
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
NIRS	Near Infrared Spectroscopy
PCA	Principal Component Analysis
PET	Position Emission Tomography
PSD	Power Spectral Density
QRcp	QR factorization with column pivoting
RBF	Radial Basis Function
RFE	Recursive Feature Elimination
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SVD	Singular Value Decomposition
SVM	Support Vector Machines

CHAPTER 1

INTRODUCTION

Emotions are an important part in the communication process between people. When we are speaking, our emotion has a huge impact on what the others will understand and feel. The word “OK” with an emotion of anger or discontent will make others feel that we are forced to accept. On the other hand, an emotion of happiness will make them feel that we are really satisfied. Despite the importance of emotions in people communications, most of currently Human-Computer Interaction (HCI) systems lack the ability to recognize and understand human emotions [Koelstra et al., 2012].

Humans express their emotion implicitly by many ways, such as facial expression, voice intonation and body movement. By monitoring these phenomena, the human emotion can be recognized. However, humans can fake those phenomena and express a false emotion. Another way to recognize human emotion is by monitoring internal physiological signals of the human body, such as heart rate, respiration and brain signals. Because it is harder to fake these physiological signals, they are considered more reliable for emotion recognition [Petrantonakis et al., 2010].

Brain-Computer Interface (BCI) is a new research area that interested in interpreting user intention based on monitoring brain activities, which can be used in recognizing user emotion. Brain activities can be monitored by many methods, such as Electroencephalography (EEG), Position Emission Tomography (PET), Magnetoencephalography (MEG) and functional Magnetic Resonance Imaging (fMRI). EEG is the most used modality to monitor brain signals. It measures the electrical activities of the brain through a set of electrodes placed on the scalp. It has a high temporal resolution with no risks, and it is relatively cheap. During the last few years, many researchers and companies became interested in the field of BCI. Many commercial BCI systems have been developed, most or all those BCI systems are based on EEG. Those BCI systems are relatively cheap and easy to setup and use by ordinary users [Nicolas-Alonso et al., 2012].

EEG-based emotion recognition involves few steps to be accomplished. Firstly, a preprocessing step is needed to reduce noise in EEG signals and separate them to their underlying components. Then, EEG signals are mapped into feature vectors which are more suitable for classification. A huge number of features can be extracted from the preprocessed EEG signals, so a feature reduction method is used to reduce the number of extracted features. Finally, user emotion is recognized by a classifier in the classification step [Koelstra et al., 2012][Nicolas-Alonso et al., 2012].

1.1. Problem Definition

Emotion recognition is considered as a hot topic for research recently, but most current approaches have a low emotion recognition accuracy. Even though some researches show good accuracies in emotion recognition, they built their models depending on special group of users, i.e., each individual has his own model, or a model was built for each gender. The problem with these models is that, they can't be reused directly with new users. The approaches that built a separated model for each individual will require a new user to retrain the system before using it, while the approaches that built models depending on gender or other characteristics will require a configuration step before using them.

There is a need to build an effective approach that generally can recognize the human emotion with higher accuracy, and on the same time, is not depending on special characteristics of users.

1.2. Research Objective

The main objective of this research is building a model that improves emotion recognition using the recorded brain signals by EEG devices. The provided model must be able to handle different kinds of users without the need to be retrained or reconfigured. This can be achieved by exploring the most common EEG-based emotion recognition methods, evaluating these methods and adopting the best methods for emotion recognition.

1.3. Thesis Organization

The rest of this thesis is organized as follows:

- **Chapter 2:-** discusses some problems related to human emotions such as: what is the emotion definition, what is affective computing, different approaches to represent emotions, and how emotions can be observed.
- **Chapter 3:-** gives an overview of how to monitor brain signals. It shows the human brain anatomy and different ways to monitor brain activities. The concept of Brain-Computer Interface (BCI) is introduced with the focus on Electroencephalography (EEG) as the most common way to monitor brain signals.
- **Chapter 4:-** introduces a background on machine learning techniques that commonly used to recognize human emotions using brain signals.
- **Chapter 5:-** presents the used dataset in the thesis and the previous work on that dataset.
- **Chapter 6:-** presents the proposed model for emotion recognition using brain signals.
- **Chapter 7:-** shows the experiments to evaluate the proposed model and discusses the results of these experiments.
- **Chapter 8:-** concludes this thesis and suggests future work.

CHAPTER 2

EMOTION RECOGNITION

2.1. Introduction

Emotion recognition is the process of identifying human emotion. Giving computers the ability to recognize and express human emotions will enhance their performance in assisting humans. Many fields will be affected by enhancing the ability of computers to recognize emotions, such as, social media, multimedia auto-tagging, computer-assisted learning, entertainment and healthcare. As example of benefits that obtained from enhancement of emotion recognition: the ability of attaching our real emotion with a Facebook post, the ability to auto-tag a YouTube video with the user emotion which will give better search results, the ability to skip boring contents of a lecture while providing computer-assisted learning courses, a strong impact in making videogames industry, and the ability to recognize the emotion of patients that have disabilities and can't express their health status [Picard, 1995].

This chapter discusses some problems related to human emotions such as: what is the emotions definition, what is affective computing, different approaches to represent emotions, and how emotions can be observed.

The organization of this chapter is as follows: section 2.2 gives some basic definitions. Emotion representation approaches are presented in section 2.3 while section 2.4 gives a summary of the ways of emotion observation. Finally, section 2.5 summarizes this chapter.

2.2. Basic Definitions

The following subsections discuss the definition of emotion and affective computing.

2.2.1. Emotion Definition

Emotion refers to the changes in the psychological and physical state as a response to internal or external stimulus event. However, there is no widespread consensus on the definition of emotion. Not just that, but also there is an overlapping among the concepts of emotion, feeling and mood [Scherer, 2005].

Affective state is more general term that is used by many researchers interchangeably with emotional state [Picard, 1995], and it can include the concepts of emotion, mood, etc.

2.2.2. Affective Computing Definition

Affective computing is a new research field that recently has an increasing interest. Affective computing is concerned with studying and designing systems that can recognize, interpret and simulate the affective state of humans [Picard, 1995].

2.3. Emotion Representation (Measurement)

One of the important issues in that research area is how to represent emotions. Although there are many defined models for emotion representation, there is no global agreement on what model must be used. Most defined models for emotion representation fall under one of two major approaches, the simplest one is to use distinct words for each emotion, and the other one is to represent emotions through multidimensions scales [Scherer, 2005]. The following subsections discuss those two approaches.

2.3.1. Discrete Categories Approach

In this approach emotions are represented with discrete categories, such as anger, fear and happiness. It is close to common sense of human, but the main limitation of this approach is that there is no global agreement on what categories should be used [Scherer, 2005]. In addition, there are difficulties in translating these categories between different cultures, the word that represents an emotion in a culture may have no

equivalent in another culture [Soleymani et al., 2011], e.g. the “disgust” emotion in English has no exact equivalent in Polish [Russell, 1991].

An example of researchers that try to define these categories is *Ekman et al.*, where they defined six basic emotions: happiness, surprise, sadness, fear, disgust and anger [Ekman et al., 1987].

Emotions recognition in this approach can be considered as a classification problem.

2.3.2. Multi-dimensions Space Approach

In this approach, emotions are represented through a set of scales, each scale is considered as a dimension in a multi-dimensions space. Each scale has a minimum and maximum value, and it can be continuous or discrete. A specific emotion can be defined by a combination of values for each scale or a point in the multi-dimensions space [Scherer, 2005], so the researcher can concentrate his attention on the emotion recognition problem without worrying about what categories of emotion should be used.

One of the most used models in this approach is the valence-arousal model, which designed by *Russell* [Russell, 1980], in this model, emotions are represented by a space of two dimensions, the first one is the valence scale ranged from unpleasant to pleasant and the second one is the arousal scale ranged from inactive to active. A third scale can be added to that model [Russell et al., 1977][Mehrabian, 1996], the dominance (or potency) scale ranged from submissive to dominant. An example for the use of dominance scale is to distinguish between “anger” and “fear” emotions, because they are close to each other in terms of the valence and arousal scales, but they are different in terms of the potency scale, i.e. “anger” has an extreme value in the direction of dominant, whereas “fear” has an extreme value in the direction of submissive [Fontaine

et al., 2007]. Figure 2.1 shows the valence-potency space (a) and the valence-arousal space (b) with some examples of emotion categories mapped on them.

In this approach, researchers can consider each scale as a regression problem, or split each scale to a set of levels, and consider it as a classification problem. A common

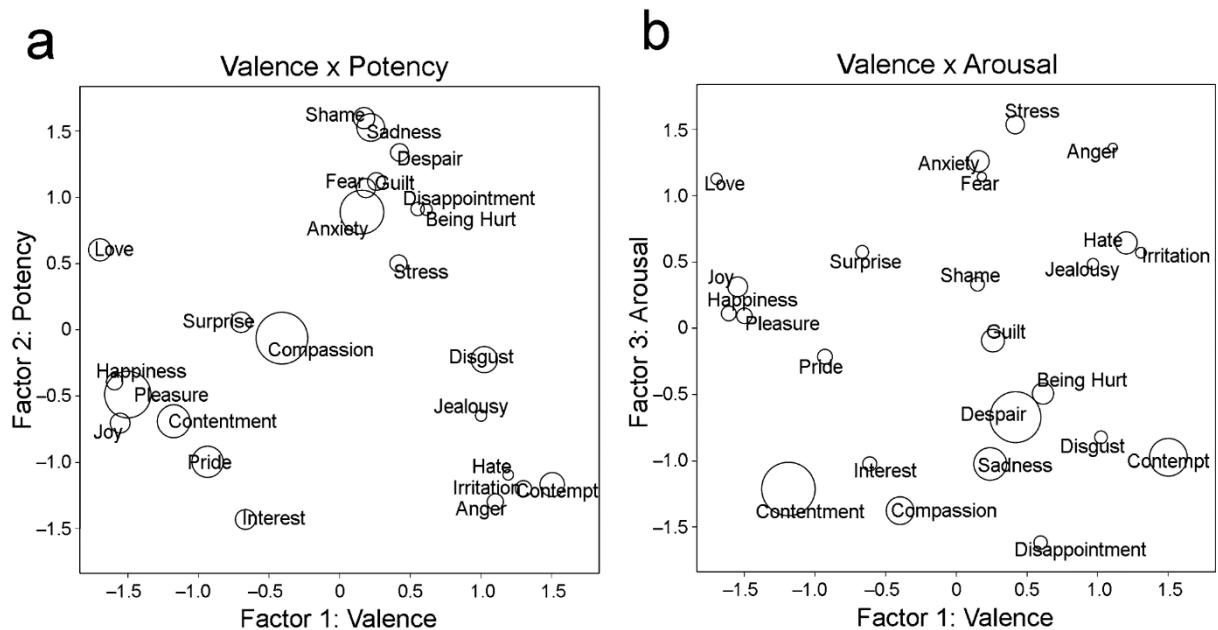


Figure 2.1: Valence-Arousal and Valence-Potency spaces [Fontaine et al., 2007]

example of splitting each scale is to split each scale into two levels: high and low, so the valence-arousal space will be split into four quadrants: High Valence with High Arousal (HVHA), High Valence with Low Arousal (HVLA), Low Valence with High Arousal (LVHA) and Low Valence with Low Arousal (LVLA) [Koelstra et al., 2012]. Figure 2.2 shows these four quadrants, which resulting from splitting the valence-arousal space.

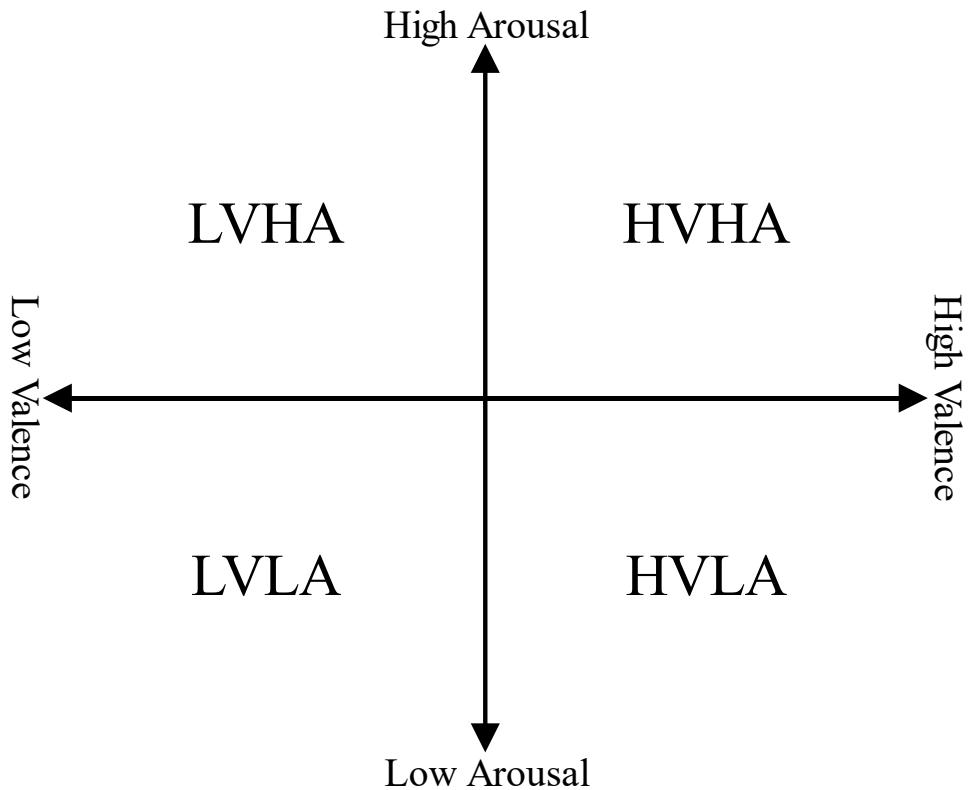


Figure 2.2: Four quadrants of valence-arousal space

2.4. Emotion Observation

Emotions can be observed through many non-verbal ways, such as facial expressions, voice intonation and body movement. Emotions can also be observed through internal physiological signals, such as heart rate, skin conductance, respiration, Galvanic Skin Response (GSR), Electroencephalography (EEG), Positron Emission Tomography (PET), Magnetoencephalography (MEG) and functional Magnetic Resonance Imaging (fMRI). The ways that based on physiological signals are considered more reliable than other ways specially the signals from Central Nerves System (CNS), such as EEG, MEG, PET and fMRI [Petrantonakis et al., 2010]. Figure 2.3 shows different ways that can be used to observe emotions.

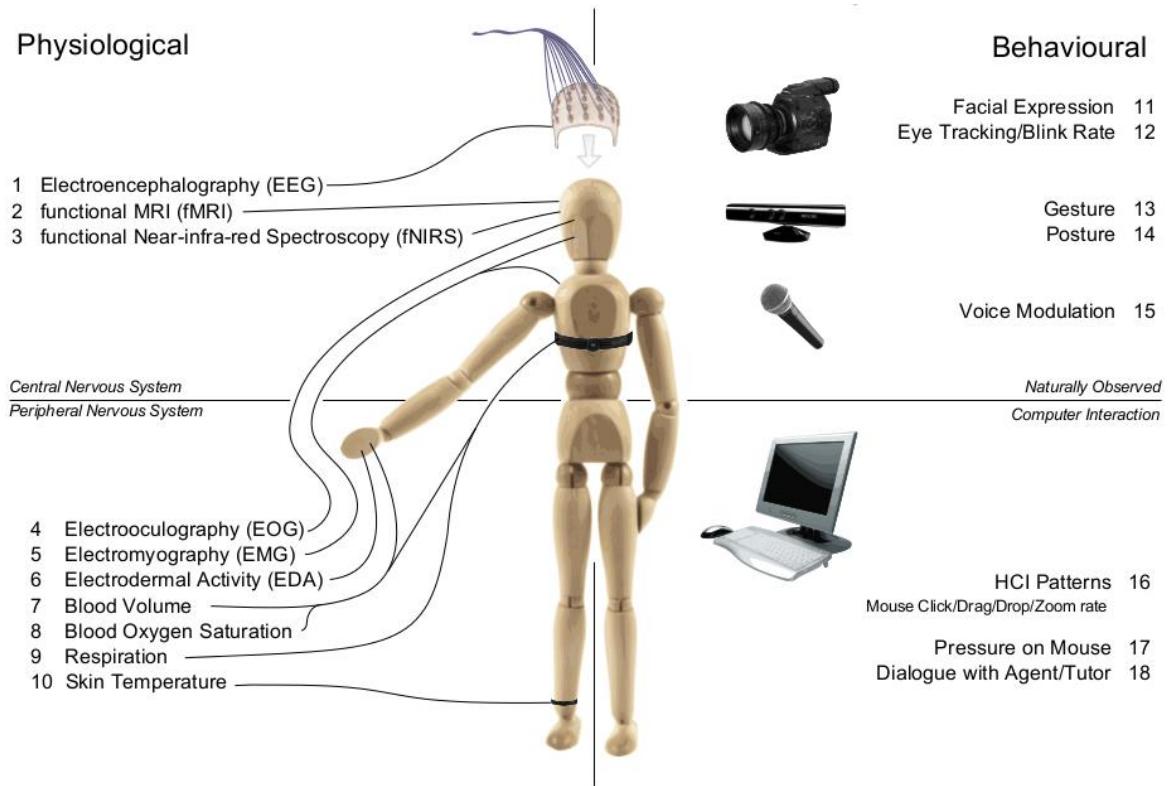


Figure 2.3: Different ways to observe emotion [Christy et al., 2012]

2.5. Summary

Emotion recognition has become a hot topic for research nowadays. Emotion is an important part of the communication process between people, but most HCI systems lack the ability to recognize it. There are many defined models to represent emotions, and most of these models can be categorized into one of two major approaches: the first one is to use distinct words for each emotion, and the second is to represent emotions through multidimensions scales. There are many non-verbal ways to observe emotions, such as heart rate, skin conductance, respiration, GSR, EEG, PET, MEG and fMRI. The ways that based on physiological signals specially from CNS are considered more reliable than other ways.

CHAPTER 3

MONITORING BRAIN SIGNALS

3.1. Introduction

The brain controls most of other organs of the human body, brain usually sends signals through the nerves to organs telling the muscles to contract, for example, nerves send lots of electrical impulses to different muscles in hand allowing a person to move his hand with extreme precision. By monitoring and analysis brain signals, user intentions can be predicted. Brain-Computer Interface (BCI) is a research field that interested in interpreting user intentions based on monitoring brain activities. It is very helpful for patients with severe motor disabilities and may be the only hope for communicating with Complete Locked-In State (CLIS) patients who have lost all motor control. There are many applications of BCI for motor disable patients, such as word processors, web browsers, motor restoration and brain control of a wheelchair or other environmental devices. Moreover, BCI is increasingly used by healthy people in neuromarketing and video games as a tool to reveal emotional information that cannot be reported easily. BCI can also be used with people that lack the ability to express their emotional state such as patients that suffer from autism, schizophrenia or depression [Nicolas-Alonso et al., 2012].

This chapter gives an overview of brain signals monitoring. It shows the human brain anatomy and different ways to monitor the brain activities. The concept of Brain-Computer Interface (BCI) is introduced with the focus on EEG.

This chapter is organized as follows: section 3.2 briefly discuss the brain anatomy. The concept of brain-computer interface is introduced in section 3.3. Electroencephalography is discussed in section 3.4. Finally, section 3.5 summarizes this chapter.

3.2. Brain Anatomy

Brain is the most mysterious organ of human body. It controls most of other organs. It controls our movements, behavior and of course our emotions. Human brain consists of three parts: the cerebrum, the brainstem and the cerebellum. The cerebrum is the largest part of the human brain, and it is divided into two cerebral hemispheres that are separated by a groove. Each cerebral hemisphere has an outer layer of gray matter called cerebral cortex and an inner layer of white matter. The cerebral cortex of each hemisphere is generally divided into four areas or lobes, named after the four skull bones protecting them: the frontal lobe, the temporal lobe, the parietal lobe, and the occipital lobe. Each area is associated with a specific function [Rosdahl et al., 2011]. Figure 3.1 shows different parts of human brain.

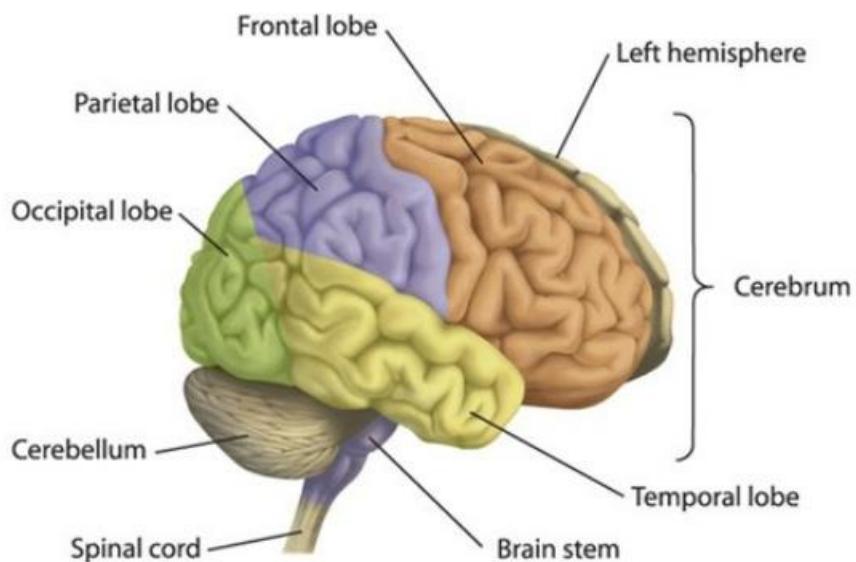


Figure 3.1: Parts of a human brain [Inside The Brain, 2013]

The cerebral cortex contains approximately 15–33 billions of neurons [Pelvig et al., 2008]. Each neuron is connected by synapses to several thousand other neurons. These neurons communicate with each other by means of long protoplasmic fibers called axons which carry trains of signal pulses called action potentials targeting cells of other areas of the brain. Activating synapses by action potential releases chemicals called neurotransmitters which

attached to the receptor molecules on the membrane of the synapse's target cell, and so alter the electrical or chemical properties of the receptor molecules. These electrochemical processes used by neurons when communicate with each other make the cerebral cortex generates electric fields. This electric field can be large enough to be detected outside the skull [Kandel et al., 2013].

3.3. Brain-Computer Interface (BCI)

Brain-Computer Interface (BCI) is a communication system that offers a direct interface between human brain and the computer without the need to use other body organs. The traditional target of this technology was the individuals with severe motor disabilities, because it would improve the quality of their lives and reduce the cost of intensive care. BCI now not only used for locked-in people but also for normal people in many life fields such as entertainment and marketing. Through the last two decades BCI has spread widely and attracted a lot of researchers recently [Nicolas-Alonso et al., 2012].

The process of recognizing user intentions by interpreting brain signals in a typical BCI system consists of the following stages: signal acquisition, preprocessing, feature extraction and selection, classification, and control interface [Khalid et al., 2009]. In the signal acquisition stage, the brain signals are captured. In the preprocessing stage, a noise reduction and artifact removal are performed, and the signals are prepared for further processing. In the feature extraction stage, the signals are mapped into a vector of features that is more suitable for the classification stage. It is important for the feature vector dimensions to be as low as possible, but without the loss of relevant information. In the classification stage, the feature vector is classified, and the user's intentions are interpreted. Finally, in the control interface stage, the classification results are used as commands for any connected devices. Figure 3.2 shows typical stages of a BCI system.

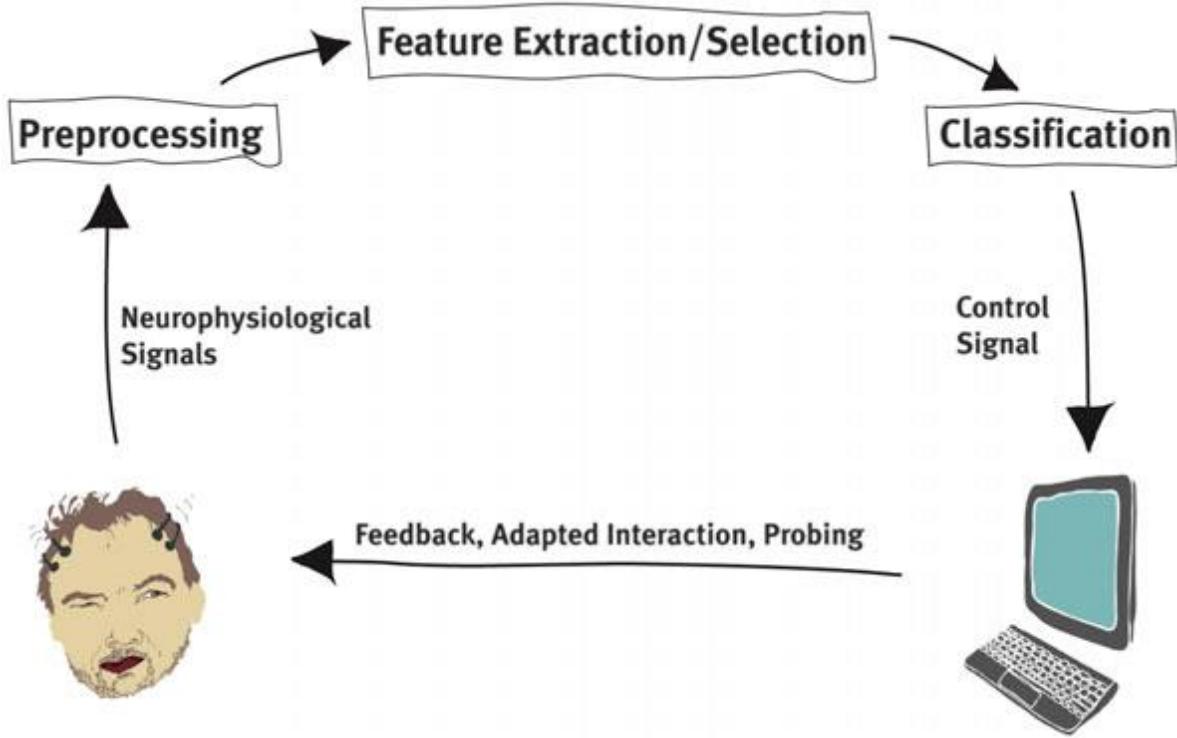


Figure 3.2: Stages of a BCI system [Mühl et al., 2014]

3.3.1. Neuroimaging Approaches

BCI systems rely on monitoring brain activities to gather information on the user intentions. There are two types of brain activities that can be monitored: electrophysiological and hemodynamic.

Electrophysiological activities are generated by the electrical potentials of impulse transmission in the nervous system while exchanging information between neurons. By monitoring the electrophysiological activities of the brain, electrophysiological based approaches monitor the neuronal activity directly. It can be monitored by Electroencephalography (EEG), Electrocorticography (ECoG), Magnetoencephalography (MEG) and intracortical neuron recording [Baillet et al., 2001] [Nicolas-Alonso et al., 2012].

Electrophysiological based approaches can be invasive or non-invasive. In invasive approaches, micro-electrodes are implanted inside the skull which involves significant health risks. While in non-invasive approaches, signals are captured by external devices

(e.g. Placing electrodes on surface of scalp), so there are no risks in these methods, but they produce poor resolution signals [Nicolas-Alonso et al., 2012].

Hemodynamic activities are generated due to the difference between the rate of glucose released by blood to active neurons and inactive neurons. Hemodynamic based approaches monitor the neuronal activity indirectly. It can be monitored by functional Magnetic Resonance Imaging (fMRI) and Near Infrared Spectroscopy (NIRS) [Laureys et al., 2009][Nicolas-Alonso et al., 2012]. Table 3-1 compares between different neuroimaging approaches that are used in BCI systems. Note that, approaches with smaller temporal and spatial resolutions are better than the larger ones.

EEG is now the most used modality in the field of BCI and has a great attention recently [Nicolas-Alonso et al., 2012], so the rest of this work focus on the use of EEG for emotion recognition.

Table 3-1: Approaches of BCI systems [Nicolas-Alonso et al., 2012]

Neuroimaging method	Activity measured	Direct/Indirect	Temporal resolution	Spatial resolution	Risk	Portability
EEG	Electrical	Direct	~0.05 s	~10 mm	Non-invasive	Portable
MEG	Magnetic	Direct	~0.05 s	~5 mm	Non-invasive	Non-portable
ECoG	Electrical	Direct	~0.003 s	~1 mm	Invasive	Portable
Intracortical Neuron Recording	Electrical	Direct	~0.003 s	~0.1 mm	Invasive	Portable
fMRI	Metabolic	Indirect	~1 s	~1 mm	Non-invasive	Non-portable
NIRS	Metabolic	Indirect	~1 s	~5 mm	Non-invasive	Portable

3.4. Electroencephalography (EEG)

Electroencephalography (EEG) is a method to measure the electrical activities of the brain and it can be recorded through a set of electrodes placed on the scalp [Baillet et al., 2001], EEG first appearance was in 1924 [Baillet et al., 2001], it is usually used in medical fields, such as studying epilepsy or sleep disorders, it is a non-invasive method with a high temporal resolution [Nicolas-Alonso et al., 2012].

During the last few years, many companies became interested in BCI systems, specially systems that based on EEG. So, many commercial EEG devices have been produced. Those EEG devices are easier to setup and use compared to the EEG devices that are used at laboratories, where the number of electrodes is reduced according to the objective of each system and dry electrodes are used, which do not need gel unlike normal electrodes [Nicolas-Alonso et al., 2012]. Figure 3.3 shows some EEG devices: Normal EEG devices that are used in a laboratory (A and B), Emotive EPOC headset by Emotive (C) and Neurosky Mindwave headset by Neurosky (D).



Figure 3.3: Some EEG devices

3.4.1. 10-20 System

The electrodes, channels, are placed on the scalp according to a standard system called the international 10-20 system, which introduced by the American Electroencephalographic Society [Nicolas-Alonso et al., 2012].

The numbers “10” and “20” refer to the fact that the distance between any adjacent electrodes are either 10% or 20% of the total distance from front to back or right to left of the skull, and each electrode has a name that identifies it from other electrodes [Nicolas-Alonso et al., 2012]. Other versions of this system with higher resolution are

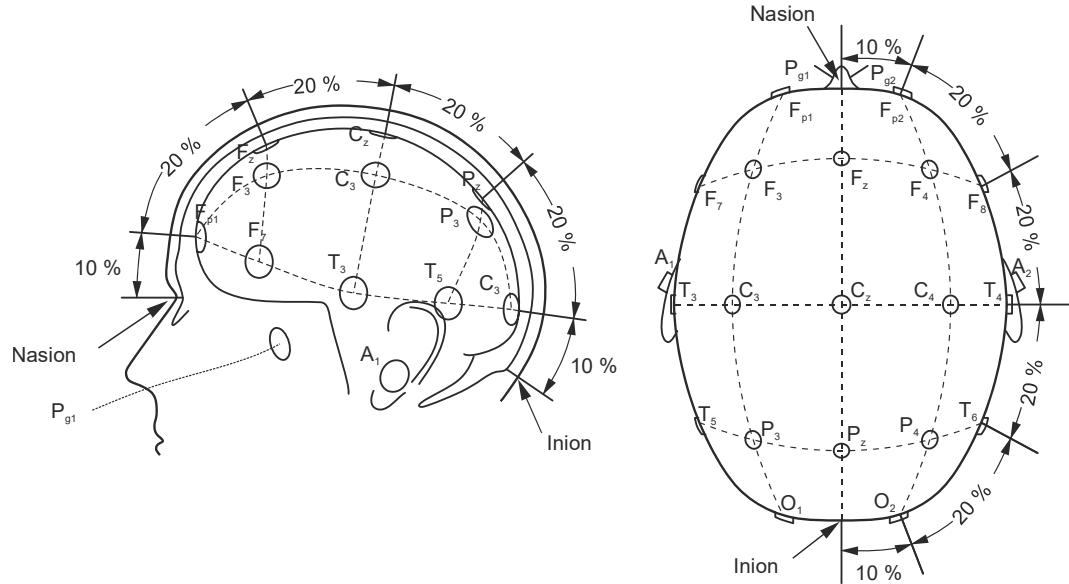


Figure 3.4: 10-20 system [Nicolas-Alonso et al., 2012]

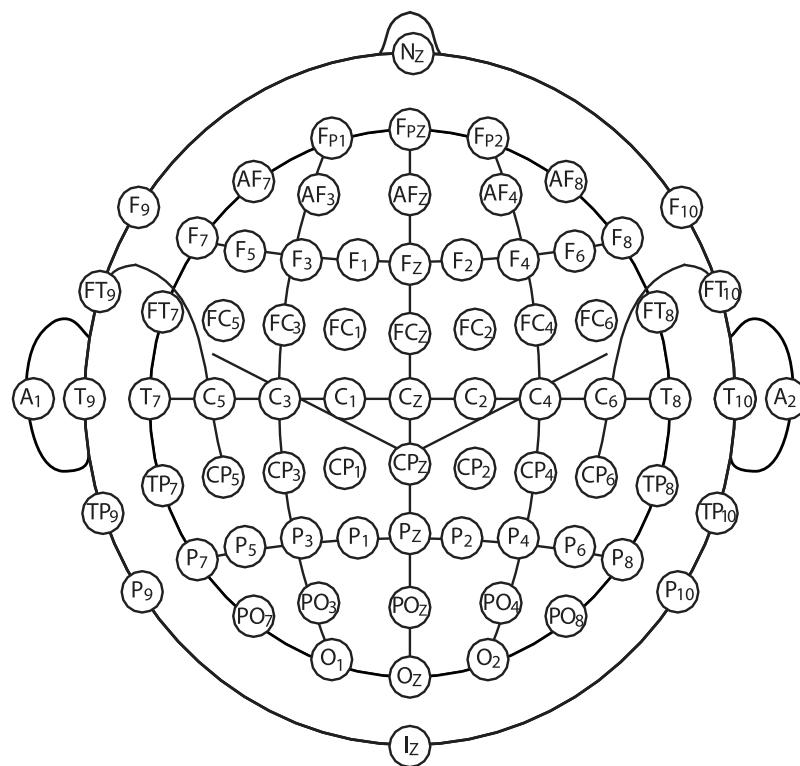


Figure 3.5: 10-10 system [ACNS, 2006]

defined, such as 10-10 system [ACNS, 2006], where distance between adjacent electrodes is only 10%. Figure 3.4 shows locations of electrodes according to 10-20 system while Figure 3.5 shows the locations according to 10-10 system.

From Figure 3.5 the one can notice that each electrode has a distinct name which consists of one or two letters postfixed by a number. Those letters are corresponding to the brain area where the electrode location exists, for example the letter F is for frontal, P for parietal, O for occipital, T for temporal and C for central. Electrode names on the left hemisphere have odd numbers while electrode names on the right hemisphere have even numbers.

Each electrode on the left hemisphere has identical opposite one on the right hemisphere with the reference number incremented by one, and each electrode with the identical opposite on the other hemisphere are called symmetrical pair of electrodes.

3.4.2. Rhythms of EEG signals

EEG signals comprises a set of rhythms which are related to some biological phenomena. These rhythms may be categorized according to their frequency band. Classical frequency bands are delta (0–4 Hz), theta (4–8 Hz), alpha (8–12 Hz), beta (12–30 Hz) and gamma (30+ Hz) [Nicolas-Alonso et al., 2012].

Delta frequency bands are below 4 Hz. Usually, delta waves are observed in babies and can be observed in adults in deep sleep. It is easy to confuse delta signals, due to their low frequency, with noise caused by the muscles of neck or jaw [Kübler et al., 2001].

Theta frequency bands lie between 4 and 8 Hz. Theta waves are seen usually in children, and they can be seen in adults in drowsy, sleep states, meditative concentration and many other cognitive processes [Kübler et al., 2001][Anand et al., 1961][Aftanas et al, 2001][Nicolas-Alonso et al., 2012].

Alpha frequency bands are in the range from 8 to 12 Hz. Alpha waves are related to visual processing, mental activities and may also be related to memory functions [Klimesch, 1997][Venables et al., 2009].

Beta frequency bands lie in 12 to 30 Hz range. Beta waves are associated with motor movement activities, where they distributed symmetrically when no motor activity exists but their distribution change during real movement or motor imagery [Pfurtscheller et al., 2001].

Gamma frequency bands are above 30 Hz. Gamma waves are related to certain motor activities and the perception of visual or auditory stimuli [Lee et al., 2003][Lutzenberger at al., 1995]. Gamma waves are usually affected by electromyography (EMG) or electrooculography (EOG) artifacts [Zhang et al., 2010].

3.4.3. First study to analysis EEG and Emotions

Hoagland et al. are the first researchers who studied the relationship between emotions and EEG in 1938, when they noticed in one of their patients on several occasions a sudden mark rises in Delta Index following emotionally disturbing experiences, so they made a separated study on a group of subjects to investigate that relationship. Some of the subjects are normal people and the others are patients with depression or schizophrenia. They noticed no significant difference between normal people and patients, and the results confirmed the relationship between EEG and emotions [Hoagland et al., 1937][Hoagland et al., 1938].

3.5. Summary

A human brain controls most of other body organs. It can be divided into many parts. Each part is responsible for doing a specific function. By monitoring the brain, user intentions can be reviled. BCI is a communication system that offers a direct interface between human brain and the computer without the need to use other body organs. A typical BCI system consists of five stages: signal acquisition, preprocessing, feature extraction, classification, and the control interface. Many methods can be used for acquisition brain signals, such as fMRI, NIRS, EEG, MEG, ECoG and Intracortical Neuron Recording.

EEG is a method to measure the electrical activities of the brain through a set of electrodes placed on the scalp. It is a non-invasive method with a high temporal resolution, and it is now the most used method in the field of BCI. The electrodes are placed on the scalp according to a standard system called the international 10-20 system. EEG signals comprises a set of rhythms which are related to some biological phenomena which are delta, theta, alpha, beta and gamma rhythms (frequencies).

CHAPTER 4

BACKGROUND KNOWLEDGE FOR EEG-BASED EMOTION RECOGNITION

4.1. Introduction

The process of recognizing human emotion using EEG brain signals is a very complicated process. EEG signals have noise and artifacts that needed to be removed, then the EEG signals are decomposed into their sub-frequency bands, next, feature extraction step is performed to convert EEG signals into a feature vector that is more suitable to the classification step, before the classification step, feature reduction techniques are applied to reduce the feature vector size, so the complexity and cost of the classification step are reduced. Classification step is then performed to interpret and estimate the user emotions. To validate the results of the used model or to compare results of different models, a validation scheme is used to measure the model performance [Koelstra et al., 2012][Nicolas-Alonso et al., 2012].

This chapter presents signal processing techniques that are used to reduce signal noise and split a signal to its sub-frequency, and common features that are extracted from EEG signals. It also provides a background knowledge on machine learning techniques that commonly used to recognize human emotions using the extracted features from EEG signals.

This chapter is organized as follows: Section 4.2 shows signal processing techniques components. Section 4.3 shows the common EEG-based features. Feature reduction techniques are discussed in section 4.4. The most common classifiers are presented in section 4.5. How to validate the built model is discussed in section 4.6. Section 4.7 shows how to measure the model performance. Finally, section 4.8 summarizes this chapter.

4.2. Signal Processing

The first step towards interpreting user intentions from brain signals is to reduce signal noise and process artifacts then EEG signals are decomposed into the previously mentioned rhythms. Signal processing techniques are used to do this task. The following subsections present the common signal processing techniques that are used to reduce the signal noise and to decompose a signal to its sub-frequency bands [Nicolas-Alonso et al., 2012].

4.2.1. Fourier Transform

Most of signals are functions of time, so they are represented in time domain. Another representation of signals is frequency domain, where signals are represented as a function of frequency, which is a sinusoidal wave. *Jean Baptiste Joseph Fourier*, a French mathematician and physicist interested in heat propagation, presented a paper on the use of sinusoids to represent temperature distributions. The paper contained the controversial claim that any continuous periodic signal could be represented as the sum of properly chosen sinusoidal waves [Smith, 2003]. Figure 4.1 shows a signal in time domain (a) and its frequency domain representation (f), where (b, c, d, and e) are the sinusoidal waves that construct the signal.

Fourier transform is a family of mathematical techniques, named after *Jean Baptiste Joseph Fourier*, that are based on decomposing signals into sinusoidal waves. Because most of nowadays signals are saved on a computer to deal and manipulate them, Discrete Fourier Transform (DFT) is the most used member of Fourier transform family. It is a member of the Fourier transform family that deals with discrete and periodic time domain signals. DFT and other Fourier transform members take a huge time to perform the transformation from time domain signals to frequency domain signals. In the light of this issue, an efficient algorithm is introduced to calculate the DFT. This algorithm is called Fast Fourier Transform (FFT) and it reduces the execution time by a huge amount, maybe by hundreds in some cases [Smith, 2003].

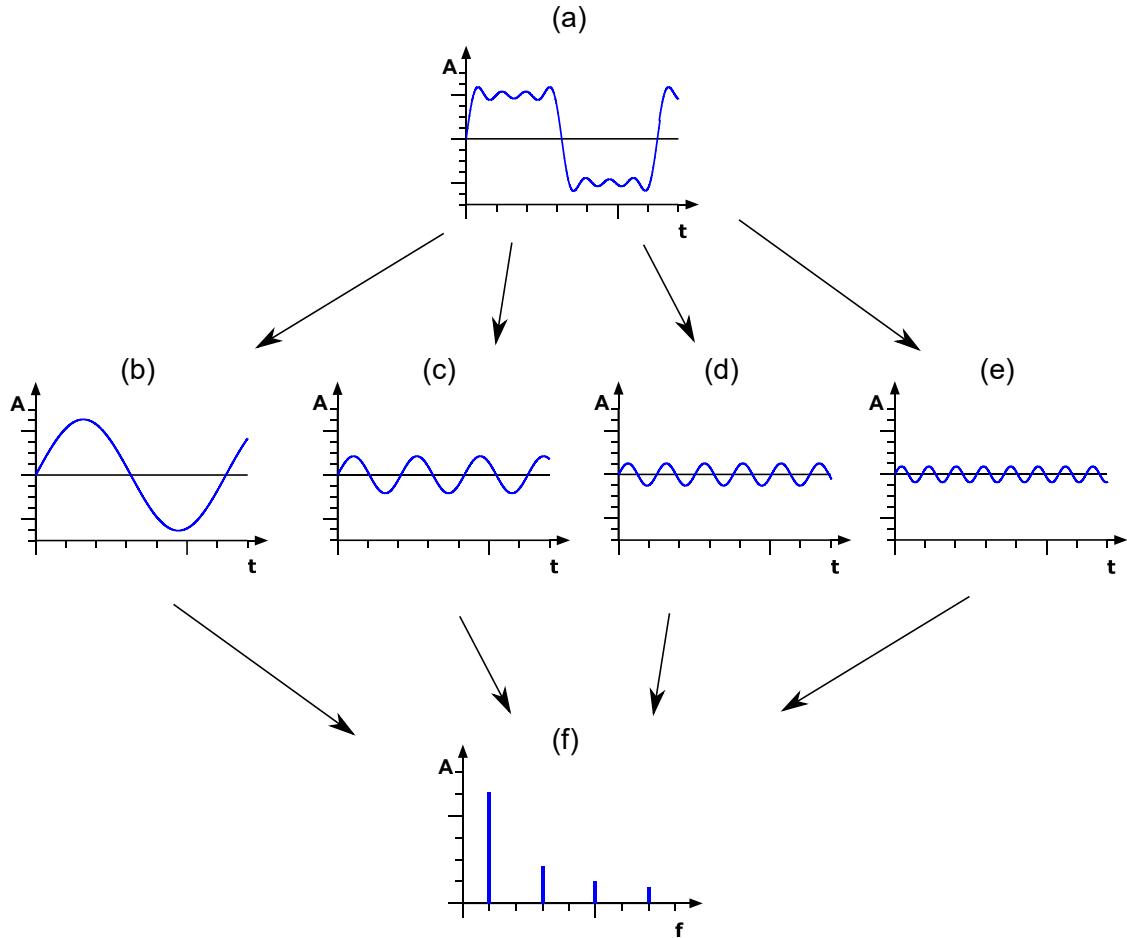


Figure 4.1: Time domain and frequency domain of a signal

4.2.2. Digital Filters

In Digital Signal Processing (DSP), filters are used for two general purposes: separating signals that have been combined and repairing signals that have been distorted. Two types of filters exist to do such tasks: analog (electric) filters and digital filters. Analog filters are cheap, fast, and have a large dynamic range in both amplitude and frequency. On the other hand, digital filters are vastly superior in the level of performance that can be achieved, which can be thousands of times better than analog filters. The extraordinary performance of digital filters is one of the key reasons that DSP has become so popular [Smith, 2003].

Digital filters can be studied and designed based on their response to the input signals. This response is called impulse response and it describes the reaction of the

filter as a function of time, when the impulse response is of a finite number of nonzero values, the designed filter is called a Finite Impulse Response (FIR) filter and it is implemented by convolving the input signals with the digital filter's impulse response. Another way to implement a digital filter is by recursion where the filter may have an internal feedback and may continue to respond indefinitely, this type of filters is called Infinite Impulse Response (IIR) filters. FIR filters can have far better performance than IIR filters, but they execute much more slowly. To solve this issue, FFT based convolution is used to allow FIR filters to execute faster [Smith, 2003].

Another aspect that can be used to study and design a filter is using its frequency response. Frequency response describes the filter as a function of frequency, and it can be found by taking the Fourier transform of the impulse response. Using the frequency response, a filter can allow some frequencies to pass unaltered, while completely blocking other frequencies. Frequencies that will pass are called passband, while others that will be blocked are called stopband. Based on passband and stopband frequencies, digital filters can be classified into four common types, which are low-pass filters, high-pass filters, band-pass filters and band-reject (or band-stop) filters [Smith, 2003]. Figure 4.2 shows these four common types of digital filters.

FIR filters that are used to separate frequency bands are called windowed-sinc filters. They are very stable, produce few surprises, and can be pushed to incredible performance levels. The ideal impulse response of windowed-sinc filters oscillates continually in both directions forever without dropping to zero amplitude, which is infeasible to be done with a computer program. To get around this problem, one solution is that, the ideal impulse response signal is to be truncated to $(M+1)$ points, symmetrically chosen around the main lobe, and other points outside this range are set to zero. Unfortunately, doing this will result in a poor frequency response and increasing the length of the impulse response (M) does not reduce this problem. To solve this issue, the truncated impulse response is multiplied by a smoothly tapered curve window, this will improve the frequency response of the filter. Several different windows are available such as: Hamming window, Blackman window, Bartlett window, and Hanning window [Smith, 2003].

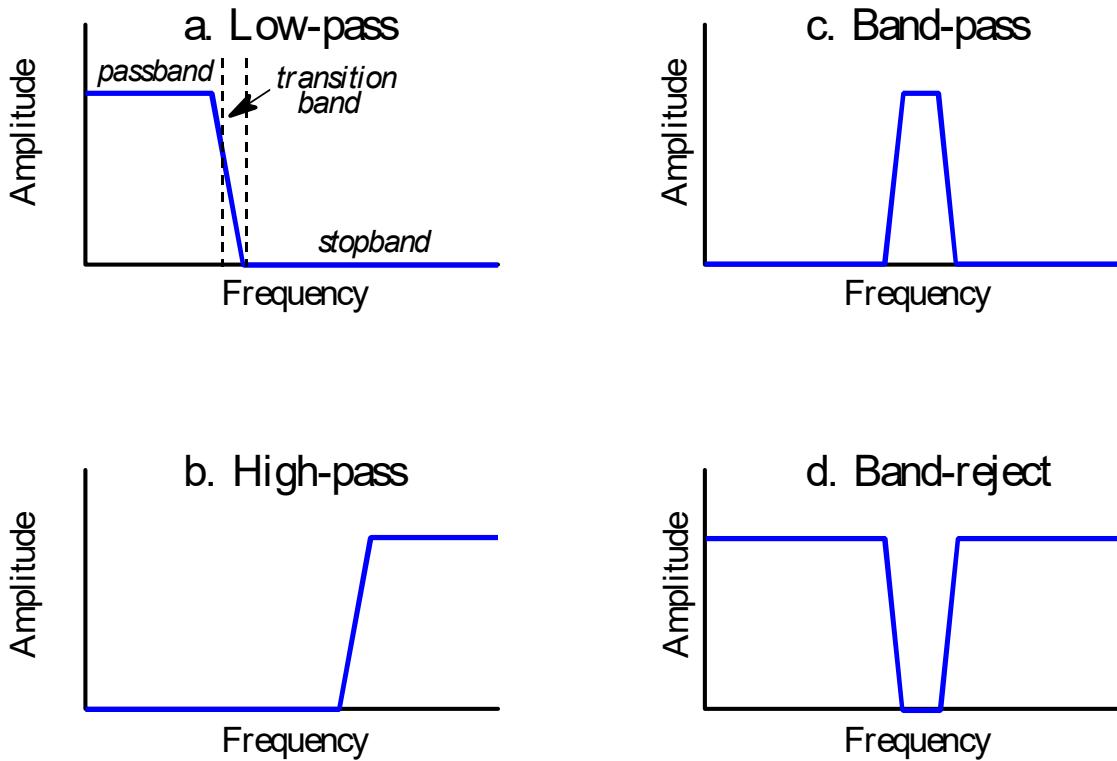


Figure 4.2: Common filter types based of frequency response [Smith, 2003]

4.2.3. Wavelet Transform

Wavelet transform is a family of transform functions that can be used to decompose signals. It looks like Fourier transform but it decomposes a signal into a set of wavelets. Unlike sinusoid waves which are symmetric, smooth and regular, wavelets can be either symmetric or asymmetric, sharp or smooth, regular or irregular, and they are localized in both time and frequency domains. wavelet based signal processing is suitable for nonstationary signals [Burrus et al., 1998].

There are two commonly used implementations of wavelet transformation. They are Discrete Wavelet Transform (DWT) and Continuous Wavelet Transform (CWT). DWT uses a discrete set of wavelet scales and translations, it decomposes the signal into mutually orthogonal set of wavelets. On the other hand, CWT uses arbitrary scales and decomposes the signal into wavelets which are not orthogonal and almost arbitrary [Burrus et al., 1998].

Other members of the family of wavelet transformation were proposed later to enhance or overcame shortages in older members. These members are Discrete Wavelet Packet Transform (DWPT), Dual-Tree Complex Wavelet Transform (DT-CWT) and Dual-Tree Complex Wavelet Packet Transform (DT-CWPT) [Bayram et al., 2008][Kingsbury, 2001].

4.3. Feature Extraction

After decomposing EEG signals into their sub-frequency bands, the EEG signals are mapped into feature vectors which are more suitable for the classification step. Figure 4.3 shows the process of decomposing EEG signals into sub-frequency bands and extracting features.

The following subsections present some types of features that are commonly extracted from EEG signals.

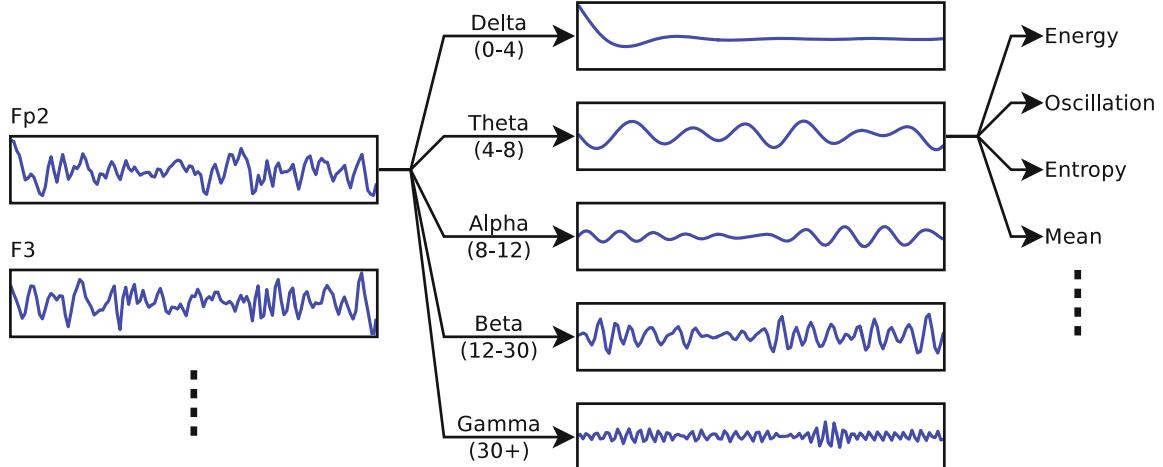


Figure 4.3: The process of extract features of EEG signals

4.3.1. Energy

Energy has many definitions and mostly related to the concept of energy in physics. In signal processing, energy is a measure of the signal strength. It is viewed as the area under the square of a signal. The following equation is used to calculate the energy of a signal.

$$E = \sum_{i=1}^n |x_i|^2 \quad (4-1)$$

Where x_i is the value of signal at sample i , it is notable that, if x_i is real, the modulus operator does not matter [Oppenheim et al., 2015][Stein, 2000][Smith, 2003].

4.3.2. Spectral Power

The spectral power, also called power spectrum or Power Spectral Density (PSD), is used to show how the power of a signal is distributed over frequencies. It uses windowing, averaging, and Fourier transforms to calculate the average power per frequency [Oppenheim, 1969]. A common method to extract this feature is Welch's method [Welch, 1967].

Other features that are commonly calculated based on the spectral power feature are the ratio of power of a specific frequency band signals to the total power of the electrode signals [Chen et al., 2015][Gao et al., 2015], and the ratio of a frequency band power to another frequency band power, such as the ratio of beta power to theta power [Chen et al., 2015].

4.3.3. Entropy

Entropy was originally defined for information theory by Shannon [Shannon, 1948], It is a measure of disorganization or uncertainty in a random variable. Johnson et al. apply entropy to a power spectrum of a signal [Johson et al., 1984]. In this context, entropy is used to measure the irregularity, complexity, or unpredictability characteristics of a signal. Since EEG signals have a near random distribution, entropy can be a very useful feature to be extracted from EEG signals [Chen et al., 2015]. Many

types of entropy can be calculated for EEG signals, such as, Shannon entropy, Spectral entropy and Kolmogorov entropy [Chen et al., 2015]. The following equation is used to calculate the Shannon entropy for a signal.

$$SE = - \sum_{i=1}^n p(x_i) \ln p(x_i) \quad (4-2)$$

Where $p(x_i)$ is a probability of a random phenomenon x_i .

4.3.4. Oscillation

Matiko et al. proposed the oscillation feature to give an insight of how signal power is related to activation and inactivation of certain areas of brain. It was obtained by counting all local maxima and local minima of the signal, and then dividing the count of signal samples by the sum of local maxima and local minima counts [Matiko et al., 2014].

4.3.5. Peak-to-peak amplitude

Peak-to-peak amplitude is the difference between the highest amplitude value (the top of the wave signal) and the lowest amplitude value (the bottom of the wave signal). Figure 4.4 illustrates how the peak-to-peak amplitude is measured.

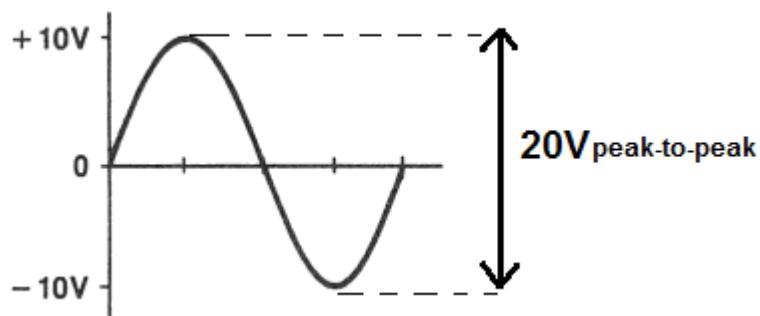


Figure 4.4: Peak-to-peak amplitude [Learning about Electronics, 2017]

4.3.6. Statistical Features

Many statistical features can be calculated from a signal, such as: mean, variance, standard deviation, skewness which measures the asymmetry of a signal about its mean and kurtosis which measures the tailedness of a signal. The following statements are equations of the common statistical features.

Mean:

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (4-3)$$

Variance:

$$V = \frac{1}{n+1} \sum_{i=1}^n |x_i - \mu|^2 \quad (4-4)$$

Standard Deviation:

$$\sigma = \sqrt{V} = \sqrt{\frac{1}{n+1} \sum_{i=1}^n |x_i - \mu|^2} \quad (4-5)$$

Skewness:

$$s = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^3}{\left(\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \right)^3} \quad (4-6)$$

Kurtosis:

$$k = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^4}{\left(\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \right)^2} \quad (4-7)$$

4.3.7. Compare left and right hemispheres

Another aspect of feature extraction is to measure the difference between left and right hemispheres across all symmetrical pairs of electrodes. This can be done after extracting a feature of the previously discussed features, and then calculate the difference between the feature value of one electrode on the left hemisphere and the value of the opposite electrode on the right. The first study that compared the left and right hemispheres in context of emotional state is *Harman and Ray*, where a significant difference between both hemispheres was found [Harman et al, 1977].

4.4. Feature Reduction

Extracting one kind of features from EEG will result in a huge set of features. For example, extracting the spectral power only will result in 230 features (32 electrodes \times 5 rhythms + 14 symmetrical pairs of electrodes \times 5 rhythms). Extracting more kinds of features will multiply this number. This huge set of features will affect the complexity of the classification step. Not only will the computational costs increase but also classification performance may be decreased [Guyon et al., 2003][Jain et al., 2000]. In this case, reducing the extracted features from EEG signals becomes a mandatory step.

Methods that are used for feature reduction can be categorized into two categories: dimensionality reduction and feature selection. Feature selection methods select a subset of the original attributes, on the other hand dimensionality reduction methods produce a linear combination of the original attributes [Janecek et al., 2008].

Dimensionality reduction methods create a new attribute space as combinations of the original attribute space to reduce the dimensionality of the data. The size of the new attribute space is less than the original attribute space without a lot of information losing. The main disadvantages of dimensionality reduction are that the new attribute space is not interpretable, and it is not possible to know how much the contribution of the original attributes. Dimensionality reduction methods can be unsupervised such as Principal Component Analysis (PCA) and Factor Analysis (FA) or supervised such as Linear Discriminant Analysis (LDA) [Liu et al., 1998][Janecek et al., 2008].

Feature selection methods select a subset of the original attributes and remove redundant or irrelevant attributes. The main advantage of feature selection methods is that the information of the importance of each feature is not lost. Feature selection can be done by a filtering-based method, where a subset of features is selected independently of the learning method, or by a wrapper-based method, where a subset of the features is selected based on its usefulness to learning method [Guyon et al., 2003].

The following subsections introduce some feature reduction methods that can be used to reduce the number of extracted features of EEG signals.

4.4.1. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is an unsupervised dimensionality reduction method. PCA maximizes the variance of the data without using the output information. PCA is interested in finding a projection of the inputs in the original d -dimensional space into a new ($k < d$)-dimensional space with the minimum loss of information. Consider a data matrix X with n rows that represent different observations and d columns that represent different features. We seek a transformation that maps X into a new matrix Z in such a way that, the first column is the most spread out so the difference between the observation points becomes most apparent. In other words, it should have the maximum possible variance. The second column should also maximize the variance as much as possible with the constraint to be orthogonal to the first one and so on with other succeeding weight vectors [Alpaydin, 2014]. This transformation is defined by $d \times d$ dimensional matrix W such that:

$$Z = X W^T \quad (4-8)$$

In such a way that, each weight vector w_i of W is successively inherit the maximum possible variance from X and constrained to be a unit vector. As a result, the dimensions with the too small variance of the output matrix Z can be ignored.

For the first principal component, we seek the weight vector w_1 that satisfy the following condition:

$$w_1 = \arg \max_{\|w\|=1} (w^T X^T X w) \quad (4-9)$$

Further components can be found by first subtracting the previous $k-1$ principal components from X as follows:

$$\hat{X}_k = X - \sum_{i=1}^{k-1} X w_i w_i^T \quad (4-10)$$

Then find the weight vector that maximize variance of the new data as follows:

$$w_k = \arg \max_{\|w\|=1} (w^T \hat{X}_k^T \hat{X}_k w) \quad (4-11)$$

Figure 4.5 shows an example of the mapping process to reduce dimensions, where PCA centers the samples and then rotates the axes to line up with the directions of the highest variance. If the variance of z_2 is too small then it can be ignored, and the dimensions are reduced from two to one [Alpaydin, 2014].

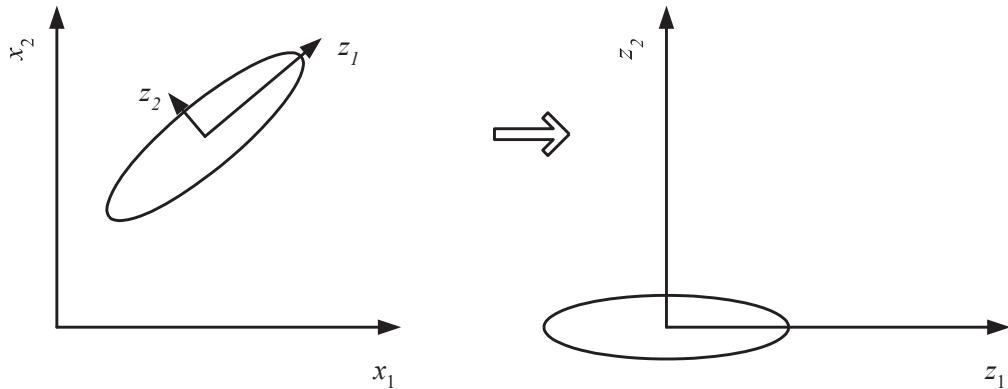


Figure 4.5: Example of the mapping process by PCA [Alpaydin, 2014]

4.4.2. Correlation coefficients

One of the common methods to select the most important features is to measure the correlation between the input feature vectors and their output labels. The correlation coefficient is measured for each feature separately and the features with a significant correlation coefficient is selected. The most common correlation methods are Spearman and Pearson correlations. Pearson correlation coefficient is a measure of the linear

correlation between two variables X and Y while Spearman correlation coefficient is a nonparametric measure between the ranking of two variables. The following equation is used to calculate Pearson correlation (R) between two variables X (values of a feature) and Y (the samples output):

$$R = \frac{cov(X, Y)}{\sqrt{var(X)var(Y)}} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2 \sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (4-12)$$

Where n is the number of samples, \bar{X} is the mean of one feature values and \bar{Y} is the mean of the samples output [Guyon et al., 2003][Dodge, 2008].

Spearman correlation (ρ) can be calculated by:

$$\rho = \frac{cov(R_X, R_Y)}{\sqrt{var(R_X)var(R_Y)}} \quad (4-13)$$

Where R_X and R_Y are the ranking of X and Y . If all ranks are distinct integers, Spearman correlation can be computed using the following equation:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (4-14)$$

Where n is the number of samples, d_i is the difference between the two ranks of each observation ($d_i = R_{X_i} - R_{Y_i}$) [Myers et al., 2003][Dodge, 2008].

4.4.3. Analysis of variance (ANOVA)

The purpose of Analysis of variance (ANOVA) is to determine whether data from several groups have common mean or not. It tests the hypothesis that all group means are equal versus the hypothesis that at least one group is different from the others. To get valid results of the ANOVA test, these assumptions need to be satisfied: all sample populations are normally distributed and independent of each other and the population standard deviations of the groups are all equal (known as the homoscedasticity property) [Wu et al., 2000][McDonald, 2009][Lowry, 2014].

4.4.4. Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) is a wrapper-based feature selection method. It is an instance of backward feature elimination. First, it starts with a set of all the features that a learning method is trained on it. Then, each feature is ranked based on its importance to the learning method. Finally, one or more features with the lowest rank are eliminated. This procedure is repeated recursively on the remaining features until desired number of features is reached. According to Guyon et al. RFE is much more robust to overfitting than other feature selection methods [Guyon et al., 2002][Kohavi et al., 1997].

It is common to use Support Vector Machines (SVM) as a learning method for RFE. In that case, RFE that is based on SVM is called SVM-RFE. Feature coefficients are used to measure the importance of each feature, and hence rank the features. Features with coefficient closer to zero are less important than other features [Guyon et al., 2002].

4.5. Emotion Classification

The aim of the emotion classification step is to recognize the user emotion using the feature vector that was extracted from EEG signals. To achieve this goal, either a classifier or regressor can be used. A classifier is an algorithm that assign the category of a new observation from a set of categories, while a regressor is an algorithm that can estimate the output of a system as a depended variable. Depending on how emotions are represented and outputs of the system that being built, a classification algorithm or a regression algorithm will be selected. If emotions are represented as multidimension scales and each scale treated as a continuous variable, then a regression algorithm will be used, but if emotions are represented as separated categories or the emotional scales are split into a set of levels, then a classification algorithm would be more suitable. In this work, emotional scales are split into two levels, so that classification algorithms are used. The following subsections present the most popular algorithms that are used to classify user intentions based on EEG signals.

4.5.1. Support Vector Machine (SVM)

A Support Vector Machine (SVM) classifier is one of the classifiers that uses a discriminant hyperplane to identify classes. The great advantage of SVM compared to other linear discrimination classifiers is that the selected hyperplane by SVM is the one that maximizes the margins between the two classes which increases the generalization capabilities [Burges, 1998][Bennett et al., 2000]. Figure 4.6 shows the optimal hyperplane that is selected by SVM.

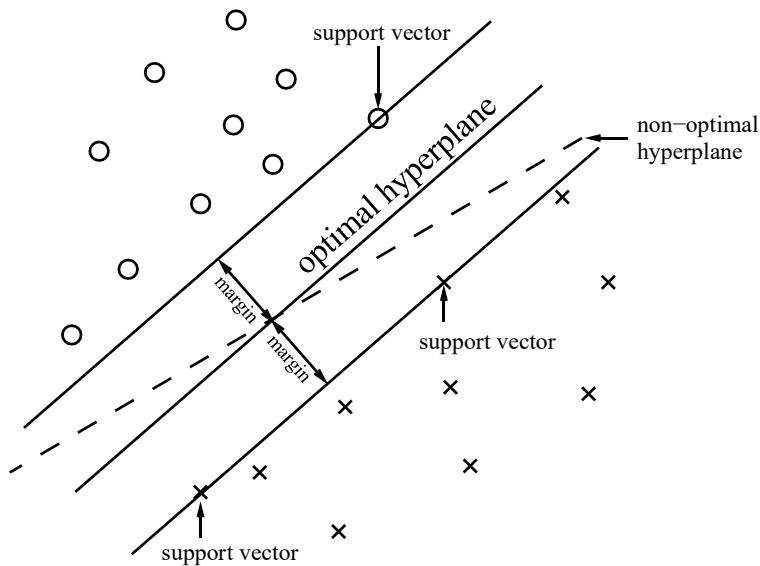


Figure 4.6: The optimal hyperplane obtained by SVM [Lotte et al., 2007]

SVM depends on a kernel function to obtain the discriminant hyperplane. Linear SVM uses a linear kernel to identify a discriminant linear hyperplane. Other kernel functions can be used to obtain a nonlinear discriminant hyperplane, such as the Gaussian or Radial Basis Function (RBF) kernel and the corresponding SVM is called RBF SVM [Burges, 1998][Bennett et al., 2000].

4.5.2. k-Nearest Neighbors (k-NN)

k-Nearest Neighbors (k-NN) is used to build a predictive model by storing the entire dataset, so there is no learning required. To predict the label of a new instance, k-NN searches through the entire dataset and selects the most similar k instances, then assigns

the most dominant class of these k instances to the new instance. The most similar k instances are determined by measuring the distance between the new instance and each instance in stored dataset. Many methods can be used to measure the distance, such as Euclidean distance, Hamming distance and Manhattan distance [Duda et al., 2001].

k -NN is very sensitive to the curse-of-dimensionality. It works well with low-dimensional feature vectors but struggles with high-dimensional feature vectors. For that reason, k -NN is not very common in BCI systems [Friedman, 1997][Borisoff et al., 2004].

4.5.3. Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) also uses a hyperplane to separate the data and identify classes. For a two-classes problem, LDA assumes the data to be linearly separable and has normal distribution with equal covariance matrix for both classes.

LDA obtains the separating hyperplane by seeking the projection that maximizes the distance between the means of the classes and minimizes the inter-classes variance [Fukunaga, 1990]. Figure 4.7 shows a hyperplane obtained by LDA to separate two classes: the “circles” and the “crosses”.

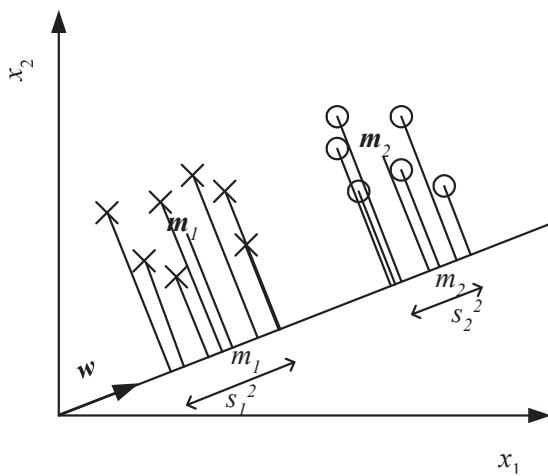


Figure 4.7: The obtained hyperplane by LDA [Alpaydin, 2014]

4.5.4. Multi-Layer Perceptron (MLP)

Multi-Layer Perceptron (MLP) is one of most well-known Artificial Neural Network (ANN) structures. It is a non-linear classifier that has been used in many applications. ANN is inspired by how the brain processes the information. ANN comprises a set of nodes and connections that are modified during the training process. After the training process, ANN can recognize any patterns that are related to the training data. Because of that ability of learning, ANN are widely used in pattern recognition problems [Nicolas-Alonso et al., 2012].

The structure of the neural network in MLP is composed of many layers of neurons. These layers are classified into: an input layer, one or many hidden layers, and an output layer. The outputs of all neurons of one layer is used as input to each neurons of the next layer, and the outputs of the neurons of the output layer are the classes of the input feature vector. MLP can classify any number of classes [Bishop, 1996]. Figure 4.8 show the structure of an MLP neural network.

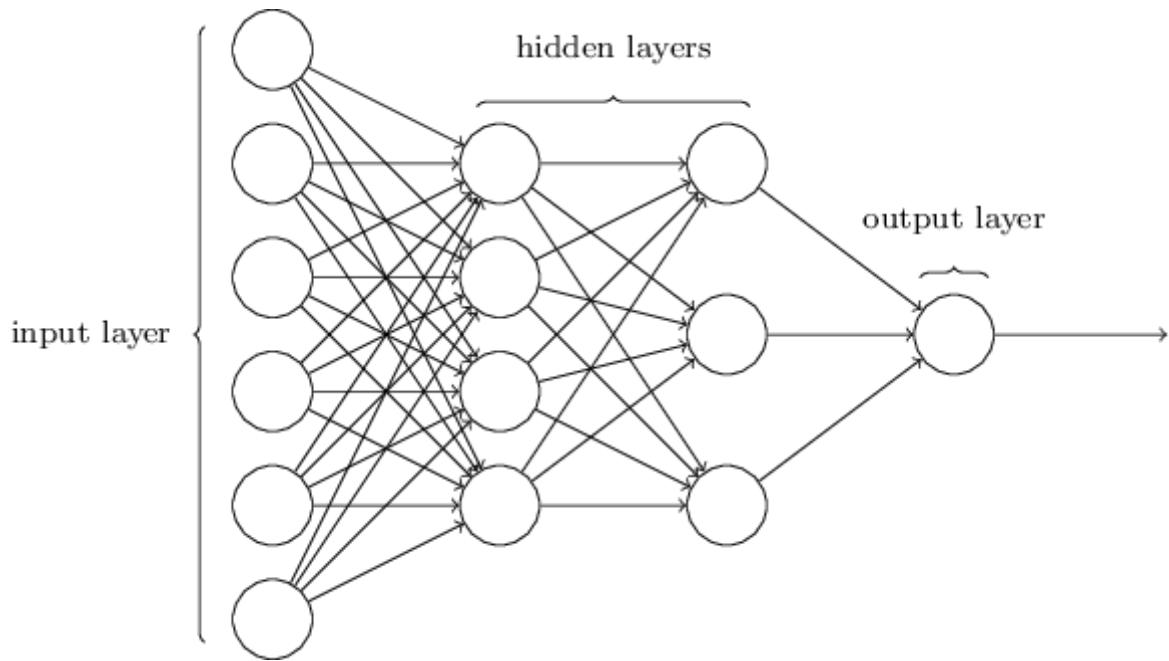


Figure 4.8: A neural network structure of MLP [Cazala, 2017]

4.5.5. Naïve Bayes

Naïve Bayes classifier is a statistical supervised learning method based on Bayes' theorem. Naïve Bayes assigns to a feature vector the class, it belongs to with the highest probability. It is called "Naïve" because of the naïve assumption of independence between every pair of features. The Bayes rule is used to compute a posteriori probability $P(y|x)$ that a feature vector has of belonging to a given class. The class with the highest probability is selected as the class of a feature vector using the Maximum A Posteriori (MAP) rule [Duda et al., 2001][Fukunaga, 1990]. Given a class y and a dependent feature vector x_1 through x_n , the following classification rule can be used to predict class of the feature vector x :

$$\hat{y} = \operatorname{argmax}_y P(y) \prod_{i=1}^n P(x_i | y) \quad (4-15)$$

Gaussian naïve Bayes classifier can be used when dealing with continuous features, where the features values associated with each class are assumed to be distributed according to the Gaussian distribution [John et al., 1995], so the likelihood of the features is:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (4-16)$$

4.5.6. Decision Trees

A decision tree is a hierarchical model for supervised learning, it is composed of internal decision nodes and terminal leaves. Each decision node labels the branches using a test function. By recursively applying the test functions of the decision nodes until reaching the terminal leaves, the local region is split into smaller regions. Given an input into the root of a decision tree, a test is applied and one of the branches is selected depending on the test result, then the input is passed into the next node in the tree to apply another test. This process is recursively repeated until a leaf node is reached, where the output class of the input is written in that leaf node [Alpaydin, 2014].

Figure 4.9 shows an example of a dataset and its corresponding decision tree, the oval nodes are the decision nodes and the rectangles are leaf nodes.

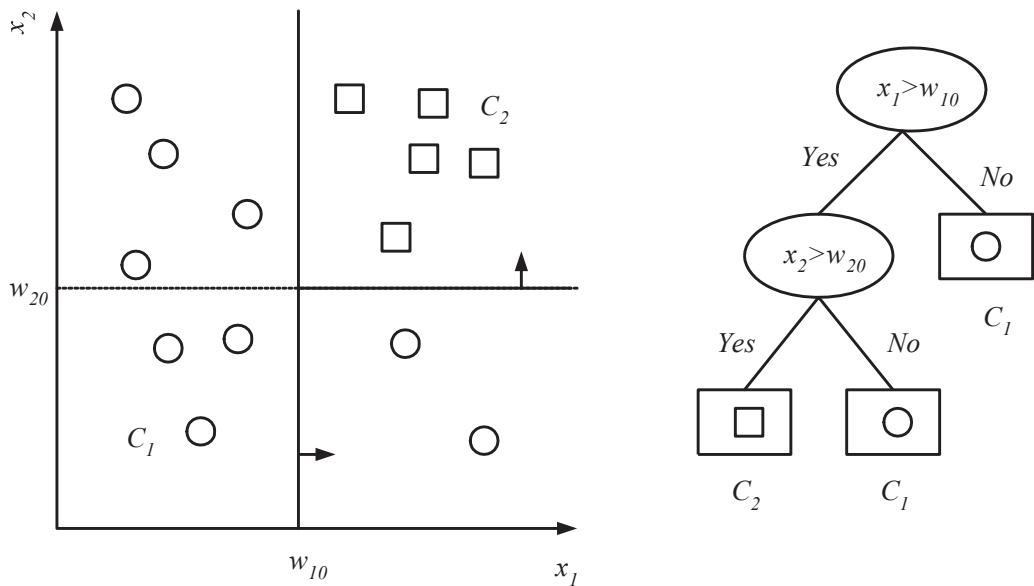


Figure 4.9: A dataset and its corresponding decision tree [Alpaydin, 2014]

4.6. Validation Scheme

One of the major difficulties in machine learning is how to validate the results of a predictive model and measure its generalization capability and how to compare the results of different models to choose the most suitable one for the problem in hand.

To solve this issue, a model must be tested on another dataset different from the one it trained on. So, one should have two datasets, one for training and another for testing, otherwise one should split the available dataset into two parts, the training set and the test set. A stronger technique to measure the generalization capability of a model is to split the available dataset into three parts: the training set to fit the model, the validation set to test the generalization ability of a model fit on the training set while tuning model hyperparameters, and the test set to provide an unbiased evaluation of a final model fit on the training dataset [Alpaydin, 2014].

If the available dataset is small, the test set can be considerably large to the point where different test sets may produce very different results and the model will most likely suffer from overfitting due to the small size of the training set. Unfortunately, all available EEG-based datasets are not large enough to overcome these issues [Kuhn et al., 2013][Nicolas-Alonso et al., 2012].

Cross-validation is a method used to overcome above issue. It uses every part of the available data to train and test the model. This is done by repeatedly using the same data split differently. Cross-validation can be done in different ways, the following subsections present the common ways to perform cross-validation.

4.6.1. K-fold cross-validation scheme

In K-fold cross-validation, the dataset is divided randomly into K equal size parts. One of the K parts is kept out as a test set and the remaining $(K-1)$ parts are combined to form the training set. This process is repeated K times, and the selected K part as a test set is changed in every time. The common value of K is 10, and in that case, it is called 10-fold cross-validation scheme [Alpaydin, 2014].

4.6.2. Leave-p-out cross-validation scheme

In Leave-p-out cross-validation scheme, p observations are used as a test set and the remaining observations as a training set. The process of splitting the dataset into training and test sets is repeated C_p^n times, where n is the number of observations. So, all possible combinations of the observations are presented one time in the test set. If $p > 1$ and n is not small, leave-p-out cross-validation scheme will be very expensive [Shao et al., 1993].

Leave-one-out cross-validation scheme is the typical case of leave-p-out cross-validation, where $p=1$. This validation scheme can also be achieved by K-fold cross-validation scheme if K is equal to the number of observations [Shao et al., 1993][Alpaydin, 2014].

4.7. Measuring Model Performance

Measuring the model performance is an important issue in machine learning research field. A variety of measures are existing. The suitable measure is selected depending on the model target, characteristics of the used dataset and different aspects of the problem that the model solves.

For a regression problem, the performance of the model can be measured by: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Squared Error (MSE), etc. While, a classifier model performance can be measured by: accuracy, precision, recall, etc.

In this work, the emotion recognition problem is treated as a classification problem by splitting the emotional scales. The following subsections discuss the most common measures for a classification problem and how to calculate them.

4.7.1. Confusion Matrix

A confusion matrix is a technique for summarizing the performance of the used model. Calculating a confusion matrix gives a better understanding of what the classification model is getting right and what is wrong? In other words, it shows what makes the model confused when making predictions [Alpaydin, 2014][Manning et al., 2008]. A confusion matrix is shown in Table 4-1.

Table 4-1: Confusion matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Such that:

- TP: number of correctly predicted items as relevant.
- FP: number of incorrectly predicted items as relevant.

- TN: number of correctly predicted items as irrelevant.
- FN: number of incorrectly predicted items as irrelevant.

Confusion matrix can be used to measure other classification performance measurements as shown in each of the following measurement equations.

4.7.2. Accuracy

Accuracy measures the correctness of a model, in other words, it is the ratio of the correct results to the total number of results.

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions} \quad (4-17)$$

Using the confusion matrix, the following equation can be used.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4-18)$$

The most used measure is accuracy, but accuracy alone can be misleading if the number of observations in each class is not equal or close to each other. So, the other measures are used to give a better indication of the model performance [Manning et al., 2008].

4.7.3. Precision

Precision measures the ability of a model to find results that are relevant to the needed information [Manning et al., 2008].

$$Precision = \frac{relevant\ items\ retrieved}{retrieved\ items} = \frac{TP}{TP + FP} \quad (4-19)$$

4.7.4. Recall

Recall measures the ability of a model to find all the relevant information [Manning et al., 2008].

$$Recall = \frac{\text{relevant items retrieved}}{\text{relevant items}} = \frac{TP}{TP + FN} \quad (4-20)$$

4.7.5. F-Score

F-score, or F-measure, is the weighted harmonic mean of the precision and recall [Manning et al., 2008].

$$F = \frac{(\beta^2 + 1) \times Precision \times Recall}{(\beta^2 \times Precision) + Recall} \quad (4-21)$$

F-score trades off precision versus recall. If $\beta < 1$, precision is emphasized, while if $\beta > 1$, recall is emphasized, the default is to balance precision and recall by making $\beta = 1$, In this case, it is commonly written F₁-score (because β is equal 1).

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4-22)$$

F₁-score takes the class balance into account which makes the reported results of F₁-score reliable in case of unbalanced classes. For multi-classes problems, F₁-score is calculated to each class and then the average of all classes is calculated [Koelstra et al., 2012].

4.8. Summary

Emotion recognition using brain signals involves a number of steps. Firstly, EEG signals are preprocessed by applying signal processing techniques to remove noise and artifacts from signals and to decompose EEG signals into their underlying components. Secondly, feature engineering is done on the preprocessed signals, where various

features are extracted which are more suitable for the classification step. To enhance the performance of the classification step, feature reduction is applied to remove unneeded features. Finally, features are classified into emotion classes in the classification step. A validation scheme is used to validate the results of the used model. Different models are compared based on a performance measuring method.

CHAPTER 5

DEAP DATASET AND PREVIOUS WORK

5.1. Introduction

Emotion recognition using brain signals become a hot topic nowadays which attracts many researchers. Machine learning techniques are used intensively in this research area. The first issue that faces researches when applying machine learning techniques is using a large dataset with enough data. This dataset can serve as a benchmark to compare research results with each other [Alpaydin, 2014].

This chapter gives an overview for the most commonly used dataset, DEAP dataset, which also adopted in this thesis, in addition, an overview and a discussion of some previous works that have been performed on that dataset.

This chapter is organized as follows: section 5.2 presents DEAP dataset, the used dataset in this work. Section 5.3 shows the previous work that have been performed on DEAP dataset. A discussion on the previous work is provided in section 5.4. Finally, section 5.5 summarizes this chapter.

5.2. DEAP Dataset

One of the important issues in this research area is collecting enough data for training and testing the model. *Kolestra et al.* provided a relatively large dataset for emotion analysis called DEAP dataset (a Database for Emotion Analysis using Physiological signals) [DEAP, 2015]. In this dataset, the EEG, GSR, blood volume pressure, respiration amplitude, skin temperature, electrocardiogram (ECG), electromyogram (EMG) and electrooculogram (EOG) for 32 participants were recorded as each participant watched 40 one-minute long excerpts of music videos. In addition to these physiological signals, the face videos of 22 participants was recorded by a camera. A total number of 1280 of trials were provided for studying. After each trial, the participant was asked to set his levels of valence, arousal, dominance and liking.

The level of each scale can be set on the range from 1 to 9 [Koelstra et al., 2012]. Figure 5.1 shows a participant in DEAP dataset while he is watching an excerpt of a music video.

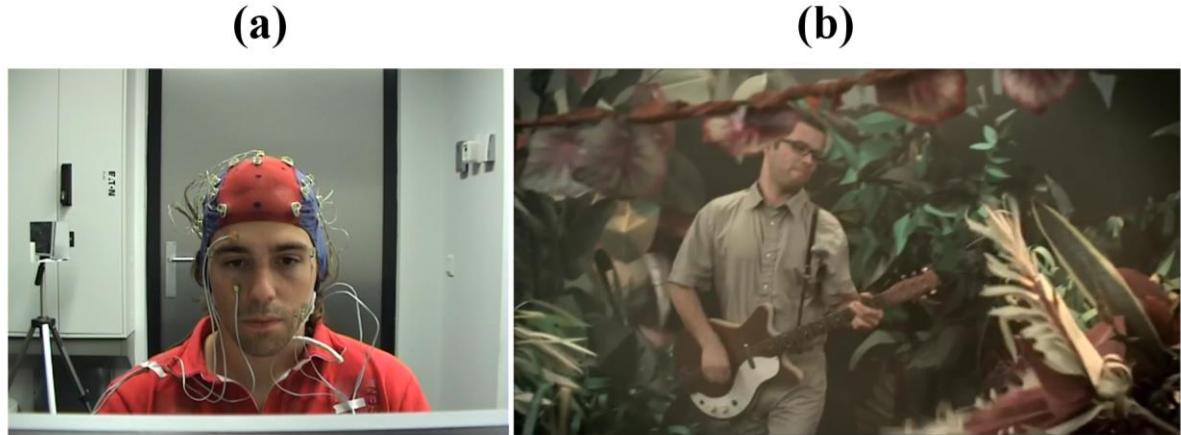


Figure 5.1: A participant watching a music video

EEG devices that were used in DEAP dataset have only 32 electrodes. Four electrodes are on the center, and the other 28 electrodes are on either the left or the right hemispheres, which forming 14 symmetrical pairs of electrodes. Figure 5.2 shows

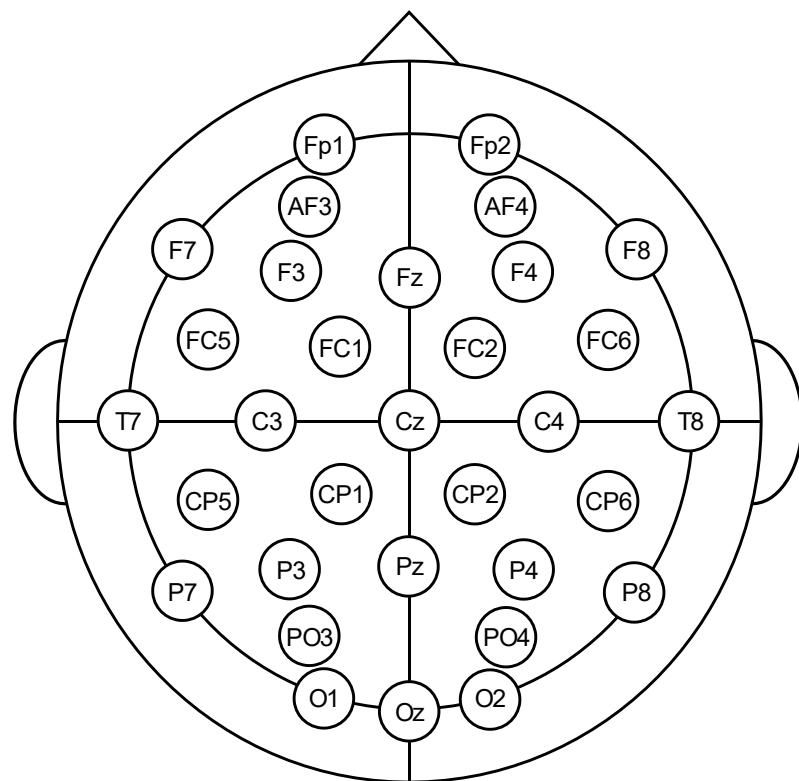


Figure 5.2: 10–20 system with 32 channels

electrodes of the used EEG devices for collecting DEAP dataset, see Appendix A for a sample part of DEAP dataset.

5.3. Previous Work

This subsection presents a set of studies for EEG-based emotion recognition using DEAP dataset.

5.3.1. Koelstra et al. (2012)

Koelstra et al., the collectors of DEAP dataset, investigated the correlation between EEG signal frequencies and participant ratings. They performed a single-trial classification for the scales of valence, arousal and liking using features extracted from the EEG signals, peripheral physiological signals and multimedia content analysis modalities. For EEG modality, features such as the spectral power of theta, slow alpha, alpha, beta and gamma bands for each electrode were extracted using Welch's method. In addition, the spectral power asymmetry between all symmetrical pairs of electrodes in the four bands of alpha, beta, theta and gamma was also extracted. The total number of extracted features of EEG signals was 216 features. Fisher's linear discriminant was used for feature selection phase with a threshold at 0.3. The three scales of valence, arousal and liking were split into two classes (low and high), and a Gaussian naïve Bayes classifier was used to deal with those three different binary classification problems. Due to the existence of unbalanced classes in some scales, F₁-scores in addition to accuracy were used to evaluate the classification performance in a leave-one-out cross-validation scheme for each participant separately. The experiment showed that, the average accuracies were 57.6%, 62.0% and 55.4% for valence, arousal and liking respectively, and the F₁-scores were 56.3%, 58.3% and 50.2%. The results of EEG-based classification were slightly better than random classification [Koelstra et al., 2012].

5.3.2. Matiko et al. (2014)

Matiko et al. presented a fuzzy based classification algorithm of positive and negative emotions. In this work, fuzzy rules are defined based on previous studies showing that, there is a strong correlation of negative and positive emotions with activation of right and left hemispheres of the human brain [Wheeler et al., 1993][Schmidt et al., 2001]. Alpha band was filtered for all symmetrical pairs of electrodes using a finite impulse response filter with 127 filter coefficients, and for each electrode four statistical features were computed, which are mean, standard deviation and mean of the absolute values of the first and second differences. In addition to statistical features, the signal power of the alpha band was also computed. The authors also proposed a new feature that referred as the oscillation feature, which obtained by finding all local maxima and local minima of the signal. After feature extraction step, Fisher's LDA was used to reduce the high dimensions feature space into low dimensions space. The results of feature reduction step show that, the signal power and oscillation features have a higher discrimination ratio than other features. Each fuzzy rule has two inputs: the value of a specific feature for an electrode and the same value of the other corresponding pair electrode, and one output, which is the valence. Three linguistic variables: low, medium and high were used for the input features and five linguistic variables: very low, low, medium, high and very high were used as linguistic terms for the output valence. The average accuracy was 62.62% in a 10-fold cross-validation scheme. The used fuzzy based classifier was compared to Gaussian naïve Bayes and SVM classifiers. The results show that, the fuzzy based algorithm is better than the Gaussian naïve Bayes and SVM classifiers [Matiko et al., 2014].

5.3.3. Jirayucharoensak et al. (2014)

Jirayucharoensak et al. investigated the usage of Deep Learning Network (DLN) for emotion recognition. The input features of the network are the power spectral densities of all electrodes in five frequencies (theta, lower alpha, upper alpha, beta and gamma), and the difference between the power spectral of all symmetrical pairs of electrodes in the same five frequencies. The power spectral density was calculated using

FFT with a Hanning window of size 128 samples. The baseline power was subtracted from all the extracted features and then all features were rescaled into the range 0.1 to 0.9. The total number of extracted features was 230. Principle component analysis (PCA) is used to handle the over-fitting problem of the DLN by selecting the most important features. The 50 most important features were extracted by PCA and were fed into the DLN with 50 hidden nodes in each layer. Covariate Shift Adaptation (CSA) concept is applied to solve the non-stationarity problem in EEG signals. The DLN was implemented with a stacked auto-encoder using hierarchical feature learning approach. The outputs of the network are valence and arousal scales, and each one has been split into three levels. The classification accuracy was measured with a special cross-validation scheme, where in each iteration of the cross-validation loop, all trails of one participant are kept for testing and the other trails are used to train the network. The average accuracy for valence and arousal was 53.42% and 52.03% respectively. The used DLN classifier outperformed the SVM classifier which compared to the DLN classifier [Jirayucharoensak et al., 2014].

5.3.4. Daimi et al. (2014)

Daimi et al. presented an approach for emotion classification using Dual-Tree Complex Wavelet Packet Transform (DT-CWPT) based energy features from EEG signals. First, energy features are extracted by decomposing each channel of EEG using DT-CWPT, and difference between energy features of all symmetrical pairs of electrodes on right and left cortical hemisphere are extracted. Then, feature selection was performed to eliminate weak and redundant features through Singular Value Decomposition (SVD), QR factorization with column pivoting (QRcp) and F-Ratio based feature selection method. Then, the selected features are used to classify emotion using SVM. Finally, F₁-score and accuracy are used to evaluate classification performance in a leave-one-out cross-validation scheme for each participant separately. The average accuracies were 65.3%, 66.9%, 69.1% and 71.2% for valence, arousal, dominance and liking respectively, and the F₁-scores were 55.0%, 57.0%, 55.2% and 50.9% [Daimi et al., 2014].

5.3.5. Chen et al. (2015)

Chen et al. proposed an EEG-based emotion assessment system. They combined ontologies for the management of EEG- and emotion- related information, and data mining techniques to recognize emotions. Previous studies [Bradley et al., 2001][Mesquita, 2003] have pointed out that there are gender differences in the emotional responses, so they used a gender-specific analysis mechanism. The proposed system was designed to give two pairs outputs: low/high valence and low/high arousal for each gender separately. Many EEG features have been investigated for the classification purpose, including: the absolute and relative power of theta, alpha and beta bands; the absolute ratio of beta power to theta power; peak-to-peak amplitude; alpha asymmetry between the channels F3-F4, C3-C4, P3-P4 and O3-O4; entropy features (Shannon entropy, Spectral entropy and Kolmogorov entropy); C₀-complexity; statistical measurements (Skewness, Kurtosis and Variance; and the Hjorth parameters (activity, mobility and complexity). Two statistical methods, Spearman correlation and ANOVA, were exploited to explore the correlation between EEG features and each emotional dimension. After selecting the most correlated features found by statistical methods, classification is performed to predict the emotional states. Four classifiers were investigated for classification step, which are C4.5 decision tree algorithm, SVM, MLP, and k-NN. C4.5 classifier obtained the best classification results in a 10-fold cross-validation scheme. The accuracies of C4.5 were 67.89% for valence and 69.09% for arousal, and the F₁-scores were 67.83% for valence and 68.96% for arousal, all results were averaged across both genders [Chen et al., 2015].

5.3.6. Gao et al. (2015)

Gao et al. used the fact that emotions have different characteristics from subject to another, so the EEG signals for different subjects may vary a lot, based on this fact, they introduced a novel emotion recognition method using Hierarchical Bayesian Network (HBN) that handles general and specific characteristics of emotions simultaneously by considering subject id as input during training, and ignoring it during testing. The used EEG features are power spectrum of five frequency bands for 32 electrodes, power

spectrum asymmetry between 14 pairs of electrodes from four frequency bands, and the ratio of the power in each frequency band to the overall power. PCA was used to reduce the dimension of the features using 85% principle components. A special cross-validation scheme was used, where in each iteration, trials of on video is kept for testing and the other trails was used for training. The obtained accuracies are 58.0% and 58.4% for valence and arousal respectively, and the F₁-scores are 55.2% and 48.8% [Gao et al., 2015].

5.3.7. Li et al. (2016)

Li et al. proposed a preprocessing method and a hybrid deep learning framework for emotion recognition. The preprocessing method encapsulates the multi-channel neurophysiological signals into grid-like frames through wavelet and scalogram transform. Firstly, Continuous Wavelet Transform (CWT) is conducted for each channel signal, before transforming it into a wavelet coefficients based time-scale representation, then transform it into an energy based time-scale representation, namely ‘scalogram’. Secondly, a 2D frame based representation is constructed for each scalogram. A hybrid deep learning model was designed for emotion classification. That model combines both Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) methods. For CNN, two stacked convolutional layers were adopted as the basic structure of the CNN. The convolutional filters in the first convolutional layer was set with the purpose of mining cross-channel correlation information followed by an average pooling layer for aggregating the correlation information. Finally, a Long-Short Term Memory (LSTM) based RNN is used for learning contextual information from the feature sequence that extracted through the front CNN. A dropout approach was adopted to prevent overfitting. Stratified K-folds cross-validation scheme with K=5 is used to evaluate the model performance which compared with Random Forest (RF) and SVM. The achieved accuracies are 72.06% for valence and 74.12% for arousal [Li et al., 2016].

5.3.8. Tripathi et al. (2017)

Tripathi et al. explore two different neural models for emotion classification, which are Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN). The targeted emotional scales for classification were valence and arousal scales. Both scales were split into two parts in one experiment and three in another.

The used DNN is comprising of four neural layers, the input neural layer contains 5000 dimensions followed by layers of 500 neurons and 1000 neurons respectively, the output layer contains two or three nodes depending on the classification classes. All layers were fully connected with softmax activation function, and dropout technique was used before rectifying the outputs to the next layers.

The CNN based model uses two convolutional layers with Tan Hyperbolic activation function, followed by Max Pooling, then dropout is applied before feeding the result to a fully connected neural network which in turn feeds it to the output layer. The output layer is that same as DNN, it contains two or three nodes that apply softmax activation function.

In case of split the emotional scales into two parts, the used DNN based model achieves accuracy of 75.78% and 73.13% for valence and arousal respectively, while the CNN based model achieves accuracy of 81.41% and 73.36% respectively. On the other hand, in case of splitting them into three parts, the DNN based model achieves accuracy of 58.44% and 55.70% while the CNN one achieves 66.79% and 57.58% for valence and arousal respectively [Tripathi et al., 2017].

5.3.9. Chao et al. (2018)

Chao et al. propose an integrated deep learning framework based on improved deep belief networks (DBN) with glia chains. First, many types of EEG features were extracted, including mean, variance, approximate entropy and spectral power for theta, slow-alpha, alpha, beta and gamma bands. The total number of extracted features for each trial is 664 features. A DBN, which composed of many stacked Restricted Boltzmann Machines (RBM) layers, is trained on the extracted features. The used DBN

is improved by glia chains to make it easy for mining inter-channel and inter-frequency correlation information from multichannel EEG signals. The used model is trained and tested via 10-fold cross-validation scheme with a participant-specific style. The average accuracies were 75.92% and 76.83% for valence and arousal respectively, while the F₁-scores were 69.31% and 70.15% [Chao et al., 2018].

Table 5-1 shows a comparison of these studies according to feature types, number of features, feature reduction method, classification type, output emotion scales, accuracy and F₁-scores.

Table 5-1: The summary of the presented studies

Study	Extracted Features	# of feat.	Reduction Method	Class. Method	Output	Acc. (%)	F ₁ . (%)
[Koelstra et al., 2012]	Spectral power	216	Fisher's DA	G. NB	V (2)	57.60	56.30
					A (2)	62.00	58.30
					L (2)	55.40	50.20
[Matiko et al., 2014]	- Spec. Power - Oscillation - ...	84	Fisher's DA	Fuzzy	V (2)	62.62	-
				G. NB	V (2)	59.64	-
				SVM	V (2)	50.62	-
[Jirayucharoensak et al., 2014]	Spectral power	230	- PCA - CSA	DLN	V (3)	53.42	-
					A (3)	52.03	-
			-	DLN	V (3)	47.87	-
					A (3)	45.50	-
			- PCA - CSA	SVM	V (3)	40.26	-
					A (3)	32.46	-
			-	SVM	V (3)	41.12	-
					A (3)	39.02	-
[Daimi et al., 2014]	DT-CWPT based Energy	552	- SVD - QRcp - F-Ratio	SVM	V (2)	65.30	55.00
					A (2)	66.90	57.00
					D (2)	69.10	55.20
					L (2)	71.20	50.90
[Chen et al., 2015]	- Spec. Power - P2P ampl.	580	- Spearman correlation	C4.5	V (2)	67.89	67.83
					A (2)	69.09	68.96

	- Entropy - C ₀ -compl. - Skewness - Kurtosis - Variance - ...		- ANOVA	k-NN	V (2)	66.45	-	
				A (2)	65.00	-		
				MLP	V (2)	64.65	-	
					A (2)	62.51	-	
				SVM	V (2)	59.56	-	
					A (2)	63.39	-	
[Gao et al., 2015]	Spectral power	216	PCA	HBN	V (2)	58.00	55.20	
					A (2)	58.40	48.80	
[Li et al., 2016]		-	-		CNN+RNN	V (2)	72.06	-
					A (2)	74.12	-	
	- Spec. power - P2P ampl. - Entropy - ...	384	-	Random Forest	V (2)	59.65	-	
					A (2)	61.88	-	
				SVM	V (2)	60.46	-	
					A (2)	64.97	-	
[Tripathi et al., 2017]		-	-	DLN	V (2)	75.78	-	
					A (2)	73.13	-	
					V (3)	58.44	-	
					A (3)	55.70	-	
				CNN	V (2)	81.41	-	
					A (2)	73.36	-	
					V (3)	66.79	-	
					A (3)	57.58	-	
				DBN with glia chains	V (2)	75.92	69.31	
					A (2)	76.83	70.15	
[Chao et al., 2018]	- Entropy - Spec. power - ...	664	-	DBN with glia chains	V (2)	75.92	69.31	

* V: valence, A: arousal, D: dominance, L: liking, Acc: Accuracy, F₁: F₁-score, G. NB: Gaussian

Naïve Bayes.

5.4. Discussion of Previous Works

Based on the literature review, the following remarks can be elicited:

- (1) All the presented studies split emotional scales into 2 levels except *Jirayucharoensak et al.* split them into 3 levels, which is mostly the reason why it has

the worst accuracy among other studies, which leads to that, it is unfair to judge DLN method without considering the number of levels.

(2) The method used by *Matiko et al.* to apply different classifiers is inappropriate. The common method is using the most important features from all pairs of electrodes as inputs to the classifier, but they used the features of each pair of electrodes separately and after calculating the accuracy of each pair, the mean of accuracies is calculated and used.

(3) Both *Koelstra et al.* and *Matiko et al.* have applied the Gaussian naïve Bayes classifier with the same feature selection method, but each of them used different extracted features. *Matiko et al.* study achieved a slightly better accuracy than the study by *Koelstra et al.* There is not enough information to differentiate between these two methods because they used a different validation scheme.

(4) Both *Koelstra et al.* and *Gao et al.* used the spectral power features, but *Koelstra et al.* used Gaussian naïve Bayes classifier with Fisher's LDA for feature selection, while *Gao et al.* used HBN classifier with PCA for feature selection. No significant difference between both studies in case of valence, but *Koelstra et al.* outperformed *Gao et al.* in case of arousal.

(5) Both *Matiko et al.* and *Daimi et al.* have adopted SVM as a classifier, but with different extracted features. *Daimi et al.* outperformed *Matiko et al.* with 15% difference in accuracy. Although they used a different validation scheme, the difference in accuracy is higher to be affected only by the validation scheme. The main reason for that difference is the features that have been used by *Daimi et al.*

(6) Both *Matiko et al.* and *Chen et al.* have applied SVM with 10-fold validation scheme, but each one used different feature set. *Chen et al.* outperformed *Matiko et al.* with 9% higher accuracy. This difference is due to the features that have been used in *Chen et al.*

(7) The reason for the good results that achieved by *Daimi et al.* is the method that has been used for feature extraction, DT-CWPT, compared to the use of predefined

frequency bands of theta, alpha, beta and gamma. It is worth mentioning that, most of the presented studies skip using delta frequency band, except *Gao et al.* Table 5-2 summarizes the used frequency bands in each study.

(8) *Chen et al.* have achieved the best results among other presented studies in terms of valence and arousal. The reasons for that high accuracy are the variant types of extracted features.

Table 5-2: Summary of the used frequency bands in the presented studies

Study	Delta	Theta	Alpha	Beta	Gamma
[Koelstra et al., 2012]	-	Yes	Yes	Yes	Yes
[Matiko et al., 2014]	-	-	Yes	-	-
[Jirayucharoenak et al., 2014]	-	Yes	Yes	Yes	Yes
[Daimi et al., 2014]	Decomposed by DT-CWPT				
[Chen et al., 2015]	-	Yes	Yes	Yes	-
[Gao et al., 2015]	Yes	Yes	Yes	Yes	Yes
[Li et al., 2016]	Directly use raw EEG signals				
[Tripathi et al., 2017]	Directly use raw EEG signals				
[Chao et al., 2018]	-	Yes	Yes	Yes	Yes

(9) *Koelstra et al.* and *Daimi et al.* divided the dataset into 32 parts based on participants, which means 32 models were built, one for each participant. So, a training step will be required for any new users that will use their approach. *Chen et al.* also divided the dataset based on gender, so a new user for their approach must specify his/her gender. On the other hand, other studies built one model for all the dataset, because no division was done on the dataset. Therefore, a new user for their systems can use it immediately. Table 5-3 compares the previous studies in terms of dataset partitioning and validation scheme.

Table 5-3: Comparing the previous studies in terms of dataset partitioning and validation scheme

Study	Dataset Partitioning	Validation Scheme	Notes
[Koelstra et al., 2012]	32 parts based on participant	Leave-one-out	A training step is need for new users
[Matiko et al., 2014]	-	10-Fold	-
[Jirayucharoensak et al., 2014]	-	Leave-one-person-out	-
[Daimi et al., 2014]	32 parts based on participant	Leave-one-out	A training step is need for new users
[Chen et al., 2015]	Two parts based on gender	10-Fold	New users need reconfiguration
[Gao et al., 2015]	-	Leave-one-video-out	Only 38 videos are used
[Li et al., 2016]	-	5-Fold	-
[Tripathi et al., 2017]	-	Leave-one-person-out	-
[Chao et al., 2018]	32 parts based on participant	10-Fold	A training step is need for new users

(10) The used features can be ranked by their effect on the accuracy as follows (see Figure 5.3):

1. Energy features using DT-CWPT.
2. Entropy features, C₀-complexity, Hjorth parameters, Statistical features, Peak-to-peak amplitude, absolute and relative power of theta, alpha and beta bands, absolute ratio of beta power to theta power and alpha asymmetry.
3. Oscillation feature, signal power, mean and standard deviation.
4. Spectral power.

(11) For the sake of ranking classifiers, the effects of the used features need to be removed. SVM is the most used classifier in the literature, so other classifiers were compared to it. The accuracies of all classifiers are computed as a relative ratio to the

accuracy of SVM in terms of valence (see Figure 5.4). The used classifiers can be ranked by its effect on the accuracy as follows:

1. DLN.
2. Fuzzy based classification algorithm.
3. HBN.
4. Gaussian naïve Bayes.
5. C4.5 decision tree algorithm.
6. k-NN.
7. MLP.
8. SVM.

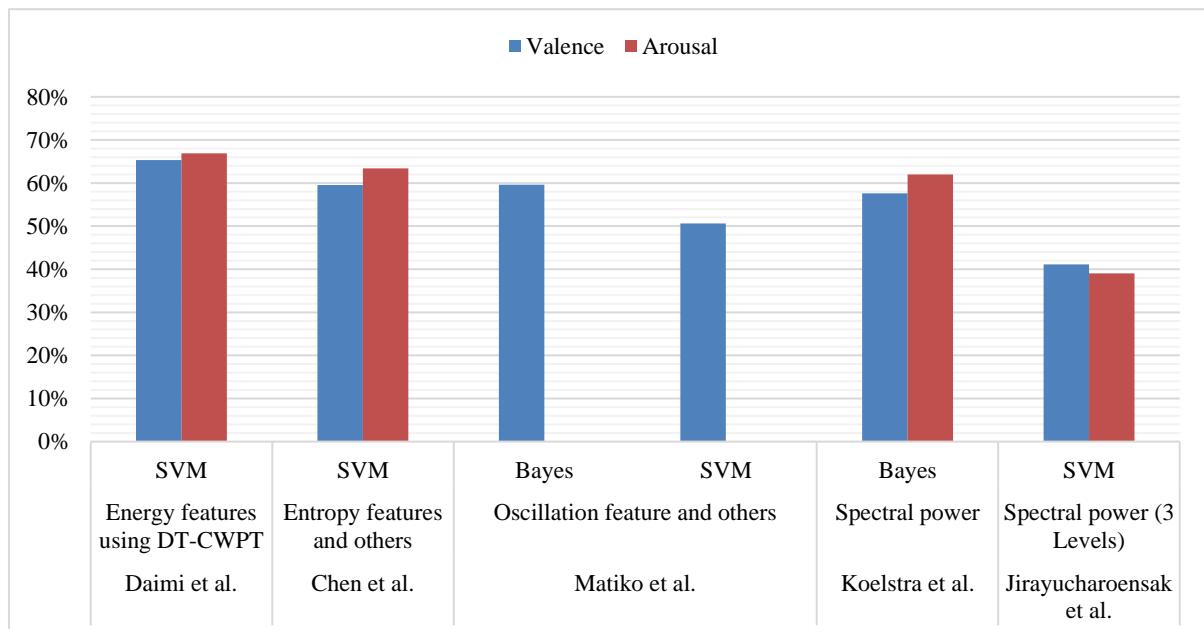


Figure 5.3: Accuracies of valence and arousal for different features

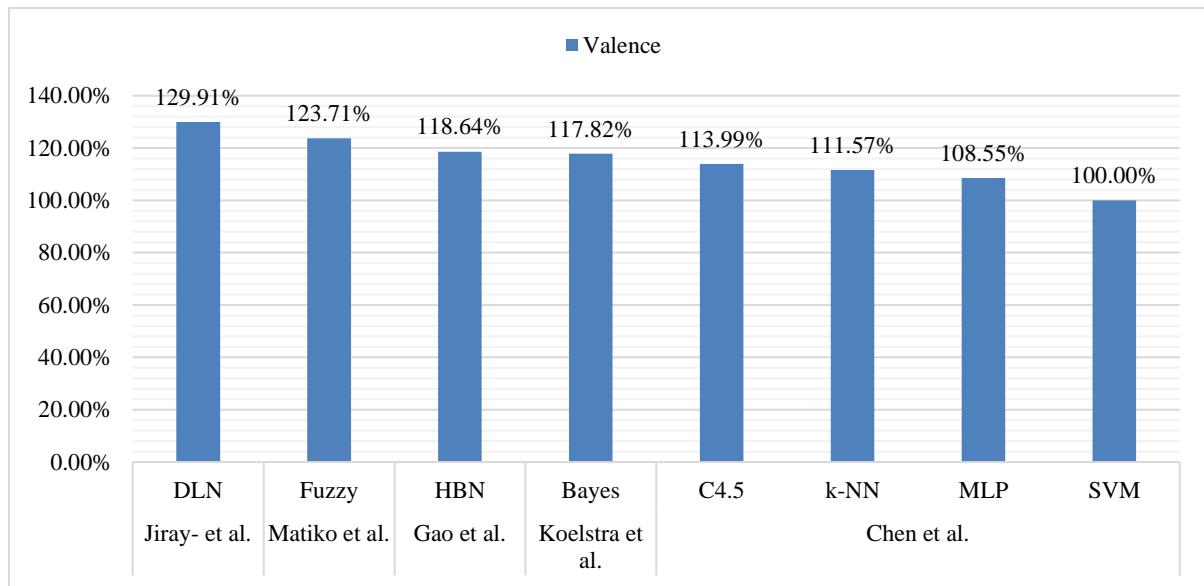


Figure 5.4: Accuracy ratios of different classifiers relative to SVM

5.5. Summary

In this chapter, the used dataset in this thesis, DEAP dataset, is illustrated. Also, previous works for emotion recognition using EEG signals on this dataset are showed and discussed.

DEAP dataset provides a relatively large dataset for emotion recognition using EEG and other physiological signals. It considered as a benchmark dataset where researchers can apply their work on it and compare the results with each other.

The presented previous work showed how EEG signals are processed and how machine learning techniques are applied in different steps of emotion recognition process.

CHAPTER 6

PROPOSED MODEL FOR EMOTION RECOGNITION USING BRAIN SIGNALS

6.1. Introduction

In this chapter, the proposed model for brain signals based emotion classification is presented. The proposed model consists of four steps, which are EEG signals preprocessing, feature extraction, feature reduction, and emotion classification.

This chapter is organized as follows: section 6.2 presents the steps of the proposed model. section 6.3 discusses the preprocessing step of EEG signals. Section 6.4 and section 6.5 discuss the feature extraction and feature reduction steps respectively. Emotions classification is discussed in section 6.6. Section 6.7 presents the used tools for implementing the proposed model. finally, section 6.8 summarizes this chapter.

6.2. The Proposed Model

The proposed model for EEG-based emotion recognition adopts extracting the five major rhythms of delta, theta, alpha, beta and gamma using FIR digital filters, and extracting spectral power, oscillation and Shannon entropy features from these rhythms. By testing many classifiers, such as SVM, K-NN, MLP, LDA, Naïve Bayes and Decision Trees, and comparing them in terms of accuracy and F₁-score, it turns out that, SVM as a classifier scores more accurate results if it used with RFE as a feature reduction method. Therefore, the proposed model adopts RFE for feature reduction and SVM for emotion classification. Figure 6.1 illustrates the steps of the proposed model.

6.3. EEG Signals Preprocessing

The first step in the proposed model is the preprocessing step. In this step, EEG signals are prepared for the next step, the feature extraction. First, noise reduction and

artifacts removing is done. Then, EEG signals are separated into the underlying frequency band components.

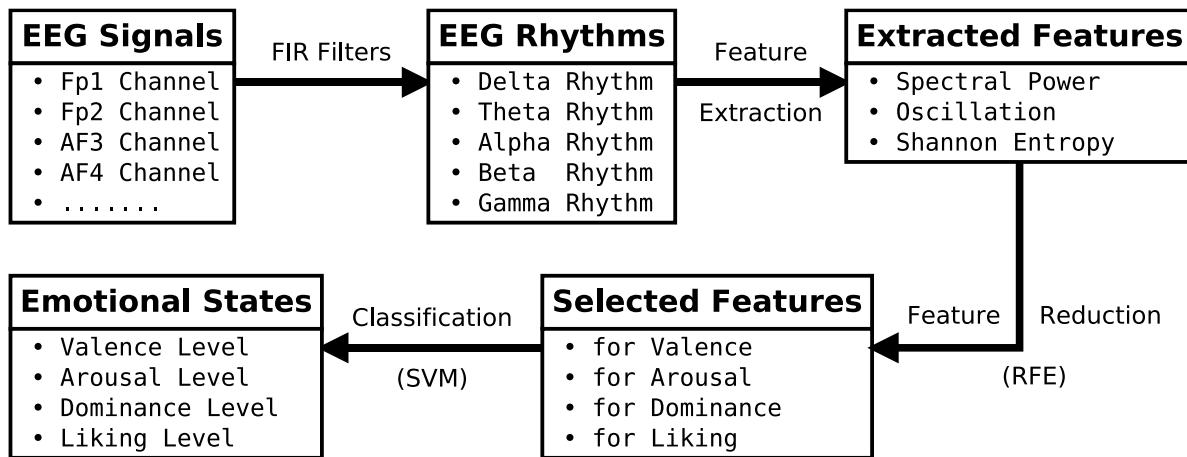


Figure 6.1: Proposed Model for EEG-based Emotion Recognition

DEAP dataset collectors provided a preprocessed version of physiological signals in addition to the raw signals' version. EEG signals were down-sampled to 128 Hz, EOG artifacts were removed and noise was reduced by applying a bandpass frequency filter (4–45 Hz). Then, data was averaged to a common reference. The preprocessed version is well-suited to quickly test machine learning techniques without the hassle of preprocessing [DEAP, 2015], so in this study the preprocessed version of the dataset is used.

Five FIR filters were designed to separate rhythms of delta (0-4 Hz), theta (4-8 Hz), alpha (8-12 Hz), beta (12-30 Hz), and gamma (30+ Hz). All the designed FIR filters are using a hamming window with a filter order of 127 points. Figure 6.2, Figure 6.3, Figure 6.4, Figure 6.5 and Figure 6.6 show impulse responses (a) and frequency responses (b) of the designed digital filters for extracting delta, theta, alpha, beta, and gamma waves respectively.

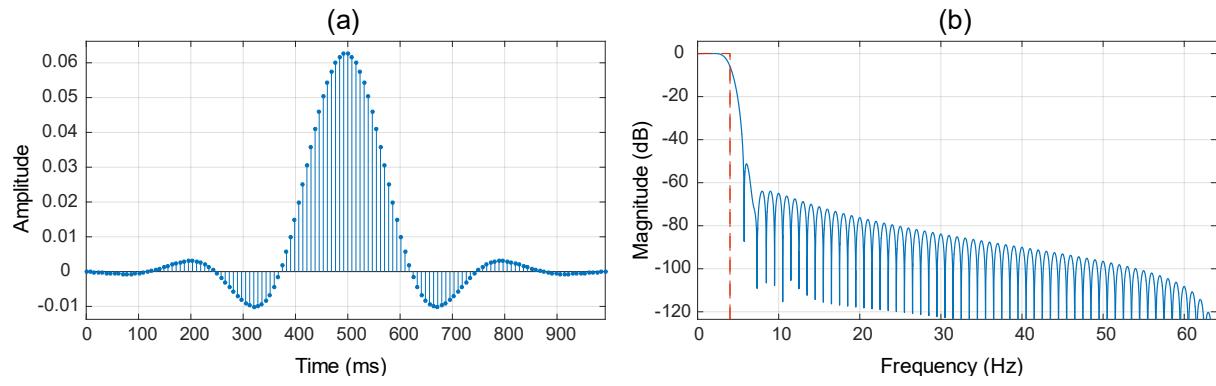


Figure 6.2: Digital filter for Delta waves

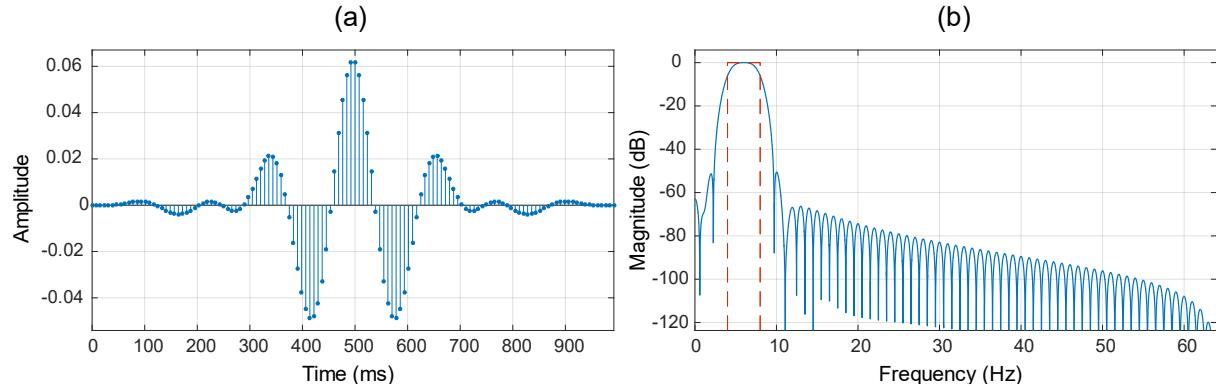


Figure 6.3: Digital filter for Theta waves

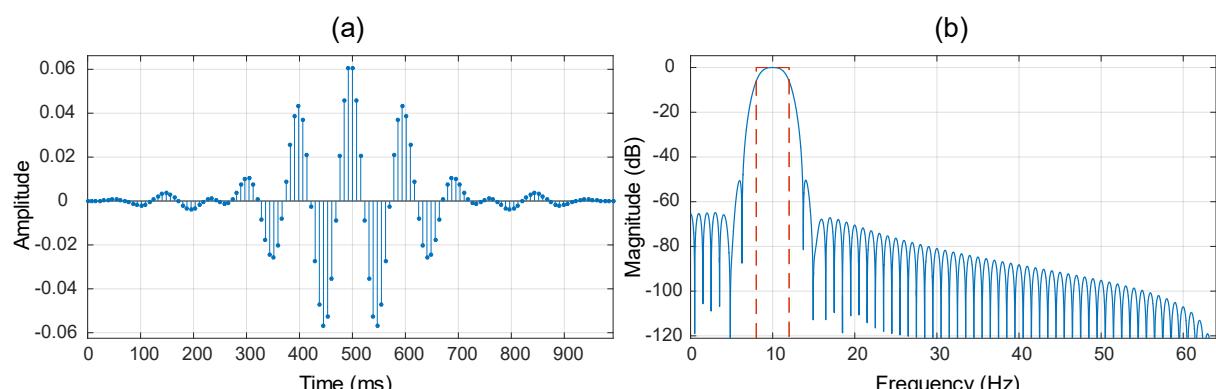


Figure 6.4: Digital filter for Alpha waves

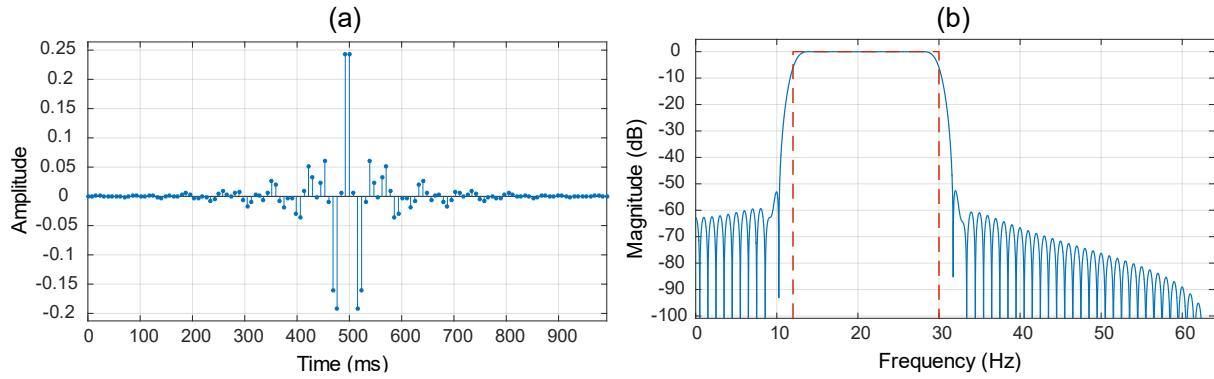


Figure 6.5: Digital filter for Beta waves

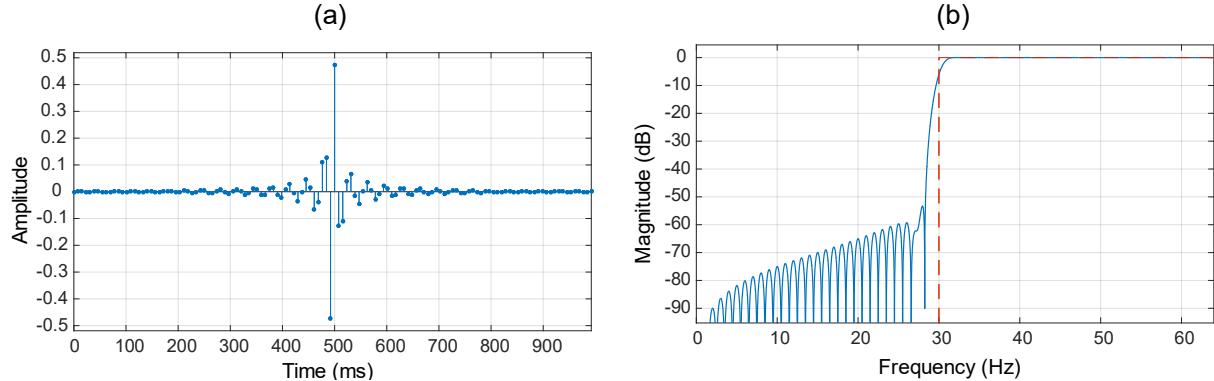


Figure 6.6: Digital filter for Gamma waves

6.4. Feature Extraction

In this step, the preprocessed EEG signals are mapped into feature vectors. The following subsection presents the extracted features to be used in the proposed model.

6.4.1. Spectral Power features

The Welch's method is used to extract the spectral power features for each frequency band of the five major frequency bands. Usually, the difference among the spectral powers of different frequency bands is small, so the logarithms of the spectral power features are calculated to increase that difference. Next, the differences between all symmetrical pairs of electrodes are calculated. In addition to the previous spectral

power based features, the ratio of the spectral power of beta-to-theta, beta-to-alpha and alpha-to-theta bands are also calculated. The following list summarizes all the extracted features based on the spectral power.

1. Logarithms of the spectral power (5 rhythms of 32 electrodes).
2. Spectral power asymmetry (5 rhythms of 14 pairs of electrodes).
3. Ratio of beta-to-theta spectral power (32 electrodes).
4. Ratio of beta-to-alpha spectral power (32 electrodes).
5. Ratio of alpha-to-theta spectral power (32 electrodes).

6.4.2. Oscillation features

The oscillation features by *Matiko et al.* are calculated for the five major rhythms of EEG signals. According to *Matiko et al.*, oscillation and power features have higher discrimination ratios than other statistical features. Algorithm 6.1 shows a pseudocode for the used algorithm to calculate the oscillation feature for the signals of only one frequency band. The same algorithm is repeated with each frequency band of each electrode [Matiko et al., 2014].

Algorithm 6.1: Calculating the oscillation feature [Matiko et al., 2014]

Input: X: signals of one frequency band of one electrode

Output: O: the oscillation value of the input signal

```

01: N ← length_of(X)
02: Lmin ← 0 // counter of local minima
03: Lmax ← 0 // counter of local maxima
04: t ← 1
05: while t ≤ N-2 do
06:   if X(t) > X(t+1) and X(t+2) > X(t+1) then
07:     Lmin ← Lmin + 1
08:   end if
09:   if X(t) < X(t+1) and X(t+2) < X(t+1) then

```

```
10:      Lmax ← Lmax + 1
11:  end if
12:  t ← t + 1
13: end while
14: O ← N / (Lmax+Lmin)
```

In addition to the oscillation feature of each frequency band, the difference between all symmetrical pairs of electrodes is calculated for all the five frequency bands.

Algorithm 6.2: Calculating the Shannon entropy feature based on [Shannon, 1948]

Input: X: signals of one frequency band of one electrode

Output: SE: The Shannon entropy of the input signal

```
01: MIN ← minimum(X)
02: MAX ← maximum(X)
03: W ← calc_proper_bin_width(X)
04: edge ← minimum (W * floor (MIN / W), MIN)
05: t ← 1
    // build a histogram for X
06: while edge ≤ MAX do
07:   px(t) ← 0
      // count number of points in current bin
08:   for each xi in X do
09:     if xi ≥ edge and xi < edge + W then
10:       px(t) ← px(t) + 1
11:     end if
12:   end for
13:   edge ← edge + W // next bin
14:   t ← t + 1
15: end while
16: SE ← 0
17: T ← sum(px)
```

```
18: for each x in px do
19:   if x ≠ 0 then
20:     p ← x / T // get the probability of x
21:     k ← p * ln(p)
22:   else
23:     k ← 0
24:   end if
25:   SE ← SE + k
26: end for
27: SE ← -SE
```

6.4.3. Shannon entropy

Shannon entropy is calculated for each frequency band of each electrode. As done before with spectral power and oscillation features, the differences of Shannon entropy between all symmetrical pairs of electrodes were also calculated. Algorithm 6.2 shows a pseudocode for the algorithm that has been used to calculate the Shannon entropy feature for the signals of one frequency band of one electrode.

6.5. Feature Reduction

The total number of extracted features is 786. This huge number of features will affect both computational costs and the performance of the classification step [Guyon et al., 2003][Jain et al., 2000]. So, it's very important to reduce the extracted features with a feature reduction method.

Feature reduction can be done by many methods. Dimension reduction methods and filter-based feature selection methods were tested in the previous work. Wrapper-based feature selection methods, on the other hand, haven't been tested on DEAP dataset.

Since RFE is a wrapper-based feature selection method which more robust to overfitting than other feature selection methods as previously discussed. Motivated by the advantages of RFE and also it hasn't been tested on the used dataset, RFE was adopted in this work to perform the feature reduction on the extracted features.

SVM with a linear kernel [Chang et al., 2011] was used as a learning method for RFE where features are ranked based on their weights in the coefficient matrix. Features with a coefficient value close to zero are less important for SVM than features with coefficient value far from zero, in both positive and negative directions. A ranking criterion is used to rank features by powering their coefficient value by 2. So, features with a coefficient value far from zero will have a greater value than the close to zero features. Only one feature is eliminated in each iteration of RFE until 25% of the total number of features remains. A pseudocode of RFE algorithm is shown in Algorithm 6.3.

Algorithm 6.3: SVM-RFE algorithm [Guyon et al., 2002]

Inputs: X_0 : the extracted feature vectors

Y : class labels

Outputs: s : indices of the selected features

```

01: TRGT ← length_of( $X_0$ ) * 0.25 // # of target features
02: STEP ← 1 // # of eliminated features in each step
03:  $s \leftarrow [1, 2, 3, \dots \text{length\_of}(X_0)]$ 
04: while length-of( $s$ ) ≤ TRGT do
05:    $X \leftarrow X_0(:, s)$  // select only good features
06:   cls ← train_SVM( $X, Y$ ) // train the learning method
07:   W ← coefficients(cls) // get weights of features
08:   for each  $w_i$  in  $W$  do
09:      $c(i) \leftarrow (w_i)^2$  // calculate the ranking criteria
10:   end for
11:    $f \leftarrow \text{argmin}(c)$  // feature with smallest rank
12:    $s \leftarrow \text{remove}(s, f)$  // remove the smallest ranked feature
13: end while
```

6.6. Emotions Classification

Using SVM as a learning method for RFE (called SVM-RFE), the performance of SVM is expected to be enhanced. In addition to SVM, other popular classifiers have been compared. The evaluated classifiers in this study are: SVM with Radial Basis Function (RBF) kernel, SVM with a linear kernel, LDA, k-NN, Gaussian naïve Bayes, MLP and Decision Tree.

6.7. Tools

Signal processing toolbox of MATLAB (R2015b) is used to design the FIR filters that have been used in the preprocessing step. Feature extraction is also implemented using MATLAB, see Appendix B.

Python 3 is used for feature reduction and classification steps. Scikit-learn toolkit, that is available for Python, have a lot of pre-implemented algorithms for feature reduction, classification, regression, and validation schemes [Pedregosa et al., 2011], see Appendix C.

6.8. Summary

This chapter presents the proposed model for emotions recognition using brain signals. It explains the adopted methods in each step of EEG-based emotion recognition process. Those steps are: signals preprocessing, feature extraction, feature reduction, and emotion classification. Five FIR digital filters were designed to extract the five major frequency bands in the preprocessing step. In the feature extraction step, spectral power, oscillation and Shannon entropy features were extracted from the preprocessed signals. RFE was adopted for feature reduction and SVM was adopted for emotion classification. The proposed model was implemented by MATLAB and Python.

CHAPTER 7

EXPERIMENTAL RESULTS AND DISCUSSION

7.1. Introduction

To ensure the significance of the proposed model, it should be tested in a real-life experiment. The data that were given by DEAP dataset must go through several steps, which are signals preprocessing, feature extraction, feature reduction and emotions classification. The higher accuracy that showed by these experiments determine the enhancement in emotion recognition.

In this chapter, the results of evaluating the proposed model are shown and compared with the previous work that have been done on the same dataset, described in Chapter 4. Examples of the effect of applying digital filters to separate EEG signals into their major frequency bands are illustrated. Experiments that have been done to validate the results of the preprocessing and feature extraction steps are explained, and their results are reported. The effect of separating the dataset into parts is also discussed.

This chapter is organized as the following: Section 7.2 shows examples of applying digital filters of the preprocessing step while section 7.3 summarizes the results of feature extraction step. A study of the extracted feature correlation with emotional scales is presented in section 7.4. Section 7.5 shows the result of verifying the previous steps by reproducing one of the previous works. Section 7.6 discusses an experiment to check the worthy of partitioning the dataset based on gender of participant. Evaluating the proposed model for emotion recognition is presented in section 7.7. Section 7.8 provide a discussion of the experimental results. Finally, section 7.9 provides a summary for this chapter.

7.2. EEG signals preprocessing

The preprocessed signals of DEAP dataset are provided in 32 separated files. Each file contains the physiological signals of one participant for all the 40 excerpts of music

videos (trials). For each trial, 40 channels exist. The first 32 channels are the EEG electrodes. Each channel contains 63 seconds. The first three seconds are the removed pre-trial baseline and the remaining 60 seconds are the trial signals [DEAP, 2015]. Table 7-1 shows an example of EEG data as read from a dataset file for one trial only.

Table 7-1: Part of EEG data of one trial

Time	Channels								
	Fp1	AF3	F3	F7	FC5	FC1	C3	T7	CP5
0.000 s	0.948	0.125	-2.217	1.006	5.096	1.207	3.867	1.862	3.689
0.008 s	1.653	1.390	2.292	1.298	5.001	0.599	3.280	7.191	4.701
0.016 s	3.014	1.835	2.746	2.368	4.176	0.869	0.656	5.907	1.713
0.023 s	1.495	-1.111	-2.365	-0.232	0.354	1.974	-3.077	-3.654	-4.313
0.031 s	-1.265	-2.591	-2.310	-1.662	-4.154	-0.091	-2.858	-1.046	-4.378
0.039 s	-1.970	-1.844	-0.668	-0.015	-3.981	-3.140	-0.102	3.618	-1.608
0.047 s	-2.169	-0.740	-0.837	0.737	-0.554	-1.927	2.197	-0.448	1.873
0.055 s	-0.215	2.396	3.655	1.422	1.403	1.206	3.133	-0.260	5.128

The three baseline seconds are skipped and the remaining 60 seconds of the 32 channels of EEG signals are used in our study. Next, EEG signals are separated to their major sub-frequency band components using Finite Impulse Response (FIR) filters. Figure 7.1 shows an example of EEG signals of one electrode. Figure 7.2 shows (a) the time domain of one second and (b) the single-sided spectrum of the frequency domain of the same electrode before applying the digital filters, while the results of applying the five digital filters of delta, theta, alpha, beta and gamma are shown in Figure 7.3, Figure 7.4, Figure 7.5, Figure 7.6 and Figure 7.7 respectively.

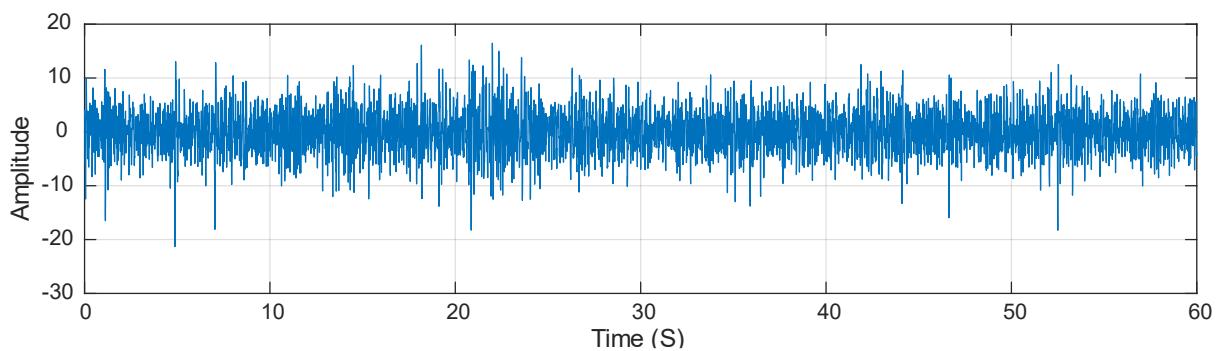


Figure 7.1: EEG signals of one electrode

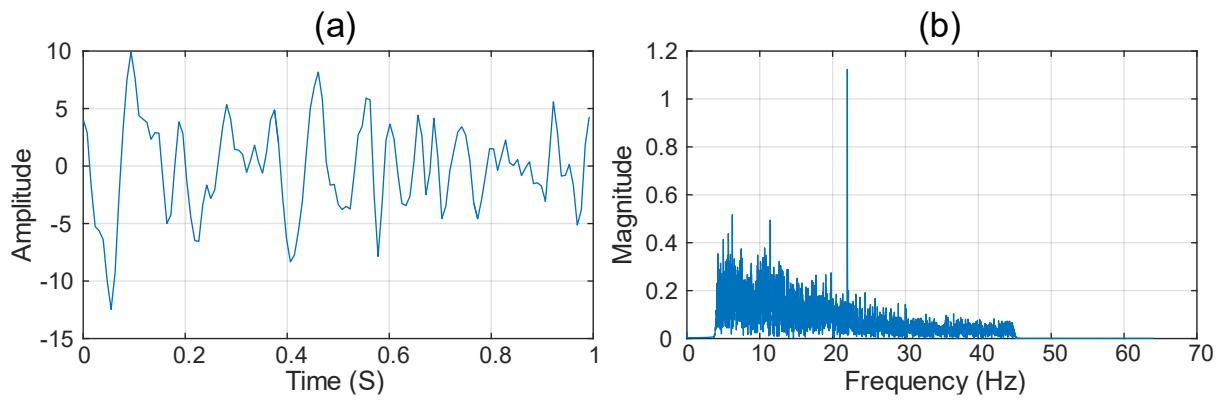


Figure 7.2: Time domain & Frequency domain of EEG signal

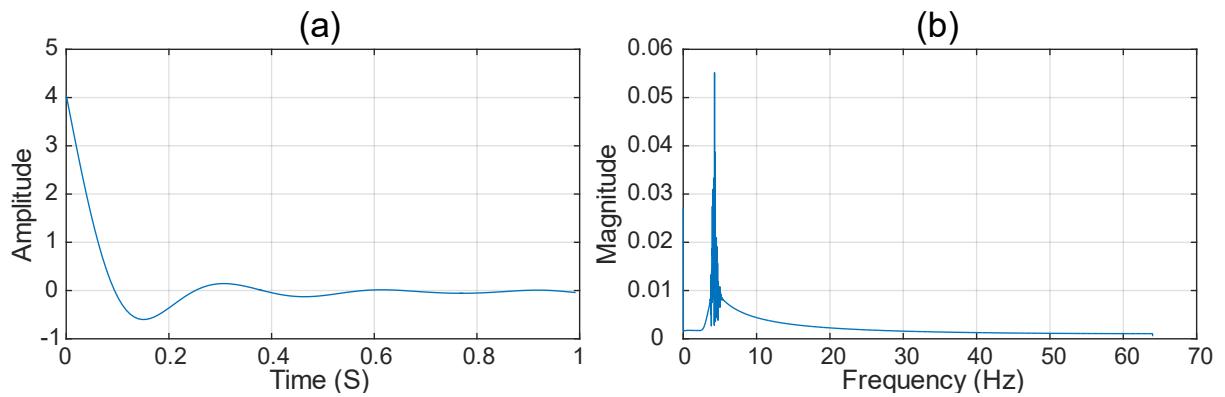


Figure 7.3: Time domain & Frequency domain of Delta signal

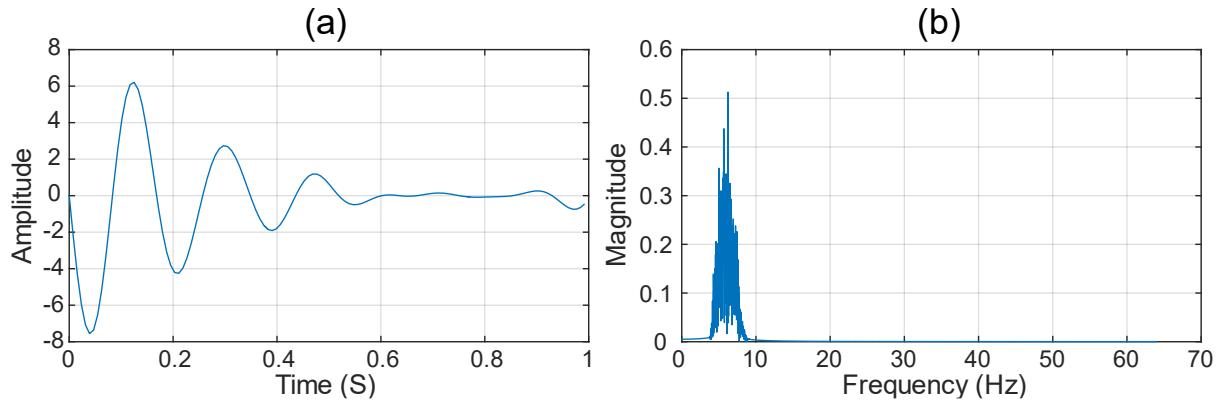


Figure 7.4: Time domain & Frequency domain of Theta signal

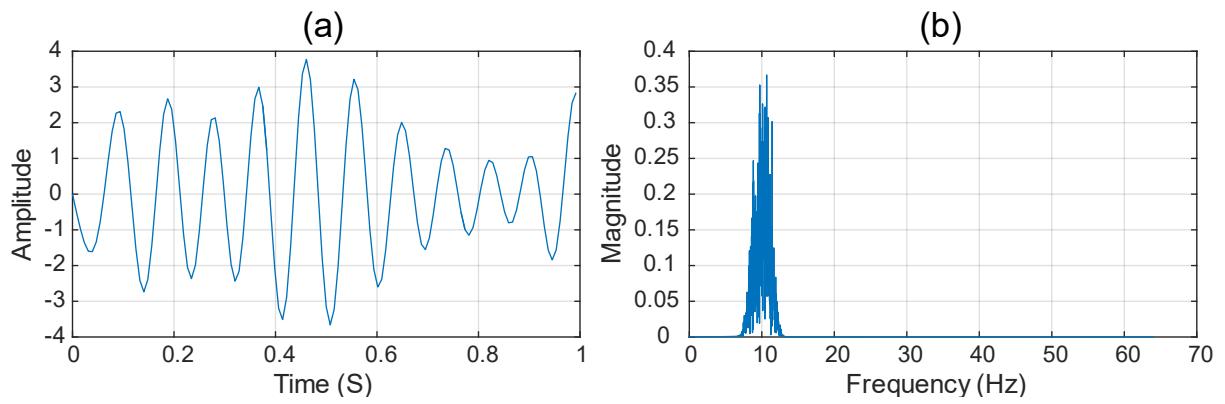


Figure 7.5: Time domain & Frequency domain of Alpha signal

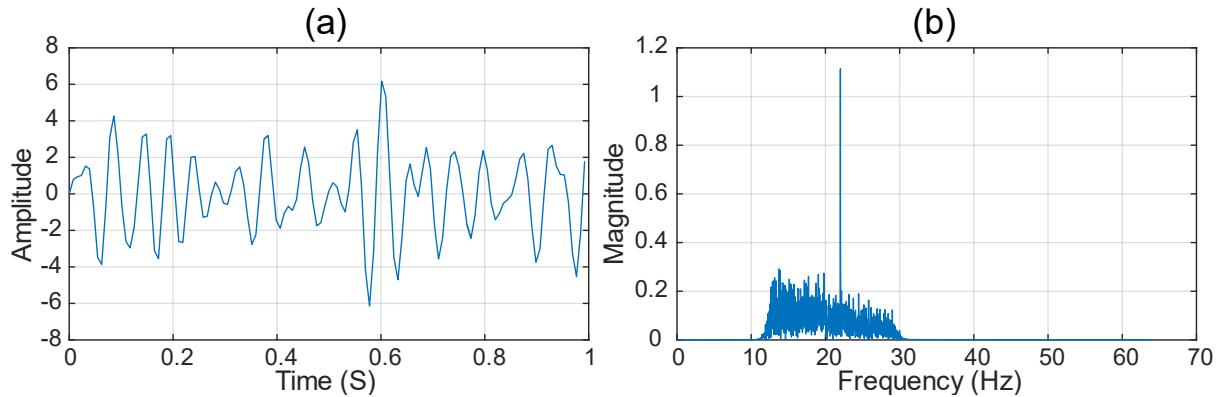


Figure 7.6: Time domain & Frequency domain of Beta signal

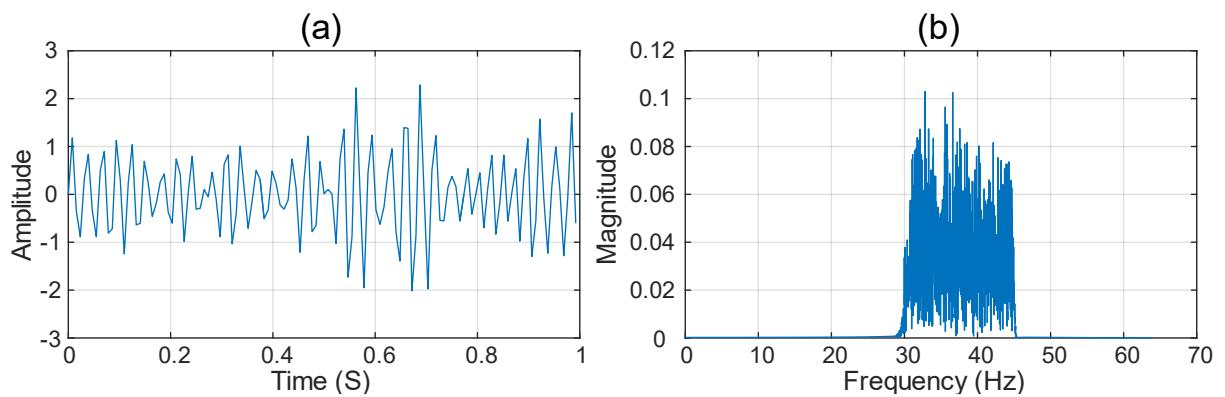


Figure 7.7: Time domain & Frequency domain of Gamma signal

7.3. Feature Extraction

The total number of extracted features was 786. Table 7-2 summarizes the extracted features from EEG signals and shows the total number of features for each kind of features.

Table 7-2: Summary of the extracted features

Feature name	# of electrodes	# of rhythms	Sum
Spectral power	32	5	160
Spectral power asymmetry	14 (pairs)	5	70
Ratio of beta-to-theta power	32	1	32
Ratio of beta-to-alpha power	32	1	32
Ratio of alpha-to-theta power	32	1	32
Oscillation	32	5	160
Oscillation asymmetry	14 (pairs)	5	70
Shannon entropy	32	5	160
Shannon entropy asymmetry	14 (pairs)	5	70
Total			786

7.4. Features correlation with emotional scales

Measuring the correlation of extracted features with the associated labels is a common way to investigate the significance of these features. Spearman correlation is adopted to measure the correlation of the extracted features with participants' ratings because it is robust to outliers and does not assume a linear relation. For each feature and scale, Spearman correlation between the feature values of all participants and their ratings of the scale is calculated, and the p-value is reported. The p-value is used to test the hypothesis that the two sets of data are uncorrelated. Table 7-3 shows the significantly correlated channels ($p < 0.01$, * = $p < 0.001$) of each rhythm and feature with the valence scale. Table 7-4, Table 7-5 and Table 7-6 shows that same results but for arousal, dominance and liking scales respectively.

Table 7-3: Significant feature correlations ($p < 0.01$, * = $p < 0.001$) with Valence scale

Feature	Rhythm	Channels
Spectral Power	Delta	O1-O2, Fp1, PO3, O1, Oz*, P4, P8*
	Theta	O1-O2*, O1, Oz, P4, P8
	Alpha	O1-O2
	Beta	O1-O2, Fp1, P8
	Gamma	O1-O2*
	Alpha to Theta	F3*, PO3, Oz, F4, P4*, P8
	Beta to Alpha	-----
	Beta to Theta	F7, Oz, P8
Oscillation	Delta	AF3, F3, PO3, F4
	Theta	Fp1, PO3, Oz*, F4, T8, P4
	Alpha	T7-T8, Fp1, F3, F7*, FC1, C3*, CP1*, P3, PO3, Pz, Fp2, AF4, F4*, F8*, FC6, FC2*, Cz*, T8*, CP6*, P4*, P8*
	Beta	Fp1-Fp2
	Gamma	FC1*, C3, T7, CP5*, CP1, P3, F4, FC2, C4*, T8, CP6, PO4*, O2
Shannon Entropy	Delta	F3-F4*, F3*, FC1, P8
	Theta	F3*, F7*, PO3, Oz*, P4*, P8
	Alpha	F3*, CP6*
	Beta	P3-P4*, P3, Fp2*, P4
	Gamma	Fp1-Fp2, T7

Table 7-4: Significant feature correlations ($p < 0.01$, * = $p < 0.001$) with Arousal scale

Feature	Rhythm	Channels
Spectral Power	Delta	FC1-FC2*, T7-T8, FC1, P4*
	Theta	FC1-FC2*, T7-T8, CP5-CP6, FC1, T7, P4*
	Alpha	FC1-FC2*, CP5-CP6*, P3-P4, FC1, T7*, Pz, P4*
	Beta	FC1-FC2, CP5-CP6*, P3-P4, F3, T7*, P3, Pz*, FC2, CP6*, P4*, P8
	Gamma	FC1-FC2, T7-T8, CP5-CP6*, T7*, FC2, P4*
	Alpha to Theta	-----

	Beta to Alpha	FC1, P3, Oz, CP2, P8, PO4
	Beta to Theta	FC1
Oscillation	Delta	FC5
	Theta	-----
	Alpha	FC1-FC2, FC1
	Beta	Fp1-Fp2, T7
	Gamma	Fp1-Fp2, FC1-FC2, T7-T8*, CP5-CP6, F3, T7*, Fp2, CP2, O2*
Shannon Entropy	Delta	F8
	Theta	-----
	Alpha	T7, FC2
	Beta	T7-T8
	Gamma	AF3-AF4, AF3*

Table 7-5: Significant feature correlations ($p < 0.01$, * = $p < 0.001$) with Dominance scale

Feature	Rhythm	Channels
Spectral Power	Delta	AF3-AF4, C3-C4*, P3-P4*, PO3-PO4*, Fp1*, F3*, C3*, T7, CP1, P7, PO3*, O1, Oz*, Fp2*, AF4*, Fz*, F4, F8, FC2, Cz*, CP2*, P4*, P8
	Theta	AF3-AF4*, C3-C4*, CP5-CP6, P3-P4*, PO3-PO4*, Fp1*, F3*, C3*, T7, CP1, P7, PO3*, Oz*, Fp2*, AF4*, Fz*, F4, F8, FC2, Cz*, CP2*, P4*, P8
	Alpha	AF3-AF4*, C3-C4*, CP5-CP6*, CP1-CP2, P3-P4*, Fp1*, FC1, C3*, T7, CP1, P3, P7, PO3, Oz*, Fp2*, AF4*, Fz*, FC2, Cz*, T8, CP2*, P4*, P8, O2
	Beta	AF3-AF4*, C3-C4*, CP5-CP6*, CP1-CP2, P3-P4*, Fp1*, C3*, Oz, Fp2*, AF4, Cz, CP6*, P4
	Gamma	AF3-AF4*, C3-C4*, CP5-CP6*, CP1-CP2, P3-P4*, PO3-PO4*, Fp1*, PO3, Oz*, Fp2*, Cz*, P4
	Alpha to Theta	Fp1*, F3*, FC1, P7, PO3*, O1, Oz*, Fp2*, Fz*, F4*, F8*, FC2*, Cz*, P4*
	Beta to Alpha	Fp1, F3*, F7, FC1*, C3, T7*, CP5*, CP1*, P3*, P7*, Pz*, Fp2, AF4*, Fz*, F8, FC2*, Cz*, T8, CP6, CP2*, P4, P8, PO4

	Beta to Theta	Fp1*, F3*, F7*, FC1*, C3, T7*, CP5, CP1, P3, P7*, PO3*, O1, Oz*, Pz, Fp2*, AF4*, Fz*, F4*, F8*, FC2*, Cz*, CP2*, P4*, P8*, PO4
Oscillation	Delta	Fp1*, AF3*, F3*, PO3*, Oz, Fp2*, F4, F8*, Cz*, CP6, P4
	Theta	F7-F8*, PO3-PO4, Fp1*, F3*, P3, PO3*, O1*, Oz*, Fp2*, F4, F8*, Cz*, CP6*, CP2, P4*
	Alpha	Fp1-Fp2, AF3-AF4, F3-F4*, C3-C4*, PO3-PO4, Fp1*, AF3*, F3*, F7*, FC5, FC1*, C3*, T7*, CP5*, CP1*, P3, P7*, PO3*, O1*, Oz*, Fp2*, AF4*, Fz*, F4*, F8*, FC6, FC2*, Cz*, T8*, CP6*, CP2*, P4*, P8*, PO4, O2*
	Beta	F7-F8*, FC5-FC6*, T7-T8, PO3-PO4*, Fp1*, F3, PO3, O1*, Oz*, Fp2*
	Gamma	CP5-CP6*, CP1-CP2*, P7-P8, CP5, O1, Cz, O2
Shannon Entropy	Delta	P3-P4
	Theta	Fp1*, F3, FC1, PO3, Oz*, F8, Cz*, P4
	Alpha	PO3-PO4, AF3, PO3, CP6, P8
	Beta	AF3-AF4*, CP5-CP6, P3-P4*, PO3-PO4, AF3, FC1, PO3*, Oz*, AF4*, CP6*, P4
	Gamma	F7-F8, T7, Pz

Table 7-6: Significant feature correlations ($p < 0.01$, * = $p < 0.001$) with Liking scale

Feature	Rhythm	Channels
Spectral Power	Delta	F7-F8*, Fp1*, F7*, FC5*, C3*, PO3*, O1*, Oz*, Pz, F4*, FC6*, FC2*, Cz*, T8*, CP6*, CP2, P4*, P8*
	Theta	F7-F8*, Fp1, F3, F7*, FC5*, C3, PO3*, O1, Oz*, Pz, F4*, FC6*, FC2*, Cz*, T8*, CP6*, CP2, P4*, P8*
	Alpha	F7-F8*, O1-O2, F7, FC5, PO3*, O1, Oz, FC6, FC2*, Cz, CP6, P8
	Beta	F7-F8*, FC1-FC2, T7-T8, F7, PO3*, Oz, FC2*, T8, CP6, P8
	Gamma	F7-F8*, FC1-FC2*, T7-T8, PO3, FC2*
	Alpha to Theta	Fp1, F7, FC5*, C3*, PO3*, Oz*, F4, FC6, Cz, T8, P4*, P8
	Beta to Alpha	FC6, CP2
	Beta to Theta	Fp1, F3, F7*, FC5, FC1, C3, CP5, PO3*, O1, Oz*, Pz, Fz, F4, FC6, FC2, Cz, CP6, CP2, P4*, P8*

Oscillation	Delta	Oz*
	Theta	Fp1-Fp2, T7-T8, Fp1*, F3, FC5*, PO3*, O1*, Oz*, F4*, FC6, Cz, C4, T8*, CP6*, P4*, P8*
	Alpha	Fp1*, AF3, F3*, F7*, FC5, FC1, C3*, T7, CP5*, CP1, P3*, P7, PO3*, O1*, Oz, Pz*, Fp2, AF4*, F4*, F8*, FC6*, FC2*, Cz*, C4*, T8*, CP6*, CP2*, P4*, P8*, O2
	Beta	Cz
	Gamma	-----
Shannon Entropy	Delta	-----
	Theta	Fp1-Fp2, F3, Oz
	Alpha	F3, FC1, T8, CP6
	Beta	P3-P4*, P3*, P8
	Gamma	AF3-AF4*, AF4

Table 7-7 summarizes the previous tables, where the number of electrodes with a significant correlation ($p<0.01$) are reported.

Table 7-7: Number of electrodes with significant correlation ($p<0.01$) for each feature and frequency band

Feature	Scale	Delta	Theta	Alpha	Beta	Gamma
Spectral Power	Valence	7	5	1	3	1
	Arousal	4	6	7	11	6
	Dominance	23	23	24	13	12
	Liking	18	19	12	10	5
Oscillation	Valence	4	6	21	1	32
	Arousal	1	0	2	2	9
	Dominance	11	15	35	10	7
	Liking	1	16	30	1	0
Shannon Entropy	Valence	4	6	2	4	2
	Arousal	1	0	2	1	2
	Dominance	1	8	5	11	3
	Liking	0	3	4	3	2

By reviewing the previous tables, the one can note that, all computed features and frequency bands have significant channels, which imply the importance of all of them, so it is not recommended to exclude any feature or frequency band, even delta rhythm which believed to include a hug noise, and it was excluded by many researchers.

7.5. Verifying the preprocessing and feature extraction steps

For the sake of verifying the results of the preprocessing and feature extraction steps, a reproduction of one of the previous works is done with the extracted features of the previous steps. The work of *Koelstra et al.*, collectors of DEAP dataset, is adopted to be reproduced for the verification [Koelstra et al., 2012].

The proposed model by *Koelstra et al.* depends on the spectral power features. Fisher's linear discriminant was used for feature selection with a threshold at 0.3 and a Gaussian naïve Bayes classifier was used for classification. The model was validated by a leave-one-out cross-validation scheme for each participant separately [Koelstra et al., 2012].

The same experiment of *Koelstra et al.* was set up for reproduction with the extracted features in this study. Fisher's linear discriminant used by *Koelstra et al.* was different than the already implemented LDA by scikit learn toolkit or by MATLAB. So, Fisher's linear discriminant was reimplemented using the provided equations by *Koelstra et al.*, on the other hand, the existing Gaussian naïve Bayes classifier in scikit learn toolkit was used. The validation scheme of *Koelstra et al.* was also reimplemented. Table 7-8 shows the results of reproducing *Koelstra et al.* work.

Table 7-8: Results of reproduction Koelstra et al. work

	Accuracy (%)				F1-Score (%)			
	V	A	D	L	V	A	D	L
<i>Koelstra et al.</i>	57.60	62.00	-	55.40	56.30	58.30	-	50.20
Reproduction	61.41	59.38	65.47	64.22	58.80	52.58	55.77	55.58

* V: valence scale, A: arousal scale, D: dominance scale, L: liking scale.

Despite having some differences between the results of *Koelstra et al.* and the reproduction that have been made, the results of the reproduction show that the extracted features resulting from the previous steps can be used for the next steps of emotion recognition process.

7.6. Checking the worthiness of partitioning the Dataset

As discussed in Chapter 4, many studies divide the dataset into parts, and a model was built for each part. This experiment has been made to check the effect of partitioning the dataset on the model performance. The extracted spectral power features of the major five frequency bands of all EEG electrodes are used in this experiment. Also, the difference in the spectral power of the five frequency bands between all symmetrical pairs of electrodes are calculated. Pearson correlation coefficient is used to select the most significant features ($p < 0.05$) and Gaussian naïve Bayes is used for classification. The classification step is validated using 10-fold cross-validation scheme. The achieved accuracy by the used classifier are reported in Table 7-9.

Table 7-9: Results of dataset partitioning experiment

Partitions	Valence (%)	Arousal (%)	Dominance (%)	Liking (%)
None	56.49	60.15	60.98	62.12
By gender	57.72	61.08	59.14	61.31
2 random groups	59.30	59.70	61.40	60.01
By participant	71.23	70.09	72.45	72.16
32 random groups	65.54	64.68	66.28	66.63

By looking at Table 7-9, it is obvious that there is nearly no effect for partitioning the dataset by gender on the classification accuracy. A slightly better accuracy was achieved by partitioning the dataset into two random parts. This means that, there is no point in partitioning the dataset based on gender, and the good results that reached by *Chen et al.* are related to the used features in their proposed model.

Partitioning the dataset into 32 parts based on participant improve the classification accuracy by nearly 12%. On the other hand, partitioning the dataset into 32 random parts also improves the accuracy but with a smaller value (6%). So, partitioning the dataset by participant is making sense, but partitioning the dataset into smaller parts affects the improved results. The drawback of these works is it is must to build a separated model for each participant or for each group of participants which leads to the new comer will not benefit from the previous models and it is needed to build a new one. So, it's better to build a model that can deal with all data of different participants than build a model for some kinds of participants or for each participant separately. The weak point in partitioning the dataset is that, the built model will need to be applied on users with the same characteristics. So, if a model was built for each participant separately, a new user must retrain the model before using it, and if the dataset was separated based on gender or other characteristics, a new user must configure the system before using it.

7.7. Evaluating the Proposed Model

SVM-RFE was applied to reduce the extracted features. 75% of the extracted features were eliminated by SVM-RFE and 25% were left to be used for the classification of emotion classes. The number of features that were left for classification is $25\% \times 786 = 196$ features. This elimination was done for each emotion scale separately.

Several famous classifiers were trained on the selected features by RFE including SVM with RBF and with linear kernels, LDA, k-NN, Gaussian naïve Bayes, MLP and Decision Tree. To evaluate the performance of these classifiers, a 10-fold cross-validation scheme was used and both accuracy and F₁-score were calculated. Some algorithms use randomness and give different results each time they are executed, so the process of evaluating each classifier was repeated 10 times and the means of accuracy and F₁-score were reported. The accuracy and F₁-score of the used classifiers for each emotional scale are reported in Table 7-10. Figure 7.8 compares the

performance of each classifier for each scale in terms of accuracy while Figure 7.9 compares the performance in terms of F₁-score.

Table 7-10: Results of evaluating different classifiers

Classifier	Accuracy (%)				F ₁ -score (%)			
	V	A	D	L	V	A	D	L
SVM (RBF)	61.22	65.05	66.36	68.68	45.52	56.06	55.58	42.32
SVM (Linear)	76.84	77.66	79.99	79.45	74.91	75.38	76.92	74.12
LDA	74.14	74.10	76.17	76.77	72.24	71.74	72.88	71.85
k-NN	63.91	63.31	66.64	66.47	60.70	57.76	60.56	57.22
G. naïve Bayes	54.89	63.06	61.46	60.63	54.32	58.99	60.26	56.43
MLP	60.29	65.55	66.24	67.14	51.21	58.18	56.27	48.88
Decision tree	57.68	59.93	60.95	58.77	55.34	57.18	56.91	52.42

* V: valence scale, A: arousal scale, D: dominance scale, L: liking scale.

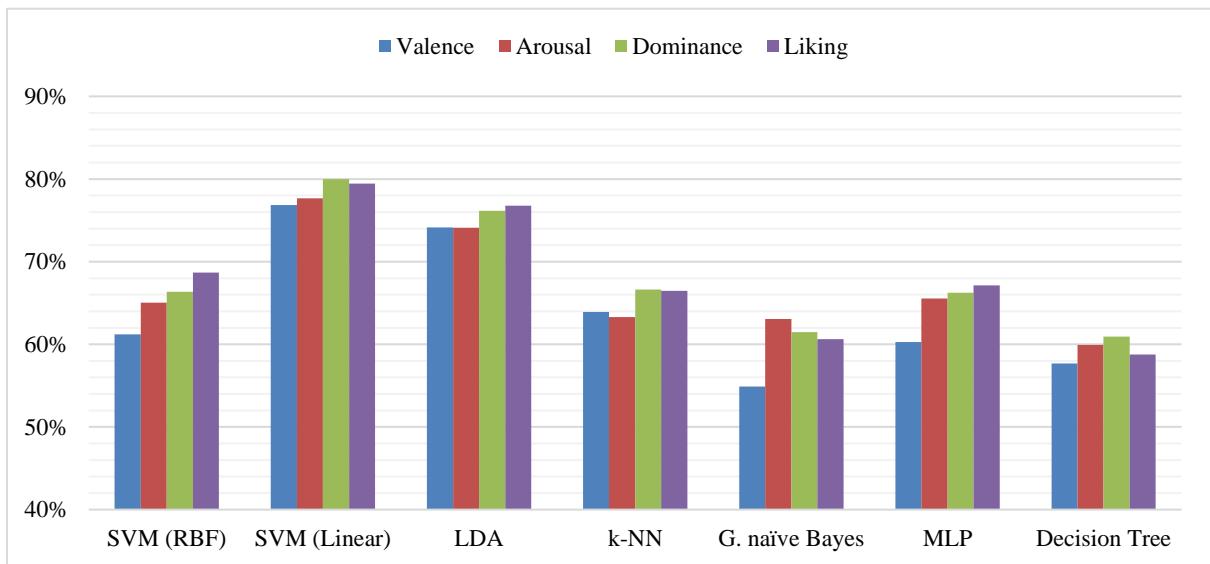


Figure 7.8: Accuracy of different classifiers for each emotional scale

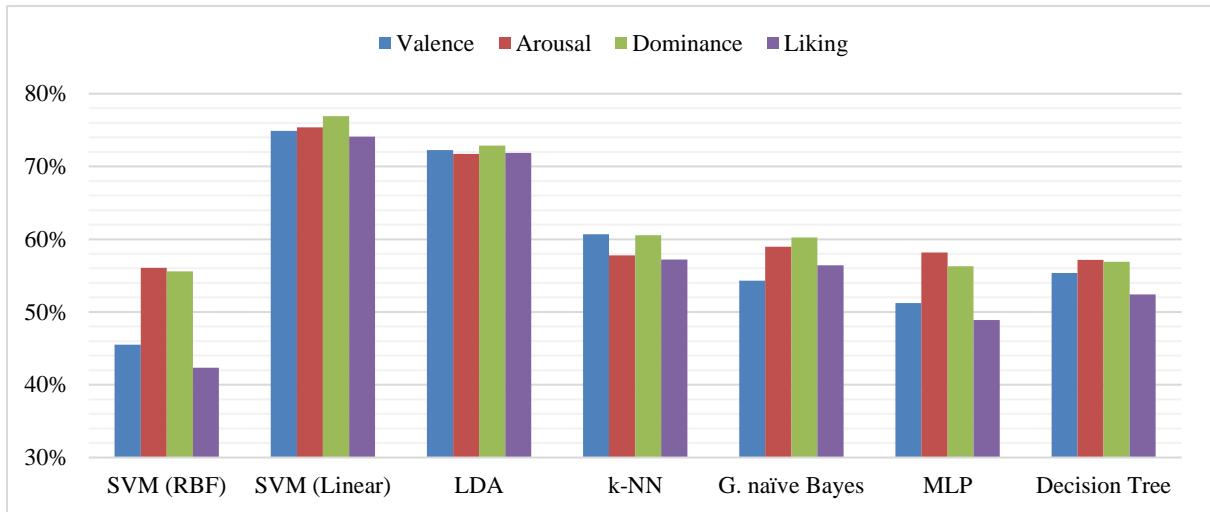


Figure 7.9: F₁-score of different classifiers for each emotional scale

SVM with a linear kernel achieved best results among other used classifiers in this study. So, it was selected to be compared with results of studies that mentioned in the previous work. Only previous studies that separate emotional scales into two classes are compared with the proposed model. Table 7-11 compares the results of the previously discussed related work with the proposed model while Figure 7.10 and Figure 7.11 illustrate this comparison in terms of accuracy and F₁-score respectively.

Table 7-11: Comparing the proposed model with previous studies

Method	Accuracy (%)				F ₁ -Score (%)			
	V	A	D	L	V	A	D	L
[Koelstra et al., 2012]	57.60	62.00	-	55.40	56.30	58.30	-	50.20
[Matiko et al., 2014]	62.62	-	-	-	-	-	-	-
[Daimi et al., 2014]	65.30	66.90	69.10	71.20	55.00	57.00	55.20	50.90
[Chen et al., 2015]	67.89	69.09	-	-	67.83	68.96	-	-
[Gao et al., 2015]	58.00	58.40	-	-	55.20	48.80	-	-
[Li et al., 2016]	72.06	74.12	-	-	-	-	-	-
[Tripathi et al., 2017]	81.41	73.36	-	-	-	-	-	-
[Chao et al., 2018]	75.92	76.83	-	-	69.31	70.15	-	-
The proposed model	76.84	77.66	79.99	79.45	74.91	75.38	76.92	74.12

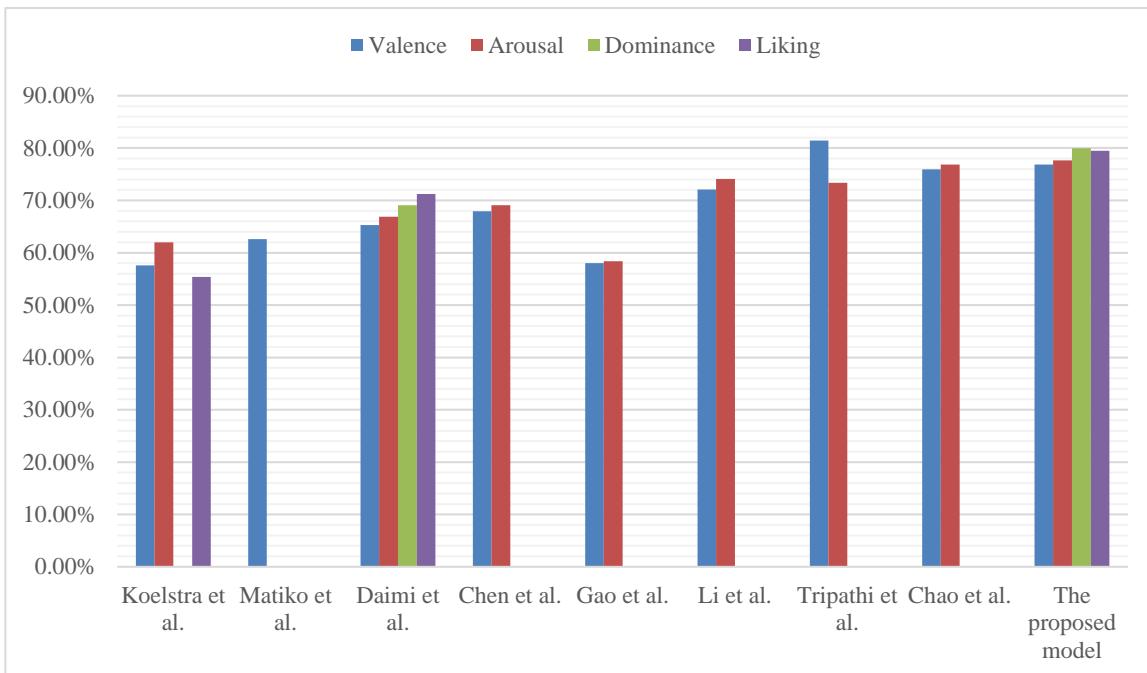


Figure 7.10: Comparing accuracies with the previous studies

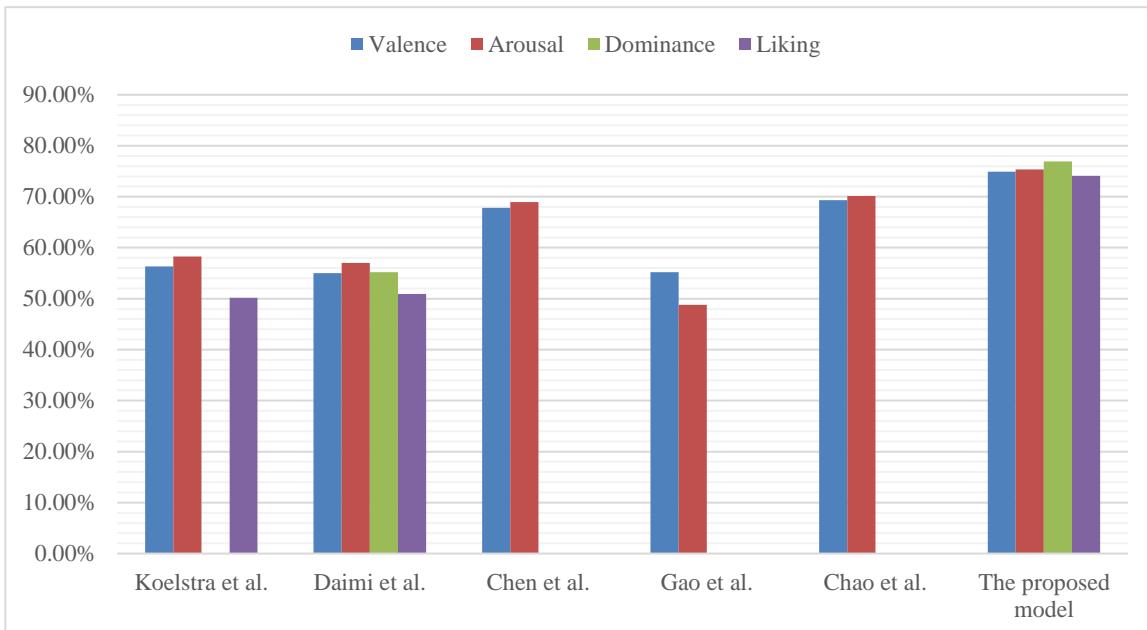


Figure 7.11: Comparing F₁-scores with the previous studies

7.8. Discussion

As expected, linear SVM outperformed other classifiers where it achieved accuracy of 76.84%, 77.66%, 79.99% and 79.45% for valence, arousal, dominance and liking respectively. LDA also as a linear classifier achieved great results (accuracy of 74.14%, 74.10%, 76.17% and 76.77%), which imply that SVM-RFE selects features that maximize the linear separability of data, so the performance of linear classifiers is enhanced. Linear SVM also achieved good F₁-score of 74.91%, 75.38%, 76.92% and 74.12% for valence, arousal, dominance and liking respectively which confirms the consistent of RFE performance and its robustness to overfitting.

Previous studies that were provided in Chapter 5 can be categorized into two categories. The first category is studies that just applied classical machine learning techniques, while the second is studies applied deep learning techniques. The proposed model in this study belongs to the first category. The following subsections compare the proposed model with studies of both categories.

7.8.1. Classical Machine Learning Studies

In terms of valence and arousal scales, *Chen et al.* achieved the best results among other studies with accuracy of 67.89% and 69.09% respectively while the proposed model achieved better accuracy of 76.84% and 77.66% (8.95% and 8.57% higher) for valence and arousal respectively. In terms of dominance and liking scales, *Daimi et al.* achieved good accuracy of 69.10% and 71.20% but low F₁-score of 55.20% and 50.90% respectively. The proposed model achieved better accuracy of 79.99% and 79.45% (10.89% and 8.25% higher) and good F₁-score of 76.92% and 74.12% for dominance and liking respectively. The proposed model achieved an average of 9% higher accuracy than the best of other models.

7.8.2. Deep Learning Studies

Chao et al. and *Tripathi et al.* achieved the best results among other studies. Comparing to *Chao et al.*, In terms of accuracy, the proposed model achieved the same

results approximately (less than 1% higher accuracy for both scales), but in terms of F₁-score, the proposed model achieved a better F₁-score (more than 5% higher F₁-score for both scales). *Tripathi et al.* report the model accuracy only, while the proposed model achieved a lower accuracy in terms of valence (4.57% lower), it achieved a better accuracy in terms of arousal (4.30% higher).

The comparison with other studies confirms the superiority and robustness of the proposed model. Another advantage of the proposed model is that, it was not built for each participant separately, such that new users will not need a training step before use it. Also, it was not built for some kinds of users, such that no configuration steps will be needed before using it either.

7.9. Summary

This chapter has presented an experimental study to evaluate the significance of the proposed model for emotion recognition using brain signals. Several classifiers were tested on the reduced features set by SVM-RFE as a wrapper-based feature reduction method. The best results were compared with previous models of emotion recognition and a discussion of this results was provided. Other experiments with the extracted features were also made and reported in this chapter.

The significant results that brought by these experimental studies showed the success of the proposed model for emotion recognition using brain signals.

CHAPTER 8

CONCLUSION AND FUTURE WORK

This thesis introduced a model for improving emotion recognition based on brain signals that recorded by EEG devices using a wrapper-based feature selection method. The proposed model is participant free, not constrained to specific users or specific groups of users, and it will become a convergence (trained) model for any new user, so no training or configuration steps are required before using it by new users.

The proposed model was tested on a bench mark dataset with a relatively large number of participants. Three emotional scales of valence, arousal and dominance in addition to liking scale were targeted by the proposed model to be recognized. The experimental results showed that linear classifiers achieved high accuracy compared to other classifiers which imply that SVM-RFE selects features that maximize the linear separability of data. SVM achieved the highest classification accuracy which are 76.84%, 77.66%, 79.99% and 79.45% for valence, arousal, dominance and liking respectively. SVM also achieved high F_1 -scores of 74.91%, 75.38%, 76.92% and 74.12% which confirms the consistent of RFE performance and its robustness to overfitting.

The proposed model achieved accuracy of 8.95%, 8.57%, 10.89% and 8.25% (9% on average) higher than the best results that were achieved by previous EEG-based emotion recognition approaches that applied classical machine learning techniques. In case of the studies which applied the modern deep learning techniques, the proposed approach shows a distinguish competitor, since the proposed model achieved close results in terms of accuracy and higher results in terms of F_1 -score (5% higher F_1 -score). The comparison with other studies confirms the superiority and robustness of the proposed model.

8.1. Thesis Contribution

The following items summarize the contributions of this thesis:

1. Finding out that the best combination of feature reduction and classification methods to improve the accuracy of EEG-based emotion recognition were RFE and SVM.
2. Finding out that partitioning the dataset based on participant, gender or even random will give better results but they are misleading results.
3. Finding out that one of the reasons behind low accuracy of EEG-based emotion recognition, that is found in other researches work, is neglecting delta rhythms from feature extraction.

8.2. Future Work

As extension of this research, using other wrapper-based feature selection methods, such as genetic algorithm or swarm optimization methods, is planned as future work. In addition, the following points are possible future work in this research area:

- Using DT-CWPT based energy features with SVM-RFE for feature reduction or any other wrapper-based feature reduction method.
- Using advanced approaches of Deep Learning Networks as a classifier.
- Applying the proposed model on online datasets.

REFERENCES

[Aftanas et al, 2001]

Aftanas, L.I., and S.A. Golocheikine. “Human anterior and frontal midline theta and lower alpha reflect emotionally positive state and internalized attention: high-resolution EEG investigation of meditation.” *Neuroscience Letters* 310, no. 1, pp. 57–60, September 2001.

[Alpaydin, 2014]

Alpaydin, Ethem. “*Introduction to machine learning.*” MIT press, 2014.

[ACNS, 2006]

American Clinical Neurophysiology Society. “Guideline 5: Guidelines for Standard Electrode Position Nomenclature.” *Journal of Clinical Neurophysiology* 32, no. 2, pp. 107–110, April 2006.

[Anand et al., 1961]

Anand, B.K., G.S. Chhina, and Baldev Singh. “Some aspects of electroencephalographic studies in Yogis.” *Electroencephalography and Clinical Neurophysiology* 13, no. 3, pp. 452–456, June 1961.

[Baillet et al., 2001]

Baillet, Sylvain, John C. Mosher, and Richard M. Leahy. “Electromagnetic brain mapping.” *IEEE Signal Processing Magazine* 18, no. 6, pp. 14–30, November 2001.

[Bayram et al., 2008]

Bayram, İlker, and Ivan W. Selesnick. “On the Dual-Tree Complex Wavelet Packet and M-Band Transforms.” *IEEE Transactions on Signal Processing* 56, no. 6, pp. 2298–2310, May 2008.

[Bennett et al., 2000]

Bennett, Kristin P., and Colin Campbell. “Support vector machines: hype or hallelujah?” *ACM SIGKDD Explorations Newsletter* 2, no. 2, pp. 1–13, December 2000.

[Bishop, 1996]

Bishop, Christopher M. “*Neural Networks for Pattern Recognition.*” Oxford University Press, 1995.

[Borisoff et al., 2004]

Borisoff, Jaimie F., Steve G. Mason, and Ali Bashashati. “Brain-computer interface design for asynchronous control applications: improvements to the LF-ASD asynchronous brain switch.” *IEEE Transactions on Biomedical Engineering* 51, no. 6, pp. 985–992, May 2004.

[Bradley et al., 2001]

Bradley, Margaret M., Maurizio Codispoti, Dean Sabatinelli, and Peter J. Lang. “Emotion and motivation II: Sex differences in picture processing.” *Emotion* 1, no. 3, pp. 300–319, September 2001.

[Burges, 1998]

Burges, Christopher J.C. “A Tutorial on Support Vector Machines for Pattern Recognition.” *Data Mining and Knowledge Discovery* 2, no. 2, pp. 121–167, June 1998.

[Burrus et al., 1998]

Burrus, C. Sidney, Ramesh A. Gopinath, Haitao Guo, Jan E. Odegard, and Ivan W. Selesnick. “*Introduction to wavelets and wavelet transforms: a primer.*” Vol. 1. New Jersey: Prentice hall, 1998.

[Cazala, 2017]

Cazala, Juan. “Architect - Cazala/Synaptic Wiki.” GitHub. 2015. Accessed November 18, 2017. <https://github.com/caza la/synaptic/wiki/Architect>.

[Chang et al., 2011]

Chang, Chih-Chung, and Chih-Jen Lin. “LIBSVM: a library for support vector machines.” *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, no. 3, pp. 27, April 2011.

[Chao et al., 2018]

Chao, Hao, Huilai Zhi, Liang Dong, and Yongli Liu. “Recognition of Emotions Using Multichannel EEG Data and DBN-GC-Based Ensemble Deep Learning Framework.” *Computational Intelligence and Neuroscience*, 2018.

[Chen et al., 2015]

Chen, Jing, Bin Hu, Philip Moore, Xiaowei Zhang, and Xu Ma. “Electroencephalogrambased emotion assessment system using ontology and data mining techniques.” *Applied Soft Computing* 29, pp. 663–674, January 2015.

[Christy et al., 2012]

Christy, Thomas, Ludmila I. Kuncheva, and Kerry W. Williams. “*Selection of Physiological Input Modalities for Emotion Recognition.*” Technical report CSTR-002-2012. Bangor University, UK, 2012.

[Daimi et al., 2014]

Daimi, Syed Naser, and Goutam Saha. “Classification of emotions induced by music videos and correlation with participants’ rating.” *Expert Systems with Applications* 41, no. 13, pp. 6057–6065, October 2014.

[DEAP, 2015]

“DEAP: A Dataset for Emotion Analysis using Physiological and Audiovisual Signals.” Queen Mary University of London, UK. 2012. Accessed March 28, 2015. <http://www.eecs.qmul.ac.uk/mmv/datasets/deap/readme.html>.

[Dodge, 2008]

Dodge, Yadolah. “*The concise encyclopedia of statistics.*” Springer Science+Business Media, 2008.

[Duda et al., 2001]

Duda, Richard O., Peter E. Hart, and David G. Stork. “*Pattern Classification.*” Vol. 2. New York: John Wiley & sons, 2001.

[Ekman et al., 1987]

Ekman, Paul, Wallace V. Friesen, Maureen O’Sullivan, Anthony Chan, Irene Diacoyanni-Tarlatzis, Karl Heider, Rainer Krause, William Ayhan LeCompte, Tom Pitcairn, Pio E. Ricci-Bitti, Klaus Scherer, Masatoshi Tomita, and Athanase Tzavaras. “Universals and Cultural Differences in the Judgments of Facial Expressions of Emotion.” *Journal of personality and social psychology* 53, no. 4, pp. 712–717, October 1987.

[Fontaine et al., 2007]

Fontaine, Johnny R.J., Klaus R. Scherer, Etienne B. Roesch, and Phoebe C. Ellsworth. “The World of Emotions is not Two-Dimensional.” *Psychological Science* 18, no. 12, pp. 1050–1057, December 2007.

[Friedman, 1997]

Friedman, Jerome H.K. “On bias, variance, 0/1—loss, and the curse-of-dimensionality.” *Data mining and knowledge discovery* 1, no. 1, pp. 55–77, 1997.

[Fukunaga, 1990]

Fukunaga, Keinosuke. “*Introduction to Statistical Pattern Recognition.*” 2nd ed. Academic Press, October 1990.

[Gao et al., 2015]

Gao, Zhen, and Shangfei Wang. “Emotion Recognition from EEG Signals using Hierarchical Bayesian Network with Privileged Information.” In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pp. 579–582, ACM, June 2015.

[Guyon et al., 2002]

Guyon, Isabelle, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. “Gene Selection for Cancer Classification using Support Vector Machines.” *Machine learning* 46, no. 1, pp. 389–422, January 2002.

[Guyon et al., 2003]

Guyon, Isabelle, and André Elisseeff. “An Introduction to Variable and Feature Selection.” *The Journal of Machine Learning Research* 3, pp. 1157–1182, 2003.

[Harman et al, 1977]

Harman, David W., and William J. Ray. “Hemispheric activity during affective verbal stimuli: An EEG study.” *Neuropsychologia* 15, no. 3, pp. 457–460, 1977.

[Hoagland et al., 1937]

Hoagland, Hudson, D. Ewen Cameron, and Morton A. Rubin. “The electroencephalogram of schizophrenics during insulin treatments.” *The American Journal of Psychiatry* 94, no. 1, pp. 183–208, July 1937.

[Hoagland et al., 1938]

Hoagland, Hudson, D. Ewen Cameron, and Morton A. Rubin. “Emotion in man as Tested by the Delta Index of the Electroencephalogram: I.” *The Journal of General Psychology* 19, no. 2, pp. 227–245, 1938.

[Inside The Brain, 2013]

Inside The Brain. “What is ‘Attention’ and where is it in the brain.” inside-the-brain.com. March 2013. Accessed November 6, 2017. <https://inside-the-brain.com/2013/03/07/what-is-attention-and-where-is-it-in-the-brain/>.

[Jain et al., 2000]

Jain, Anil K., Robert P.W. Duin, and Jianchang Mao. “Statistical Pattern Recognition: A Review.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, no. 1, pp. 4–37, January 2000.

[Janecek et al., 2008]

Janecek, Andreas, Wilfried Gansterer, Michael Demel, and Gerhard Ecker. “On the relationship between feature selection and classification accuracy.” In *New Challenges for Feature Selection in Data Mining and Knowledge Discovery*, pp. 90–105. 2008.

[Jirayucharoensak et al., 2014]

Jirayucharoensak, Suwicha, Setha Pan-Ngum, and Pasin Israsena. “EEG-Based Emotion Recognition Using Deep Learning Network with Principal Component Based Covariate Shift Adaptation.” *The Scientific World Journal*, September 2014.

[John et al., 1995]

John, George H., and Pat Langley. “Estimating continuous distributions in Bayesian classifiers.” In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pp. 338–345, Morgan Kaufmann Publishers Inc., 1995.

[Johnson et al., 1984]

Johnson, Rodney, and J Shore. “Which is the better entropy expression for speech processing: -S log S or log S?” *IEEE transactions on acoustics, speech, and signal processing* 32, no. 1, pp. 129–137, February 1984.

[Kandel et al., 2013]

Kandel, Eric R., James H. Schwartz, Thomas M. Jessell, Steven A. Siegelbaum, and A. J. Hudspeth. “*Principles of Neural Science.*” McGraw-hill New York, 2013.

[Khalid et al., 2009]

Khalid, Muhammad Bilal, Naveed Iqbal Rao, Intisar Rizwan-i-Haque, Sarmad Munir, and Farhan Tahir. “Towards a Brain Computer Interface Using Wavelet Transform with Averaged and Time Segmented Adapted Wavelets.” In *2nd international conference on Computer, control and communication*, pp. 1–4, IEEE, February 2009.

[Kingsbury, 2001]

Kingsbury, Nick. “Complex Wavelets for Shift Invariant Analysis and Filtering of Signals.” *Applied and Computational Harmonic Analysis* 10, no. 3, pp. 234–253, May 2001.

[Klimesch, 1997]

Klimesch, Wolfgang. “EEG-alpha rhythms and memory processes.” *International Journal of psychophysiology* 26, nos. 1-3, pp. 319–340, 1997.

[Koelstra et al., 2012]

Koelstra, Sander, Christian Mühl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. “DEAP: A Database for Emotion Analysis Using Physiological Signals.” *IEEE Transactions on Affective Computing* 3, no. 1, pp. 18–31, January 2012.

[Kohavi et al., 1997]

Kohavi, Ron, and George H. John. “Wrappers for feature subset selection.” *Artificial intelligence* 97, nos. 1-2, pp. 273–324, 1997.

[Kübler et al., 2001]

Kübler, Andrea, Boris Kotchoubey, Jochen Kaiser, Jonathan R. Wolpaw, and Niels Birbaumer. “Brain-computer communication: unlocking the locked in.” 127, no. 3, pp. 358–375, 2001.

[Kuhn et al., 2013]

Kuhn, Max, and Kjell Johnson. “Applied predictive modeling.” Vol. 26. Springer, 2013.

[Laureys et al., 2009]

Laureys, Steven, Melanie Boly, and Giulio Tononi. “Functional Neuroimaging.” In *The Neurology of Consciousness*, pp. 31–42, 2009.

[Learning about Electronics, 2017]

Learning about Electronics. “How to Build a Peak-to-Peak Detector Circuit.” learningaboutelectronics.com. 2017. Accessed February 22, 2018.
<http://www.learningaboutelectronics.com/Articles/Peak-to-peak-detector-circuit.php>.

[Lee et al., 2003]

Lee, Kwang-Hyuk, Leanne M. Williams, Michael Breakspear, and Evian Gordon. “Synchronous gamma activity: a review and contribution to an integrative neuroscience model of schizophrenia.” *Brain Research Reviews* 41, no. 1, pp. 57–78, January 2003.

[Li et al., 2016]

Li, Xiang, Dawei Song, Peng Zhang, Guangliang Yu, Yuexian Hou, and Bin Hu. “Emotion recognition from multi-channel EEG data through Convolutional Recurrent Neural Network.” In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 352-359. IEEE, 2016.

[Liu et al., 1998]

Liu, Huan, and Hiroshi Motoda. *Feature selection for knowledge discovery and data mining*. Kluwer Academic Publishers, 1998.

[Lotte et al., 2007]

Lotte, Fabien, Marco Congedo, Anatole Lécuyer, Fabrice Lamarche, and Bruno Arnaldi. “A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces.” 4, no. 2, 2007.

[Lowry, 2014]

Lowry, Richard. “Concepts and applications of inferential statistics,” 2014. Accessed July 21, 2018. <http://vassarstats.net/textbook/>.

[Lutzenberger at al., 1995]

Lutzenberger, Werner, Friedemann Pulvermüller, Thomas Elbert, and Niels Birbaumer. “Visual stimulation alters local 40-Hz responses in humans: an EEG-study.” *Neuroscience letters* 183, no. 1-2, pp. 39–42, 1995.

[Manning et al., 2008]

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. “*Introduction to information retrieval*.” Cambridge University Press, 2008.

[Matiko et al., 2014]

Matiko, Joseph W., Stephen P. Beeby, and John Tudor. “Fuzzy logic based emotion classification.” In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4389–4393. IEEE, May 2014.

[McDonald, 2009]

McDonald, John H. “*Handbook of biological statistics.*” Vol. 2. Sparky House Publishing Baltimore, 2009.

[Mehrabian, 1996]

Mehrabian, Albert. “Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in Temperament.” *Current Psychology* 14, no. 4, pp. 261–292, December 1996.

[Mesquita, 2003]

Mesquita, Batja. “Emotions as dynamic cultural phenomena.” In *Handbook of Affective Sciences*, pp. 871–890, Oxford University Press, 2003.

[Mühl et al., 2014]

Mühl, Christian, Brendan Allison, Anton Nijholt, and Guillaume Chanel. “A survey of affective brain computer interfaces: principles, state-of-the-art, and challenges.” *Brain-Computer Interfaces* 1, no. 2, pp. 66–84, May 2014.

[Myers et al., 2003]

Myers, Jerome L., and Arnold D. Well. “*Research Design and Statistical Analysis.*” Lawrence Erlbaum Associates, Inc, 2003.

[Nicolas-Alonso et al., 2012]

Nicolas-Alonso, Luis Fernando, and Jaime Gomez-Gil. “Brain Computer Interfaces, a Review.” *Sensors* 12, no. 2, pp. 1211–1279, January 2012.

[Oppenheim, 1969]

Oppenheim, Alan V. “*Papers on Digital Signal Processing.*” MIT Press, 1969.

[Oppenheim et al., 2015]

Oppenheim, Alan V., and George C. Verghese. “*Signals, systems and inference.*” Pearson, 2015.

[Pedregosa et al., 2011]

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. “Scikit-learn: Machine learning in Python.” *Journal of Machine Learning Research* 12, pp. 2825–2830, 2011.

[Pelvig et al., 2008]

Pelvig, D.P., H. Pakkenberg, A.K. Stark, and B. Pakkenberg. “Neocortical glial cell numbers in human brains.” *Neurobiology of Aging* 29, no. 11, pp. 1754–1762, November 2008.

[Petrantonakis et al., 2010]

Petrantonakis, Panagiotis C., and Leontios J. Hadjileontiadis. “Emotion Recognition from Brain Signals Using Hybrid Adaptive Filtering and Higher Order Crossings Analysis.” *IEEE Transactions on Affective Computing* 1, no. 2, pp. 81–97, August 2010.

[Pfurtscheller et al., 2001]

Pfurtscheller, Gert, and Christa Neuper. “Motor imagery and direct braincomputer communication.” *Proceedings of the IEEE* 89, no. 7, pp. 1123–1134, July 2001.

[Picard, 1995]

Picard, Rosalind W. “*Affective Computing*.” Technical report 321. MIT Media Laboratory Perceptual Computing Section, April 1995.

[Rosdahl et al., 2011]

Rosdahl, Caroline Bunker, and Mary T. Kowalski. “*Textbook of basic nursing*.” Lippincott Williams & Wilkins, 2011.

[Russell et al., 1977]

Russell, James A., and Albert Mehrabian. “Evidence for a three-factor theory of emotions.” *Journal of Research in Personality* 11, no. 3, pp. 273–294, September 1977.

[Russell, 1980]

Russell, James A. “A circumplex model of affect.” *Journal of Personality and Social Psychology* 39, no. 6, pp. 1161–1178, December 1980.

[Russell, 1991]

Russell, James A. “Culture and the categorization of emotions.” *Psychological Bulletin* 110, no. 3, pp. 426–450, 1991.

[Scherer, 2005]

Scherer, Klaus R. “What are emotions? And how can they be measured?” *Social Science Information* 44, no. 4, pp. 695–729, December 2005.

[Schmidt et al., 2001]

Schmidt, Louis A., and Laurel J. Trainor. “Frontal brain electrical activity (EEG) distinguishes valence and intensity of musical emotions.” *Cognition and Emotion* 15, no. 4, pp. 487–500, 2001.

[Shannon, 1948]

Shannon, Claude Elwood. “A Mathematical Theory of Communication.” *The Bell System Technical Journal* 27, no. 3, pp. 379–423, July 1948.

[Shao et al., 1993]

Shao, Jun. “Linear model selection by cross-validation.” *Journal of the American Statistical Association* 88, no. 422, pp. 486–494, 1993.

[Smith, 2003]

Smith, Steven. “*Digital signal processing: a practical guide for engineers and scientists.*” Elsevier, 2003.

[Soleymani et al., 2011]

Soleymani, Mohammad, Sander Koelstra, Ioannis Patras, and Thierry Pun. “Continuous Emotion Detection in Response to Music Videos.” In *IEEE International Conference on Automatic Face and Gesture Recognition and Workshops (FG 2011)*, pp. 803–808, IEEE, March 2011.

[Stein, 2000]

Stein, Jonathan Y. “*Digital signal processing: A computer science perspective.*” John Wiley & Sons, Inc., 2000.

[Tripathi et al., 2017]

Tripathi, Samarth, Shrinivas Acharya, Ranti Dev Sharma, Sudhanshu Mittal, and Samit Bhattacharya. “Using Deep and Convolutional Neural Networks for Accurate Emotion Classification on DEAP Dataset.” In *Proceeding of the Twenty-Ninth AAAI Conference on Innovative Applications (IAAI-17)*, pp. 4746-4752, 2017.

[Venables et al., 2009]

Venables, Louise, and Stephen H. Fairclough. “The influence of performance feedback on goal-setting and mental effort regulation.” *Motivation and Emotion* 33, no. 1, pp. 63–74, 2009.

[Welch, 1967]

Welch, Peter. “The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms.” *IEEE Transactions on audio and electroacoustics* 15, no. 2, pp. 70–73, 1967.

[Wheeler et al., 1993]

Wheeler, Robert E., Richard J. Davidson, and Andrew J. Tomarken. “Frontal brain asymmetry and emotional reactivity: A biological substrate of affective style.” *Psychophysiology* 30, no. 1, pp. 82–89, January 1993.

[Wu et al., 2000]

Wu, Chien-Fu Jeff, and Michael S. Hamada. *Experiments: Planning, Analysis, and Parameter Design Optimization*. Vol. 1. John Wiley & Sons, 2000.

[Zhang et al., 2010]

Zhang, Li, Wei He, Chuanhong He, and Ping Wang. “Improving mental task classification by adding high frequency band information.” *Journal of medical systems* 34, no. 1, pp. 51–60, February 2010.

APPENDIX A:

DEAP DATASET

DEAP Dataset: a sample part.

The screenshot shows the homepage of the DEAPdataset website. At the top, there is a navigation bar with tabs for "home" (which is highlighted in red), "dataset description", "download", and "contact". Below the navigation bar, there is a photograph of a participant wearing an EEG cap and sitting at a computer. To the right of the photo is a grid of 12 topographic EEG maps showing spatial distribution of brain activity across different electrode sites. Below the photo and maps, there is a section titled "abstract" which contains text about the dataset. To the right of the abstract is a scatter plot of Valence score (y-axis, -1.5 to 2) versus Arousal score (x-axis, -1.5 to 2). Four specific video clips are highlighted with green boxes and arrows pointing to them: "Louis Armstrong What a wonderful world", "Blur Song 2", "Napalm death Procrastination on the empty vessel", and "Sia Breathe me". At the bottom left, there is a "how to use" section with instructions and a link to the "dataset description page".

We present a multimodal dataset for the analysis of human affective states. The electroencephalogram (EEG) and peripheral physiological signals of 32 participants were recorded as each watched 40 one-minute long excerpts of music videos. Participants rated each video in terms of the levels of arousal, valence, like/dislike, dominance and familiarity. For 22 of the 32 participants, frontal face video was also recorded. A novel method for stimuli selection was used, utilising retrieval by affective tags from the last.fm website, video highlight detection and an online assessment tool.

The dataset is made publicly available and we encourage other researchers to use it for testing their own affective state estimation methods. The dataset was first presented in the following paper:

- ["DEAP: A Database for Emotion Analysis using Physiological Signals \(PDF\)"](#), S. Koelstra, C. Muehl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, I. Patras, IEEE Transaction on Affective Computing, Special Issue on Naturalistic Affect Resources for System Building and Evaluation, *in press*

how to use

If you are interested in using this dataset, you will have to print, sign and scan an EULA (End User License Agreement) and return it via email. We will then supply you with a username and password to download the data. Please head on over to the [downloads page](#) for more details.

Also, please consult the [dataset description page](#) for a complete explanation of the dataset.

credits

Figure A.1: The home page of DEAP dataset

DEAPdataset

a dataset for emotion analysis using eeg, physiological and video signals

home

dataset description

download

contact

dataset summary

The DEAP dataset consists of two parts:

1. The ratings from an online self-assessment where 120 one-minute extracts of music videos were each rated by 14-16 volunteers based on arousal, valence and dominance.
2. The participant ratings, physiological recordings and face video of an experiment where 32 volunteers watched a subset of 40 of the above music videos. EEG and physiological signals were recorded and each participant also rated the videos as above. For 22 participants frontal face video was also recorded.

For a more thorough explanation of the dataset collection and its contents, see [1]

file listing

The following files are available (each explained in more detail below):

File name	Format	Part	Contents
Online_ratings	xls, csv, ods spreadsheet	Online self-assessment	All individual ratings from the online self-assessment.
Video_list	xls, csv, ods spreadsheet	Both parts	Names/YouTube links of the music videos used in the online self-assessment and the experiment + stats of the individual ratings from the online self-assessment.
Participant_ratings	xls, csv, ods spreadsheet	Experiment	All ratings participants gave to the videos during the experiment.
Participant_questionnaire	xls, csv, ods spreadsheet	Experiment	The answers participants gave to the questionnaire before the experiment.
Face_video	Zip file	Experiment	The frontal face video recordings from the experiment for participants 1-22.
Data_original	Zip file	Experiment	The original unprocessed physiological data recordings from the experiment in BioSemi .bdf format
Data_preprocessed	Zip file for Python and Matlab	Experiment	The preprocessed (downsampling, EOG removal, filtering, segmenting etc.) physiological data recordings from the experiment in Matlab and Python(numpy) formats

file details

online_ratings

This file contains all the individual video ratings collected during the online self-assessment. The file is available in Open-Office Calc (*online_ratings.ods*), Microsoft Excel (*online_ratings.xls*), and Comma-separated values (*online_ratings.csv*) formats.

The ratings were collected using an online self-assessment tool as described in [1]. Participants rated arousal, valence and dominance using SAM mannequins on a discrete 9-point scale. In addition, participants also rated the felt emotion using an emotion wheel (see [2]).

The table in the file has one row per individual rating and the following columns:

Column name	Description																
Online_id	The video id corresponding to the same column in the video_list file.																
Valence	The valence rating (integer between 1 and 9).																
Arousal	The arousal rating (integer between 1 and 9).																
Dominance	The dominance rating (integer between 1 and 9).																
Wheel_slice	The slice selected on the emotion wheel. For some participants the emotion wheel rating was not properly recorded. In these cases, the Wheel_slice value is 0. Otherwise, the mapping of emotions on the wheel to integers given here is: <table><tr><td>1. Pride</td><td>5. Relief</td><td>9. Sadness</td><td>13. Envy</td></tr><tr><td>2. Elation</td><td>6. Hope</td><td>10. Fear</td><td>14. Disgust</td></tr><tr><td>3. Joy</td><td>7. Interest</td><td>11. Shame</td><td>15. Contempt</td></tr><tr><td>4. Satisfaction</td><td>8. Surprise</td><td>12. Guilt</td><td>16. Anger</td></tr></table>	1. Pride	5. Relief	9. Sadness	13. Envy	2. Elation	6. Hope	10. Fear	14. Disgust	3. Joy	7. Interest	11. Shame	15. Contempt	4. Satisfaction	8. Surprise	12. Guilt	16. Anger
1. Pride	5. Relief	9. Sadness	13. Envy														
2. Elation	6. Hope	10. Fear	14. Disgust														
3. Joy	7. Interest	11. Shame	15. Contempt														
4. Satisfaction	8. Surprise	12. Guilt	16. Anger														
Wheel_strength	The strength selected on the emotion wheel (integer between 0=weak and 4=strong).																

Figure A.2: The description page of DEAP dataset (part 1)

video_list

This file lists all the videos used in the online self-assessment and in the experiment in a table. The file is available in Open-Office Calc (*video_list.ods*), Microsoft Excel (*video_list.xls*), and Comma-separated values (*video_list.csv*) formats.

The table has one row per video and the following columns:

Column name	Description
Online_id	The unique id used in the online self-assessment.
Experiment_id	If this video was selected for the experiment, this lists the unique id used in the experiment. Blank if not selected.
Lastfm_tag	If this video was selected via last.fm affective tags, this lists the affective tag. Blank otherwise.
Artist	The artist that recorded the song.
Title	Title of the song.
Youtube_link	The original youtube link where the video was downloaded. Note that due to copyright restrictions we are unable to provide the videos we used and these links may have been removed or may be unavailable in your country.
Highlight_start	The time in seconds where the extracted one-minute highlight begins as determined by MCA analysis. For some videos, the highlight was manually overridden (for instance when a section of the song is particularly well-known).
Num_ratings	The number of volunteers that rated this video in the online self-assessment
VAQ_Estimate	The valence/arousal quadrant this video was selected for by the experimenters. For each quadrant, 15 videos were selected by last.fm and 15 by manual selection. The quadrants are: 1. high arousal, high valence. 2. low arousal, high valence. 3. low arousal, low valence. 4. high arousal, low valence.
VAQ_Online	The valence/arousal quadrant as determined by the average ratings of the volunteers in the online self-assessment. Note that these can and sometimes do differ from the estimated quadrants.
AVG_x, STD_x, Q1_x, Q2_x, Q3_x	Average, standard deviation and first, second and third quartile of ratings x (Valence/Arousal/Dominance) by volunteers in the online self-assessment.

participant_ratings

This file contains all the participant video ratings collected during the experiment. The file is available in Open-Office Calc (*participant_ratings.ods*), Microsoft Excel (*participant_ratings.xls*), and Comma-separated values (*participant_ratings.csv*) formats.

The start_time values were logged by the presentation software. Valence, arousal, dominance and liking were rated directly after each trial on a continuous 9-point scale using a standard mouse. SAM Mannequins were used to visualize the ratings for valence, arousal and dominance. For liking (i.e. how much did you like the video?), thumbs up and thumbs down icons were used. Familiarity was rated after the end of the experiment on a 5-point integer scale (from "never heard it before" to "listen to it regularly"). Familiarity ratings are unfortunately missing for participants 2, 15 and 23.

The table in the file has one row per participant video rating and the following columns:

Column name	Column contents
Participant_id	The unique id of the participant (1-32).
Trial	The trial number (i.e. the presentation order).
Experiment_id	The video id corresponding to the same column in the <i>video_list</i> file.
Start_time	The starting time of the trial video playback in microseconds (relative to start of experiment).
Valence	The valence rating (float between 1 and 9).
Arousal	The arousal rating (float between 1 and 9).
Dominance	The dominance rating (float between 1 and 9).
Liking	The liking rating (float between 1 and 9).
Familiarity	The familiarity rating (integer between 1 and 5). Blank if missing.

participant_questionnaire

This file contains the participants' responses to the questionnaire filled in before the experiment. The file is available in Open-Office Calc (*participant_questionnaire.ods*), Microsoft Excel (*participant_questionnaire.xls*), and Comma-separated values (*participant_questionnaire.csv*) formats.

Most questions in the questionnaire were multiple-choice and speak pretty much for themselves. Participant 26 unfortunately failed to fill in the questionnaire. This questionnaire also contains the answers to the questions on the consent forms (can the data be used for research, can your imagery be published?).

Figure A.3: The description page of DEAP dataset (part 2)

face_video.zip

Face_video.zip contains the frontal face videos recorded in the experiment for the first 22 participants, segmented into trials. In the zip file, sXX/sXX_trial_YY.avi corresponds to the video for trial YY of subject XX.

For participants 3, 5, 11 and 14, one or several of the last trials are missing due to technical issues (i.e. the tape ran out). Please note that these **videos are in the order of presentation**, so the trial numbers do **not** correspond to the **Experiment_id** columns in the **video_list** file. The mapping between trial numbers and **Experiment_ids** can be found in the **participant_ratings** file.

Videos were recorded from a tripod placed behind the screen in DV PAL format using a SONY DCR-HC27E camcorder. The videos were then segmented according to the trials and transcoded to a 50 fps deinterlaced video using the h264 codec. The transcoding was done using the mencoder software with the following command:

```
mencoder sXX.dv -ss trialYY_start_second -endpos 59.05 -nosound -of avi -ovc x264  
-fps 50 -vf yadif=1:1,hqdn3d -x264encopts bitrate=50:subq=5:8x8dct:frameref=2:bframes=3  
-noskip -ofps 50 -o sXX_trialYY.avi
```

The synchronisation of the video is accurate to approximately 1/25 second (barring human error). Synchronisation was achieved by displaying a red screen before and after the experiment at the same time as a marker sent to the EEG recording PC. The onset frame of this screen was then manually marked in the video recording. Individual trial starting times were then calculated from the trial starting markers in the EEG recording.

data_original.zip

These are the original data recordings. There are 32 .bdf files (BioSemi's data format generated by the Actiview recording software), each with 48 recorded channels at 512Hz. (32 EEG channels, 12 peripheral channels, 3 unused channels and 1 status channel). The .bdf files can be read by a variety of software toolkits, including EEGLAB for Matlab and the BIOSIG toolkit.

The data was recorded in two separate locations. Participants 1-22 were recorded in Twente and participant 23-32 in Geneva. Due to a different revision of the hardware, there are some minor differences in the format. First, the order of EEG channels is different for the two locations. Second, the GSR measure is in a different format for each location.

The table below gives the EEG channel names (according to the 10/20 system) for both locations and the indices that can be used to convert one ordering to the other:

Channel no.	Ch. name Twente	Ch. name Geneva	Geneva > Twente	Twente > Geneva
1	Fp1	Fp1	1	1
2	AF3	AF3	2	2
3	F7	F3	4	4
4	F3	F7	3	3
5	FC1	FC5	6	6
6	FC5	FC1	5	5
7	T7	C3	8	8
8	C3	T7	7	7
9	CP1	CP5	10	10
10	CP5	CP1	9	9
11	P7	P3	12	12
12	P3	P7	11	11
13	Pz	PO3	16	14
14	PO3	O1	13	15
15	O1	Oz	14	16
16	Oz	Pz	15	13

Figure A.4: The description page of DEAP dataset (part 3)

data_preprocessed_matlab.zip and data_preprocessed_python.zip

These files contain a downsampled (to 128Hz), preprocessed and segmented version of the data in Matlab (data_preprocessed_matlab.zip) and pickled python/numpy (data_preprocessed_python.zip) formats. This version of the data is well-suited to those wishing to quickly test a classification or regression technique without the hassle of processing all the data first. Each zip file contains 32 .dat (python) or .mat (matlab) files, one per participant. Some sample code to load a python datafile is below:

```
import cPickle  
x = cPickle.load(open('s01.dat', 'rb'))
```

Each participant file contains two arrays:

Array name	Array shape	Array contents
data	40 x 40 x 8064	video/trial x channel x data
labels	40 x 4	video/trial x label (valence, arousal, dominance, liking)

The videos are in the order of Experiment_id, so not in the order of presentation. This means the first video is the same for each participant. The following table shows the channel layout and the preprocessing performed:

Channel no.	Channel content	Preprocessing
1	Fp1	
2	AF3	
3	F3	
4	F7	
5	FC5	
6	FC1	
7	C3	
8	T7	
9	CP5	
10	CP1	
11	P3	
12	P7	
13	PO3	
14	O1	1. The data was downsampled to 128Hz. 2. EOG artefacts were removed as in [1]. 3. A bandpass frequency filter from 4.0-45.0Hz was applied. 4. The data was averaged to the common reference. 5. The EEG channels were reordered so that they all follow the Geneva order as above. 6. The data was segmented into 60 second trials and a 3 second pre-trial baseline removed. 7. The trials were reordered from presentation order to video (Experiment_id) order.
15	Oz	
16	Pz	
17	Fp2	
18	AF4	
19	Fz	
20	F4	
21	F8	
22	FC6	
23	FC2	
24	Cz	
25	C4	
26	T8	
27	CP6	
28	CP2	
29	P4	
30	P8	
31	PO4	
32	O2	
33	hEOG (horizontal EOG, hEOG ₁ - hEOG ₂)	
34	vEOG (vertical EOG, vEOG ₁ - vEOG ₂)	
35	zEMG (Zygomaticus Major EMG, zEMG ₁ - zEMG ₂)	
36	tEMG (Trapezius EMG, tEMG ₁ - tEMG ₂)	1. The data was downsampled to 128Hz. 2. The data was segmented into 60 second trials and a 3 second pre-trial baseline removed. 3. The trials were reordered from presentation order to video (Experiment_id) order.
37	GSR (values from Twente converted to Geneva format (Ohm))	
38	Respiration belt	
39	Plethysmograph	
40	Temperature	

Figure A.5: The description page of DEAP dataset (part 4)

DEAPdataset

a dataset for emotion analysis using eeg, physiological and video signals

home

dataset description

download

contact

dataset access

To gain access to the dataset and download the files on this page, please download the EULA (End User License Agreement) below. The EULA should be printed, signed, scanned and returned via email to i.patras@qmul.ac.uk with the subject line "DEAP account request". Please, send the request using your institutional email, i.e. not your yahoo, googlemail, etc account, unless of course you work in Yahoo, Google. Please state in your email your position and your institution. Upon receipt, a username and password will be issued that can be used to download the data files below.



The following files are available. Please see the [dataset description](#) for a thorough explanation of what is in each file.

metadata

All metadata contained in four spreadsheets ([online_ratings](#), [video_list](#), [participant_ratings](#) and [participant_questionnaire](#)). Available in the following formats:



physiological recordings

Recordings of EEG and peripheral physiological signals. Three formats are available: The original unprocessed recordings in BioSemi's .bdf format, preprocessed recordings in Matlab and Python(numpy) formats (see the [dataset description](#) for more information).



Several people have reported problems downloading these large files using download managers. If you face this problem, there is also a [multi-volume version of these files](#), where the files are split up into 100mb parts.

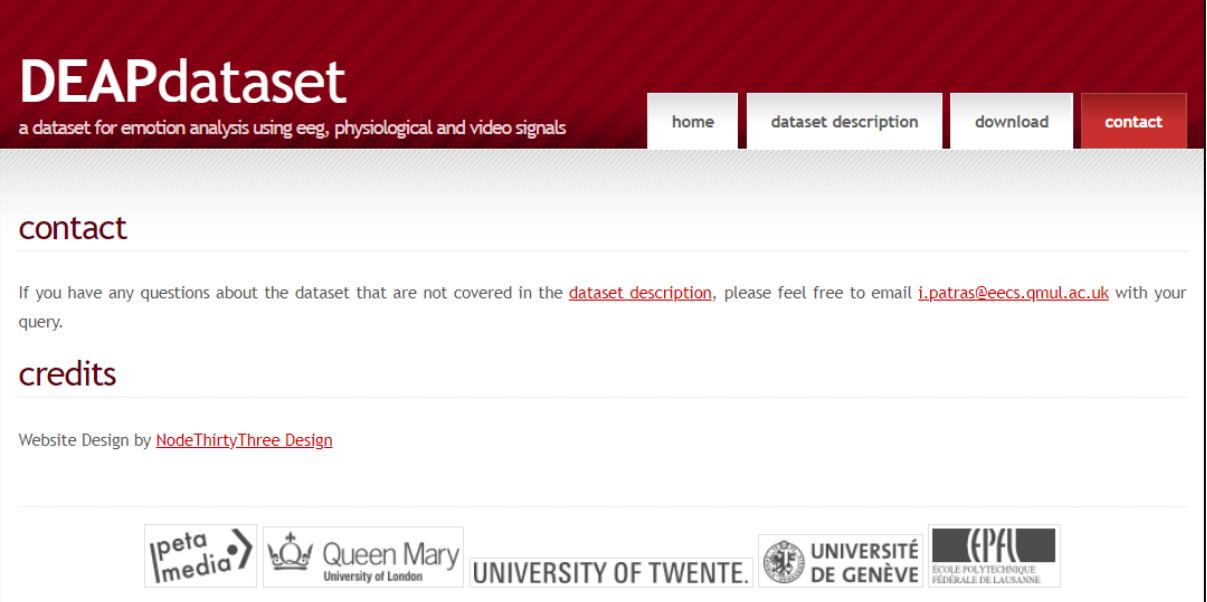
video recordings

The frontal face video recordings from the experiment for participants 1-22 in h264 format.



Several people have reported problems downloading this large file using download managers. If you face this problem, there is also a [multi-volume version of the file](#), where the files are split up into 100mb parts.

Figure A.6: The download page of DEAP dataset



The screenshot shows the 'contact' page of the DEAPdataset website. The header includes the dataset title, a subtitle 'a dataset for emotion analysis using eeg, physiological and video signals', and navigation links for 'home', 'dataset description', 'download', and 'contact'. The main content area has a heading 'contact' and a text block stating: 'If you have any questions about the dataset that are not covered in the [dataset description](#), please feel free to email i.patras@eecs.qmul.ac.uk with your query.' Below this is a section titled 'credits' with a note: 'Website Design by [NodeThirtyThree Design](#)'. At the bottom, there are logos for ipeta media, Queen Mary University of London, UNIVERSITY OF TWENTE, UNIVERSITÉ DE GENÈVE, and EPFL.

Figure A.7: The contact information of DEAP dataset

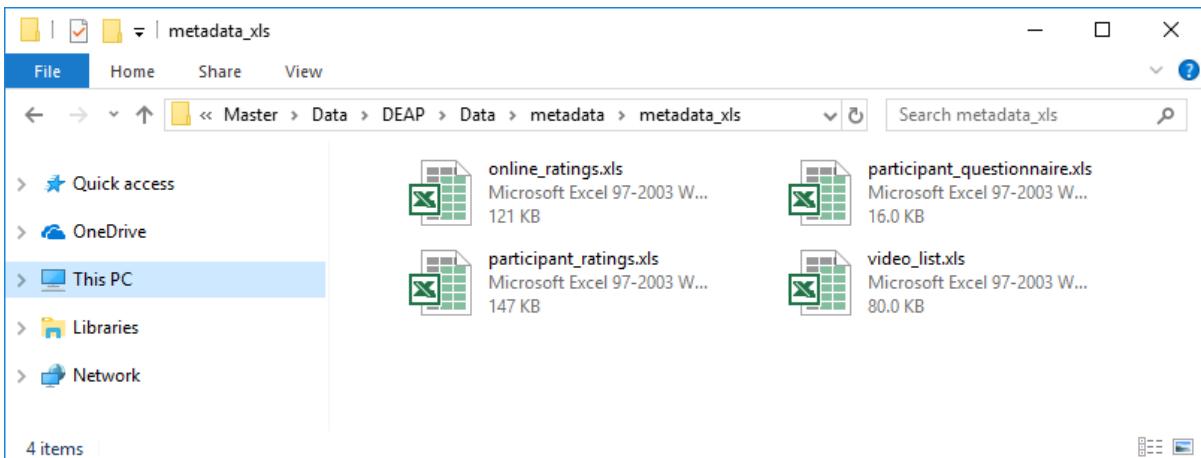


Figure A.8: Metadata spreadsheet files in Microsoft Excel (.xls) format

video_list.xls - Compatibility Mode - Excel

Mohammed Abdelaal

The screenshot shows a Microsoft Excel spreadsheet with the title bar 'video_list.xls - Compatibility Mode - Excel' and the name 'Mohammed Abdelaal'. The spreadsheet contains a single sheet named 'Sheet1'. The data is organized into columns A through F. Column A contains row numbers from 1 to 29. Columns B and C contain 'Experiment_id' values (e.g., 1, 2, 3, 4, etc.) and 'Lastfm_tag' values (e.g., fun, exciting, joy, etc.). Columns D, E, and F contain 'Artist' names, 'Title' names, and 'Youtube_link' URLs respectively. The data includes various artists like Mariana Torrini, Scotty Doesn't Know, Jackson 5, and many others, along with their corresponding song titles and YouTube video links.

	A	B	C	D	E	F	
1	Online_id	Experiment_id	Lastfm_tag	Artist	Title	Youtube_link	High
2	2	1	fun	Emiliana Torrini	Jungle Drum	http://www.youtube.com/watch?v=iZ9vkd7Rp-g	
3	7	2	exciting	Lustra	Scotty Doesn't Know	http://www.youtube.com/watch?v=51ncDQYxsm8	
4	15	3	joy	Jackson 5	Blame It On The Boogie	http://www.youtube.com/watch?v=nb1u7wmKywM	
5	62	4		The B52's	Love Shack	http://www.youtube.com/watch?v=leohcmf8kM	
6	63	5		Blur	Song 2	http://www.youtube.com/watch?v=WIAHZURxRjY	
7	70	6		Blink 182	First Date	http://www.youtube.com/watch?v=vYy9lppg1m8	
8	72	7		Benny Benassi	Satisfaction	http://www.youtube.com/watch?v=eoRIVwFP02s	
9	84	8		Lily Allen	Fuck You	http://www.youtube.com/watch?v=S0zMHF7J15g	
10	88	9		Queen	I Want To Break Free	http://www.youtube.com/watch?v=EVYgRPfC9nQ	
11	118	10		Rage Against The Machine	Bombtrack	http://www.youtube.com/watch?v=Tu1wAP2Baco	
12	9	11	happy	Michael Franti & Spearhead	Say Hey (I Love You)	http://www.youtube.com/watch?v=eoaTl7lcFs8	
13	17	12	cheerful	Grand Archives	Miniature Birds	http://www.youtube.com/watch?v=_iEnN9ip1Qk	
14	24	13	love	Bright Eyes	First Day Of My Life	http://www.youtube.com/watch?v=zwfS69nA-1w	
15	27	14	happy	Jason Mraz	I'm Yours	http://www.youtube.com/watch?v=EkHTsc9PU2A	
16	29	15	lovely	Bishop Allen	Butterfly Nets	http://www.youtube.com/watch?v=B8el64H1Cqk	
17	37	16	sentimental	The Submarines	Darkest Things	http://www.youtube.com/watch?v=ijLkoqN5_EY	
18	76	17		Air	Moon Safari	http://www.youtube.com/watch?v=kxWfyTg6mc	
19	80	18		Louis Armstrong	What A Wonderful World	http://www.youtube.com/watch?v=3orLNBS2zbU	
20	83	19		Manu Chao	Me Gustas Tu	http://www.youtube.com/watch?v=mzgjiPBCss	
21	85	20		Taylor Swift	Love Story	http://www.youtube.com/watch?v=8xg3vE8le_E	
22	108	21		Diamanda Galas	Gloomy Sunday	http://www.youtube.com/watch?v=A-BSL5Av89w	
23	31	22	sentimental	Porcupine Tree	Normal	http://www.youtube.com/watch?v=zLDPhPr5lg	
24	33	23	melancholy	Wilco	How To Fight Loneliness	http://www.youtube.com/watch?v=wVyygTKDcOE	
25	41	24	sad	James Blunt	Goodbye My Lover	http://www.youtube.com/watch?v=EDEEZ7Ov2k	
26	44	25	depressing	A Fine Frenzy	Goodbye My Almost Lover	http://www.youtube.com/watch?v=G-k19OCq7vE	
27	45	26	mellow	Kings Of Convenience	The Weight Of My Words	http://www.youtube.com/watch?v=15kWITrp5k	
28	95	27		Madonna	Rain	http://www.youtube.com/watch?v=nhPcYgnOn4Y	
29	96	28		Sia	Breathe Me		

Figure A.9: Metadata spreadsheet of the music clubs

participant_ratings.xls - Compatibility Mode - Excel

Mohammed Abdelaal

The screenshot shows a Microsoft Excel spreadsheet with the title bar 'participant_ratings.xls - Compatibility Mode - Excel' and the name 'Mohammed Abdelaal'. The spreadsheet contains a single sheet named 'Sheet1'. The data is organized into columns A through I. Column A contains 'Participant_id' values (e.g., 1, 2, 3, 4, etc.). Columns B through I contain rating scores for different dimensions: Trial, Experiment_id, Start_time, Valence, Arousal, Dominance, Liking, and Familiarity. The data shows multiple rows for each participant, indicating repeated measurements over time.

	A	B	C	D	E	F	G	H	I
1	Participant_id	Trial	Experiment_id	Start_time	Valence	Arousal	Dominance	Liking	Familiarity
2	1	1	5	1695918	6.96	3.92	7.19	6.05	4
3		1	2	18	2714905	7.23	7.15	6.94	8.01
4		1	3	4	3586768	4.94	6.01	6.12	8.06
5		1	4	24	4493800	7.04	7.09	8.01	8.22
6		1	5	20	5362005	8.26	7.91	7.19	8.13
7		1	6	31	6176062	3.03	8.14	2.86	8.04
8		1	7	40	7138735	5.1	7.12	6.17	5.97
9		1	8	39	8081417	3.24	6.18	7.87	6.15
10		1	9	13	8960934	1.95	3.12	2.87	6.18
11		1	10	33	9816492	3.81	3.85	4.78	5.13
12		1	11	6	10696693	8.27	3.92	7	8.03
13		1	12	10	11559734	1.99	4.86	2.04	7.09
14		1	13	35	12593050	2.06	8.15	8.05	5.18
15		1	14	22	13429432	7.09	2.08	7.06	7.37
16		1	15	15	14230827	3.17	8.08	2.91	5.04
17		1	16	17	15040219	2.46	6.91	6.77	6.41
18		1	17	36	15852443	2.9	6.92	6.5	3.87
19		1	18	34	16671331	2.28	7.09	7.28	6.92
20		1	19	14	17538704	4.18	2.24	3.04	5.04
21		1	20	27	18394413	7.35	6.95	7.03	7.29
22		1	21	11	21438533	2.00	2.36	3.63	6.24

Figure A.10: Metadata spreadsheet of participants' ratings

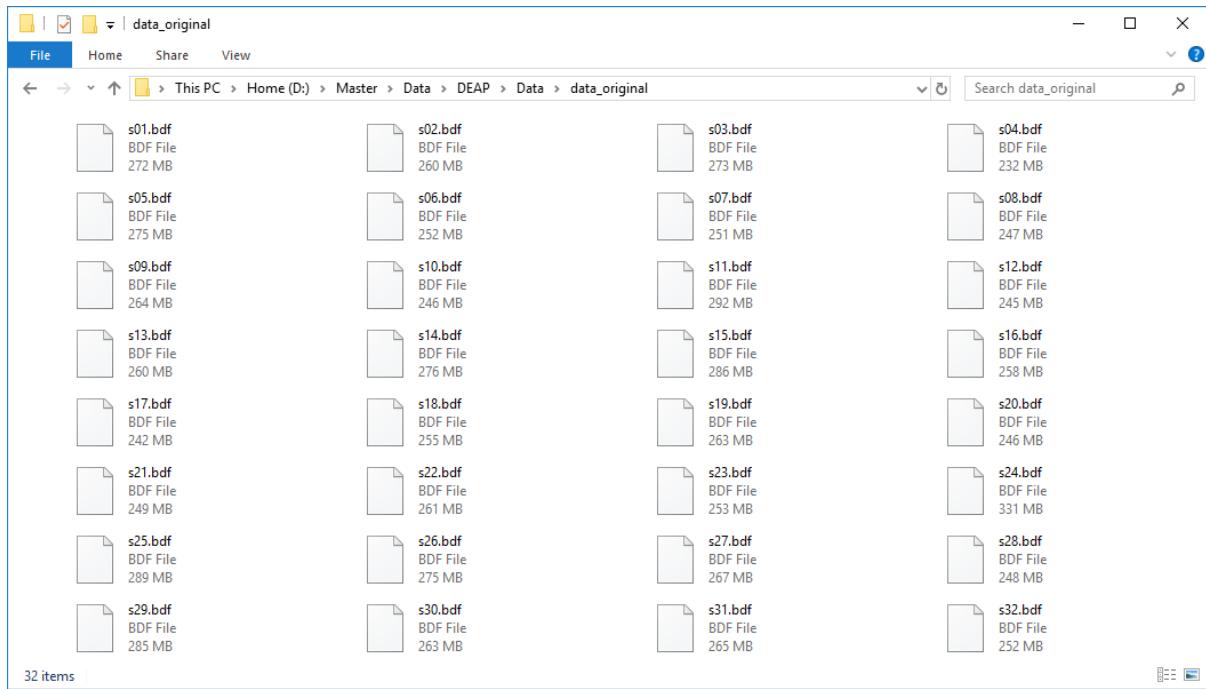


Figure A.11: Files of the unprocessed signals in BioSemi (.bdf) format

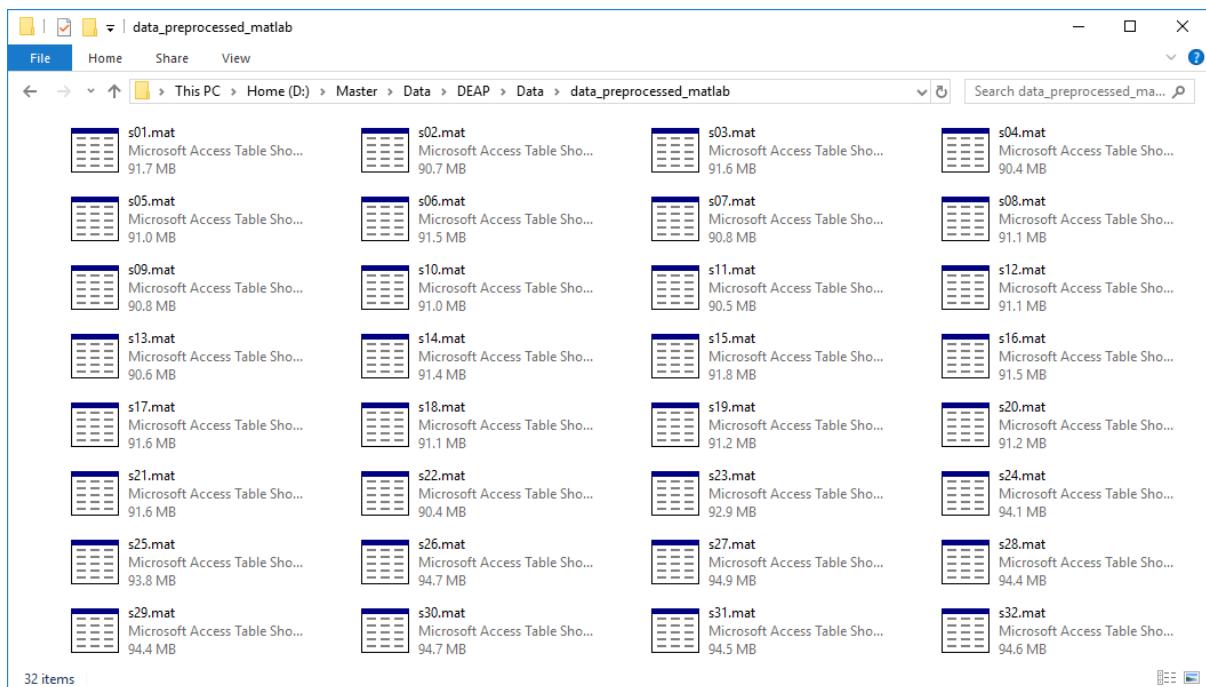


Figure A.12: Files of the preprocessed signals in MATLAB (.mat) format

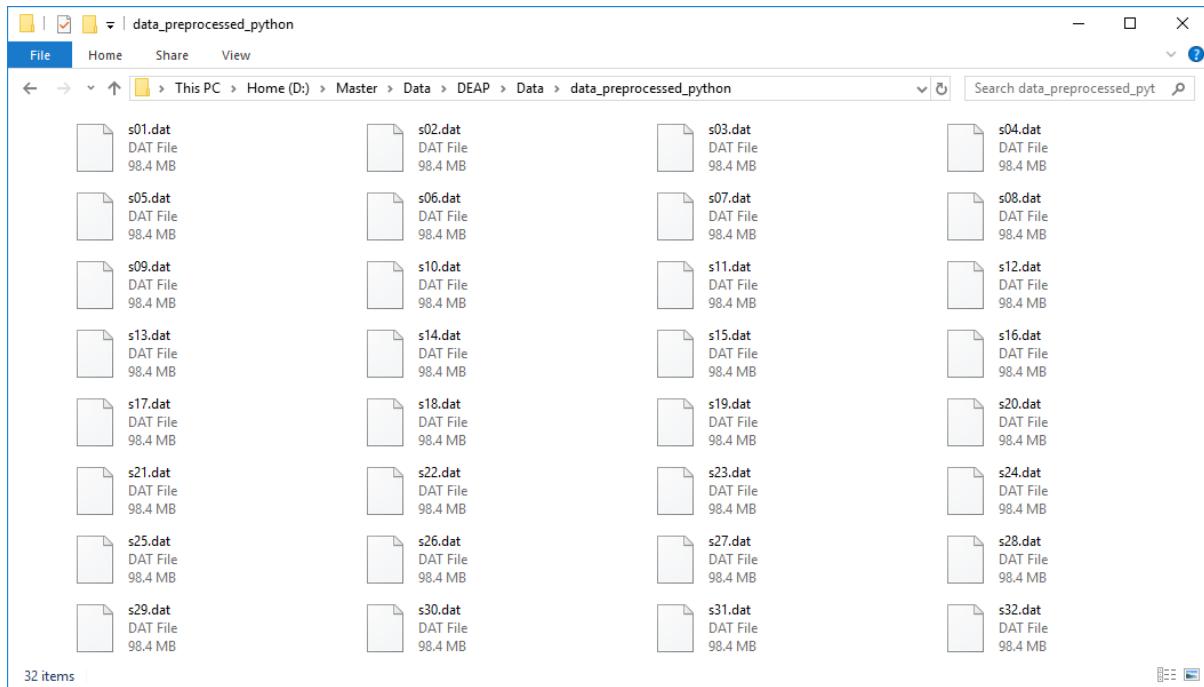


Figure A.13: Files of the preprocessed signals in Python (numpy) format

Time	Fp1	AF3	F3	F7	FC5	FC1	C3	T7	CP5	CP1	P3	P7	PO3	O1	Oz
0.898 s	-1.415	-0.852	0.068	-4.780	-2.603	-0.593	0.222	-3.001	0.041	0.108	1.130	1.180	0.733	-0.671	-3.6
0.906 s	0.506	1.812	2.624	-1.373	-1.670	0.172	-0.610	-1.164	-1.575	-1.355	-0.187	-1.087	1.735	1.498	-2.6
0.914 s	1.734	4.093	3.600	2.819	-2.347	-2.159	0.008	2.153	-0.330	0.864	-1.313	-1.538	4.970	5.416	1.3
0.922 s	3.051	4.609	4.525	4.363	-0.885	-1.830	2.321	5.291	2.391	2.702	-1.287	-1.498	4.476	4.184	2.4
0.930 s	4.498	3.601	4.760	4.396	2.527	1.853	2.421	2.063	1.400	-0.366	-0.953	-2.072	-0.589	-0.395	1.3
0.938 s	2.653	1.691	3.016	1.537	1.280	2.533	-0.882	-3.627	-3.373	-4.242	-3.026	-3.170	-5.404	-1.280	1.5
0.945 s	-0.839	-0.854	0.632	-1.570	-2.469	0.283	-3.089	-2.590	-5.453	-3.753	-5.186	-1.714	-5.711	0.526	2.5
0.953 s	-2.666	-2.163	-0.502	-0.473	-1.777	-1.478	-0.816	2.231	-1.719	-0.386	-2.960	2.431	-2.104	0.345	2.1
0.961 s	-3.227	-2.031	-0.290	0.414	1.924	-0.023	3.449	1.234	2.803	2.386	2.198	3.196	-0.202	-1.126	0.5
0.969 s	-2.354	-2.664	-0.203	-0.545	2.925	2.179	4.905	-1.385	4.315	3.216	4.125	1.569	-0.174	0.958	0.2
0.977 s	-1.076	-2.902	-0.705	0.897	0.905	-0.268	2.701	2.190	3.977	1.710	1.495	2.267	1.236	4.019	0.9
0.984 s	-0.825	-0.794	-0.506	2.690	0.170	-1.959	-0.723	4.247	1.390	-1.541	-0.698	1.791	-0.107	0.638	-1.4
0.992 s	0.995	1.730	1.345	3.592	1.892	1.283	-2.895	1.757	-1.498	-3.233	-0.934	-0.337	-3.461	-3.488	-4.6
1.000 s	2.901	2.099	2.852	4.797	2.902	0.583	-2.076	3.865	0.395	-0.866	-0.499	0.798	-0.922	-0.393	-3.1
1.008 s	0.896	0.215	1.134	2.770	1.808	-3.813	-0.275	6.152	3.462	1.927	1.336	2.441	3.501	2.371	0.9
1.016 s	-0.963	-1.566	-1.317	-1.008	-0.728	-2.837	-1.166	0.480	1.354	1.465	1.338	0.622	2.783	0.598	2.8
1.023 s	-0.486	-1.673	-1.125	-2.766	-3.787	0.621	-3.145	-5.392	-2.934	-0.098	-2.452	-1.974	0.301	0.046	2.6
1.031 s	-1.025	-0.492	-0.584	-4.319	-4.520	1.611	-2.656	-5.499	-5.505	-1.598	-4.237	-4.368	-2.444	-1.300	0.6
1.039 s	-1.134	1.641	-0.246	-3.402	-2.249	2.019	0.294	-1.807	-4.643	-2.688	-2.745	-5.573	-3.970	-4.335	-2.0
1.047 s	-1.596	1.146	0.228	-0.458	-0.098	1.602	4.052	3.851	1.852	-0.118	0.051	0.352	1.303	-1.612	-1.7
1.055 s	-4.170	-3.425	-3.022	-1.497	1.733	-0.329	5.552	5.483	7.463	3.214	5.681	9.638	7.667	4.047	1.5
1.063 s	-2.191	-4.145	-6.890	-1.715	3.880	-2.107	2.231	0.925	4.387	1.185	8.188	8.039	6.211	3.685	3.3

Figure A.14: The EEG signals of a participant while watching a music video

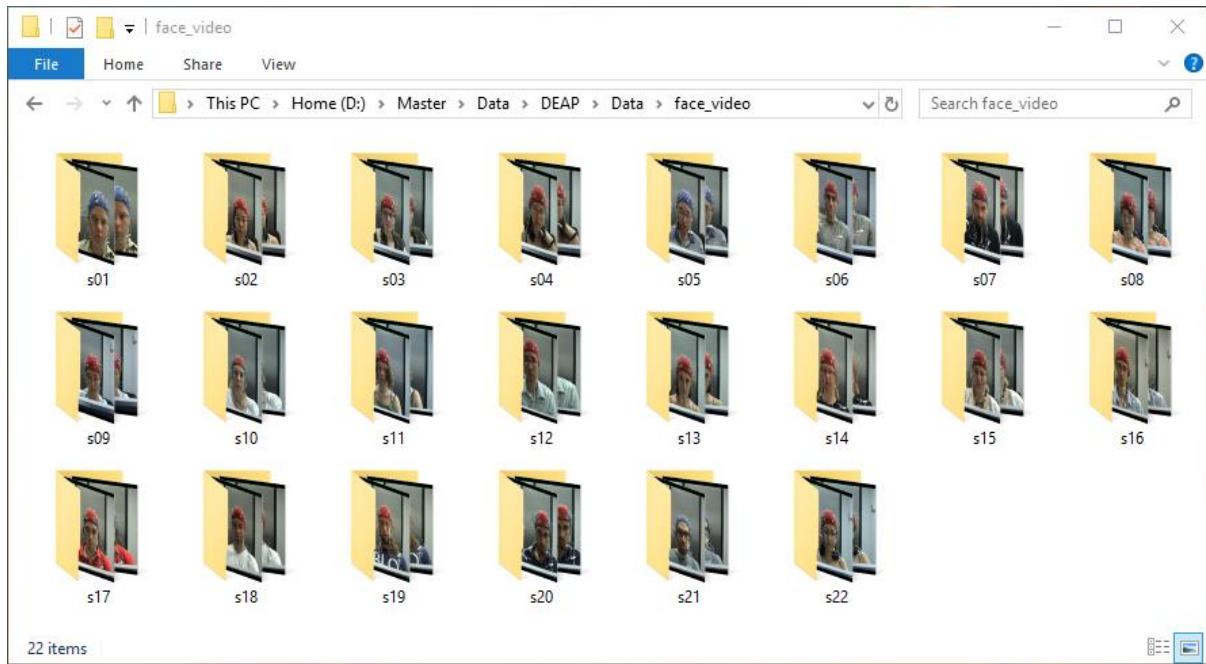


Figure A.15: Folders contain frontal face videos for participants 1-22

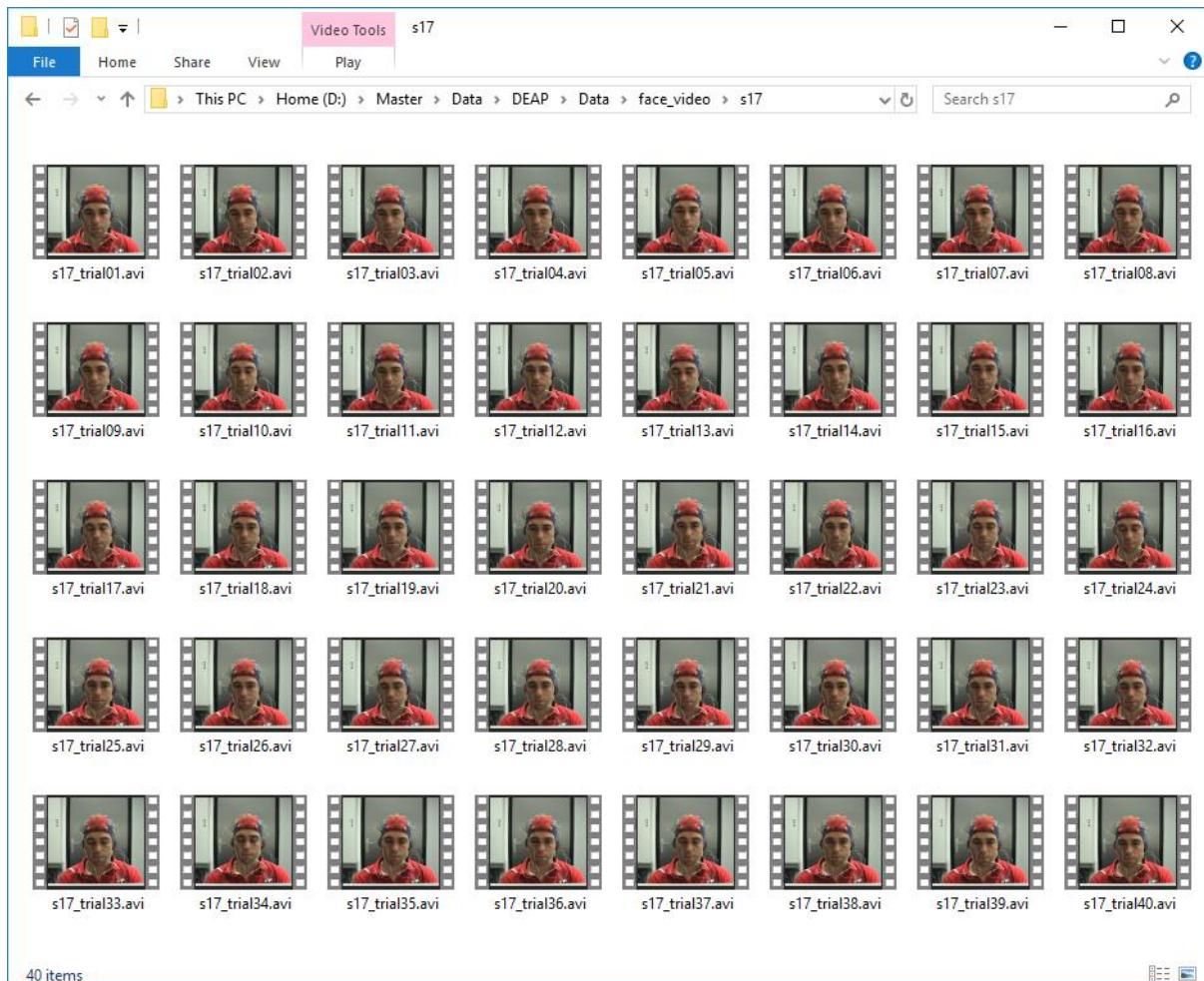


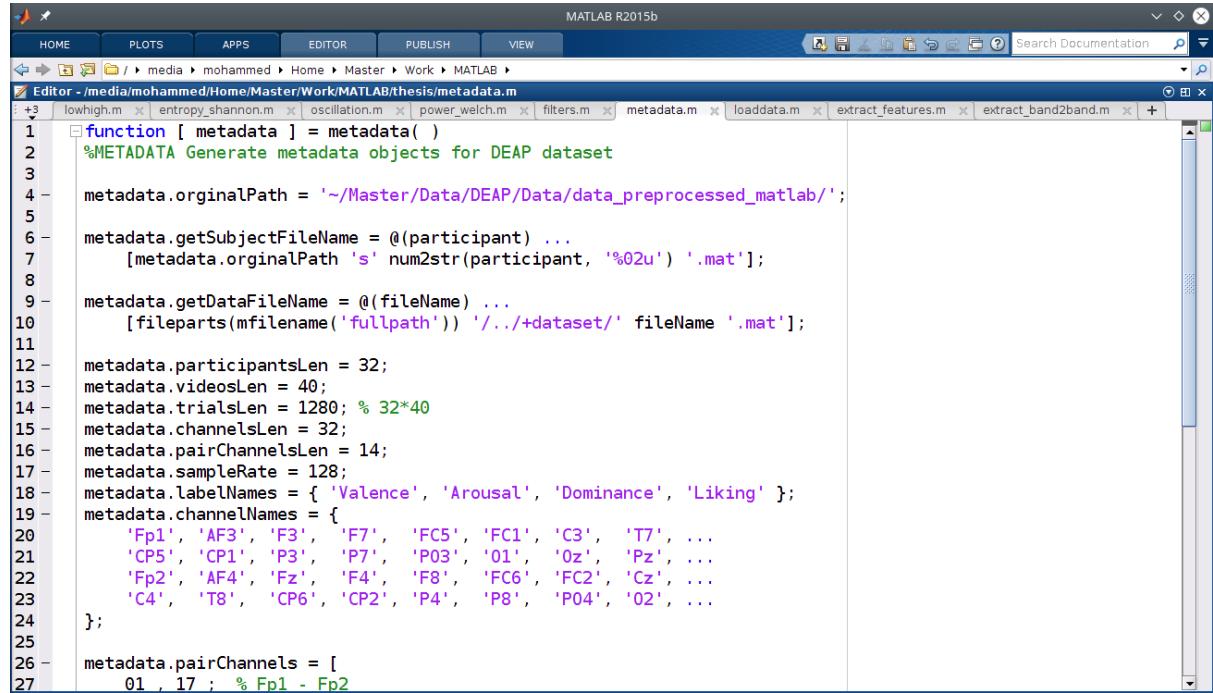
Figure A.16: Files of the frontal face videos of the 17th participant



Figure A.17: The frontal face video of the 17th Participant while doing the experiment

APPENDIX B: MATLAB SCRIPTS

Samples of implemented MATLAB scripts using signal processing toolbox.



The screenshot shows the MATLAB R2015b interface with the 'Editor' tab selected. The current file is 'metadata.m'. The code in the editor is as follows:

```
function [ metadata ] = metadata( )
%METADATA Generate metadata objects for DEAP dataset

metadata.orginalPath = '~/Master/Data/DEAP/Data/data_preprocessed_matlab/';

metadata.getSubjectFileName = @(participant) ...
    [metadata.orginalPath 's' num2str(participant, '%02u') '.mat'];

metadata.getDataFileName = @(fileName) ...
    [fileparts(fullfile('fullpath')) '/../+dataset/' fileName '.mat'];

metadata.participantsLen = 32;
metadata.videosLen = 40;
metadata.trialsLen = 1280; % 32*40
metadata.channelsLen = 32;
metadata.pairChannelsLen = 14;
metadata.sampleRate = 128;
metadata.labelNames = { 'Valence', 'Arousal', 'Dominance', 'Liking' };
metadata.channelNames = [
    'Fp1', 'AF3', 'F3', 'F7', 'FC5', 'FC1', 'C3', 'T7', ...
    'CP5', 'CP1', 'P3', 'P7', 'P03', 'O1', 'Oz', 'Pz', ...
    'Fp2', 'AF4', 'Fz', 'F4', 'F8', 'FC6', 'FC2', 'Cz', ...
    'C4', 'T8', 'CP6', 'CP2', 'P4', 'P8', 'P04', 'O2', ...
];
metadata.pairChannels = [
    01 , 17 ; % Fp1 - Fp2
```

Figure B.1: Creating a metadata object for the dataset (part 1)

The screenshot shows the MATLAB R2015b interface with the 'EDITOR' tab selected. The current file is 'metadata.m'. The code defines a 'pairChannels' matrix and 'males' and 'females' arrays, then reshapes and permutes the data.

```

24    };
25
26    metadata.pairChannels = [
27        01 , 17 ; % Fp1 - Fp2
28        02 , 18 ; % AF3 - AF4
29        03 , 20 ; % F3 - F4
30        04 , 21 ; % F7 - F8
31        05 , 22 ; % FC5 - FC6
32        06 , 23 ; % FC1 - FC2
33        07 , 25 ; % C3 - C4
34        08 , 26 ; % T7 - T8
35        09 , 27 ; % CP5 - CP6
36        10 , 28 ; % CP1 - CP2
37        11 , 29 ; % P3 - P4
38        12 , 30 ; % P7 - P8
39        13 , 31 ; % P03 - P04
40        14 , 32 ; % O1 - O2
41    ];
42
43    metadata.males = [1,5,6,7,12,16,17,18,19,20,21,23,26,27,28,29,30];
44    metadata.females = [2,3,4,8,9,10,11,13,14,15,22,24,25,31,32];
45
46    metadata.reshapedata = @(data) reshape(data, [], metadata.videosLen, metadata.participantsLen);
47    metadata.reformatdata = @(data) permute(metadata.reshapedata(data), [2 1 3]);
48
49 end
50

```

Figure B.2: Creating a metadata object for the dataset (part 2)

The screenshot shows the MATLAB R2015b interface with the 'EDITOR' tab selected. The current file is 'filters.m'. The code designs five frequency band filters: Delta, Theta, Alpha, Beta, and Gamma.

```

1 function [ filters ] = filters( sampleRate, filterOrder )
2 %FILTERS create a filter for each frequency band
3
4 if nargin < 3
5     filterOrder = sampleRate - 1;
6 end
7
8 filters = cell(1, 5);
9
10 %% Delta filter
11 filters{1} = designfilt('lowpassfir', 'Window', 'hamming', 'FilterOrder', filterOrder, ...
12 'SampleRate', sampleRate, 'CutoffFrequency', 4);
13 %% Theta filter
14 filters{2} = designfilt('bandpassfir', 'Window', 'hamming', 'FilterOrder', filterOrder, ...
15 'SampleRate', sampleRate, 'CutoffFrequency1', 4,'CutoffFrequency2', 8);
16 %% Alpha filter
17 filters{3} = designfilt('bandpassfir', 'Window', 'hamming', 'FilterOrder', filterOrder, ...
18 'SampleRate', sampleRate, 'CutoffFrequency1', 8, 'CutoffFrequency2', 12);
19 %% Beta filter
20 filters{4} = designfilt('bandpassfir', 'Window', 'hamming', 'FilterOrder', filterOrder, ...
21 'SampleRate', sampleRate, 'CutoffFrequency1', 12, 'CutoffFrequency2', 30);
22 %% Gamma filter
23 filters{5} = designfilt('highpassfir', 'Window', 'hamming', 'FilterOrder', filterOrder, ...
24 'SampleRate', sampleRate, 'CutoffFrequency', 30);
25 end
26

```

Figure B.3: Design the five major frequency band filters

The screenshot shows the MATLAB R2015b interface with the Editor window open. The current file is 'power_welch.m'. The code defines a function to compute the spectral power of a signal using the Welch method.

```
function [ y ] = power_welch( x, sampleRate, freqrange )
%POWER_WELCH compute the spectral power of a signal
if nargin < 3
    freqrange = [0, sampleRate/2];
end
y = bandpower(x, sampleRate, freqrange);
end
```

Figure B.4: Compute the spectral power of a signal

The screenshot shows the MATLAB R2015b interface with the Editor window open. The current file is 'entropy_shannon.m'. The code defines a function to compute the Shannon entropy of a signal.

```
function [ y ] = entropy_shannon( x )
%ENTROPY_SHANNON compute the shannon entropy of a signal
%% get probability of signal values
px = histcounts(x, 'Normalization', 'probability');

%% skip 0 values (for log to work)
px = px(px~=0);

%% compute shannon entropy
y = -sum(px .* log(px));
end
```

Figure B.5: Compute Shannon entropy of a signal

The screenshot shows the MATLAB R2015b interface with the 'Editor' window open. The current file is 'oscillation.m'. The code calculates the oscillation feature of a signal 'x'. It initializes variables N, Lmin, and Lmax to 0. A for loop iterates from 1 to N-2. Inside the loop, two if statements check for local minima and maxima. If a local minimum is found, Lmin is incremented. If a local maximum is found, Lmax is incremented. After the loop, an if statement checks if both Lmin and Lmax are greater than 0. If true, y is set to N divided by the sum of Lmin and Lmax. Otherwise, y is set to 0.

```
function [ y ] = oscillation( x )
%OSCILLATION compute the oscillation of a signal

N = length(x);
Lmin = 0;
Lmax = 0;

for t = 1:(N-2)
    if x(t) > x(t+1) && x(t+2) > x(t+1)
        Lmin = Lmin + 1;
    end
    if x(t) < x(t+1) && x(t+2) < x(t+1)
        Lmax = Lmax + 1;
    end
end

if Lmin + Lmax > 0
    y = N / (Lmin + Lmax);
else
    y = 0;
end
```

Figure B.6: Compute the oscillation feature of a signal

```

1 function extract_features()
2 %Extract EEG features from DEAP dataset
3
4 metadata = dataset.metadata();
5 rhythmRangs = { [0,4], [4,8], [8,12], [12,30], [30,64] };
6 filters = signal.filters(metadata.sampleRate, rhythmRangs);
7 data = zeros(metadata.channelsLen, length(rhythmRangs), ...
8               metadata.videosLen, metadata.participantsLen);
9
10 %% extract data
11 for pId = 1:metadata.participantsLen
12     fileName = metadata.getSubjectFileName(pId);
13     subject = load(fileName, 'data'); % data => subject.data
14     for vId = 1:metadata.videosLen
15         for cId = 1:metadata.channelsLen
16             eegSig = squeeze(subject.data(vId, cId, :));
17             %% extract from 4th to 63rd second (skip 1st three seconds)
18             eegSig = eegSig((3*metadata.sampleRate+1):end);
19             data(cId, :, vId, pId) = featcalc...
20                 (eegSig, metadata.sampleRate, filters, rhythmRangs);
21         end
22     end
23 end
24
25 %% save data into files
26 save(metadata.getDataFileName('data32'), 'data');
27 end

```

Figure B.7: Extract a specific feature from EEG signals (part 1)

```

28
29 function [ y ] = featcalc( x, fs, filters, rhythmRangs ) %#ok<INUSL>
30 % y = calcsigfeat(x, @mean, filters);
31 % y = calcsigfeat(x, @std, filters);
32 % y = calcsigfeat(x, @var, filters);
33 % y = calcsigfeat(x, @skewness, filters);
34 % y = calcsigfeat(x, @kurtosis, filters);
35 % y = calcsigfeat(x, @signal.hjorth_mobility, filters);
36 % y = calcsigfeat(x, @signal.hjorth_complexity, filters);
37 % y = calcsigfeat(x, @signal.oscillation, filters);
38 % y = calcsigfeat(x, @signal.peak2peak, filters);
39 % y = calcsigfeat(x, @signal.entropy_shannon, filters);
40 % y = calcsigfeat(fft(x), @signal.entropy_spectral, rhythmRangs, fs);
41 % y = calcsigfeat(x, @signal.energy, filters); % absolute energy
42 % y = calcsigfeat(x, @signal.power_welch, rhythmRangs, fs); % absolute power
43 y = log(calcsigfeat(x, @signal.power_welch, rhythmRangs, fs)); % log of power
44 end
45
46 function y = calcsigfeat( x, func, filters, fs )
47 L = length(filters); y = zeros(L, 1);
48 for r = 1:L
49     if nargin==4, y(r) = func( x, fs, filters{r} );
50     else, y(r) = func( filtfilt(filters{r}, x) );
51     end
52     if isnan(y(r)), error('# Feature value is NaN !!!'); end
53 end
54 end

```

Figure B.8: Extract a specific feature from EEG signals (part 2)

```

1  %%> get metadata of DEAP
2  metadata = dataset.metadata();
3
4  %%> set data files
5  fileName = '+welch/+psd/+bands/+32/power_log';
6
7  %%> load data
8  channels = load(metadata.getDataFileName(fileName));
9
10 %%> Calc data of pair channels
11 pair1Data = channels.data(metadata.pairChannels(:, 1), :, :, :);
12 pair2Data = channels.data(metadata.pairChannels(:, 2), :, :, :);
13
14 data = pair1Data - pair2Data;
15
16 info = channels.info;
17 info{4, 2} = '14 pairs of channels';
18
19 %%> save data
20 fileName = strrep(fileName, '+32/', '+14/');
21 save(metadata.getDataFileName(fileName), 'data', 'info');
22

```

Figure B.9: Compute the symmetrical pair of electrodes features

```

1  %%> get metadata of DEAP
2  metadata = dataset.metadata();
3
4  b1 = 6; % beta index
5  b2 = 5; % alpha index
6
7  %%> load data
8  fileName = '+welch/+psd/+bands/+32/power_log';
9  bands = load(metadata.getDataFileName(fileName));
10
11 %%> calc band to band features
12 data = bands.data(:, b1, :, :) ./ bands.data(:, b2, :, :);
13
14 %%> set data info
15 info = bands.info;
16 info{2, 2} = 'Beta to Alpha';
17
18 %%> save data
19 fileName = strrep(fileName, '+32/', '+b2b/');
20 fileName = [fileName '_beta2alpha'];
21
22 save(metadata.getDataFileName(fileName), 'data', 'info');
23

```

Figure B.10: Compute the band to band features

APPENDIX C: SCIKIT-LEARN TOOLKIT

Brief description of scikit-learn toolkit for Python scripts for feature reduction and emotion classification, in addition to samples of implemented Python scripts using scikit-learn toolkit.

The screenshot shows the official website for scikit-learn. At the top, there is a navigation bar with links for Home, Installation, Documentation, Examples, Google Custom Search, and a search icon. Below the navigation bar, there is a grid of 12 small plots illustrating various machine learning models. To the right of the grid, the text "scikit-learn" and "Machine Learning in Python" is displayed, followed by a bulleted list of features: Simple and efficient tools for data mining and data analysis, Accessible to everybody, and reusable in various contexts, Built on NumPy, SciPy, and matplotlib, and Open source, commercially usable - BSD license.

Classification
Identifying to which category an object belongs to.
Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ...
— Examples

Regression
Predicting a continuous-valued attribute associated with an object.
Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ...
— Examples

Clustering
Automatic grouping of similar objects into sets.
Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ...
— Examples

Dimensionality reduction
Reducing the number of random variables to consider.
Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization.
— Examples

Model selection
Comparing, validating and choosing parameters and models.
Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics.
— Examples

Preprocessing
Feature extraction and normalization.
Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction.
— Examples

Figure C.1: Part of the home page of scikit-learn toolkit

The screenshot shows the official scikit-learn documentation page for installation. At the top, there's a navigation bar with links for Home, Installation, Documentation, Examples, Google Custom Search, and a search icon.

Installing scikit-learn

Note: If you wish to contribute to the project, it's recommended you [install the latest development version](#).

Installing the latest release

Scikit-learn requires:

- Python (>= 2.7 or >= 3.3),
- NumPy (>= 1.8.2),
- SciPy (>= 0.13.3).

If you already have a working installation of numpy and scipy, the easiest way to install scikit-learn is using `pip`

```
pip install -U scikit-learn
```

or `conda`:

```
conda install scikit-learn
```

If you have not installed NumPy or SciPy yet, you can also install these using conda or pip. When using pip, please ensure that *binary wheels* are used, and NumPy and SciPy are not recompiled from source, which can happen when using particular configurations of operating system and hardware (such as Linux on a Raspberry Pi). Building numpy and scipy from source can be complex (especially on Windows) and requires careful configuration to ensure that they link against an optimized implementation of linear algebra routines. Instead, use a third-party distribution as described below.

If you must install scikit-learn and its dependencies with pip, you can install it as `scikit-learn[alldeps]`. The most common use case for this is in a `requirements.txt` file used as part of an automated build process for a PaaS application or a Docker image. This option is not intended for manual installation from the command line.

Third-party Distributions

If you don't already have a python installation with numpy and scipy, we recommend to install either via your package manager or via a python bundle. These come with numpy, scipy, scikit-learn, matplotlib and many other helpful scientific and data processing libraries.

Available options are:

Canopy and Anaconda for all supported platforms

Canopy and Anaconda both ship a recent version of scikit-learn, in addition to a large set of scientific python library for Windows, Mac OSX and Linux.

Anaconda offers scikit-learn as part of its free distribution.

Warning: To upgrade or uninstall scikit-learn installed with Anaconda or `conda` **you should not use the pip command**. Instead:

To upgrade `scikit-learn`:

```
conda update scikit-learn
```

To uninstall `scikit-learn`:

```
conda remove scikit-learn
```

Upgrading with `pip install -U scikit-learn` or uninstalling `pip uninstall scikit-learn` is likely fail to properly remove files installed by the `conda` command.

pip upgrade and uninstall operations only work on packages installed via `pip install`.

WinPython for Windows

The WinPython project distributes scikit-learn as an additional plugin.

For installation instructions for particular operating systems or for compiling the bleeding edge version, see the [Advanced installation instructions](#).

Figure C.2: Installation instructions for scikit-learn toolkit

 [Home](#) [Installation](#) [Documentation](#) [Examples](#) [Google Custom Search](#) [🔍](#)

1.13. Feature selection

The classes in the `sklearn.feature_selection` module can be used for feature selection/dimensionality reduction on sample sets, either to improve estimators' accuracy scores or to boost their performance on very high-dimensional datasets.

1.13.1. Removing features with low variance

`VarianceThreshold` is a simple baseline approach to feature selection. It removes all features whose variance doesn't meet some threshold. By default, it removes all zero-variance features, i.e. features that have the same value in all samples.

As an example, suppose that we have a dataset with boolean features, and we want to remove all features that are either one or zero (on or off) in more than 80% of the samples. Boolean features are Bernoulli random variables, and the variance of such variables is given by

$$\text{Var}[X] = p(1 - p)$$

so we can select using the threshold `.8 * (1 - .8)`:

```
>>> from sklearn.feature_selection import VarianceThreshold
>>> X = [[0, 0, 1], [0, 1, 0], [1, 0, 0], [0, 1, 1], [0, 1, 0], [0, 1, 1]]
>>> sel = VarianceThreshold(threshold=.8 * (1 - .8))
>>> sel.fit_transform(X)
array([[0, 1],
       [1, 0],
       [0, 0],
       [1, 1],
       [1, 0],
       [1, 1]])
```

As expected, `VarianceThreshold` has removed the first column, which has a probability $p = 5/6 > .8$ of containing a zero.

1.13.2. Univariate feature selection

1.13.3. Recursive feature elimination

Given an external estimator that assigns weights to features (e.g., the coefficients of a linear model), recursive feature elimination (`RFE`) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a `coef_` attribute or through a `feature_importances_` attribute. Then, the least important features are pruned from current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.

`RFECV` performs RFE in a cross-validation loop to find the optimal number of features.

Examples:

- [Recursive feature elimination](#): A recursive feature elimination example showing the relevance of pixels in a digit classification task.
- [Recursive feature elimination with cross-validation](#): A recursive feature elimination example with automatic tuning of the number of features selected with cross-validation.

1.13.4. Feature selection using SelectFromModel

Figure C.3: Part of the user guide page for feature selection

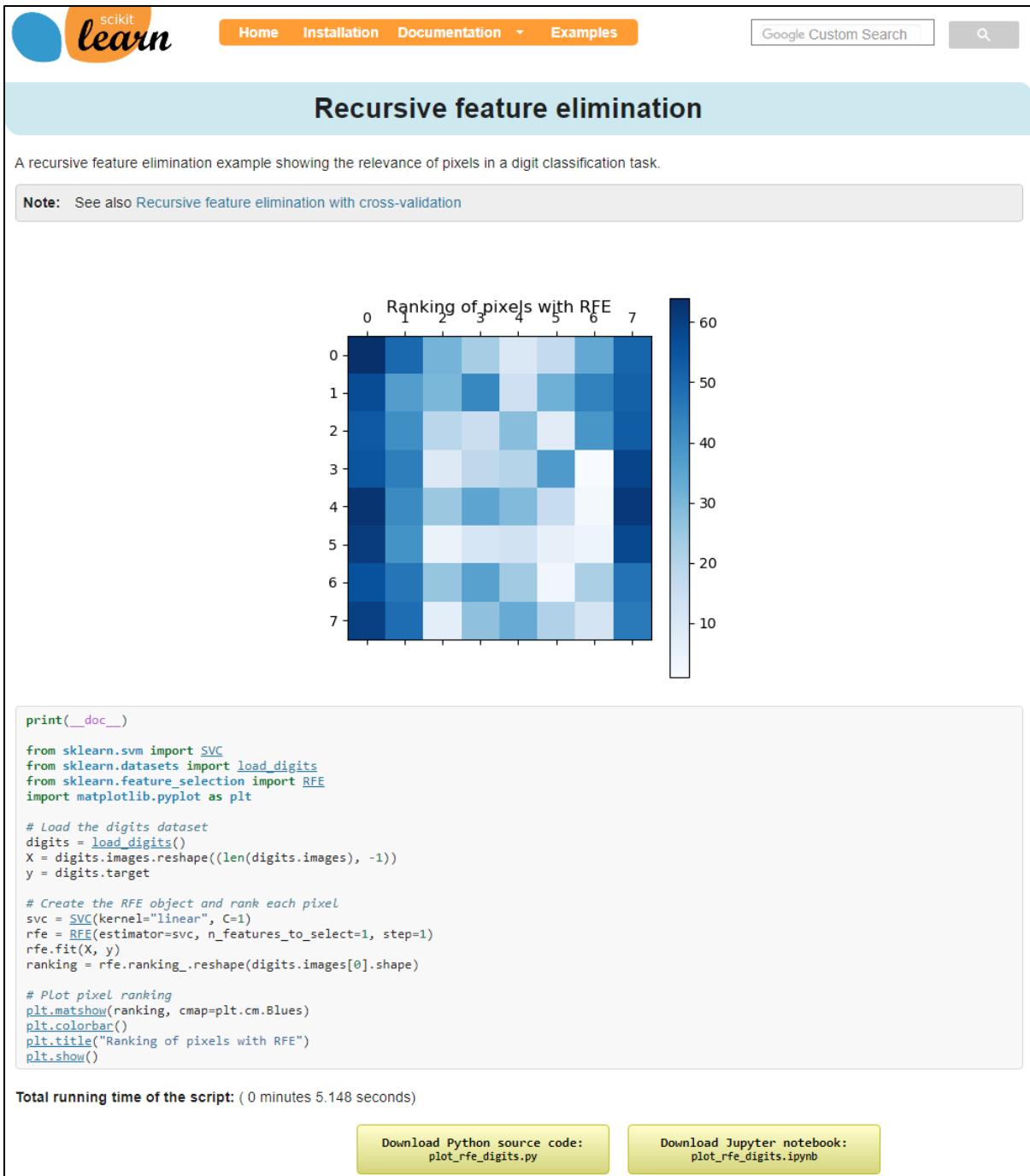
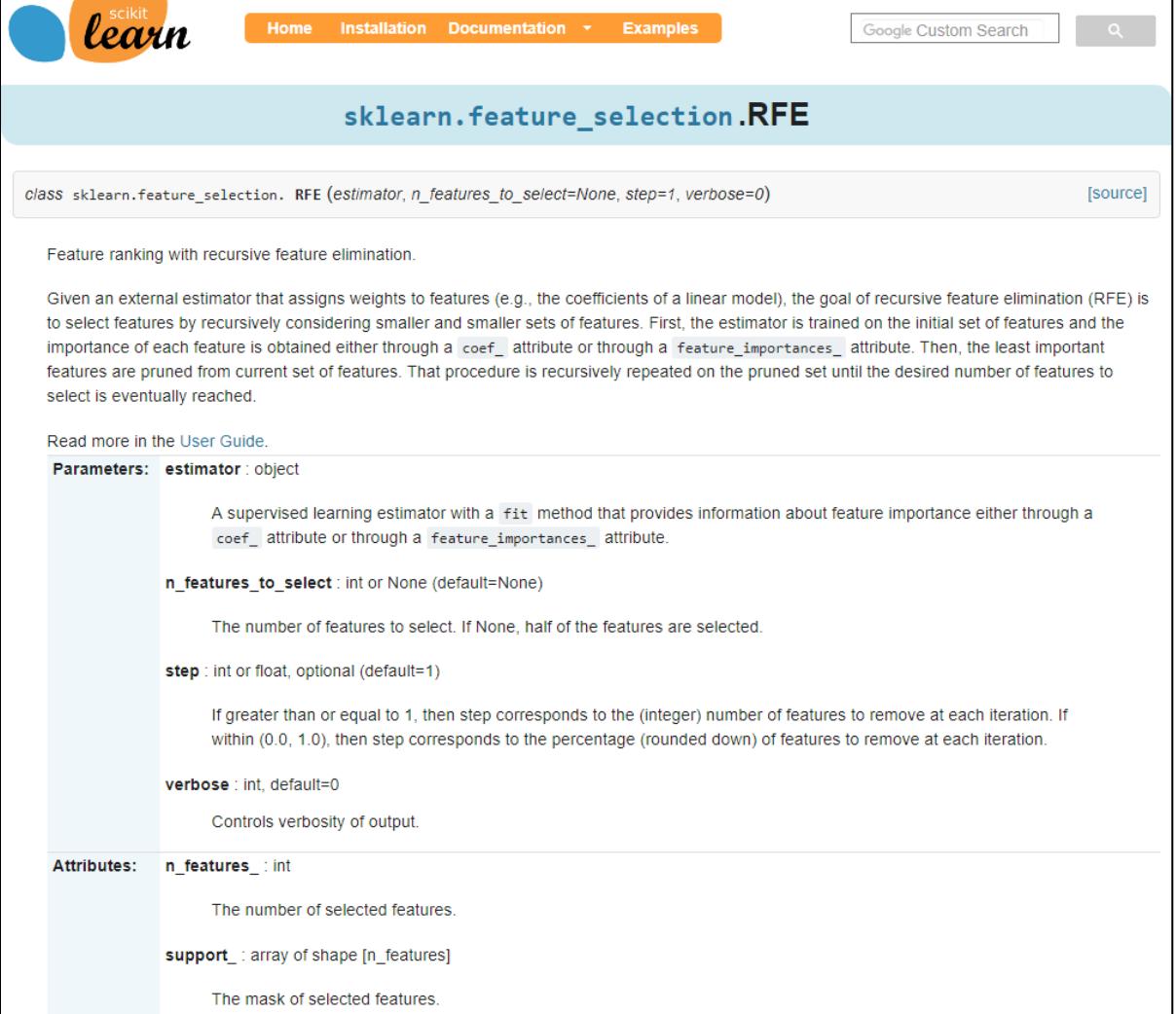


Figure C.4: An example for using RFE



The screenshot shows a section of the scikit-learn API reference for the `sklearn.feature_selection.RFE` class. At the top, there's a navigation bar with links for Home, Installation, Documentation (with a dropdown menu), Examples, Google Custom Search, and a search icon.

sklearn.feature_selection.RFE

`class sklearn.feature_selection.RFE (estimator, n_features_to_select=None, step=1, verbose=0)` [source]

Feature ranking with recursive feature elimination.

Given an external estimator that assigns weights to features (e.g., the coefficients of a linear model), the goal of recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a `coef_` attribute or through a `feature_importances_` attribute. Then, the least important features are pruned from current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.

Read more in the [User Guide](#).

Parameters:

- estimator** : object
- A supervised learning estimator with a `fit` method that provides information about feature importance either through a `coef_` attribute or through a `feature_importances_` attribute.
- n_features_to_select** : int or None (default=None)
- The number of features to select. If None, half of the features are selected.
- step** : int or float, optional (default=1)
- If greater than or equal to 1, then step corresponds to the (integer) number of features to remove at each iteration. If within (0.0, 1.0), then step corresponds to the percentage (rounded down) of features to remove at each iteration.
- verbose** : int, default=0
- Controls verbosity of output.

Attributes:

- n_features_** : int
- The number of selected features.
- support_** : array of shape [n_features]
- The mask of selected features.

Figure C.5: Part of the API reference page of RFE (1)

ranking_: array of shape [n_features]
The feature ranking, such that `ranking_[i]` corresponds to the ranking position of the i-th feature. Selected (i.e., estimated best) features are assigned rank 1.

estimator_: object
The external estimator fit on the reduced dataset.

References
[R170] Guyon, I., Weston, J., Barnhill, S., & Vapnik, V., "Gene selection for cancer classification using support vector machines", *Mach. Learn.*, 46(1-3), 389–422, 2002.

Examples

The following example shows how to retrieve the 5 right informative features in the Friedman #1 dataset.

```
>>> from sklearn.datasets import make_friedman1
>>> from sklearn.feature_selection import RFE
>>> from sklearn.svm import SVR
>>> X, y = make_friedman1(n_samples=50, n_features=10, random_state=0)
>>> estimator = SVR(kernel="linear")
>>> selector = RFE(estimator, 5, step=1)
>>> selector = selector.fit(X, y)
>>> selector.support_
array([ True,  True,  True,  True,
       False, False, False, False, False], dtype=bool)
>>> selector.ranking_
array([1, 1, 1, 1, 1, 6, 4, 3, 2, 5])
```

Methods

<code>decision_function (X)</code>	
<code>fit (X, y)</code>	Fit the RFE model and then the underlying estimator on the selected features.
<code>fit_transform (X[, y])</code>	Fit to data, then transform it.
<code>get_params ([deep])</code>	Get parameters for this estimator.
<code>get_support (indices)</code>	Get a mask, or integer index, of the features selected
<code>inverse_transform (X)</code>	Reverse the transformation operation
<code>predict (X)</code>	Reduce X to the selected features and then predict using the underlying estimator.
<code>predict_log_proba (X)</code>	
<code>predict_proba (X)</code>	
<code>score (X, y)</code>	Reduce X to the selected features and then return the score of the underlying estimator.
<code>set_params (**params)</code>	Set the parameters of this estimator.
<code>transform (X)</code>	Reduce X to the selected features.

Figure C.6: Part of the API reference page of RFE (2)

1.4. Support Vector Machines

Support vector machines (SVMs) are a set of supervised learning methods used for [classification](#), [regression](#) and [outliers detection](#).

The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different [Kernel functions](#) can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

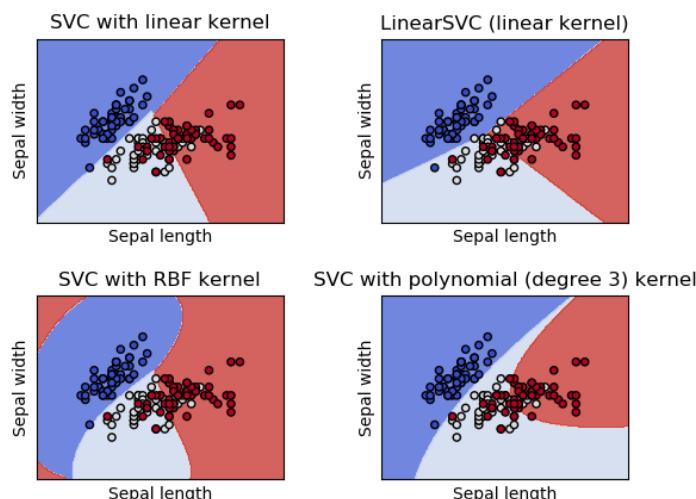
The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing [Kernel functions](#) and regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see [Scores and probabilities](#), below).

The support vector machines in scikit-learn support both dense (`numpy.ndarray`) and convertible to that by `numpy.asarray`) and sparse (any `scipy.sparse`) sample vectors as input. However, to use an SVM to make predictions for sparse data, it must have been fit on such data. For optimal performance, use C-ordered `numpy.ndarray` (dense) or `scipy.sparse.csr_matrix` (sparse) with `dtype=float64`.

1.4.1. Classification

`SVC`, `NuSVC` and `LinearSVC` are classes capable of performing multi-class classification on a dataset.



`SVC` and `NuSVC` are similar methods, but accept slightly different sets of parameters and have different mathematical formulations (see section [Mathematical formulation](#)). On the other hand, `LinearSVC` is another implementation of Support Vector Classification for the case of a linear kernel. Note that `LinearSVC` does not accept keyword `kernel`, as this is assumed to be linear. It also lacks some of the members of `SVC` and `NuSVC`, like `support_`.

As other classifiers, `SVC`, `NuSVC` and `LinearSVC` take as input two arrays: an array `X` of size `[n_samples, n_features]` holding the training samples, and an array `y` of class labels (strings or integers), size `[n_samples]`:

```
>>> from sklearn import svm
>>> X = [[0, 0], [1, 1]]
>>> y = [0, 1]
>>> clf = svm.SVC()
>>> clf.fit(X, y)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

After being fitted, the model can then be used to predict new values:

```
>>> clf.predict([[2., 2.]])
array([1])
```

Figure C.7: Part of the user guide page for Support Vector Machines



scikit
learn

[Home](#)
[Installation](#)
[Documentation](#)
[Examples](#)

Google Custom Search

🔍

sklearn.svm.SVC

```
class sklearn.svm. SVC (C=1.0, kernel='rbf', degree=3, gamma='auto', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200,
class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', random_state=None)
```

[\[source\]](#)

C-Support Vector Classification.

The implementation is based on libsvm. The fit time complexity is more than quadratic with the number of samples which makes it hard to scale to dataset with more than a couple of 10000 samples.

The multiclass support is handled according to a one-vs-one scheme.

For details on the precise mathematical formulation of the provided kernel functions and how gamma, coef0 and degree affect each other, see the corresponding section in the narrative documentation: [Kernel functions](#).

Read more in the [User Guide](#).

Parameters:

- C** : float, optional (default=1.0)

Penalty parameter C of the error term.
- kernel** : string, optional (default='rbf')

Specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. If none is given, 'rbf' will be used. If a callable is given it is used to pre-compute the kernel matrix from data matrices; that matrix should be an array of shape `(n_samples, n_samples)`.
- degree** : int, optional (default=3)

Degree of the polynomial kernel function ('poly'). Ignored by all other kernels.
- gamma** : float, optional (default='auto')

Kernel coefficient for 'rbf', 'poly' and 'sigmoid'. If gamma is 'auto' then $1/n_features$ will be used instead.
- coef0** : float, optional (default=0.0)

Independent term in kernel function. It is only significant in 'poly' and 'sigmoid'.
- probability** : boolean, optional (default=False)

Whether to enable probability estimates. This must be enabled prior to calling fit, and will slow down that method.
- shrinking** : boolean, optional (default=True)

Whether to use the shrinking heuristic.
- tol** : float, optional (default=1e-3)

Tolerance for stopping criterion.
- cache_size** : float, optional

Specify the size of the kernel cache (in MB).
- class_weight** : {dict, 'balanced'}, optional

Set the parameter C of class i to $class_weight[i] * C$ for SVC. If not given, all classes are supposed to have weight one. The "balanced" mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as $n_samples / (n_classes * np.bincount(y))$
- verbose** : bool, default: False

Enable verbose output. Note that this setting takes advantage of a per-process runtime setting in libsvm that, if enabled, may not work properly in a multithreaded context.
- max_iter** : int, optional (default=-1)

Hard limit on iterations within solver, or -1 for no limit.

Figure C.8: Part of the API reference page for SVM classification (1)

decision_function_shape : {‘ovo’, ‘ovr’, ‘default’=‘ovr’}

Whether to return a one-vs-rest (‘ovr’) decision function of shape (n_samples, n_classes) as all other classifiers, or the original one-vs-one (‘ovo’) decision function of libsvm which has shape (n_samples, n_classes * (n_classes - 1) / 2).

Changed in version 0.19: `decision_function_shape` is ‘ovr’ by default.

New in version 0.17: `decision_function_shape='ovr'` is recommended.

Changed in version 0.17: Deprecated `decision_function_shape='ovo'` and `None`.

random_state : int, RandomState instance or None, optional (default=None)

The seed of the pseudo random number generator to use when shuffling the data. If int, `random_state` is the seed used by the random number generator; If RandomState instance, `random_state` is the random number generator; If None, the random number generator is the RandomState instance used by np.random.

Attributes:

- support_** : array-like, shape = [n_SV]
Indices of support vectors.
- support_vectors_** : array-like, shape = [n_SV, n_features]
Support vectors.
- n_support_** : array-like, dtype=int32, shape = [n_class]
Number of support vectors for each class.
- dual_coef_** : array, shape = [n_class-1, n_SV]
Coefficients of the support vector in the decision function. For multiclass, coefficient for all 1-vs-1 classifiers. The layout of the coefficients in the multiclass case is somewhat non-trivial. See the section about multi-class classification in the SVM section of the User Guide for details.
- coef_** : array, shape = [n_class-1, n_features]
Weights assigned to the features (coefficients in the primal problem). This is only available in the case of a linear kernel.
`coef_` is a readonly property derived from `dual_coef_` and `support_vectors_`.
- intercept_** : array, shape = [n_class * (n_class-1) / 2]
Constants in decision function.

See also:

- SVR**
Support Vector Machine for Regression implemented using libsvm.
- LinearSVC**
Scalable Linear Support Vector Machine for classification implemented using liblinear. Check the See also section of LinearSVC for more comparison element.

Examples

```
>>> import numpy as np
>>> X = np.array([[-1, -1], [-2, -1], [1, 1], [2, 1]])
>>> y = np.array([1, 1, 2, 2])
>>> from sklearn.svm import SVC
>>> clf = SVC()
>>> clf.fit(X, y)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
     decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
     max_iter=-1, probability=False, random_state=None, shrinking=True,
     tol=0.001, verbose=False)
>>> print(clf.predict([[-0.8, -1]]))
[1]
```

Methods

<code>decision_function (X)</code>	Distance of the samples X to the separating hyperplane.
<code>fit (X, y[, sample_weight])</code>	Fit the SVM model according to the given training data.
<code>get_params ([deep])</code>	Get parameters for this estimator.
<code>predict (X)</code>	Perform classification on samples in X.
<code>score (X, y[, sample_weight])</code>	Returns the mean accuracy on the given test data and labels.
<code>set_params (**params)</code>	Set the parameters of this estimator.

Figure C.9: Part of the API reference page for SVM classification (2)

```

1 import numpy
2 from sklearn import metrics
3 from sklearn import model_selection
4 from sklearn import feature_selection
5 from sklearn import svm
6
7 import dataset
8
9 # Load data and labels
10 data_files = [
11     'entropy_shannon_14', 'entropy_shannon_32',
12     'oscillation_14',      'oscillation_32',
13     'power_log_14',        'power_log_32',
14     'power_alpha2theta',   'power_beta2alpha',  'power_beta2theta'
15 ]
16 data = dataset.load_data(data_files)
17 lbls = dataset.load_labels()
18
19 # map labels (1-9) into classes
20 clss = (lbls >= 5).astype(float)
21 # get distinct classes
22 dist_clss = numpy.unique(clss)
23 # get cross-validation object
24 cv = model_selection.StratifiedKFold(n_splits=10, shuffle=True, random_state=None)
25 # get classifier
26 classifier = svm.SVC(kernel='linear')
27 # get dimension reduction object
28 dim_reduction = feature_selection.RFE(
29     estimator=svm.SVC(kernel='linear'), n_features_to_select=data.shape[1]//4, step=1)
30
31 # init results matrices with zeros
32 accuracy = numpy.zeros((4, 10))
33 f1_score = numpy.zeros((4, 10))
34
35 # Loop on scales
36 for scale in (0, 1, 2, 3):
37     # select classes of current scale
38     scl_clss = clss.take(scale, axis=1)
39     # apply dimension reduction
40     scl_data = dim_reduction.fit_transform(data, scl_clss)
41     # repeat the process by 10 times
42     for repeat in range(10):
43         # get cross-validation object
44         cv = model_selection.StratifiedKFold(n_splits=10, shuffle=True, random_state=repeat)
45         # apply cross-validation and get prediction of test sets
46         results = model_selection.cross_val_predict(classifier, scl_data, scl_clss, cv=cv)
47         # calc results
48         accuracy[scale, repeat] = metrics.accuracy_score(scl_clss, results)
49         f1_score[scale, repeat] = metrics.f1_score(
50             scl_clss, results, labels=dist_clss, average='macro')
51
52 print(accuracy)
53 print(f1_score)

```

Figure C.10: Python code to classify DEAP Dataset participants' emotions

ملخص

التعرف على المشاعر أصبح من العوامل المهمة لجعل تفاعل البشر مع الحاسوب الآلي أسهل وأكثر فاعلية. على الرغم من أهمية المشاعر في التواصل بين البشر فإن معظم نظم التفاعل بين الإنسان والآلة تفتقر القدرة على التعرف على المشاعر وتفسيرها.

يمكن التعرف على المشاعر عن طريق مراقبة المظاهر الخارجية للإنسان. مثل مراقبة تعبيرات الوجه أو نبرة الصوت أو حركة الجسم. كذلك يمكن التعرف على المشاعر عن طريق مراقبة الإشارات الفيسيولوجية الداخلية لجسم الإنسان. مثل معدل ضربات القلب أو معدل التنفس أو إشارات المخ. تعتبر الطرق المعتمدة على مراقبة الإشارات الفيسيولوجية لجسم الإنسان صعبة الخداع وبالتالي يمكن الوثوق بها أكثر خصوصاً تلك التي تعتمد على مراقبة إشارات المخ والجهاز العصبي. خلال السنوات القليلة الماضية، ازداد الاهتمام بتفسير نوايا المستخدمين عن طريق مراقبة إشارات المخ من قبل العديد من الباحثين والشركات الخاصة.

رسم المخ الكهربائي هو أكثر الطرق المستخدمة في مراقبة إشارات المخ. هذه الطريقة تقيس النشاط الكهربائي للمخ عن طريق مجموعة من الأقطاب الكهربائية توضع على فروة الرأس. تتميز هذه الطريقة بأن لها دقة زمانية عالية وأنها آمنة بدون أي مخاطر كما أنها رخيصة التكلفة. أثناء العقدين الماضيين قد تم اختراع العديد من الأجهزة التجارية لقياس التخطيط الكهربائي للمخ، تتميز تلك الأجهزة بسهولة الاعداد والاستخدام مقارنة بنظيرتها التي تستخدم في المعامل والمخبرات.

التعرف على المشاعر بالاعتماد على مراقبة إشارات المخ ليست بال مهمة السهلة. فمثيل المشاعر يتضمن بعض الصعوبات كما أن تفسير إشارات المخ صعب للغاية. معظم الدراسات السابقة مسبقاً تفتقر للدقة العالية للتعرف على المشاعر، كما أن بعضها يقوم ببناء نموذج للتعرف على المشاعر بشكل منفصل لكل مستخدم أو لكل نوع من المستخدمين مما يتطلب إعادة عملية التدريب أو التهيئة عند استخدام النموذج المقترن من قبل مستخدمين جدد.

هدف هذا البحث هو تحسين التعرف على المشاعر بالاعتماد على إشارات المخ المسجلة باستخدام أجهزة رسم المخ الكهربائي. النموذج المقترن قادر على التعامل مع أنواع المستخدمين المختلفة دون الحاجة لإعادة عملية التدريب أو التهيئة.

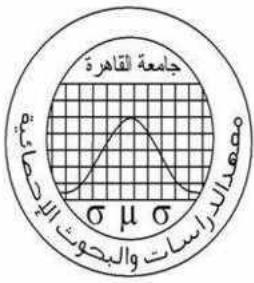
يتضمن التعرف على المشاعر بالاعتماد على إشارات المخ مجموعة من الخطوات لكي يتحقق. أولاً هناك تداخل في الموجات يحدث عند تسجيل إشارات المخ حيث تتأثر أجهزة التخطيط الكهربائي للمخ بحركة العين وعضلات الوجه والرقبة وبالتالي يجب إزالة أو تقليل الضجيج الناتج عن ذلك التداخل في إشارات المخ قبل معالجتها، كما أن إشارات المخ تتضمن مجموعة من الإشارات أو المكونات الداخلية والتي تحتاج إلى فصلها عن

بعضها البعض. بعد تجهيز إشارات المخ يتم استخراج الصفات التي تستخدم في التعرف على المشاعر. عدد ضخم جداً من الصفات يمكن استخراجه من إشارات المخ مما يتسبب في تعقيد عملية التعرف على المشاعر، لذلك تستخدم وسائل تقليل الصفات لتقليل عدد الصفات المستخرجة. أخيراً، يتم استخدام أحد وسائل التصنيف للتعرف على مشاعر المستخدم المتعلقة بالصفات التي تم استخراجها.

تعرض هذه الأطروحة الطرق المختلفة المستخدمة في كل خطوة من الخطوات السابق ذكرها. كما أنها تناقش بعض النماذج المقترحة مسبقاً للتعرف على المشاعر بالاعتماد على إشارات المخ وتقارن بينهم.

يتبنى هذا البحث استخراج صفات القوة الطيفية والتذبذب ومقدار الفوضى (الإنتروبيا) من إشارات المخ، كما يتبنى تقليل الصفات المستخرجة باستخدام طريقة حذف الصفات المتكرر وتصنيف المشاعر باستخدام طريقة شعاع الدعم الآلي.

النتائج التجريبية أظهرت مدى تفوق ومتانة النموذج المقترح مقارنة مع نتائج الدراسات السابقة على نفس قاعدة البيانات. حيث حقق النموذج المقترح دقة أعلى من الدراسات الأخرى.



جامعة القاهرة

معهد الدراسات والبحوث الإحصائية

قسم علوم الحاسوب

تحسين التعرف على المشاعر باستخدام إشارات المخ

بحث مقدم للحصول على درجة الماجستير في علوم الحاسوب

مقدم من

محمد أحمد عبدالعال أحمد

إشراف

دكتور / عاصم أحمد الصاوي

أستاذ دكتور / هشام أحمد حفني

أكاديمية الإسكندرية للإدارة

قسم علوم الحاسوب

والمحاسبة

معهد الدراسات والبحوث الإحصائية

وزارة التعليم العالي والبحث العلمي

جامعة القاهرة