Deep Learning-based Design of Robust Brain-Computer Interfaces

Capstone Report II

Dinmukhamed Sagynbay

Nazarbayev University
Department of Electrical and Computer Engineering
School of Engineering and Digital Sciences

NAZARBAYEV UNIVERSITY

Electrical and Computer Engineering
Nazarbayev University
http://www.nu.edu.kz

Title:
Deep Learning-Based Design of Robust Brain-Computer Interfaces

Theme:
Deep Learning based Brain-Computer Interfaces

Project Period:
Spring Semester 2021

Participants:
Dinmukhamed Sagynbay

Supervisors:
Amin Zollanvari

Copies: 1

Page Numbers: 22

Date of Completion:
May 2, 2021

Abstract:

The field of brain-computer interface has been gaining increasingly more attention recently. The technology has a potential to be of extreme importance for people who suffer from various forms of physical impairment which restrict the control over their bodies. This project focuses on transfer learning based approach for binary motor imagery EEG signal classification. Two deep neural network architectures (ResNet and DenseNet) were applied on four different datasets. The evaluation of the models was carried out using a leave-one-out fashion â€“ signals corresponding to one subject were not included in the training phase, to be later used for validation. According to the obtained results, the proposed approach demonstrates a somewhat better performance than the random classifier. However, more scrutiny is recommended to reach a definite conclusion about the method.

Contents

Preface **iv**

1 Introduction **1**

2 Literature Review **3**

2.1 Transfer Learning . . . 3

2.2 State-of-the-art Results . . . 3

3 Methodology **7**

3.1 Dataset description . . . 7

3.1.1 Weibo2014 . . . 7

3.1.2 Physionet . . . 7

3.1.3 BCI Competition VI - Dataset 2a (BCI-DataSet2A) . . . 8

3.1.4 BCI Competition VI - Dataset 2b (BCI-DataSet2b) . . . 8

3.2 Models . . . 8

3.2.1 ResNet . . . 8

3.2.2 DenseNet . . . 9

3.3 Pre-processing . . . 9

3.4 Stacked Autoencoder . . . 10

4 Results and Discussion **14**

5 Concluding Remarks **18**

Bibliography **19**

Preface

This work was completed during one semester in the framework of an undergraduate level senior project. In this project the author attempts at constructing a deep learning-based classifier model for brain-computer interface purposes, using the transfer learning paradigm.

I would like to express my gratitude to my supervisor, Prof. Amin Zollanvari, for his guidance and continuous feedback throughout the semester. In addition, his course, "Introduction to Big Data" was extremely helpful and relevant for the project.

Also, the project could not have been implemented without a Nazarbayev University alumnus Kassymzhomart Kunanbayev, who kindly offered his expertise in the topic, and was eager to help on every occasion.

Lastly, I would like to thank every staff member at the university and the student government for trying their best to create the most convenient studying environment for the students during these challenging times. Thanks to everyone who creates free educational content on the platform â€œYouTubeâ€, they deserve equal credits as my professors at the university.

Nazarbayev University, May 2, 2021

Dinmukhamed Sagynbay
dinmukhamed.sagynbay@nu.edu.kz

Chapter 1

Introduction

Brain-computer interfaces (BCI) or brain machine interfaces (BMI) are systems that enable people to control various external devices by detecting, processing and decoding brain signals. The signals are also known as brain waves. The nature of these brain waves is characterized by the kind of cognitive processes that take place at a given time. Therefore, by measuring, recording, and processing those signals corresponding commands can be implemented with no use of muscles [1].

BCI applications have the potential to be useful for a large number of people throughout the world. According to [2], in the United States alone there are at least 1.6 million people who live with a limb loss, and more than 5.5 million people who have some form of paralysis, and 50,000 thousand of them suffer from the most severe case - tetraplegia (loss of control over arms, legs and torso). Through BCI specific applications it is possible to alleviate, to a certain degree, the issues these people experience due to physical impairment. As examples of those applications wheelchair control, spelling systems, cursor control and etc. can be mentioned. Nevertheless, application of BCI is not limited by the field of medicine, many other areas nowadays are also making use of the concept such as gaming, smart house design, biometrics etc. [1].

Basically BCI systems comprise five different stages, which are signal detection, preprocessing of the signal, feature extraction, and classification and implementation of the corresponding command [3]. First, the signal is measured utilizing a certain paradigm (EEG, ECoG, fMRI). The second stage, preprocessing, focuses on removing redundant artifacts from the raw signals, in order to eliminate the possibility of errors [4]. The artifacts can be of various origins: surrounding electrical devices and appliances, internal non-cognitive processes of the body, varying resistance of the skin can be mentioned as some of them. Preprocessing stage may utilize a variety of methods to deal with the artifacts. The third stage, feature extraction aims at deconstructing or interpreting the acquired signal as a series of "features", which are specific parameters and values corresponding to a signal [4]. In the next phase, classification, the features are fed into Machine learning and Deep learning algorithms to classify the signal. Eventually, the identified signal is translated into a command [5] [6].

BCI systems can be classified into many different categories with regards to such properties as dependability, recording method, and mode of operation [7]. Dependent systems require some sort of motor control from the user or capable people around them, whereas

independent systems do not impose such requirements, which makes it suitable for people with extreme cases of physical impairment [8]. As for recording methods utilized, there are two techniques for data acquisition: invasive and non-invasive. The non-invasive method has been prominent because of relative simplicity and convenience, with electroencephalography (EEG) acquired data being the most common modality. Lastly, BCI systems might also be classified as synchronous and asynchronous. In synchronous BCIs the interaction has to be implemented within a certain time limit after the cue from the system has been given. Conversely, in the case of asynchronous BCIs, the user can attempt a command at any time. Thus, asynchronous systems are more flexible, however the level of complexity of the design is commensurate [9].

In this work, the focus has been put on the independent modality called motor imagery (MI). MI is a prominent subclass of spontaneous type of signals. As the name suggests, a concept behind the signal is based on the fact that any actual muscle movement or imagining it will stimulate the brain to produce certain oscillations known as sensorimotor rhythms (SMR). As different movements by different body part can cause activation of different regions of the brain it is important that these regions are distinguishable from one another in order for the system to be viable. To illustrate, left hand and right hand movement cause activation in two different hemispheres, hence the signal can be differentiated, whereas EEG signals caused by left foot and right foot are extremely difficult, because the corresponding regions are very close to each other [10].

Building an efficient classifier based on machine learning (ML) has been one of the greatest challenges in the field of BCI. However, a significant progress in the form of new techniques for feature extraction, feature selection and classification for SMR-based BCIs have been made over the last decade. Despite this, surveys on the state-of-the-art of the field indicate that increasingly more research has been focused on deep learning-based classifiers [11]. This might be substantiated by the superior performance demonstrated by deep learning-based systems, which at the same time requires less (or no) pre-processing of the data and liberates from manual feature extraction steps.

In the scope of the project, a transfer learning-based subject-independent BCI classifier model has been built and tested. The term subject-independent is applied when a model is tested (validated) and trained based on data acquired from different individuals. To illustrate, if the leave-one-out approach is used, out of the whole sample, the data corresponding to one of the subjects is "left out", to be later used for the accuracy test of the model that has been trained on the rest of the dataset.

The report is structured as follows. First, a literature survey on transfer learning based and deep learning based BCI's will be given, with state-of-the-art performances outlined. It will be followed by the methodology section in which datasets as well utilized models will be of interest. Next, the results obtained will be presented and discussed. Finally, the work will be summarized in the conclusion section.

Chapter 2

Literature Review

This chapter provides with a background information on transfer learning paradigm as it was of utility in the framework of the project. In addition, state-of-the-art classification results for each dataset used are summarized as well.

2.1 Transfer Learning

Transfer learning (TL) is a machine learning paradigm wherein the idea is to use the knowledge obtained while solving a particular problem and to incorporate it into the solution of a different but related problem. The formal definition can be described as follows. Transfer learning aims at improving the learning rate $f_T(\cdot)$. corresponding to target application given a source domain Ds and a learning task Ts, a target domain Dt and a corresponding learning task Tt, where $Ds \neq Dt$ and$Ts \neq Tt$ [12].

In BCI, the motivation behind making use of transfer learning is based on tackling the issue of calibration. BCI based applications are hindered by the need of time consuming calibrations that are necessary before an actual utilization. This is substantiated by such factors as the high dimensional and non-stationary nature of brain signals and the abundance of noises.

In [13], the authors did an exhaustive survey on transfer learning in the framework of BCI, offering categorizations based on several relevant aspects. However, the project attempts at applying transfer learning in a more general sense. The idea is to use models who have shown good performance in image processing (object recognition) in an attempt to build an EEG signal classifier. Along with the architecture, pre-trained weights of the model will be utilized as initial trained values. In the methodology section, the approach is described in more details.

2.2 State-of-the-art Results

The state-of-the-art classification accuracies for Physionet and BCI competition (2A) datasets are summarized in Tables 2.1 and 2.2, respectively. It can be noticed that a plethora of approaches with varying success levels have been suggested. Most of the research attention

has focused on the subject-dependent paradigm, however, there are still several papers devoted to semi and fully subject-independent classifiers.

Table 2.1: State-of-the-art accuracies obtained for the Physionet dataset

Research paper	Movements	Classification	Subject-independent	Best accuracy
[14]	Right, left hands	Support vector machine (SVM)	No	77.7%
[15]	Right hand, both feet	LDA	No	81.96%
[16]	Right,left hands, both fists, both feet	Recurrent Neural Network (RNN) architecture: Long short term memory (LSTM),	No	93.25%
[17]	Right,left hands, both fists, both feet	Shallow CNN	Yes	65.73%
[18]	Right, left hands, both feet, both fists	Graph Convolutional Neural Networks (GCN)	Semi (not fully subject specific, model was trained with mixed data from n subjects, with CV testing)	88.57%
[19]	Right hand, left hand	Model selection: Standard Decoder (SD), Transfer Learning decoder, TL decoder with data transformation	Semi (not fully subject-specific, test subjectsâ€™ data is used for calibration)	64-92.5 % (depending on the degree of calibration)

Table 2.2: State-of-the-art accuracies obtained for the BCI Competition â€“ 2A dataset

Research paper	**Movements**	**Classification**	**Subject-independent**	**Best accuracy**
[20]	Right, left hands, feet, tongue	CNN ShallowConvNetModel	No	69%
[21]	Right, left hands, feet, tongue	Deep ConvNet architecture	No	70.9%
[22]	Left vs right hand; Foot vs. tongue	SVM	No	82.5; 84 %
[23]	Right, left, hands	ANN	Yes	71.41%
[24]	Right, left, hands	SVM	Yes	84.4%
[25]	Right, left, hands	CNN, Meta Update Strategy (MUPS)	Yes	76.6%

Chapter 3

Methodology

This chapter elaborates on methods that were used to accomplish the task of the project. The section presents the datasets, models as well pre-processing techniques that were of utility. To work with the data and the models "Python" programming language and the "Pytorch" libraries were used.

3.1 Dataset description

3.1.1 Weibo2014

The dataset corresponds to EEG data recorded from 10 healthy right-handed individuals. Neuroscan SynAmps2 amplifier was utilized to obtain the 64-channel EEG signals, with electrodes being placed according to the international 10-20 system with reference to the nose and grounded prefrontal lobe. The acquired signals were then band-pass filtered into the range of 0.5-50 Hz and down-sampled at 200 Hz. The authors of the original study worked on identifying discriminatory EEG patterns of simple and complex limb MI. Each trial took eight seconds: after two seconds a visual cue was shown to participants, which is followed by a command indicating right or left hand motor imagery. The subjects performed the mental exercise for about four seconds. Overall, nine sessions, consisting of 60 trials for each task, were conducted [26].

3.1.2 Physionet

The dataset consists of MI EEG data obtained from 109 subjects. The signals were obtained using the BCI2000 system with 64 channels at the sampling frequency of 160 Hz. The electrodes were placed in accordance with the international 10-10 placement convention. As the project focuses on binary classification, only the data corresponding to right and left hand MI were taken. The trials lasted for eight seconds: four seconds of mental imagery followed by four seconds of rest. Overall, 46 trials were recorded for each subject [27].

3.1.3 BCI Competition VI - Dataset 2a (BCI-DataSet2A)

The dataset corresponds to MI EEG data obtained from nine healthy individuals. The subjects were asked to perform four different forms of motor imagery: movements of right hand, left hand, both feet and tongue. Two separate sessions were conducted in each of them recording 288 trials for each category of MI. As in the case of the Physionet dataset, EEG data corresponding only to right and left hand movement is derived for this project. The instrumentation consisted of 22 Ag/AgCl electrodes fixed on the subject according to 10-20 placement system. The EEG signals were acquired at the sampling rate of 200 Hz, and consequently band-pass filtered to the range 0.5 Hz to 100 Hz [28].

3.1.4 BCI Competition VI - Dataset 2b (BCI-DataSet2b)

The dataset was created based on EEG recordings generated by 9 healthy,right-handed individuals. Overall, each participant took part in 5 sessions, where the first two the training was without feedback, and the last three with the feedback. Three electrodes were employed in the experiment: C3, Cz, and C4, and the electrode Fz served as ground. The data was recorded at the sampling frequency of 250 Hz, then bandpass filtered on the interval 0.5 Hz to 100 Hz. Motor imagery for two classes was performed: left hand, right hand. The trials were initiated by visual as well as audio signal, which was followed by a cue that lasted for 1.25 seconds. During the next 4 seconds the participants had to imagine the corresponding hand movement. In total, each individual performed 120 trials per session [29].

3.2 Models

As it has been discussed in previous sections, in the transfer learning paradigm, the model to be applied to certain purposes is pre-trained based on different data or domains. In this project, two architectures with pre-trained weights have been of major focus: "ResNet" and "DenseNet", which were originally intended to improve state-of-the-art image classification performance. This section gives an overview of these architectures.

3.2.1 ResNet

The architecture was first introduced and discussed in [30]. The authors stated that the motivation behind this work was to generate a feasible solution to the issue of degradation in deep neural networks: as the number of layers increase the accuracy of the model saturates and starts to decline rapidly after a certain point. Surprisingly, such a behavior is not caused by overfitting as integrating more layers into the network results in the decreased training accuracy [31].

To tackle the issue, the authors propose a deep residual learning framework. The novelty is represented by introducing skip connections, or shortcuts over some layers. Double or triple-layer skips can be utilized with a nonlinear function and batch normalization integrated in them. The authors also highlight that the solution does not significantly increase the complexity of the deep network as it does not introduce new parameters or add computational load.

Using the proposed architecture, the authors produced significantly better results than the one had been recorded earlier. To illustrate, in 2015, the solution won first place in a classification competition, and was able to achieve the error rate as low as 3.57% on the ImageNet dataset.

3.2.2 DenseNet

The paradigm of dense convolutional networks (DenseNet) was first introduced and described by [32]. The motivation behind the new network architecture was based on the observation that if shorter connections between layers close to the input as well as the output were used, deep convolutional networks can offer improved accuracy and training efficiency.

In DenseNet each layer is connected to every other layer in a feedforward fashion. To be more precise, if in traditional neural networks there are N connections for N layers, in the case of DenseNet architecture there will be $\frac{N(N+1)}{2}$ connections. Thus, feature-maps corresponding to each layer are used as inputs to all the following layers.

The architecture offers numerous benefits: it can mitigate the vanishing-gradient problem, reinforce feature propagation, support feature use, and considerably reduce the number of parameters. The proposed solution was tested utilizing common public object recognition tasks such as CIFAR-10, CIFAR-100, ImageNet and SVHN. As a result, DenseNet demonstrated extremely promising performance: it was able to surpass state-of-the-art accuracies on most of them.

3.3 Pre-processing

To be able to utilize the aforementioned models, one-dimensional EEG data had to be transformed into two dimensional input arrays. To be more specific, in-built pytorch models corresponding to ResNet and DenseNet accept inputs of size (3, 224, 224), whereas depending on the dataset EEG signals are one dimensional arrays with different lengths. To accomplish this task a common technique called short time Fourier transform was used.

STFT is a Fourier transform of a windowed signal. Commonly used windows are Hann and Gaussian windows. The method is particularly useful when the frequency component of the signal varies over time, as only a certain range is transformed at a time. If the input is a one dimensional signal the output will be a 2D pattern, corresponding to time-frequency data. For discrete signal the equations are given as follows [33]:

$$X_{STFT}\left[m,n\right] = \sum_{k=0}^{L-1} x\left[k\right] g\left[k-m\right] e^{-j2\pi nk/L}, \tag{3.1}$$

$$x\left[k\right] = \sum_{m}\sum_{n} X_{STFT}\left[m,n\right] g\left[k-m\right] e^{j2\pi nk/L}. \tag{3.2}$$

The outcome can also be visualized as a spectrogram, which is an intensity plot of STFT magnitude on the time-frequency plane. It has to be noted that there exists a tradeoff between the resolution in frequency and the resolution in time, that is, a high resolution in one parameter will lead to a lower resolution in the other. In Figure 3.2 one can observe

the application of STFT on a signal from Weibo2014 dataset corresponding to left hand and right hand MI.

When applying STFT there are two parameters that can be adjusted: window size and the overlap size. It is important to choose the optimal setting for both parameters as they are responsible for the final frequency and time resolutions. As in three datasets that were analyzed in the project different lengths of signals one cannot use the same set of parameters on all of them. Therefore, the optimal parameters were derived empirically, by trial and error.

Before applying STFT each of the dataset signals were subjected to bandpass-filtering in the interval of 7-32 Hz, as it corresponds to (with small margin) alpha and beta frequencies which are significant for left hand, right hand MI. Another important information is that only three channels from each dataset were used. This decision was inspired by the approach taken in [34], where the aforementioned channels were used to create an image with three planes, which can be thought as a counterpart to an RGB image. However, in the in [34] the authors did not have to aim for a certain size for the input image, as the CNN-based architecture was custom made. In our case, the input size is constrained to (3,224,224), hence one must undertake several preprocessing steps to satisfy this condition.

Firstly, STFT is applied in such a way that the resulting image is of a quadratic form, this is substantiated by the fact that, otherwise during resizing, the data might get distorted thus adding undesirable artifacts. After applying STFT to each of the three channels, it has to be resized to fit (56x56). This is done using the bilinear interpolation algorithm, which can be described as a linear interpolation applied in both dimensions. Finally, the obtained image is replicated 4 times to reach the necessary size - (224,224). This, of course, is repeated for the rest of the electrodes resulting in the final image of (3,224,224). The process is illustrated in Figure 3.1. Also, in Table 4.5 the window and overlap size of STFT are specified for each dataset.

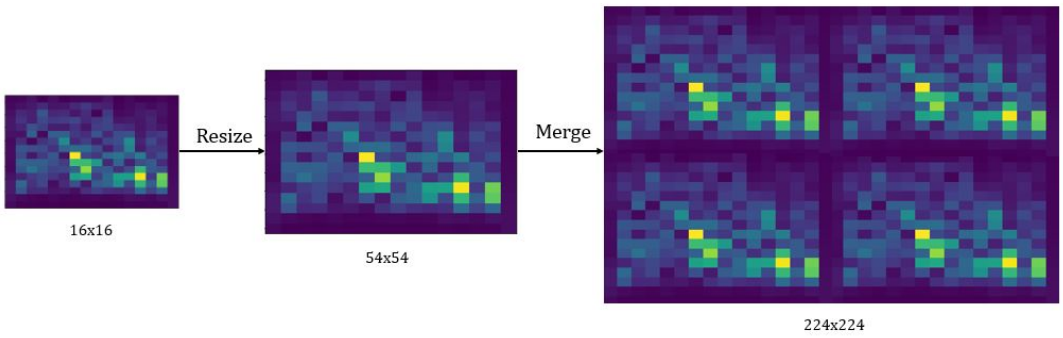

Figure 3.1: The preprocessing steps illustrated for single channel from 3 available

3.4 Stacked Autoencoder

An autoencoder is a type of neural network which consists of an input layer, one hidden layer, and one output layer [35]. The neural network is distinguished from other neural

Table 3.1: Window and overlap size for each dataset

Parameter	Weibo2014	BCI-DataSet2A	BCI-DataSet2b	Physionet
Window size	30	30	30	25
Overlap size	8	8	8	7

networks by the fact the output layer has the same number of layers as the input layer. The input-output relationship can be expressed as follows:

$$y = f(W_y x + b_y)$$

$$o = f(W_o y + b_o)$$

Here, y is the values stored in the hidden layer, and o represents the output of the whole network. f is an activation function. W and b represent the weights and the biases of respective layers.

AE is trained by forcing the output of the network to approximate the input. Once the network is trained, the structure can be interpreted as an encoder and a decoder connected together. The hidden layer will be able to store a higher level representation of the input. AE can be used in a variety of manners in an attempt to improve the performance of a certain model. The most prominent example of utilization is calculating initial values of weights and biases in this manner, after which the model can be finetuned over the whole neural network. Stacked autoencoder (SAE) is, in turn, a combination of autoencoders. When using SAE, the layers are trained iteratively, gradually moving to higher levels of representation. It is important to keep in mind that while a certain layers is being trained, the preceding layers are kept unchanged, the layer can be considered isolated.

In this project, SAE will be used in combination with the ResNet18 architecture to assess the feasibility of this approach. In [36], the authors utilized the same approach, however, they made use of their own convolutional network architecture. The method is substantiated by the reasoning that it can potentially mitigate the effects of artifacts and other inconsistencies inherent to EEG data. The preprocessing stage remains the same, spatio-temporal data is extracted and interpreted using STFT. Firstly, the CNN module (ResNet18) is trained using supervised learning. Once the training is complete the SAE module is concatenated and each corresponding layer is trained iteratively, with finetuning carried out in the end. The SAE part consists of an input layer, 6 hidden layers and single output layer. The whole architecture can be observed in Figure 3.4.

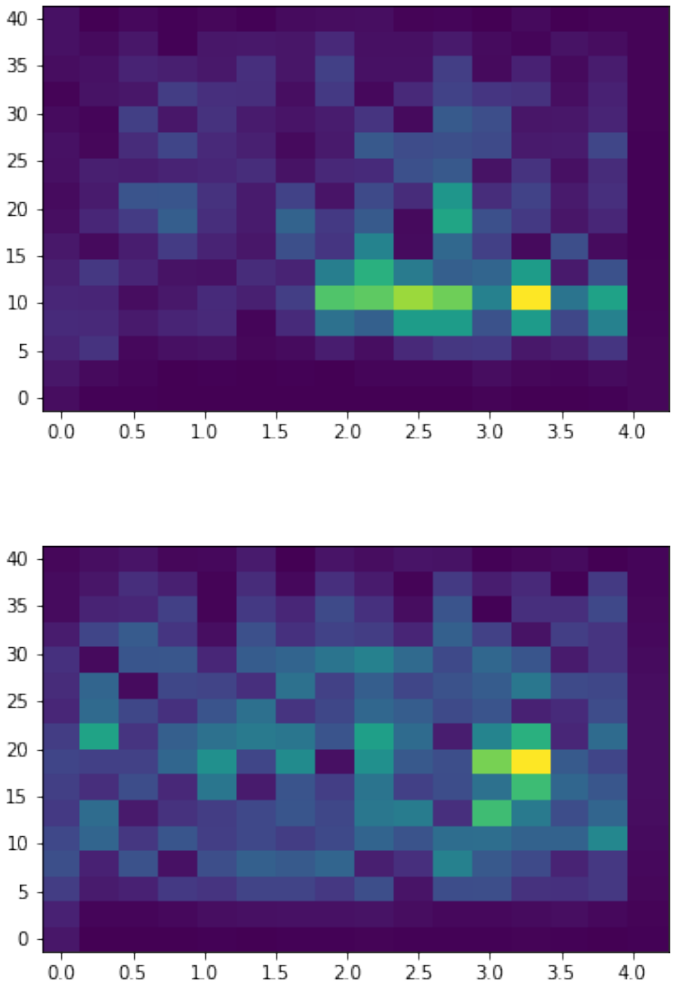

Figure 3.2: An example of STFT application: left hand vs. right hand MI

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2 $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Figure 3.3: ResNet18 Architecture

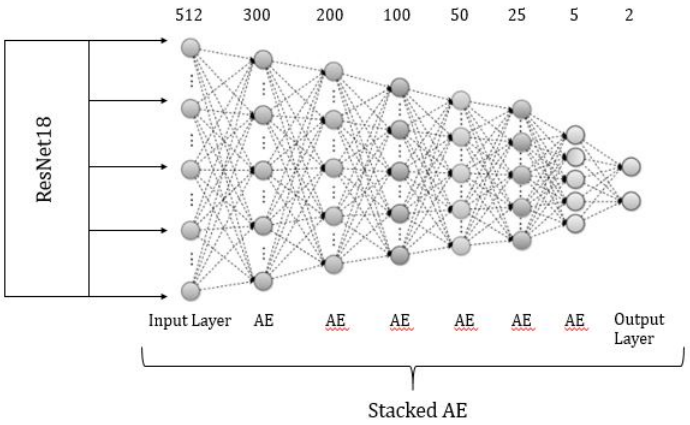

Figure 3.4: Stacked autoencoder architecture

Chapter 4

Results and Discussion

In this chapter results produced by the models will be presented for 4 different datasets: Weibo2014, Physionet, BCI-DataSet2A, and BCI-DataSet2B. Before narrowing down the scope of the project to two described above architectures, other models had been considered as well, such as VGG, AlexNet, SqueezNet. However, it was revealed that they were completely inefficient: not only the validation accuracies but also training accuracies did not show any meaningful improvement. Therefore, it was decided to ignore those models and apply the remaining two (ResNet and DenseNet) on every dataset.

Results obtained for the dataset : BCI-DataSet2b are presented in the Table 4.1. The models were trained and validated in leave-one-out fashion, and accuracies corresponding to which subject are displayed. Here, on average, DenseNet161 and ResNet101 showed significantly better results that ResNet50, with the differences in the average accuracy being around 5%.

In Table 4.2 one can find similar results for Weibo2014 dataset. Here, interestingly, the models performed at almost the same level with all reaching an average accuracy of around 0.6. For BCI-DataSet2a (Table 4.3), ResNet101 was shown to be superior with an average accuracy 0.6440 and with highest value 0.6639.

Similar to Weibo2014, validation accuracy results for the dataset BCI-DataSet2A have been found displayed in Table 4.2.

For the Physionet dataset (Table 4.4), in terms of classification accuracy the differences between models are more pronounced. Once again, ResNet101 was shown to be more accurate with the proportion of correctly identified events being 0.6484. ResNet50 and DenseNet161 follow with 0.6238 and 0.5884 accuracy values, respectively.

Based on the findings, the best performance across datasets was demonstrated by ResNet101, followed by ResNet50 and DenseNet161. However, the discrepancies between are not significant.

In Table 4.5, we can see the result obtained from the hybrid architecture. From the accuracy values, it is difficult to state definitely that the additional SAE module brought justifiable improvement.

Table 4.1: BCI-DataSet2b

	DenseNet161	ResNet50	Resnet101
Subject1	0.549408	0.564164	**0.638665**
Subject2	**0.637035**	0.506770	0.617755
Subject3	0.608830	0.566689	0.590521
Subject4	0.652854	0.525466	0.615054
Subject5	0.624049	**0.601705**	0.580494
Subject6	0.636302	0.543860	0.572080
Subject7	0.614724	0.552849	0.620149
Subject8	0.622659	0.560260	0.542533
Subject9	0.601210	0.559785	0.592822
Average	*0.616341*	*0.553505*	*0.596675*

Table 4.2: Weibo2014

	DenseNet161	ResNet50	Resnet101
Subject1	0.557488	**0.646951**	0.637520
Subject2	0.603327	0.634777	**0.678463**
Subject3	0.609069	0.630526	0.603956
Subject4	0.560278	0.641593	0.533333
Subject5	0.570449	0.586771	0.586098
Subject6	0.594779	0.543203	0.593982
Subject7	0.557885	0.688521	0.595450
Subject8	**0.650684**	0.550926	0.564674
Subject9	0.609245	0.601446	0.581356
Subject10	0.613914	0.544200	0.665425
Average	*0.592712*	*0.606891*	*0.604026*

Table 4.3: BCI-DataSet2a

	DenseNet161	ResNet50	Resnet101
Subject1	0.514436	0.595256	0.619238
Subject2	0.549740	0.647667	0.638290
Subject3	0.568316	0.558225	0.637977
Subject4	0.537294	**0.666197**	0.643703
Subject5	0.536957	0.579248	0.659074
Subject6	0.556347	0.555109	**0.663868**
Subject7	**0.591942**	0.592489	0.684746
Subject8	0.505312	0.639880	0.637204
Subject9	0.579002	0.706363	0.612236
Average	*0.548816*	*0.615604*	*0.644038*

Table 4.4: Physionet. Only first 20 subject are show, average is over the entire dataset

	DenseNet161	ResNet50	Resnet101
Subject1	0.554238	0.648454	0.656977
Subject2	0.546533	0.659343	0.641073
Subject3	0.500663	0.665736	0.601475
Subject4	0.610117	0.571563	0.633977
Subject5	0.646476	0.476922	0.642144
Subject6	0.573716	0.543180	0.638289
Subject7	0.529089	0.618539	0.658940
Subject8	0.551747	0.613853	0.638357
Subject9	0.621803	0.594931	0.639759
Subject10	0.525617	0.633911	0.622544
Subject11	0.576051	0.676132	0.642877
Subject12	0.572698	0.658641	0.627200
Subject13	0.591654	0.589316	0.635288
Subject14	0.622384	0.631151	0.669467
Subject15	0.583385	0.596742	0.629217
Subject16	0.574051	0.577586	0.632839
Subject17	0.590697	0.620418	0.663668
Subject18	0.524593	0.593678	0.602312
Subject19	0.678278	0.579806	0.660766
Subject20	0.547853	0.609144	0.635585
Average	*0.588418*	*0.623839*	*0.648375*

Table 4.5: ResNet18 - SAE. Average accuracy values.

Parameter	**Weibo2014**	**BCI-DataSet2A**	**BCI-DataSet2b**	**Physionet**
Accuracy	0.5981	0.6223	0.6412	0.6544

Chapter 5

Concluding Remarks

It is evident that BCI technology is of extreme importance for a considerable proportion of the human population. Although significant progress has been and is being made in the field, there is still a variety of challenges.

This project attempted to utilize a relatively novel approach of transfer learning in the BCI paradigm. Two deep neural network architectures, ResNet and DenseNet, were chosen as they could demonstrate acceptable initial results. The project was confined to three datasets: Physionet, Weibo2014, BCI-DataSet2A, BCI-DataSet2B. Corresponding EEG signals were pre-processed by applying STFT. Lastly, the models were assessed based on the leave-on-out validation approach. Thus, the obtained results correspond to subject-independent performance.

Based on the acquired results, the system can be described as somewhat better than the random classifier. Narrowing the choice of electrodes seem to be helpful while exporting spatio-temporal information, since when compared to previous results the current accuracy values were higher. It also might be accounted to the resizing method. In addition a hybrid architecture uniting ResNet18 and SAE was investigated. The results were ambiguous in terms the improvement the given modification can propose as the average values increased only slightly.

Additionally, some ideas on how to improve the performance were discussed. One might attempt different pre-processing procedures and incorporate optimization techniques to achieve better results.

Bibliography

[1] I. Volosyak, F. Gembler, and P. Stawicki, "Age-related differences in ssvep-based bci performance," *Neurocomputing*, vol. 250, pp. 57–64, 2017.

[2] K. Ziegler-Graham, E. J. MacKenzie, P. L. Ephraim, T. G. Travison, and R. Brookmeyer, "Estimating the prevalence of limb loss in the united states: 2005 to 2050," *Archives of physical medicine and rehabilitation*, vol. 89, no. 3, pp. 422–429, 2008.

[3] S. G. Mason and G. E. Birch, "A general framework for brain-computer interface design," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 11, no. 1, pp. 70–85, 2003.

[4] A. Bashashati, M. Fatourechi, R. K. Ward, and G. E. Birch, "A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals," *Journal of Neural engineering*, vol. 4, no. 2, R32, 2007.

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

[6] A Kubler, V. K. Mushahwar, L. R. Hochberg, and J. P. Donoghue, "Bci meeting 2005-workshop on clinical issues and applications," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 14, no. 2, pp. 131–134, 2006.

[7] F. Lotte, L. Bougrain, and M. Clerc, "Electroencephalography (eeg)-based brain–computer interfaces," *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–20, 1999.

[8] E. C. Lalor, S. P. Kelly, C. Finucane, R. Burke, R. Smith, R. B. Reilly, and G. Mcdarby, "Steady-state vep-based brain-computer interface control in an immersive 3d gaming environment," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 19, pp. 1–9, 2005.

[9] A. Bashashati, R. K. Ward, and G. E. Birch, "Towards development of a 3-state self-paced brain-computer interface," *Computational intelligence and neuroscience*, vol. 2007, 2007.

[10] A. Schlögl, F. Lee, H. Bischof, and G. Pfurtscheller, "Characterization of four-class motor imagery eeg data for the bci-competition 2005," *Journal of neural engineering*, vol. 2, no. 4, p. L14, 2005.

[11] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: A systematic review," *Journal of neural engineering*, vol. 16, no. 5, p. 051 001, 2019.

[12] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[13] P. Wang, J. Lu, B. Zhang, and Z. Tang, "A review on transfer learning for brain-computer interface classification," in *2015 5th International Conference on Information Science and Technology (ICIST)*, IEEE, 2015, pp. 315–322.

[14] C. Park, D. Looney, N. ur Rehman, A. Ahrabian, and D. P. Mandic, "Classification of motor imagery bci using multivariate empirical mode decomposition," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 21, no. 1, pp. 10–22, 2012.

[15] H. S. Kim, M. H. Chang, H. J. Lee, and K. S. Park, "A comparison of classification performance among the various combinations of motor imagery tasks for brain-computer interface," in *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)*, IEEE, 2013, pp. 435–438.

[16] X. Zhang, L. Yao, C. Huang, Q. Z. Sheng, and X. Wang, "Intent recognition in smart living through deep recurrent neural networks," in *International Conference on Neural Information Processing*, Springer, 2017, pp. 748–758.

[17] H. Dose, J. S. Møller, H. K. Iversen, and S. Puthusserypady, "An end-to-end deep learning approach to mi-eeg signal classification for bcis," *Expert Systems with Applications*, vol. 114, pp. 532–542, 2018.

[18] X. Lun, S. Jia, Y. Hou, Y. Shi, Y. Li, H. Yang, S. Zhang, and J. Lv, "Gcns-net: A graph convolutional neural network approach for decoding time-resolved eeg motor imagery signals," *arXiv preprint arXiv:2006.08924*, 2020.

[19] T. Verhoeven, B. Vuylsteker, and J. Dambre, "Model selection for subject-to-subject transfer learning in brain-computer interfaces," Dec. 2016.

[20] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "Eegnet: A compact convolutional neural network for eeg-based brain–computer interfaces," *Journal of neural engineering*, vol. 15, no. 5, p. 056 013, 2018.

[21] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for eeg decoding and visualization," *Human brain mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.

[22] Y. Zhang, C. S. Nam, G. Zhou, J. Jin, X. Wang, and A. Cichocki, "Temporally constrained sparse group spatial patterns for motor imagery bci," *IEEE transactions on cybernetics*, vol. 49, no. 9, pp. 3322–3332, 2018.

[23] S. Saha, M. Hossain, K. Ahmed, R. Mostafa, L. Hadjileontiadis, A. Khandoker, M. Baumert, *et al.*, "Wavelet entropy-based inter-subject associative cortical source localization for sensorimotor bci," *Frontiers in neuroinformatics*, vol. 13, p. 47, 2019.

[24] S. M. Hosseini, M. Bavafa, and V. Shalchyan, "An auto-adaptive approach towards subject-independent motor imagery bci," in *2019 26th National and 4th International Iranian Conference on Biomedical Engineering (ICBME)*, IEEE, 2019, pp. 167–171.

[25] T. Duan, M. Chauhan, M. A. Shaikh, J. Chu, and S. Srihari, "Ultra efficient transfer learning with meta update for cross subject eeg classification," *arXiv preprint arXiv:2003.06113*, 2020.

[26] W. Yi, S. Qiu, K. Wang, H. Qi, L. Zhang, P. Zhou, F. He, and D. Ming, "Evaluation of eeg oscillatory patterns and cognitive process during simple and compound limb motor imagery," *PloS one*, vol. 9, no. 12, e114853, 2014.

[27] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "Bci2000: A general-purpose brain-computer interface (bci) system," *IEEE Transactions on biomedical engineering*, vol. 51, no. 6, pp. 1034–1043, 2004.

[28] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. J. Miller, G. Mueller-Putz, *et al.*, "Review of the bci competition iv," *Frontiers in neuroscience*, vol. 6, p. 55, 2012.

[29] J. Allen, F Lee, C Keinrath, R Scherer, H Bischof, and G Pfurtsceller, "Brain-computer communication: Motivation, aim, and impact of exploring a virtual apartment," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 473-482, pp. 235–238, 2007.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[31] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5353–5360.

[32] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[33] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 3, pp. 235–238, 1977.

[34] D. Lee, S.-H. Park, H.-J. Lee, and S.-G. Lee, "Eeg-based motor imagery classification using convolutional neural network," *Journal of Korean Institute of Information Technology*, vol. 15, no. 6, pp. 103–110, 2017.

[35] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, *et al.*, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, p. 153, 2007.

[36] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of eeg motor imagery signals," *Journal of neural engineering*, vol. 14, no. 1, p. 016 003, 2016.