Classification of EEG Data Using Convolutional Neural Networks and the Scaled Reassigned Spectrogram

Karin Pagels

Bachelor's thesis
2021:K12

Lund University

Faculty of Science
Centre for Mathematical Sciences
Mathematical Statistics

Centrum Scientiarum Mathematicarum

Abstract

Electroencephalography (EEG) is a medical technique for measuring brain activity through several channels connected to the scalp. Interpreting EEG data is a difficult problem because of the large amount of noise contained in the data. Using spectral methods on EEG data can improve the ability to interpret the data, especially using a time-frequency method called the scaled reassigned spectrogram which has been shown to perform well on data with similar properties to a model containing Gaussian envelope transients. A technique for classification of data transformed by time-frequency methods is to use convolutional neural networks (CNN), which are known to be successful at image classification. In this thesis, spectral methods and CNNs are combined for use on EEG data in order to identify the location of a sound, whether a person hears the sound in the left or in the right ear. The best classification results obtained in this thesis were for a single channel near the right ear without transforming the data at 60.13%, and using singular value decomposition (SVD) on four channels near each ear and the scaled reassigned spectrogram with a result of 59.47%. These are both significant results.

Acknowledgement

I give my sincere gratitude to my supervisor Maria Sandsten for her patience and optimism, as well as for all the time she spent helping me with this thesis. It has been an honour to have her as my supervisor.

I would also like to express my gratitude towards Mikael Johansson, professor in psychology at Lund University, for providing excellent EEG data for this thesis.

I am very thankful to my family and friends for their support, understanding and help in keeping me motivated. I am grateful to have them in my life.

Contents

1 Introduction **1**
- 1.1 Thesis Objective . . . 2
- 1.2 Structure of the Thesis . . . 2

2 Theory **3**
- 2.1 Spectral Methods . . . 3
  - 2.1.1 The Periodogram . . . 3
  - 2.1.2 The Spectrogram . . . 4
  - 2.1.3 The Scaled Reassigned Spectrogram . . . 5
- 2.2 Signal-to-Noise Ratio . . . 6
- 2.3 Singular Value Decomposition . . . 6
- 2.4 Artificial Neural Networks . . . 7
  - 2.4.1 Convolutional Neural Networks . . . 8

3 Method **10**
- 3.1 Neural Network . . . 10
- 3.2 Simulated Data . . . 12
  - 3.2.1 Approach . . . 13
- 3.3 Real Data . . . 16
  - 3.3.1 Approach . . . 17

4 Results and Discussion **21**
- 4.1 Simulated Data . . . 21
- 4.2 Real Data . . . 25

5 Conclusion **33**
- 5.1 Final Conclusion . . . 35
- 5.2 Further Studies . . . 35

Bibliography **37**

Chapter 1

Introduction

The reassigned spectrogram is a spectral method that was first mentioned in 1994 by Auger and Flandrin [1]. The purpose of the method was to get better resolution on the peaks in power when viewing the spectrogram, and thereby be able to see the signal more clearly. By adding scaling factors to the reassignment, the scaled reassigned spectrogram is obtained. In 2015 it was shown that the method works well for data containing Gaussian envelope transient signals [2], meaning an oscillating signal with Gaussian envelopes, where an envelope is a smooth curve outlining the extremes of the signal.

Though the scaled reassigned spectrogram has been theoretically shown to work well for data holding Gaussian envelope transients, it still has to be evaluated on a number of different data sets. A type of real-world data that appears to behave like Gaussian envelope transient data is electroencephalography, often referred to as EEG.

EEG is a medical technique that measures activity in the brain. The method consists of placing a cap with several channels on a person's scalp. Each channel corresponds to a signal output, and each signal output is a result of activity in a certain part of the brain, due to the fact that the channels are evenly distributed over the scalp.

EEG was discovered in 1929 by German psychiatrist Hans Berger [3]. Since then it has been researched continuously, and still is today, as being able to interpret EEG is something that could help people with injuries and disabilities in their daily life. Currently, a system for interpretation of EEG signals called BCI (Brain-Computer Interface) is being researched, which aims to create a direct communication pathway between the brain and an external device. An example of a use of BCI [4] is to develop compensating systems to help people with severe motor control disability to recover mo-

bility. There are many other areas where interpreting EEG can have a large impact. The BCI project that is driven at Lund University [5] sees opportunities like smart hearing aids, forensics tools and gaming-devices, along with other health-related applications, such as rehabilitation. An important part to be able to do this is to classify the EEG data.

A great tool for classification is the use of artificial neural networks, as they are known to perform well as classifiers. For image classification, a convolutional neural network is typically used. A convolutional neural network, or a CNN, is a type of artificial neural network that uses a specific type of filter for image recognition. The reason for using CNNs on signals transformed by the scaled reassigned spectrogram is that the signals will become two-dimensional and therefore can be seen as images.

1.1 Thesis Objective

The aim of this thesis is simply to classify real EEG data. The goal is to be able to tell if a sound is coming in the left or the right ear on a person, only by using the EEG data set. The idea is to classify by using the scaled reassigned spectrogram, but other spectral methods will be tested and compared to this method.

The classification will be done by using a CNN for all the spectral methods, and it will be kept similar for all methods. The only difference is between the one-dimensional methods, simple raw data and the periodogram, and the two-dimensional methods, the spectrogram and the scaled reassigned spectrogram. This is because of the so-called convolutional layer which preferably should be two dimensional but has to be one-dimensional for the raw data and the one-dimensional periodogram.

1.2 Structure of the Thesis

In chapter 2 of the thesis, all theoretical concepts are specified. Secondly, in chapter 3 the method is described, where the used CNNs are specified. The rest of the method is divided into two parts, where the first part describes a simulation study and the second part the use of real data. In chapter 4 the results are presented and a discussion about the results is carried out. The first part of the chapter regards the simulated data and the second part regards the real data. The last chapter of the thesis, chapter 5, contains the conclusions.

Chapter 2

Theory

Contained in this chapter are the definitions of all the spectral methods used in the thesis. These are the periodogram, the spectrogram and the scaled reassigned spectrogram. Then the signal-to-noise ratio is defined as well as the singular value decomposition. The signal-to-noise ratio is used when creating simulated data, and singular value decomposition is used when merging several sequences of real data. Lastly, artificial neural networks are described, as well as convolutional neural networks.

2.1 Spectral Methods

In spectral analysis, a common and useful tool is time-frequency analysis. Time-frequency methods are representations of the power of the signal under consideration, over frequency and time. The scaled reassigned spectrogram is a time-frequency method, but the most common one is the spectrogram. Another used method that is not a time-frequency method is the periodogram. The periodogram is an estimate of the stationary spectral density of a signal and is defined below. For all spectral methods, integration runs from $-\infty$ to ∞ if nothing else is specified.

2.1.1 The Periodogram

The periodogram is normally defined for stationary discrete-time data [6]. The discrete-time data is replaced with the continuous-time signal $x(t)$ for consistency with all other formulas, and therefore we define the continuous-time Fourier transform

$$\mathscr{X}(f) = \int x(t)h(t)e^{-i2\pi ft}dt \tag{2.1}$$

where t is time, f is the frequency and $h(t)$ is the window. Then the estimate of the spectral density is

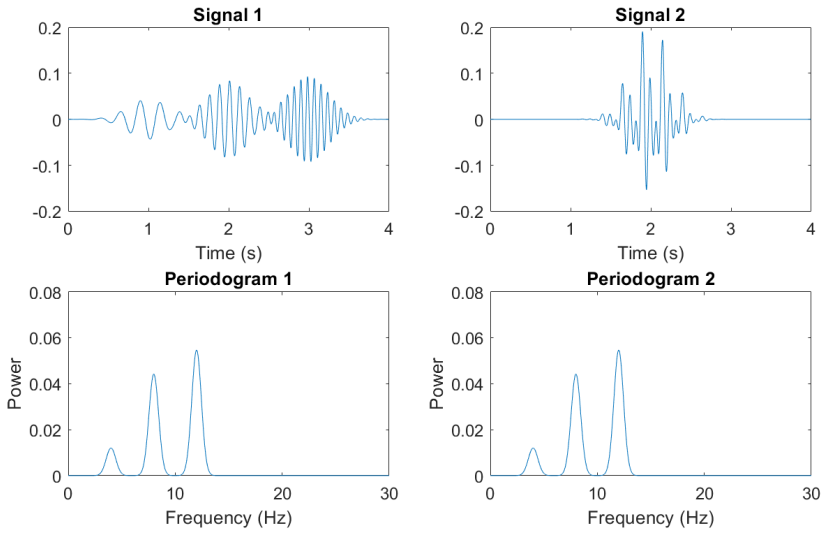

Figure 2.1: Two simulated Gaussian envelope transients and their corresponding periodograms. Signal 1 has the time centre values at 1, 2 and 3 seconds while signal 2 has all its centre values at 2 seconds.

$$\hat{R}_X(f) = |\mathscr{X}(f)|^2 \tag{2.2}$$

in concordance with the periodogram definition. Figure 2.1 shows two simulated signals, Gaussian envelope transients, with their corresponding unwindowed periodograms. The first signal has the time centre values at 1, 2 and 3 seconds, and the second signal has all the time centre values at 2 seconds. Notice that the two periodograms are identical.

2.1.2 The Spectrogram

To define the spectrogram we first define the continuous-time STFT (short-time Fourier transform) of the signal $x(t)$ as

$$F_x^h(t,f) = \int x(s)h(s-t)e^{-i2\pi fs}ds \tag{2.3}$$

where t is time, f is the frequency and $h(t)$ is the window [7]. From this we get the spectrogram as

$$S_x^h(t,f) = |F_x^h(t,f)|^2. \tag{2.4}$$

In figure 2.2 a spectrogram computed on real EEG data is shown.

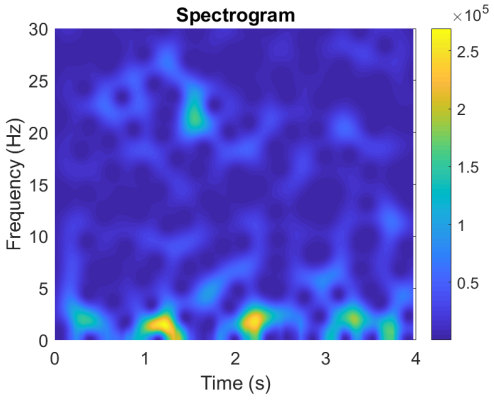

Figure 2.2: A spectrogram computed on real EEG data.

2.1.3 The Scaled Reassigned Spectrogram

The primary method considered in this thesis is the scaled reassigned spectrogram. Firstly the spectrogram values t and f are relocated to $\hat{t}_x$ and $\hat{f}_x$, and the reassigned spectrogram is found as

$$RS_x^h(t,f) = \int\int S_x^h(s,\xi)\delta(t-\hat{t}_x(s,\xi), f-\hat{f}_x(s,\xi))dsd\xi. \tag{2.5}$$

Here s is the integration variable in time, ξ the integration variable in frequency and $\delta(t,f)$ is the two dimensional Dirac impulse defined as

$$\int\int g(t,f)\delta(t-t_0, f-f_0)dtdf = g(t_0,f_0) \tag{2.6}$$

where $g(t,f)$ is a function. To get the scaled reassigned spectrogram we use the scaling factors c_t and c_f to compute the reassignment as

$$\hat{t}_x(t,f) = t + c_t\Re\left(\frac{F_x^{th}(t,f)}{F_x^h(t,f)}\right), \tag{2.7}$$

$$\hat{f}_x(t,f) = f - c_f\frac{1}{2\pi}\Im\left(\frac{F_x^{dh/dt}(t,f)}{F_x^h(t,f)}\right), \tag{2.8}$$

where $\Re$ is the real part and $\Im$ is the imaginary part [7]. F_x^{th} and $F_x^{dh/dt}$ are calculated using (2.3) with the windows $t\cdot h(t)$ and $\frac{dh(t)}{dt}$ respectively. In figure 2.3 the spectrogram and the scaled reassigned spectrogram on simulated data containing three Gaussian envelope transients are plotted. Both the spectrogram and the scaled reassigned spectrogram have their time centre at 2 seconds and the frequencies 5, 10 and 15 as well as an SNR of 20 dB (see section 2.2 for SNR).

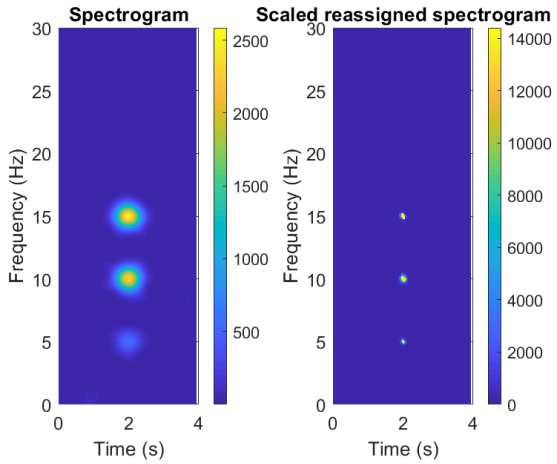

Figure 2.3: A comparison of the spectrogram and the scaled reassigned spectrogram on simulated Gaussian envelope transient data.

2.2 Signal-to-Noise Ratio

Signal-to-noise ratio, or SNR, is defined as

$$\text{SNR} = \frac{\int x^2(t)dt}{\sigma^2_{\text{noise}}} \tag{2.9}$$

where $x(t)$ is the signal assumed to be transient and σ^2_{noise} is the variance of the noise. In a decibel scale SNR is

$$\text{SNR}_{\text{dB}} = 10\log_{10}(\text{SNR}). \tag{2.10}$$

(2.9) and (2.10) indicates that a lower SNR, or SNR_{dB}, means a greater amount of noise relative to the signal.

2.3 Singular Value Decomposition

The singular value decomposition, or SVD, is a type of matrix factorization. If $A \in \mathbb{R}$ is an $m \times n$ matrix, then the full SVD is defined as

$$A = U\Sigma V^T \tag{2.11}$$

where U is an $m \times m$ orthogonal matrix, Σ is an $m \times n$ diagonal matrix and V is an $n \times n$ orthogonal matrix [8]. The entries σ_j of Σ are non-negative and in non-increasing order, meaning that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$ where $p = \min(m, n)$.

2.4 Artificial Neural Networks

An artificial neural network (ANN), often referred to simply as a neural network, is a construction that is inspired by the human brain and consists of so-called nodes. The nodes are placed in different layers where the first layer is the input layer and the last layer is the output layer. In between these, there are other layers, where the types and the sizes of the layers vary depending on the type of ANN and the desired result.

Deep learning is a specific kind of machine learning that uses artificial neural networks to learn. The definition of learning is presented below.

Definition. A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E [9].

Simply put, an ANN is an algorithm that can learn from data in some way. The task T is what we want to solve with the learning algorithm, it is not the process of learning [10]. An example of a deep learning task is classification. The performance measure P is a quantitative measure of the learning algorithm's performance. Often the performance measure is the accuracy of the model, where classification is a case where this is true.

The experience E, or the learning, can be classified into two classes; supervised learning and unsupervised learning. Unsupervised learning algorithms learn useful properties in the data set. In supervised learning, each measurement in the data set is associated with a label or target. Because of this, the algorithm can learn to classify the data into categories. The found properties can be connected to the label or target, and then sorted into categories.

When working with ANNs, an important part is that the algorithm should be able to take in new data, that it has not trained on, and still perform well. This means that the algorithm can not be too specifically created for the training data, but has to be able to handle other similar data. This data is called test data, and one of the most important parts of deep learning is to increase test accuracy. We also want to decrease the gap between the test accuracy and training accuracy. When the gap between these is too large we have something called overfitting, and when the training accuracy in very low we have underfitting [10]. There needs to be a balance in how well the model fits the training data. When we have underfitting we have a quite large bias. The bias is much lower for overfitting but is still quite low when the optimal model is achieved.

To be able to increase the accuracy of the test data using the training data there needs to be some assumptions made about the data. We assume that

all measurements are independent of each other and that the training data and the test data are identically distributed [10]. These assumptions are known as the i.i.d. assumptions.

In machine learning a popular theorem is the no free lunch theorem by Wolpert [11]. This means that there is no universally better machine learning algorithm, the model has to fit the data and the task.

Most deep learning algorithms are powered by the learning rate optimization algorithm called stochastic gradient descent [10]. This is an algorithm that uses the derivative is to minimize the loss function, which is described below. An extension of the classical stochastic gradient descent is the Adam algorithm. This is an algorithm that is viewed as being quite robust to the choice of hyperparameters. In one step of the algorithm, a mini-batch is sampled. A mini-batch is a small subset of the total amount of data, and this subset is used during one iteration in the ANN. A run through all the mini-batches is called an epoch.

When using deep learning, a loss function needs to be specified. The loss function usually includes at least one term that causes the ANN to perform statistical estimation. An example of a loss function is the cross-entropy loss, which is common for classification. It measures the difference between two probability distributions for classification ANNs, where the output is a probability value between 0 and 1.

2.4.1 Convolutional Neural Networks

In this thesis, a convolutional neural network (CNN) is used. The reason for this is that CNNs are shown to often perform well on images, due to current research on the subject [12], and a time-frequency plot is an image.

The difference between a CNN and a general ANN is that a CNN uses a convolution, which is a specialized kind of linear operation, instead of general matrix multiplication, in at least one layer. The convolution provides a more smoothed version of the input and is defined as

$$s(t) = \int x(a)w(t-a)da \tag{2.12}$$

where w is a weighting function and x is the input [10]. Convolution is typically denoted as

$$s(t) = (x * w)(t). \tag{2.13}$$

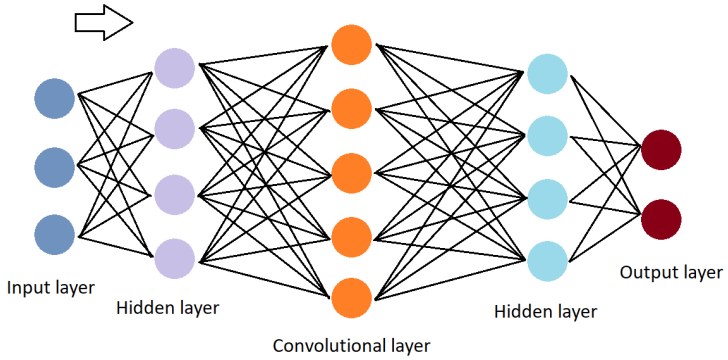

Figure 2.4: Illustration of the nodes in a CNN, where the first layer from the left is the input layer, the middle layer is the convolutional layer and the last layer is the output layer. In between these are other hidden layers.

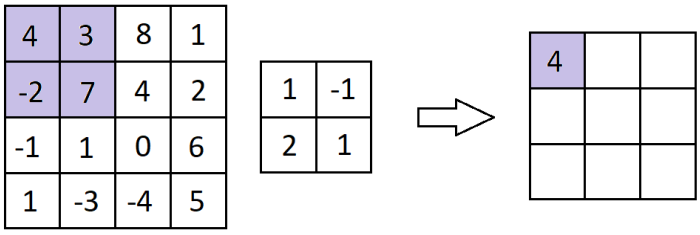

Figure 2.5: An example of a convolutional layer. The first matrix from the left is the input of the layer, the second matrix is the filter and the third matrix is the output of the layer.

In figure 2.4 an example of how the nodes in a CNN are positioned can be seen. The first layer is the input layer and the last layer is the output layer. In between these, there is at least one convolution layer, and the other used layers here referred to as hidden layers. An example of how a convolutional layer works can be seen in figure 2.5. The filter, which is the second matrix from the left, is placed in the top left corner of the input matrix. The values in the same positions are multiplied, and everything in the filter is added together. This value will be in the top left corner of the output.

Shifting the filter one step to the right, and computing a value for this placement the same way as before, gives the value to the right of the formerly placed value in the output matrix. The filter is shifted one step to the right again and the computations are done. When the first row in the output matrix is filled, the filter is moved to the next row and the values for the output matrix are computed. This is done until the whole input matrix of size $m \times n$ is covered, and the output is a matrix with size $(m-1) \times (n-1)$.

Chapter 3

Method

The thesis is divided into two parts, where the first one consists of creating simulated EEG data and inserting this data into a CNN to gain an understanding of the potential of the network, as well as a comparison of the different spectral methods. The chosen procedure for simulating EEG data was created by Barzegaran, Bosse, Kohler and Norcia [13], and implemented into Matlab by Maria Sandsten.

The second part of the thesis was to use real EEG data provided by Mikael Johansson, professor in psychology at Lund University, to classify the data, and along with this also compare the spectral methods and evaluate the usefulness of the CNN applied to EEG data. The first part of the method will be a description of the CNN that is used for both of the data types, along with other choices involving the neural network that applies to all data. All programming in this thesis was done in Matlab.

3.1 Neural Network

In this section, all choices regarding the CNNs for both simulated data and real data are specified. The layers are specified and explained.

For the training data and the test data, shuffling was done. The training data was shuffled before every training epoch and the test data was shuffled before each network validation. The validation frequency was chosen so that the network was validated about once per epoch, and so that the validation frequency times the mini-batch size equalled the size of the training data set.

The chosen learning rate optimization algorithm was Adam [15]. The stochastic gradient descent with momentum (SGDM) optimizer was also tested, but the results were not improved. Because the last layer in the network is a classification layer, the used loss function is the cross-entropy loss.

#	**Layer**	**Matlab name**
1	Image input layer	imageInputLayer
2	2-D convolutional layer	convolution2dLayer
3	Batch normalization layer	batchNormalizationLayer
4	Rectified Linear Unit (ReLU) layer	reluLayer
5	Dropout layer	dropoutLayer
6	Fully connected layer	fullyConnectedLayer
7	Softmax layer	softmaxLayer
8	Classification output layer	classificationLayer

Table 3.1: The CNN structure for all neural networks in this thesis as well as the layers and their Matlab names.

In table 3.1 the structure for all neural networks used in this thesis is presented. The first layer is the image input layer where the selected data is inserted into the CNN, and the size of the data is specified. Then we have the convolutional layer, followed by a batch normalization layer and a rectified linear unit layer, or a ReLU layer [14]. In the batch normalization layer the mini-batches are normalized, and in the ReLU layer input elements that are less than zero are set to zero. In the dropout layer randomly selected input elements are set to zero with a given probability. The fully connected layer multiplies the input by a weight matrix and adds a bias vector to it, this is where the output size of the neural network is specified. In the softmax layer a softmax function, or a normalized exponential function, is applied to the input. From this layer, we get a probabilistic output. Finally, in the classification output layer, the cross-entropy loss for the classification problem is computed.

In table 3.3 the structures with inputs for the simulated and the real data are shown. In the convolutional layer, the first input is the size of the filter and the second input is the number of filters used. After this padding, or zero-padding, is specified, where 'same' means that the zero-padding is done so that the output of the layer has the same size as the input.

The 70/30 rule was used, which is a common way to divide data sets in deep learning. This means that 70% of the data was used for training and 30% for validation.

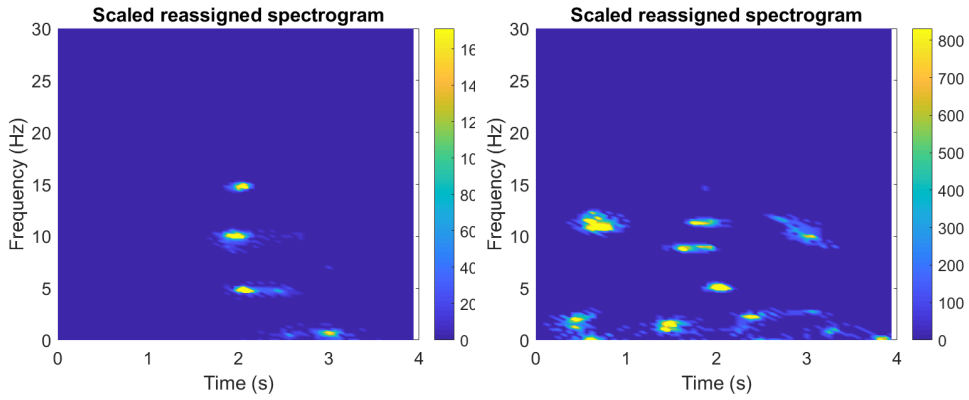

Figure 3.1: A comparison of the scaled reassigned spectrogram corresponding to two different SNRs in simulated EEG data. The first plot has an SNR of 13 dB and the second an SNR of 10 dB.

3.2 Simulated Data

The simulated data consists of Gaussian envelope transients as well as three different types of noise [13]. 1/f-noise is one of the types of noise and is also called pink noise. To create it, temporally and spatially uncorrelated white Gaussian noise is filtered to get the desired power spectral density of $\frac{1}{f}$. The other types of noise are α-noise and sensor noise. α-noise is created using white Gaussian noise signals that are filtered using a 6^{th}-order Butterworth filter with a lower cutoff frequency of 8 Hz and a higher cutoff frequency of 12 Hz. Sensor noise is modelled as spatially and temporally uncorrelated white Gaussian noise.

The amount of noise in the data is set by SNR_{dB}, the SNR in a decibel scale, and will be referred to as simply SNR from now on. An SNR of about 20 dB means a very low amount of noise while a slightly negative SNR means quite a lot of noise, and this is a level that is similar to the amount of noise in real EEG data. The signals in the simulated sequence were three Gaussian envelope transients with both random and set values that can be seen in table 3.2.

In figure 3.1 two scaled reassigned spectrograms on simulated Gaussian envelope transient data are seen. The data contains three Gaussian envelope transients with time centre values at 2 seconds and the frequencies 5, 10 and 15. The SNRs are 13 dB and 10 dB respectively.

Input parameter	C1 & C2	C3
Data vector length	1024	2000
Sample frequency	256	500
Gaussian component lengths	512	1000
Amplitudes	$U(0.5, 1.5)$	$U(0.5, 1.5)$
Time centre values (s)	2	2
Frequency values of each component (Hz)	4, 8 & 12	5, 10 & 15
Phases	$U(0, 2\pi)$	$U(0, 2\pi)$

Table 3.2: General choices for the simulated Gaussian envelope transient and the choice for the signal that is similar to the real data. C1 is case 1 where the second class has a different time, C2 is case 2 where the second class has a different frequency and C3 is case 3 where the second class does not contain any signals.

3.2.1 Approach

When simulating the data, the SNR was initially set to 13 dB. This was the lowest value for SNR where one could clearly see the three peaks in power. The lowest chosen value for SNR was -1 dB. The amplitude of the signal set to a random value uniformly distributed between 0.5 and 1.5.

The first step was to create two classes of simulated EEG data, with some small difference between the two data classes. For all choices, the first class had the input parameters seen in table 3.2, and the second class had the same input parameters unless something else is specified.

The first difference tested was a small difference in the time (case 1). The choices for the second class was 2.1 seconds instead of 2 seconds as for the first class. The second difference was in frequency (case 2), where the first class had frequencies 4, 8 and 12 and the second class had 5, 10 and 15. The last difference tested was where only the first class had a signal and the second class contained solely noise (case 3). The first class had a signal with conditions similar to those of the real data. A comparison between this and the real data can be seen in figure 3.2.

For all the cases, half of the data was of the first class and the other half was of the second class. For cases 1 and 2, the number of data points in the data sets was 200, and for case 3 it was 352. The test sets for cases 1 and 2 had the sizes 60, and for case 3 it had size 102. The Gaussian envelope transients were created for the above-specified cases using the input parameters seen in table 3.2. The data was created by adding noise to these.

CNN
imageInputLayer([m 1 1])
convolution2dLayer([60 1], 4, 'Padding', 'same')
batchNormalizationLayer
reluLayer
dropoutLayer(0.3)
fullyConnectedLayer(2)
softmaxLayer
classificationLayer

(a) CNN for the first two cases of simulated raw data and periodogram data, where m is 1024 for raw data and 2048 for the periodogram.

CNN
imageInputLayer([490 64 1])
convolution2dLayer([30 10], 4, 'Padding', 'same')
batchNormalizationLayer
reluLayer
dropoutLayer(0.3)
fullyConnectedLayer(2)
softmaxLayer
classificationLayer

(b) CNN for the first two cases of simulated spectrogram data and scaled reassigned spectrogram data.

CNN
imageInputLayer([m 1 p])
convolution2dLayer([60 1], 16, 'Padding', 'same')
batchNormalizationLayer
reluLayer
dropoutLayer(0.3)
fullyConnectedLayer(2)
softmaxLayer
classificationLayer

(c) CNN for the third case of simulated data and real data, using raw data and the periodogram, where m is 2000 for the raw data and 4096 for the periodogram.

CNN
imageInputLayer([490 125 p])
convolution2dLayer([30 10], 16, 'Padding', 'same')
batchNormalizationLayer
reluLayer
dropoutLayer(0.3)
fullyConnectedLayer(2)
softmaxLayer
classificationLayer

(d) CNN for the third case of simulated data and real data, using the spectrogram and the scaled reassigned spectrogram.

Table 3.3: CNNs using simulated and real data. p is either 1 or 2 for real data, depending on whether the data is originated from the left or right ear, or both ears. For simulated data p is 1.

The data in the three cases were first inserted into the CNN seen in table 3.3a for cases 1 and 2, and the CNN seen in table 3.3c for case 3. Secondly, the periodograms for the cases were computed using the number of fast Fourier transform samples as seen in table 3.4 and inserted into the same neural networks as the raw data (table 3.3a or 3.3c). Thereafter the spectrogram was computed, using a Matlab function created by Maria Sandsten, for the three cases using the inputs seen in table 3.4. The scaled reassigned spectrogram was computed using the same inputs as the spectrogram as well as the reassignment level also seen in table 3.4, using the same Matlab function

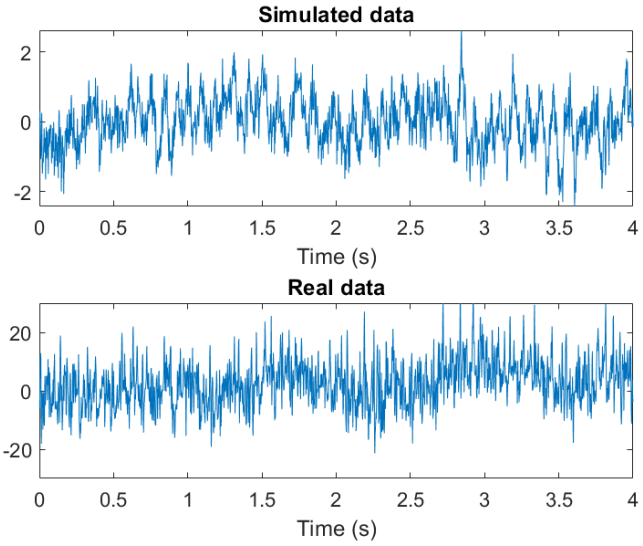

Figure 3.2: Simulated data plotted with real data. The simulated data has inputs similar to the ones for the real data and an SNR of 0 dB.

Input parameter	**C1 & C2**	**C3 & real**
Number of FFT-samples	4096	8192
Window length (Gaussian)	256	512
Time-step between two spectrum calculations	16	16
Reassignment level	0.01	0.01

Table 3.4: The inputs for the periodogram, the spectrogram and the scaled reassigned spectrogram. C1 is case 1, C2 is case 2 and C3 is case 3 for the simulated data. Real is all real data. For the periodogram the only used input parameter is the number of FFT-samples, for the spectrogram all except the reassignment level are used and for the scaled reassigned spectrogram they are all used.

as for the spectrogram. Both the spectrograms and the scaled reassigned spectrograms were inserted into the CNN in table 3.3b for cases 1 and 2, and the CNN in table 3.3d for case 3.

The chosen validation frequency for the first two cases of data was 7, the number of epochs was 20 and the mini-batch size 20. For case 3 the validation frequency was 10, the number of epochs was 50 and the mini-batch size was 25.

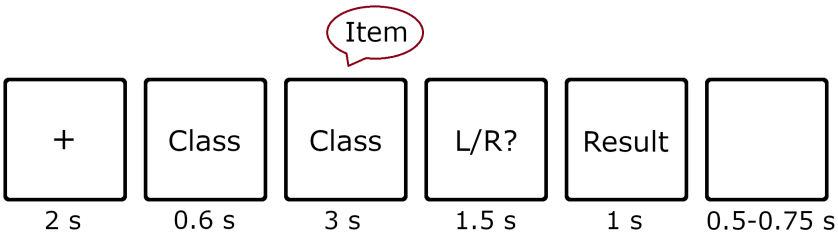

Figure 3.3: An illustration of the different parts of the EEG data. The second and third box from the left is where a word is showed on a screen, and in the third box, a word is spoken in the persons left or right ear. In the fourth box, the person gets to guess which ear they heard the word in, and in the fifth box, they get to know the correct answer.

3.3 Real Data

The data of 352 measurements are real EEG data measured from a single person. Each signal consisted of 4501 samples with a sampling frequency of 500 Hz, originally 1000 Hz but was then downsampled. The EEG data were recorded using a Neuroscan SynAmps RT (Compumedics) amplifier, Curry 7 software and 64 active Ag/AgCl electrodes. These were mounted in an EasyCap using the 10% system covering the 10/20 area with 60 channels.

The recording of the EEG was during approximately 9 seconds, where the first 2 seconds consists of solely noise. In the next 0.6 seconds, a word is showed on a screen in front of the person. The word is a class, for example, "Dance". After this, there are 3 seconds where the word is left on the screen, but now the person hears a spoken word in either the left or the right ear. The spoken word is an item in the class shown on the screen, for example, "Salsa" in the class "Dance". If the person hears the word in the left or the right ear is random, and after these 3 seconds the person has 1.5 seconds to decide whether the sound was in the left or the right ear. Under the following 1 second, the person gets to know if they answered correctly. The recording stops after an additional 0.5-0.75 seconds. The arrangement of the signal can be seen in figure 3.3.

The data was preprocessed with a high-pass filter at 0.1 Hz and baseline corrected based on the average of all the different parts of the data (figure 3.3) to minimize offsets. Line noise, or random electrical impulses, at 50 Hz were reduced, and other flaws caused by equipment were removed by visual inspection. Independent component analysis (ICA) was applied to remove artefacts caused by the eyes or muscles.

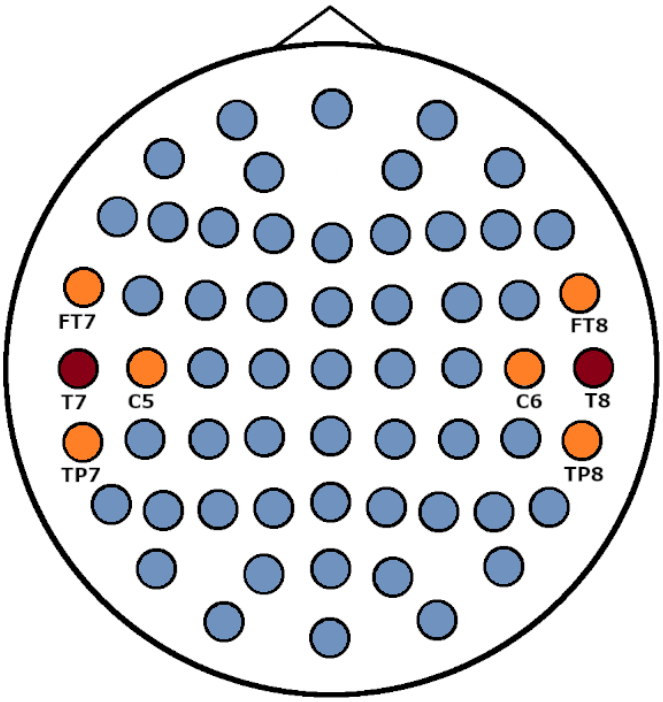

Figure 3.4: An EEG map of the 60 channels in the data set. The EEG map should be viewed as seen from the top of the head, and the arrow is placed where the nose would be. The small red circles are the main channels used in this thesis, and the orange circles are other channels that are used. The blue circles are the channels in the data set that are not used in this thesis.

In the data set used in this thesis, there are 60 channels and these are named as seen in figure 3.4. It is seen in the figure that the channels are positioned evenly over the scalp to be able to see activity in as many parts of the brain as possible. The hearing centres of the brain are positioned close to the ears, these are called the auditory cortex [16].

3.3.1 Approach

The first step was to decide how much of the original signal of 4501 samples should be used. The choice was based on where the sound occurred, which was from sample 1301 to sample 2800, and the choice was from sample 1001 to sample 3000. This was done to capture the entirety of the signal, in case of small time-shifts in the data. Firstly two cases of data were constructed; one with channel T7 near the left ear and one with channel T8 near the right ear.

Consider the dimensions of the data inputs to the CNNs. For the raw data and the periodogram, the data consists of vectors sizes m, where $m = 2000$ for the raw data and $m = 4096$ for the periodogram data. There is also the number of data measurements, 352, which leaves the 2-dimensional input $m \times 352$. For the spectrogram and the scaled reassigned spectrogram, the data consists of matrices sizes 490×125. Adding the number of measurements, the 3-dimensional input $490 \times 125 \times 352$ is obtained.

At this point, only data containing information about one ear was used for the input to the CNNs, either channel T7 near the left ear or channel T8 near the right ear. Another dimension was added, which enabled the usage of information about both ears simultaneously. For the raw data and the periodogram, the inputs were 3-dimensional and had the sizes $m \times 2 \times 352$. For the spectrogram and the scaled reassigned spectrogram, the inputs were 4-dimensional and had the sizes $490 \times 125 \times 2 \times 352$. This was firstly used for the data sets of channel T7 and channel T8.

The next step was to use more than one channel near each ear. The choice was four channels near each ear; T7, FT7, TP7 and C5 for the left ear, and T8, FT8, TP8 and C6 for the right ear. Firstly these were combined using a simple mean value for channels near the left ear, channels near the right ear as well as channels near the left and the right ear combined. Then the SVD was used by laying the four data vectors as columns in a matrix and applying SVD to this matrix. The first column in U times 1000 was the one that was used, because it looked similar to data vectors as seen in figure 3.5. This was done for channels near the left ear, channels near the right ear and channels near the left and the right ear combined. A comparison of the mean value data and the SVD data can be seen in figure 3.6.

In conclusion, nine cases of real data were created; the channel T7, the channel T8, the channels T7 and T8 combined, the mean value of channels near the left ear, the mean value of channels near the right ear, the mean value of channels near both the left and the right ear, channels near the left ear combined using SVD, channels near the right ear combined using SVD, as well as channels near both the left and the right ear combined using SVD separately. For all of these nine cases, the data was first inserted into the CNN seen in table 3.3c, using a test set size of 102. Secondly, the periodograms for the nine cases were computed, the mean values were subtracted and the periodograms were inserted into the same neural network as the raw data (3.3c). Thereafter the spectrogram was computed for the nine cases using the inputs seen in table 3.4. The scaled reassigned spectrogram was computed using the same inputs as the spectrogram as well as the reassignment level also seen in table 3.4.

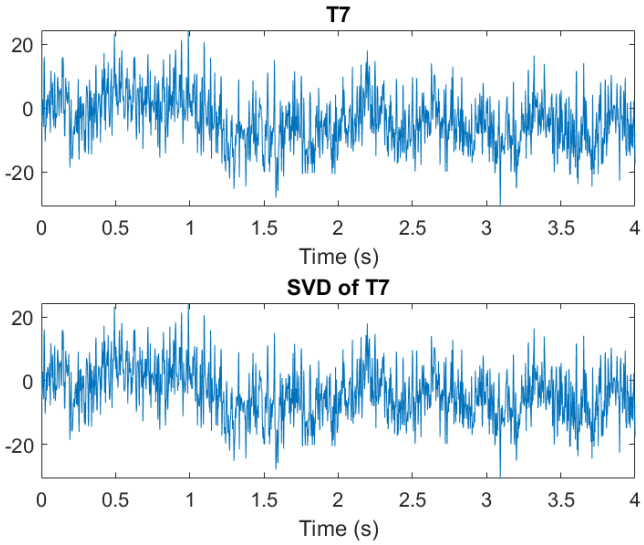

Figure 3.5: The channel T7 and the first column of U in the SVD of the channel T7 of one single data point where in the SVD plot, the first column of U is multiplied by the first singular value found in S.

After the spectrogram and the scaled reassigned spectrogram were created they were cut in frequency, from originally a maximum frequency of 250 Hz to the maximum frequency of 30 Hz. This was done because nothing of interest was seen above 30 Hz and capacity of the neural network was saved. After this both the spectrograms and the scaled reassigned spectrograms where inserted into the CNN in table 3.3d. The number of epochs used for the real data was 50, the mini-batch size was 25 and the validation frequency 10.

All values in table 3.4 were evaluated due to the performance they provided. The window length was decided by looking at plots with different window lengths and finding that 256 or 512 would be the best choices. Both were tested in the neural network and 256 performed slightly better for SVD data while 512 performed better for the rest of the cases. Because the data was not stationary, the mean of the data was subtracted before applying any spectral methods to the data. For the spectrogram and the scaled reassigned spectrogram, this was done using the Matlab functions created by Maria Sandsten.

The CNN was mainly optimised with respect to the average of both ears and the SVD of both ears. This should be the optimal CNN for all methods due to the fact that the problem is very similar for all the methods, what

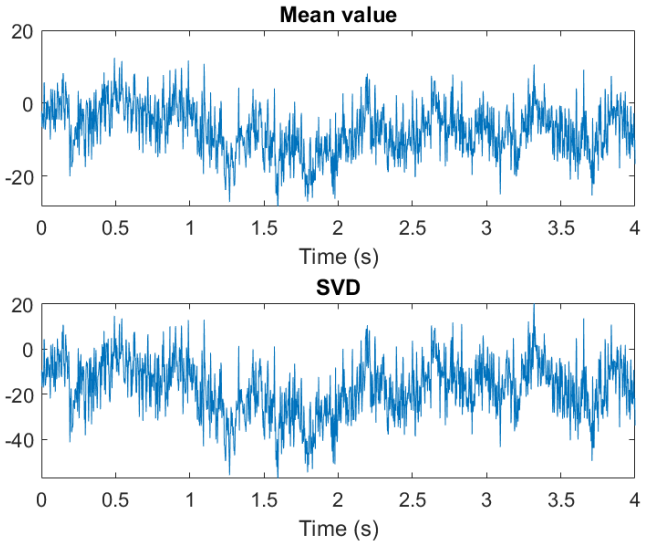

Figure 3.6: The mean value and the first column of U in the SVD of the channels T7, FT7, TP7 and C5 of one single data point. In the SVD plot the first column of U is multiplied by the first singular value found in S, and -1 for comparison reasons.

is put into the network is data sets that are very similar to each other for all methods. However, there is a difference between the raw data and the periodogram, and the spectrogram and the scaled reassigned spectrogram. This is because of the difference in dimensions of the transforms, which means that the inputs into the CNNs and the convolutional filters will have different sizes.

Chapter 4

Results and Discussion

In this chapter, the classification results are presented. In the following tables, "Raw" stands for the unprocessed data, "Per" is the periodogram, "Spec" is the spectrogram and "ScReSp" is the scaled reassigned spectrogram.

For comparison with the accuracies presented in this chapter, the upper percentiles of the Binomial distribution with a success rate of 0.5 were calculated at a significance level of 0.05. The number of trials is the same as the sizes of the test sets, which is 60 for the simulated data in cases 1 and 2, and 102 for case 3 and the real data. For the simulated data in cases 1 and 2, the percentile is 59.76%. For the simulated data in case 3 and the real data, it is 57.64%. When the real data is run three times the number of trials is 306 and the percentile is 54.54%. Classification results larger than the percentiles are considered as being significant.

4.1 Simulated Data

The simulated data can be categorized into three cases; data that has differences in time, in frequency and one that only has signals present in one of the classes. The results for SNR are presented in a logarithmic scale and has the maximum SNR at 13 dB for cases 1 and 2. 13 dB is the lowest SNR where the signal clearly can be seen with the eye.

In table 4.1 the classification results for the time difference of 0.1 seconds between the classes are presented. In table 4.2 the results for the difference in frequency is shown, where the first class had frequencies 4, 8 and 12 and the second class had frequencies 5, 10 and 15. The results for the data set where one class had signals and one class had no signals at all is presented in table 4.3.

SNR	Raw	Per	Spec	ScReSp
13	83.33%	48.33%	85.00%	80.00%
10	50.00%	50.00%	53.33%	55.00%
7	51.67%	46.67%	51.67%	46.67%
4	58.33%	53.33%	65.00%	53.33%
2	60.00%	51.67%	51.67%	43.33%
0	53.33%	30.00%	53.33%	46.67%

Table 4.1: Classification results where there is a difference in time.

SNR	Raw	Per	Spec	ScReSp
13	83.33%	100.00%	98.33%	100.00%
10	55.00%	96.67%	98.33%	90.00%
7	45.00%	70.00%	71.67%	63.33%
4	48.33%	55.00%	40.00%	48.33%
2	46.67%	48.33%	55.00%	55.00%
0	50.00%	48.33%	50.00%	50.00%

Table 4.2: Classification results where there is a difference in frequency.

From table 4.1 we get that the raw data and the spectrogram are the two transforms that gave the best classification results. The scaled reassigned spectrogram gave a slightly worse result than these two. In the table, it is seen that the periodogram should not show any recognition. This is because the periodogram does not depend on the time, which can be seen in figure 2.1.

For all the transforms in table 4.1 it can be seen that when the SNR decreases the classification results also decreases. A quite large difference in the results for SNR 13 and SNR 10 can be seen. However, this can partly depend on the random shuffling of the data, which is done when the SNR is changed. At SNR 2 there is a significant recognition for the raw data, but not for the other transforms. At SNR 0 there are no applicable results. For the scaled reassigned spectrogram there are no significant results below SNR 10.

In table 4.2 it is seen that the transforms with the best results are the periodogram, the spectrogram and the scaled reassigned spectrogram, where the scaled reassigned spectrogram performed slightly worse than the other two. The raw signal did have a significant result, but only for SNR 13. At SNR 2 the spectrogram and the scaled reassigned spectrogram has almost applicable results, but below this, there are none for all of the transforms. The last significant results are found at SNR 7.

SNR	Raw	Per	Spec	ScReSp
4	62.75%	50.98%	54.90%	52.94%
2	47.06%	57.84%	54.90%	46.08%
0	52.94%	50.98%	48.04%	48.04%
-1	52.94%	46.08%	44.12%	44.12%

Table 4.3: Classification results where one class has no signals.

Comparing table 4.1 and table 4.2 it can be seen that at SNR 13 the results in table 4.2 are better than the results in table 4.1. When the SNR decreases the classification results decrease for both, but faster for the data where one class has different frequencies, and at a low SNR both data sets have similar classification results.

The best results obtained when mimicking the real data with simulated data, seen in table 4.3, seem to be from using raw data. All of the data transforms gave at least one applicable result each, except possibly the scaled reassigned spectrogram. The results at SNR 4 are only slightly better than those for SNR 0 and SNR -1, but the only significant results in table 4.3 are found for the raw data at SNR 4 and the periodogram at SNR 2.

Comparing the results in table 4.3 to the results in table 4.1 and table 4.2 we see that the results are quite similar. Both in table 4.3 and table 4.1 some significant results for SNR 4 and SNR 2 are seen, but none for SNR 0.

Looking at the training processes for the four transforms in figure 4.1, figure 4.2, figure 4.3 and figure 4.4 it can be seen that the raw data and the scaled reassigned spectrogram seem to converge the fastest. The spectrogram does not seem to converge fully during the 50 epochs of training. The periodogram converges quite quickly. Overfitting is seen in all the plots.

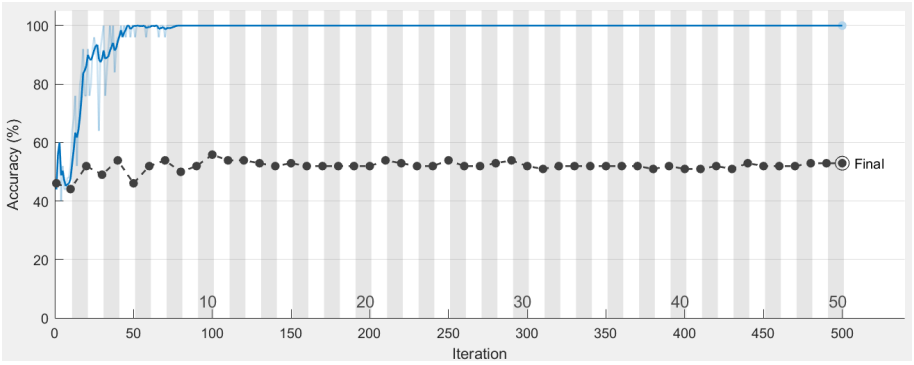

Figure 4.1: The training process of the unprocessed data for case 3 with SNR 0 dB and a classification result of 52.94%.

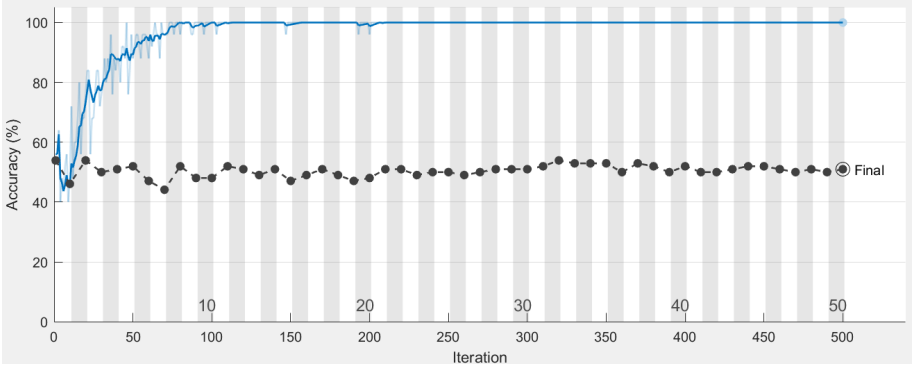

Figure 4.2: The training process of the periodogram data for case 3 with SNR 0 dB and a classification result of 50.98%.

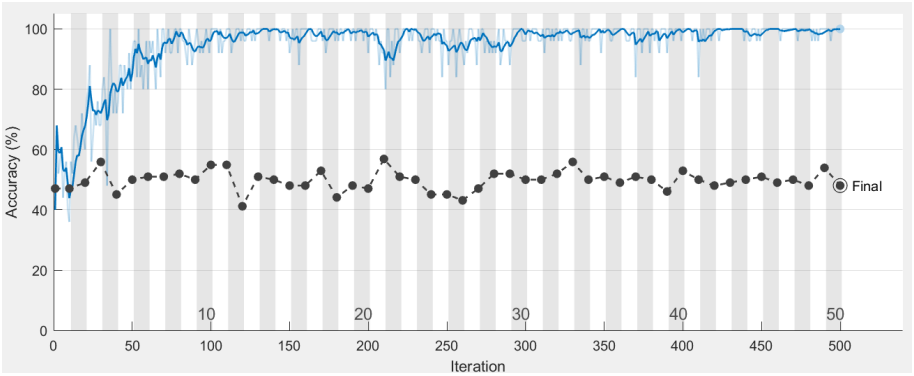

Figure 4.3: The training process of the spectrogram data for case 3 with SNR 0 dB and a classification result of 48.04%.

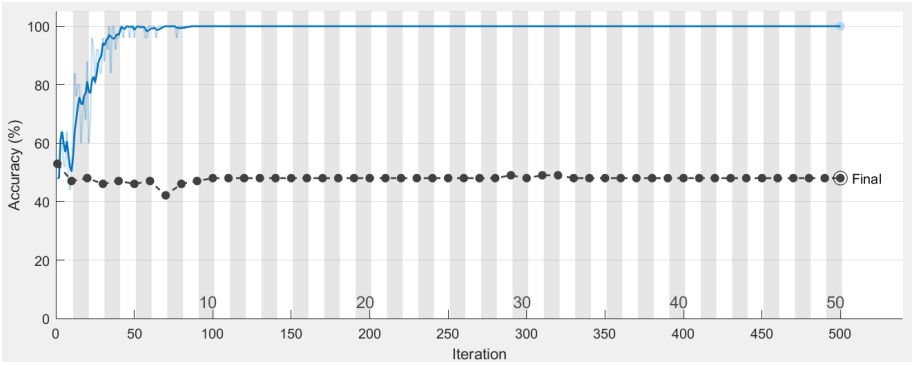

Figure 4.4: The training process of the scaled reassigned spectrogram data for case 3 with SNR 0 dB and a classification result of 48.04%.

4.2 Real Data

For all the classification results on real data, the network was run three times. All the results are presented as well as the mean values and the standard deviations of the three runs. The results for the nine cases of data are presented in the tables below. In table 4.4 the classification results for solely channel T7 is presented and in table 4.5 the results for channel T8. In table 4.6 the results for the channels T7 and T8 combined are seen.

The results for the mean values of the channels T7, FT7, TP7 and C5 are presented in table 4.7 and the mean values of T8, FT8, TP8 and C6 are shown in table 4.8. For the mean values of the channels T7, FT7, TP7 and C5 together with the mean values of T8, FT8, TP8 and C6 the results are presented in table 4.9. The results for the SVD of the channels T7, FT7, TP7 and C5 are shown in table 4.10 and in table 4.11 the results of the channels T8, FT8, TP8 and C6 are shown. In table 4.12 the results for the SVDs of the channels T7, FT7, TP7 and C5, and the channels T8, FT8, TP8 and C6 are presented.

Comparing the classification for the channels T7 and T8 near the left and the right ear, it can be seen in table 4.4 and table 4.5 that the classification for the channel T8 does better for all the four transforms. Though the classification for channel T8 has a higher standard deviation than for channel T7, it still seems to be the better choice. The difference in standard deviation is small and the difference in accuracy is quite large.

Looking at the difference between the transforms it is seen that the raw data and the scaled reassigned spectrogram does much better than the periodogram and the spectrogram. While the spectrogram has higher accu-

	Raw	**Per**	**Spec**	**ScReSp**
1	54.90%	40.20%	44.12%	52.94%
2	49.02%	47.06%	48.04%	52.94%
3	53.92%	47.06%	50.00%	50.98%
Mean	52.61%	44.77%	47.39%	52.29%
Std	3.15%	3.96%	2.99%	1.13%

Table 4.4: Classification results for channel T7 near the left ear.

	Raw	**Per**	**Spec**	**ScReSp**
1	58.82%	40.20%	50.00%	59.80%
2	57.84%	42.16%	57.84%	60.78%
3	63.73%	49.02%	48.04%	56.86%
Mean	60.13%	43.79%	51.96%	59.15%
Std	3.16%	4.63%	5.19%	2.04%

Table 4.5: Classification results for channel T8 near the right ear.

racy than the periodogram, the spectrogram only has a classification accuracy that is slightly above 50.00% (51.96%) and a large standard deviation (5.19%). It seems like none of the classifications for the periodogram and the spectrogram using the channels T7 and T8 gives any applicable results.

The raw data and the scaled reassigned spectrogram gave the best results for the channels T7 and T8, and the raw data gave slightly better results. However, the raw data had a higher standard deviation than the scaled reassigned spectrogram, and because of this, the raw data might not be the better choice of the two. For channel T7 none of the transforms gave any significant results, but for channel T8 both seem to be useful. The reason for this could be due to fault in connection between the person's scalp and the EEG cap.

The results for the classification of channel T7 and T8 combined, in table 4.6, seem to be similar to the ones for the two channels separately. One difference found is that the spectrogram performed slightly worse than the periodogram. There is no clear reason for this, and it might just be a coincident, because the standard deviation for the spectrogram is quite large. The periodogram performed almost the same as it did for the single channel T8, except with slightly higher accuracy, and with a higher standard deviation as well. The spectrogram performed worse than it did for both the single channels.

	Raw	**Per**	**Spec**	**ScReSp**
1	57.84%	44.12%	49.02%	56.86%
2	51.96%	36.27%	40.20%	51.96%
3	60.78%	41.18%	48.04%	56.86%
Mean	56.86%	40.52%	45.75%	55.23%
Std	4.49%	3.97%	4.83%	2.83%

Table 4.6: Classification results for the channels T7 and T8.

The accuracies for the raw data and the scaled reassigned spectrogram is somewhere in between the ones for the raw data and the scaled reassigned spectrogram of the two single channels. As for the single channels, the raw data has a better classification accuracy, but it also has a higher standard deviation. Overall the combination of the two channels had a higher standard deviation than for both the single channels.

Looking at the results for the mean values near the left ear and near the right ear in the tables 4.7 and 4.8, it can be seen that the raw data and the scaled reassigned spectrogram still gave the best results. The periodogram and the spectrogram gave no significant results, though they had a quite low standard deviation. For the mean value near the left ear the raw data had a somewhat higher accuracy, but a much higher standard deviation than the scaled reassigned spectrogram. For this reason, the scaled reassigned spectrogram might be the better choice here.

For the mean value near the right ear, the raw data performed a lot better than the scaled reassigned spectrogram, and it also had a lower standard deviation.

The classification of the mean values near the left ear and the right ear combined, seen in table 4.9, has no significant results for the periodogram and the spectrogram. For the raw data, the accuracy is low, meaning no significant results, and the standard deviation is fairly large. The accuracy of the scaled reassigned spectrogram is quite large, but the standard deviation is large as well. The results for the combination of the mean values are overall neither better nor worse than for the mean values separately. For the raw data, the combination is worse, but for the three other transforms it is slightly better.

Comparing the results for the mean values of four channels to the ones of single channels it seems like the single channels give more stable results. Patterns between the three versions of it can be seen while not so many patterns can be seen for the mean values. The single channels seem to give

	Raw	**Per**	**Spec**	**ScReSp**
1	56.86%	48.04%	49.02%	56.86%
2	61.76%	49.02%	50.00%	56.86%
3	52.94%	43.14%	48.04%	56.86%
Mean	57.19%	46.73%	49.02%	56.86%
Std	4.42%	3.15%	0.98%	0.00%

Table 4.7: Classification results using mean values for channels near the left ear.

	Raw	**Per**	**Spec**	**ScReSp**
1	56.86%	42.16%	45.10%	53.92%
2	52.94%	43.14%	45.10%	55.88%
3	60.78%	41.18%	45.10%	47.06%
Mean	56.86%	42.16%	45.10%	52.29%
Std	3.92%	0.98%	0.00%	4.63%

Table 4.8: Classification results using mean values for channels near the right ear.

	Raw	**Per**	**Spec**	**ScReSp**
1	51.96%	42.16%	50.98%	61.76%
2	49.02%	40.20%	52.94%	50.98%
3	53.92%	44.12%	42.16%	55.88%
Mean	51.63%	42.16%	48.69%	56.21%
Std	2.47%	1.96%	5.74%	5.40%

Table 4.9: Classification results using mean values for channels near both ears.

better accuracies than the mean values, though the difference is very small.

In table 4.10 where the results for the SVD near the left ear is showed, it can be seen that the only significant result is for the scaled reassigned spectrogram. All the others have accuracies below 50%, though the spectrogram is close to an accuracy of 50%. The raw data and the periodogram have large standard deviations and the spectrogram has a lower standard deviation. The scaled reassigned spectrogram has one of the highest accuracies in this thesis and a quite low standard deviation.

From the results for SVD near the right ear seen in table 4.11 it seems like the raw data performed somewhat better than the other methods, the standard deviation is only fairly larger than for the scaled reassigned spec-

	Raw	**Per**	**Spec**	**ScReSp**
1	42.16%	51.96%	47.06%	57.84%
2	51.96%	48.04%	50.00%	57.84%
3	50.98%	50.98%	50.00%	60.78%
Mean	48.37%	50.33%	49.02%	58.82%
Std	5.40%	2.04%	1.70%	1.70%

Table 4.10: Classification results using SVD for channels near the left ear.

	Raw	**Per**	**Spec**	**ScReSp**
1	55.88%	47.06%	49.02%	52.94%
2	53.92%	46.08%	54.90%	52.94%
3	53.92%	47.06%	49.02%	53.92%
Mean	54.57%	46.73%	50.98%	53.27%
Std	1.13%	0.57%	3.39%	0.57%

Table 4.11: Classification results using SVD for channels near the right ear.

	Raw	**Per**	**Spec**	**ScReSp**
1	54.90%	50.00%	50.00%	59.80%
2	62.75%	48.04%	58.82%	57.84%
3	50.98%	50.00%	48.04%	60.78%
Mean	56.21%	49.35%	52.29%	59.47%
Std	5.99%	1.13%	5.74%	1.50%

Table 4.12: Classification results using SVD for channels near both ears.

trogram. The scaled reassigned spectrogram performed quite well too and has a low standard deviation. For both the SVD near the left ear and the SVD near the right ear, the periodogram and the spectrogram gave no significant results.

The best classification result for the SVD near both ears, seen in table 4.12, is for the scaled reassigned spectrogram. It has an accuracy close to 60% and a fairly low standard deviation. The raw data has a quite good accuracy, but a large standard deviation. The spectrogram has an acceptable accuracy but a large standard deviation and the periodogram has low accuracy, so they are both not very beneficial.

Overall the SVD for both ears combined performed better than the SVDs separately. The most stable case of data still seems to be the single data. Comparing the results for the SVD to the results for the mean value and the single data as seen in figure 4.13, it is seen that the single channels

	Ear	Raw	Per	Spec	ScReSp
Single	Left	52.61%	44.77%	47.39%	52.29%
	Right	60.13%	43.79%	51.96%	59.15%
	Both	56.86%	40.52%	45.75%	55.23%
	Mean	56.53%	43.03%	48.37%	55.56%
Mean value	Left	57.19%	46.73%	49.02%	56.86%
	Right	56.86%	42.16%	45.10%	52.29%
	Both	51.63%	42.16%	48.69%	56.21%
	Mean	55.23%	43.68%	47.60%	55.12%
SVD	Left	48.37%	50.33%	49.02%	58.82%
	Right	54.57%	46.73%	50.98%	53.27%
	Both	56.21%	49.35%	52.29%	59.47%
	Mean	53.05%	48.80%	50.76%	57.19%
	Total mean	54.94%	45.17%	48.91%	55.95%

Table 4.13: Summary of the classification results, with mean values, within the three versions of the data and the total mean values.

performed better overall for the raw data and the periodogram, and the SVD data performed better for the spectrogram and the scaled reassigned spectrogram. The overall best results come from using the scaled reassigned spectrogram.

Figure 4.5, figure 4.6, figure 4.7 and figure 4.8 shows the training processes that gave the highest accuracies from each data transform are shown. From the training processes, it can be seen that the most stable data transform seem to be from using the scaled reassigned spectrogram. The least stable training process seems to be from using the spectrogram. All the loss functions are similar and do decrease compared to the initial loss, except for the periodogram where the final loss is greater than the initial loss.

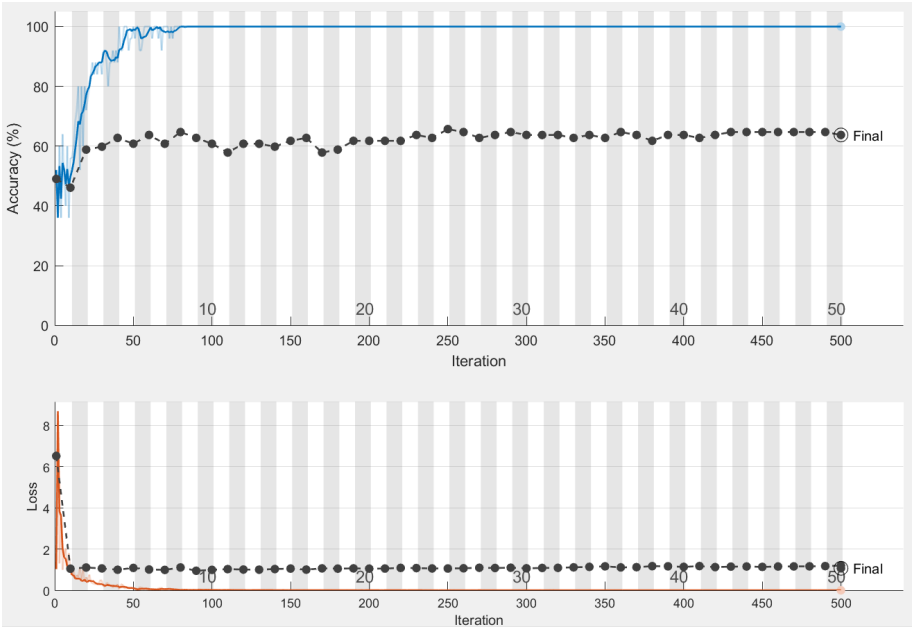

Figure 4.5: The training process of the unprocessed channel T8 with a classification result of 63.73%.

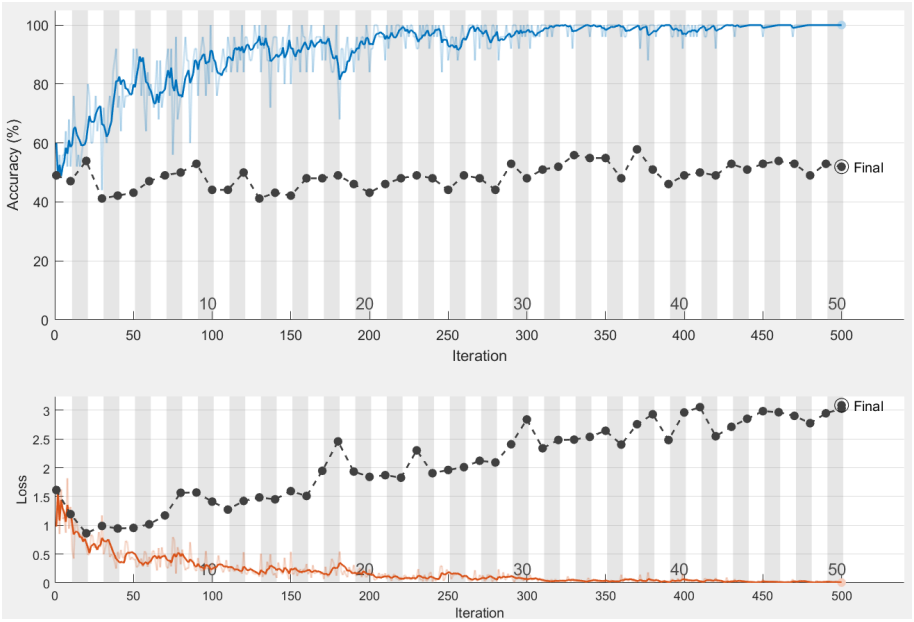

Figure 4.6: The training process of the periodogram on SVD data for the left ear with a classification result of 51.96%.

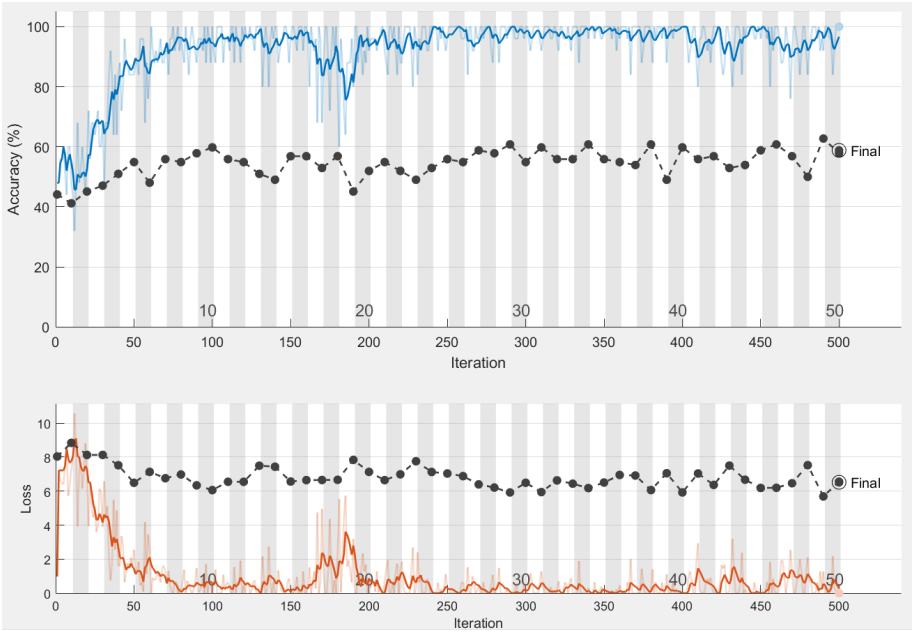

Figure 4.7: The training process of the spectrogram on SVD data for both ears with a classification result of 58.82%.

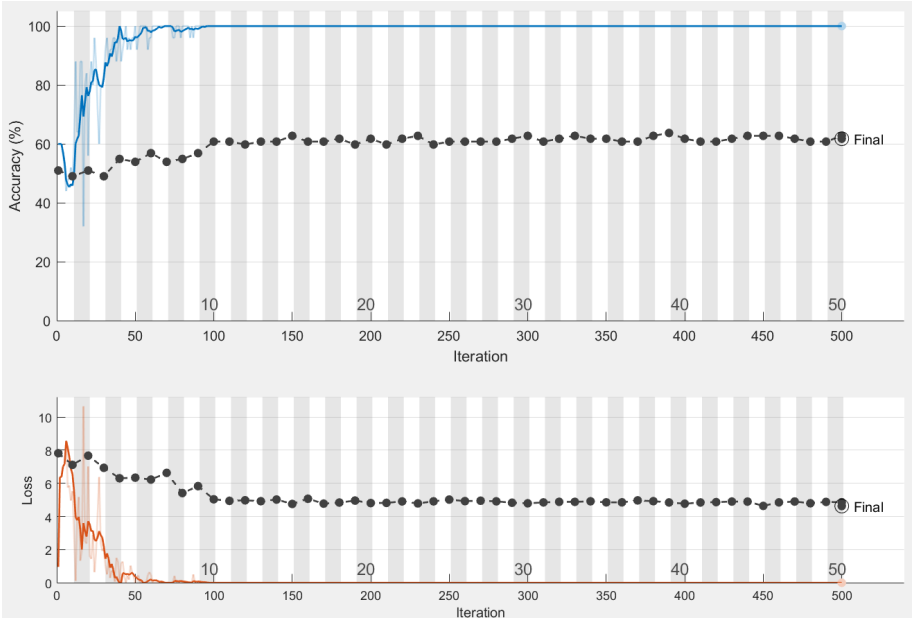

Figure 4.8: The training process of the scaled reassigned spectrogram on mean value data for both ears with a classification result of 61.76%.

Chapter 5

Conclusion

The two best results regarding the real data in this thesis are for the channel T8 using raw data and for the channels combined using SVD near both ears using the scaled reassigned spectrogram. The single channel T8 gave a slightly better classification result than the channels combined using SVD, but the single channel T8 had a higher standard deviation.

A reason for preferring the method of using channels combined using SVD near both ears with the scaled reassigned spectrogram is that it is more robust and seem to be the method that would work the best if the experiment was to be done on new but similar data. When using an EEG cap one can never be sure that all the channels are connected properly to the scalp and that no information will get lost. By using several channels near each ear and combining them using SVD it does not matter as much if a channel is missing information, because the other channels can still contain this information.

Comparing the data set with channels combined using SVD and the mean value data it can be seen that the SVD method performed better for all data transforms except for the raw data. The scaled reassigned spectrogram performed the best when using the SVD method.

Looking at the simulated data in table 4.3, which is constructed to look like the real data, some similarities and some differences between this and the real data in table 4.4 and table 4.5. A similarity is found when looking at the training processes in the tables 4.1, 4.2, 4.3 and 4.4 for the simulated data and the tables 4.5, 4.6, 4.7 and 4.8 for the real data. It can be seen that for both simulated and the real data, the training processes of the raw data and the scaled reassigned spectrogram did both converge fast, while the spectrogram did not converge properly during the entire training process. The periodogram converged faster for the simulated data, but still not as

fast as for the raw data and the scaled reassigned spectrogram.

The overall performances of the real data are better than for the simulated data with low SNRs. The raw data performed the best for both the simulated data and the real data. Some differences between the simulated data and the real data are the performances of the periodogram, the spectrogram and the scaled reassigned spectrogram. For the simulated data the periodogram was the second best method after the raw data, while the periodogram was the worst method for the real data. For the real data the scaled reassigned spectrogram performed better than the spectrogram, but it was the other way around for the simulated data.

This indicates that the simulated data might not be very similar to the real data. Indications of this can also be seen in figure 3.2 where the real data seem to look more like white noise while the simulated data looks more structured.

The purpose of working with simulated data was to find out how well the network could classify the data with different levels of noise, and as a conclusion, it was quite difficult for the CNNs at low SNRs. As EEG data is known to carry a large amount of noise, classifying it can be quite a challenge. The neural network did a good job to classify for certain transforms of the data and combinations of channels, and worse for others.

An issue with the data when training was to avoid overfitting. A solution to this is to simplify the network by removing features [17]. A neural network with only one convolutional layer gave the best test accuracy as well as less overfitting compared to more complex networks. Early stopping and using a dropout layer was also tested. Early stopping gave worse results, but adding a dropout layer improved the result, and was therefore used.

Fewer filters and higher dropout was tested but gave worse accuracy and not much less overfitting. For the scaled reassigned spectrogram with SVD on both ears, when only using two filters and a dropout of 0.6 the accuracy was 51.96%, and when using four filters and dropout 0.5 the accuracy was 56.86%. These are both worse in comparison to the accuracy of 59.47% for the used CNN.

Looking at confusion matrices the distribution of the classification can be seen for the two classes. This distribution varies quite a bit and does not seem to follow any noticeable patterns. However, in most of the training processes, the distribution between the two classes are quite even.

A difference between the training process for the spectrogram and the scaled reassigned spectrogram is that the one for the spectrogram does not reach 100% for the training data in a stable manner. It is possible that the training process for the spectrogram would reach higher results if it had been trained for a longer time.

Several windows for the periodogram were tested, including Kaiser, rectangular, Gaussian and Hamming as well as no window at all, but the Hann (Hanning) window performed the best. Different numbers of FFT-samples were also tested, and the ones that gave the best classification results are the ones that are used.

5.1 Final Conclusion

The best way to classify this type of EEG data seems to be by using the scaled reassigned spectrogram and SVD, with an accuracy of 59.47%. This is the most stable method that can be used if the experience would be redone. The scaled reassigned spectrogram along with the unprocessed data performed better for all cases of the data. The SVD data performed better than the mean value data and the single channels for most of the data transforms.

Comparing the simulated data to the real data there are a few differences. The classification results look different within the transforms, though the training processes have similar behaviours.

5.2 Further Studies

One of the greatest issues in this thesis was the overfitting that occurred in all the training processes. Further measures to prevent this could be considered. Cross-validation is a way of preventing overfitting that is not tested in this thesis, it uses the training data to generate multiple smaller train-test splits. Having more data to train on would also reduce the overfitting.

As seen in figure 4.7 the spectrogram does not converge properly. Training the spectrogram for a longer time could possibly improve the classification result.

Other time-frequency methods could be tested. An interesting example is the Wigner distribution, it has strong cross-terms unlike the time-frequency methods used in this thesis.

Improving the simulated data seem to be necessary, and is something that could be done in the future. The noise in the simulated data looked a bit different from the noise in the real data, and the classification was quite different between the two data sets. However, this could be because of how the data in this thesis looked, and that the simulated data looks similar to other EEG data.

Bibliography

[1] Francois Auger and Patrick Flandrin. *Generalization of the reassignment method to all bilinear time-frequency and time-scale representations*. Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing Acoustics, Speech, and Signal Processing, IEEE International Conference on iv:IV/317-IV/320, vol. 4, 1994.

[2] Maria Hansson-Sandsten and Johan Brynolfsson. *The Scaled Reassigned Spectrogram with Perfect Localization for Estimation of Gaussian Functions*. IEEE Signal Processing Letters, Vol. 22, No. 1, January 2015.

[3] Hans Berger. *Über das Elektrenkephalogramm des Menschen*. Archiv für Psychiatrie und Nervenkrankheiten, Vol. 87, Issue 1, December 1929, pp. 527-570.

[4] `https://www.clinatec.fr/en/research/projects/bci-project/`.

[5] `https://portal.research.lu.se/portal/sv/projects/realtime-individualization-of-brain-computer-interfaces(aacf3f2f-cc57-4fb5-89ea-060e43fc0290).html`.

[6] Georg Lindgren, Holger Rootzén and Maria Sandsten. *Stationary Stochastic Processes for Scientists and Engineers*. CRC Press, Taylor & Francis Group, 2014, pp. 222 & 238.

[7] Isabella Reinhold and Maria Sandsten. *Matched Window Estimation Using Wavelet Denoising for Spectrogram Reassignment*. Mathematical Statistics, Centre for Mathematical Sciences, Lund University, Sweden.

[8] Lloyd N. Trefethen and David Bau, III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM), 1997, pp. 25-29.

[9] Tom M. Mitchell. *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1 March 1997, pp. 2.

[10] Ian Goodfellow, Yoshua Bengio and Aaron Courville. *Deep Learning*. MIT Press, 2016, pp. 96-116, 122-128, 149-152, 243, 274-279, 305-307 & 326-330.

[11] David H. Wolpert. *The lack of a priori distinction between learning algorithms*. NeuralComputation, 1996, pp. 1341–1390.

[12] Mingyuan Xin and Yong Wang. *Research on image classification modelbased on deep convolution neural network*. EURASIP Journal on Imageand Video Processing, 2019.

[13] Elham Barzegaran, Sebastian Bosse, Peter J. Kohler and Anthony M. Norcia. *EEGSourceSim: A framework for realistic simulation of EEG scalp data using MRI-based forward models and biologically plausible signals and noise*. Journal of Neuroscience Methods 328, 2019.

[14] `https://se.mathworks.com/help/deeplearning/ug/layers-of-a-convolutional-neural-network.html`.

[15] `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7407771/`.

[16] Dale Purves, George J. Augustine, David Fitzpatrick, William C. Hall, Anthony-Samuel Lamantia, James O. Mcnamara and S. Mark Williams. *Neuroscience*. Sinauer Associates, Inc., 2004, pp. 309.

[17] `https://elitedatascience.com/overfitting-in-machine-learning`.

[18] `https://se.mathworks.com/help/deeplearning/ref/trainingoptions.html`.