Sub-grouping Schizophrenia Spectrum Disorders using Deep Learning on Resting State fMRI

Master's Thesis in Medical Technology

by

Oscar Alm Harestad

Department of Physics and Technology

University of Bergen

June 2, 2024

Scientific environment

The work in this thesis has been carried out at the Department of Physics and Technology at the Faculty of Natural Sciences and Mathematics, University of Bergen, as part of a Master's program in Medical Technology. Throughout the project there has been a strong collaboration with researchers at the Mohn Medical Imaging and Visualization Centre at the Department of Radiology, Haukeland University Hospital and the Department of Biological and Medical Physiology, University of Bergen with the ERC research group and the Bergen fMRI group. Clinical questions with respect to psychiatric applications were addressed to clinicians at the Division of Psychiatry, Haukeland University Hospital.

In the thesis, both data from an online database and already acquired local data (ERC project grants to Professor Kenneth Hugdahl) are included and used according to current guidelines. Computational resources at the University of Bergen were utilised in all machine learning analysis.

MMIV

HELSE BERGEN
Haukeland universitetssjukehus

Acknowledgements

I would like to thank everyone who has been a part of this project, both those who were crucial for its success but also those who helped along the way. I am sincerely thankful to my supervisors, Renate Grüner and Frank Riemer, for providing support, feedback, and brilliant guidance. Thank you, Renate, for being a great discussion partner, a true motivator, and a solid support, both when conducting analyses but also when writing, planning, and thinking. Thank you, Frank, for being reliable, brilliant, and positive, and for always contributing with wise words, often accompanied by a smile or a laugh. Also, a huge thank you to Lydia Brunvoll Sandøy for excellent explanations, invaluable assistance and great feedback.

This research would not have been possible without the support of the ERC research group. I would especially like to thank Kenneth Hugdahl for access to the data used in the study and for providing beautiful insights and good laughs. I would also like to thank Alex Craven for valuable aid with data access and management, your help has been mega! Thank you to Igne Sinkeviciute for tips and assistance in the grouping of patients, and general interest in the study. Also, thank you to the Department of Radiology and the radiologists, patients and healthy controls who were part of the original projects. A huge thanks to the Center for Biomedical Research Excellence for providing excellent data, and everyone at MMIV for being interested and able to help with anything, anytime. Also, a thanks to everyone involved in the MMIV conference, and thanks for letting me participate!

I would like to thank my family and friends, and my fellow training buddies on both sides of the world. Thank you to Curie, both students past and present. Thank you to all my study mates who have helped me throughout the past years, especially the ones from 534. It would not have been the same without any of you. Last but not least, thank you to my partner Ananda Tallaksen Benediktsdottir for always being there for me, for your love and support.

Oscar Alm Harestad
Bergen, June 2024

Abstract

Schizophrenia (SCZ) is a complex mental disorder which affects about 1 in 300 people worldwide [1]. Individuals with SCZ can experience psychosis in the form of hallucinations and delusions. The disorder has a severe impact on quality of life, not just for the patient but for their family and friends [2]. Personalized treatment is rare and the selection of treatment often follows a "trial and error" regime where a self-report of symptoms decides what medication is most appropriate [3]. Thus, there is a need to achieve a more personalized approach [4].

Since the discovery of functional magnetic resonance imaging (fMRI) in the 90s, functional neuroimaging has been used to investigate brain activity in patients with psychiatric disorders such as SCZ [5]. Resting-state fMRI scans can be used to classify subjects with schizophrenia among a group of both patients and healthy controls using machine learning (ML) [6–9]. However, few have tried to classify the various subgroups of the disorder using ML. The aim of this thesis is to investigate if ML based on resting-state fMRI data (4D) can aid in subgrouping patients with SCZ. Furthermore, the feasibility of this approach to distinguish patients from healthy controls is investigated.

The approach consists of implementing a deep learning (DL) pipeline to handle four dimensional data, before analysing both online (N=148) and local (N=316) data. The study also assesses how different preprocessing of images impact DL models and explores hyperparameter combinations to optimize the performance of models.

The problem addressed in the thesis is difficult and probably an ill-posed problem with more variability than what would be ideal for sub-grouping in SCZ. Therefore, the overall performance of the implemented ML models are lower than expected. However, to our knowledge, this is the first attempt on distinguishing between data based on 4D neuroimaging data directly. The project shows that minimizing preprocessing, as well as using data only from one source rather than grouping several datasets, is beneficial. Hyperparameter selection improves performance and could potentially be further optimized and explored to improve performance. The proposed approach is able to reproduce previous attempts of separation between patients and controls, even if the analysis is performed on the raw data directly (4D) rather than feature extracted fMRI data. Thus, the proposed approach could still be valuable in clinical research and clinical follow up in the future.

Contents

Abbreviations

AI	Artificial Intelligence
BOLD	Blood Oxygenation Level Dependant
CNN	Convolutional Neural Network
CSF	Cerebrospinal fluid
DL	Deep Learning
EPI	Echo Planar Imaging
FID	Free Induction Decay
fMRI	Functional Magnetic Resonance Imaging
FN	False Negative
FP	False Positive
IID	Independent and Identically Distributed
ML	Machine Learning
MRI	Magnetic Resonance Imaging
NN	Neural Network
PANSS	Positive And Negative Syndrome Scale
ReLU	Rectified Linear Unit
RF	Radio Frequency
ROI	Region Of Interest
SCZ	Schizophrenia
SGD	Stochastic Gradient Descent
SNR	Signal-to-noise ratio

SPM	Statistical Parametric Mapping
TE	Echo time
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
TR	Repetition time

List of Figures

List of Tables

Chapter 1

Introduction

Since the discovery of fMRI in the 90s, functional neuroimaging has been used to investigate brain activity in subjects with psychiatric disorders [5]. Among these are schizophrenia (SCZ), where subjects can experience psychosis in the form of hallucinations and delusions [1]. It is a genetically complex disorder and consists of a heterogeneous patient group. The disorder has a severe impact on quality of life, not just for the subject but for their family and friends [2]. Personalized treatment is rare and the selection of treatment often follows a "trial and error" regime where a self-report of symptoms decides what medication is most appropriate [3]. Thus, there is a need to achieve a more personalized approach [4]. Practitioners may use sub-groups to categorize subjects based on their symptoms, e.g. paranoid-, disorganized- and catatonic SCZ. Correct classification of the mental disorder can aid in finding a working medication for the subject in question.

In resting-state fMRI studies, the subject will not conduct any task [10]. Instead, the subject is instructed to rest but not fall asleep. This can be preferable in the study of psychiatric disorders due to the technique not requiring any subjective input. Resting-state fMRI can be used to study the functional connectivity in the brain, i.e. different anatomically separated brain regions being connected due to their simultaneous activations or fluctuations in resting-state [11]. The functional connectivity can be used to identify brain networks, such as brain regions involved in specific functions or tasks [12]. A brain network that have been studied extensively using resting-state fMRI is the default mode network [13]. Some studies have shown that functional connectivity can contain information on disorders such as SCZ [14–16].

Resting-state fMRI scans can be used to classify subjects with SCZ among patients and healthy controls using machine learning (ML) [6–9]. However, few have attempted to classify the various sub-groups of the disorder using this technique. And to our knowledge, none have used deep learning on four dimensional fMRI time series to classify subjects with SCZ.

The aim of this thesis is to investigate if ML based on resting-state fMRI data (4D) can aid in the sub-grouping of subjects with SCZ. Furthermore, the feasibility of this approach to distinguish subjects with SCZ from healthy controls is investigated. The following steps are proposed:

- To design, implement and run a deep learning (DL) pipeline that can handle 4D data.
- To optimize the ML model through hyperparameter fine-tuning.
- To investigate the effect of preprocessed imaging data on the performance of the ML model.
- To perform binary and multilabel classification of subjects with SCZ using DL models.
- To investigate and assess the performance of the models on an online dataset (N=148) and local datasets (N=316).

Chapter 2

Theory

2.1 Functional Magnetic Resonance Imaging

2.1.1 Difference between MRI and fMRI

Magnetic Resonance Imaging (MRI) is one of the main medical imaging techniques today [17]. It is one of the most advanced, versatile, and important imaging modalities used in clinical diagnostics, treatment planning, and follow up. Because MRI does not involve ionizing radiation, it is also extensively used in research applications. Using MRI, we can acquire anatomical images of the body to view different tissues and organs. The main difference between structural MRI and functional MRI (fMRI) is that structural MRI is being used for single volume captures, whereas fMRI is being used to capture a time series of multiple volumes [18]. Structural MRI can be used to differentiate between different tissues in the brain, i.e. gray matter, white matter and cerebrospinal fluid (CSF), which makes it possible to detect abnormalities in the body such as tumours. fMRI is mainly used to look at the dynamics of brain activity as a function of time. This can give us an important understanding of the function and interactions between different brain regions. fMRI can be used as a tool to compare brain activity in healthy subjects to subjects with mental disorders such as depression, bipolar disorder and SCZ. This furthers our understanding of possible underlying mechanisms that are involved in the disorders.

2.1.2 Principles of MRI

MRI uses the properties of nuclei with magnetic moment, mainly hydrogen atoms but also sodium or phosphor, to generate signals [19]. Protons and neutrons are spin $\frac{1}{2}$ particles with an associated magnetic moment μ. The magnitude of the magnetic moment μ of each nucleon is given by

$$|\mu| = \gamma\hbar\sqrt{I(I+1)} \tag{2.1}$$

where I is the spin number, $\hbar$ is the reduced Planck constant and γ is the gyromagnetic ratio, which connects the angular momentum to the magnetic moment [19]. The gyromagnetic ratio is specific for each nucleus. The induced magnetic moment for each nucleon will have a direction and strength, i.e. it has vector properties. All nuclei with these properties are called MR active nuclei.

Within the human body, the vast majority of the MR active nuclei are hydrogen, mainly the hydrogen atoms in water and fat [19]. Due to the absence of neutrons in the hydrogen atom, the single proton makes for a large magnetic moment in the hydrogen atom. Due to thermal energy, the orientation of this magnetic moment is random until an external magnetic field $\overrightarrow{B_0}$ is applied. Depending on the energy level of the nuclei, they can be oriented either with or against the magnetic field. From equation 2.1, the orientation and magnitude of the magnetic moment of the hydrogen nucleus in the direction of $\overrightarrow{B_0}$, the z-direction, is given by

$$\overrightarrow{\mu_z} = \pm\frac{1}{2}\gamma\hbar \tag{2.2}$$

The energy levels are therefore given by

$$E_\uparrow = -\frac{1}{2}\gamma\hbar B_0 \tag{2.3}$$

where $E_\uparrow$ is the lower energy state, and

$$E_\downarrow = \frac{1}{2}\gamma\hbar B_0 \tag{2.4}$$

where $E_\downarrow$ is the higher energy state. There will always be more hydrogen atoms in the low-energy state than in the high-energy state in the human body [20], leading to a net magnetization parallel to the strong external magnetic field called the net magnetization vector $\overrightarrow{M}$ which is the sum of the magnetic moments μ.

$$\overrightarrow{M} = \sum_{i=1}^{n} \mu_i \tag{2.5}$$

In even-numbered nuclei, the magnetic moments of protons and neutrons cancel each other out, resulting in a zero net magnetization vector in a strong externally applied magnetic field. In odd-numbered nuclei, however, the net magnetization is not zero. The temporal behaviour of the net magnetization vector can be described by classical mechanics and the Bloch equation:

$$\frac{d\overrightarrow{M}}{dt} = \gamma\overrightarrow{M} \times \overrightarrow{B_0} \tag{2.6}$$

The remainder of the thesis describes MRI using classical physics.

Under the influence of an external magnetic field, the hydrogen atoms in the human body will have a precessional frequency ω_0 [19]. This can be modelled as the hydrogen atoms wobbling about the magnetic field. The precessional frequency ω_0, the Larmor frequency, is determined by the Larmor equation:

$$\omega_0 = \gamma B_0 \tag{2.7}$$

where γ is the gyromagnetic ratio and B_0 is the external magnetic field.

2.1.3 Signal generation

For an MRI-image to be acquired, a radio frequency (RF) pulse has to be transmitted to the system [20]. The frequency of this RF-pulse has to match the Larmor frequency of the nuclei for a phenomenon called resonance to occur [19]. Therefore, only the hydrogen atoms in the body will provide a signal when the appropriate frequency of the RF-pulse is applied.

An excitation of the spin system can be achieved by adding an RF-pulse of the right frequency [17]. The excitation causes the net magnetization vector $\overrightarrow{M}$ to change its orientation relative to the direction of the magnetic field B_0. The resulting angle, called the flip angle, is a result of the applied duration and the amplitude of the RF-pulse.

When the RF-pulse is turned off, the processes of relaxation start [19]. This is when the net magnetization vector $\overrightarrow{M}$ returns into alignment with the magnetic field $\overrightarrow{B_0}$. Two important relaxation processes called T1 recovery and T2 decay occur. T1 relaxation is the recovery of $\overrightarrow{M}$ along the main magnetic field, while T2 relaxation is the decay of $\overrightarrow{M}$

in the transverse plane. These processes provide the definition for the T1 spin-lattice recovery time and the T2 spin-spin decay time, for tissues. The T1 time is defined as the time it takes for 63% of the magnetization to recover in the longitudinal plane, while the T2 time is the time it takes until 37% of the magnetization is left in the transverse plane [20]. The T1 time is always longer than the T2 time in human tissues. Assuming that the longitudinal direction is the z-direction ($\overrightarrow{k}$) and the transverse plane is the xy-plane ($\overrightarrow{ij}$), the relaxation process can be explained with a modified Bloch equation:

$$\frac{d\overrightarrow{M}}{dt} = \gamma\overrightarrow{M} \times \overrightarrow{B} - \frac{M_x\overrightarrow{i} + M_y\overrightarrow{j}}{T_2} - \frac{(M_z - M_z^0)\overrightarrow{k}}{T_1} \tag{2.8}$$

where M_x, M_y and M_z are the components of the net magnetization in the x, y and z directions, respectively. M_z^0 is the original orientation along B_0, typically modelled by Boltzmann statistics, i.e. dependent on field strength, spin density and temperature. The terms can be separated to provide the equations for the independent processes T1 and T2, respectively:

$$\frac{d\overrightarrow{M_z}}{dt} = -\frac{(M_z - M_z^0)\overrightarrow{k}}{T_1} \tag{2.9}$$

$$\frac{d\overrightarrow{M_{xy}}}{dt} = -\frac{\overrightarrow{M_{xy}}}{T_2} \tag{2.10}$$

which gives origin to the equations describing the T1 and T2 times, respectively:

$$M_z(t) = M_z^0(1 - e^{-\frac{t}{T_1}}) \tag{2.11}$$

$$M_{xy}(t) = M_z^0 e^{-\frac{t}{T_2}} \tag{2.12}$$

Equations 2.11 and 2.12 show how the magnetization vector in the z-direction and the xy-plane changes over time. The T1 and T2 times are shown in figures 2.1 and 2.2.

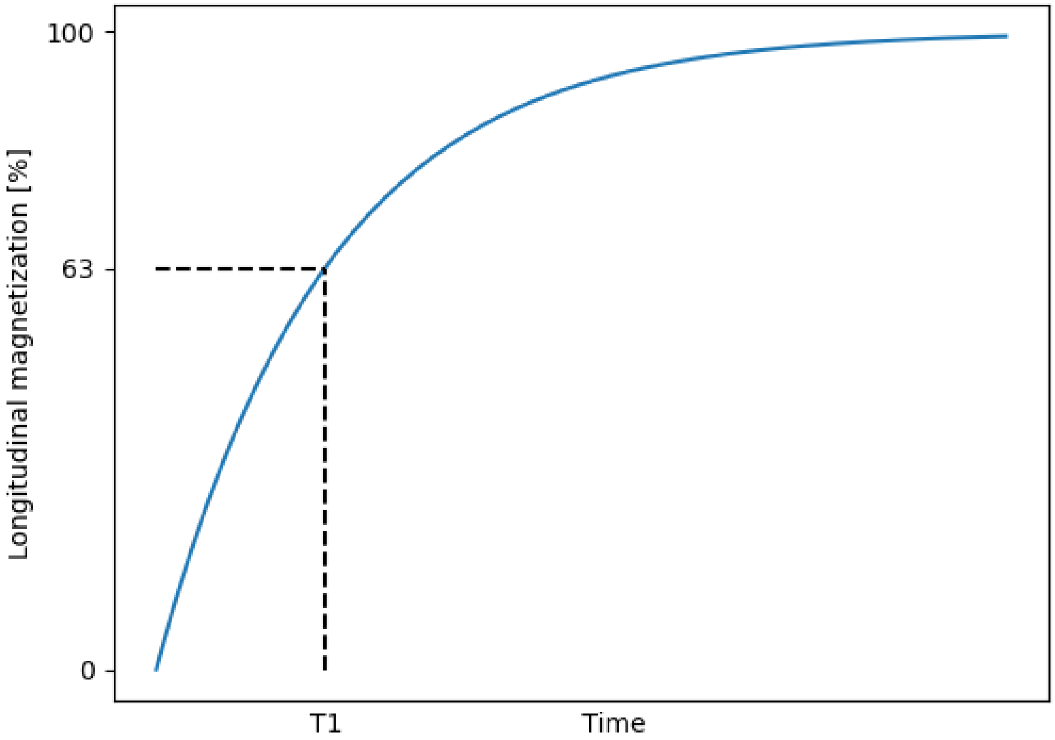


Figure 2.1: The temporal development of T1 relaxation with the T1 time of a given tissue, marked in a dashed line.

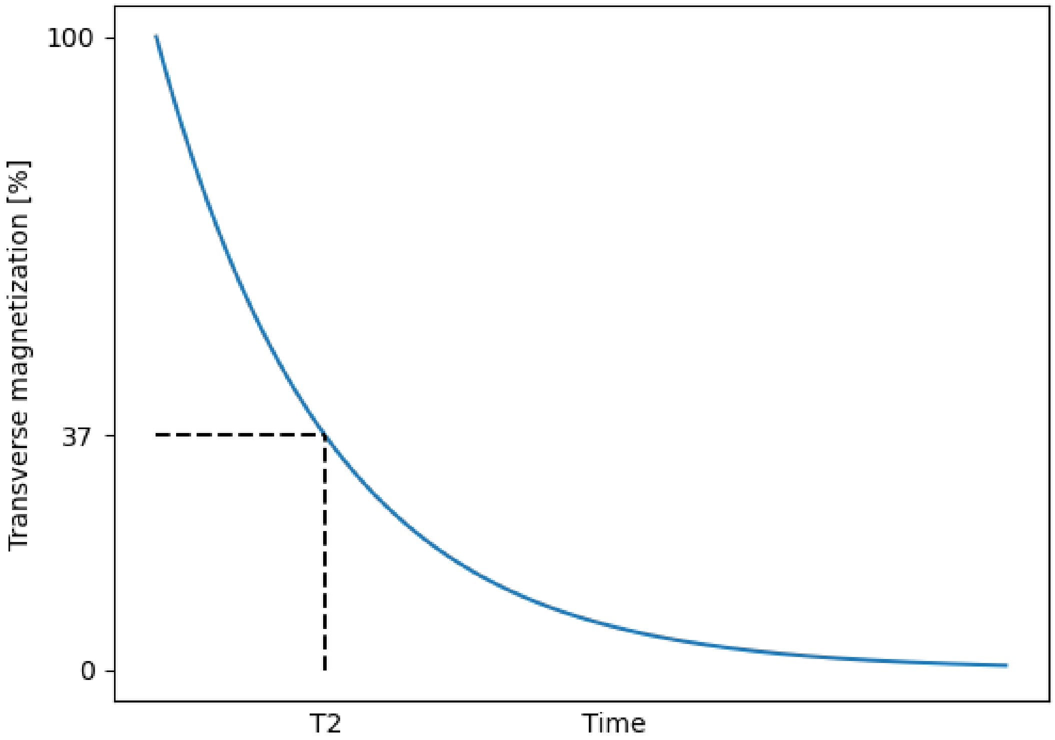


Figure 2.2: The temporal development of T2 relaxation with the T2 time of a given tissue, marked in a dashed line.

External field inhomogeneities or local variations in tissue susceptibility may lead to an added dephasing of the magnetization [17], influencing the relaxation time to seem shorter. This gives rise to the T2* time, also called the apparent spin-spin relaxation time, which will always be shorter than the T2 time [21]. Structural MRI are typically T1 or T2 weighted, while fMRI is based on utilizing the T2*.

2.1.4 Measurement and localization

In addition to the strong static main magnetic field B_0 of the MR scanner, there are also three smaller magnetic fields, one in each of the three orthogonal spatial directions $\overrightarrow{x}$, $\overrightarrow{y}$ and $\overrightarrow{z}$ [18]. These magnetic fields, G_x, G_y, and G_z, are called time-varying gradient fields and can change in strength over time. The gradient fields allow the origin of the measured signals to be positioned in the xyz-space. Receiver coils in the transverse plane will detect changes in magnetic flux perpendicular to the B_0 field. After excitation, the magnetization vector is precessing in the transverse plane, and due to Faraday's law of induction there will be an induced current in the coil. The created signal is termed the free induction decay (FID), which is the basis for all signal detection in MRI. However, when performing scans clinically, it is not the FID that is being measured but rather an echo version of it. This allows for a more efficient image acquisition.

When creating a structural MRI image, one cross-sectional slice with a certain thickness is being recorded at a time [22]. To choose a slice in the z-direction, the frequency of the RF pulse and the strength of G_z can be changed according to the Larmor equation. The thickness can be decided by changing the bandwidth of the RF pulse or the amplitude of the gradient.

To generate an MRI image, the data matrix k-space has to be filled [22]. Every slice has its k-space equivalent, and the k represents the x, y, z coordinate in the Fourier domain. The k-space is the Fourier equivalent of the image space, and is therefore in the frequency domain. This means that each coordinate in k-space will not represent the same physical coordinate in the image space, which is in the spatial domain. Each coordinate in k-space has a value which contains a signal proportional to the spin density, the T1 time and the T2 time in that specific coordinate. By changing the time-varying gradients, each coordinate in the k-space matrix can be filled with information. This way, the whole k-space will be filled after an MRI scan. To get the image space, an inverse Fourier transformation has to be performed on k-space.

2.1.5 Pulse sequences

MRI images are acquired through pulse sequences [21]. The most common sequences are spin echo and gradient echo. The sequences make use of the echo time (TE) which is the time between the initial RF pulse and the echo that is being recorded, and the repetition time (TR) which is the time between each excitation pulse. In most spin echo and gradient echo sequences, k-space is filled one line per RF-pulse.

Spin echo

The spin echo pulse sequence begins with a 90° RF excitation pulse [17]. This will align the magnetization with the transverse plane. The spins will then start to dephase in the transverse plane, before a 180 degree pulse is applied, reversing the spin phases. The following rephasing and dephasing generates the echo, and is the signal that is being recorded during these types of sequences.

Gradient echo

The gradient echo pulse sequence uses the gradients to create the echo [21]. After excitation, the gradient fields are used to cause a dephasing of the spins, which in turn will provide an echo when they rephase because the applied gradient fields are reversed. After TE, the echo will have a peak and this is the signal being recorded.

Echo planar imaging

Another often-used MRI technique is echo planar imaging (EPI). It is used for fast acquisition of images and is the technique most often used in fMRI [18]. A pulse sequence diagram for EPI can be seen on figure 2.3. The k-space is filled up as shown in figure 2.4. An EPI sequence works by capturing the whole k-space with just one excitation. Therefore, the time-varying gradients have to be strong enough to fill the entire k-space before significant T_2 or T_2^* decay can occur. The gradient in the x-direction changes between the same positive and negative value to move in the horizontal direction in k-space, while the gradient in the y-direction sends blips of the same magnitude and direction to move in the vertical direction to the next row in k-space. These blips can be seen as the small changes in the phase encoding gradient in figure 2.3. The number of rows in the k-space are determined by the number of echoes captured in the scan, and the number of columns are determined by the frequency encoding matrix, thus influencing how many echoes are captured from each slice [21].

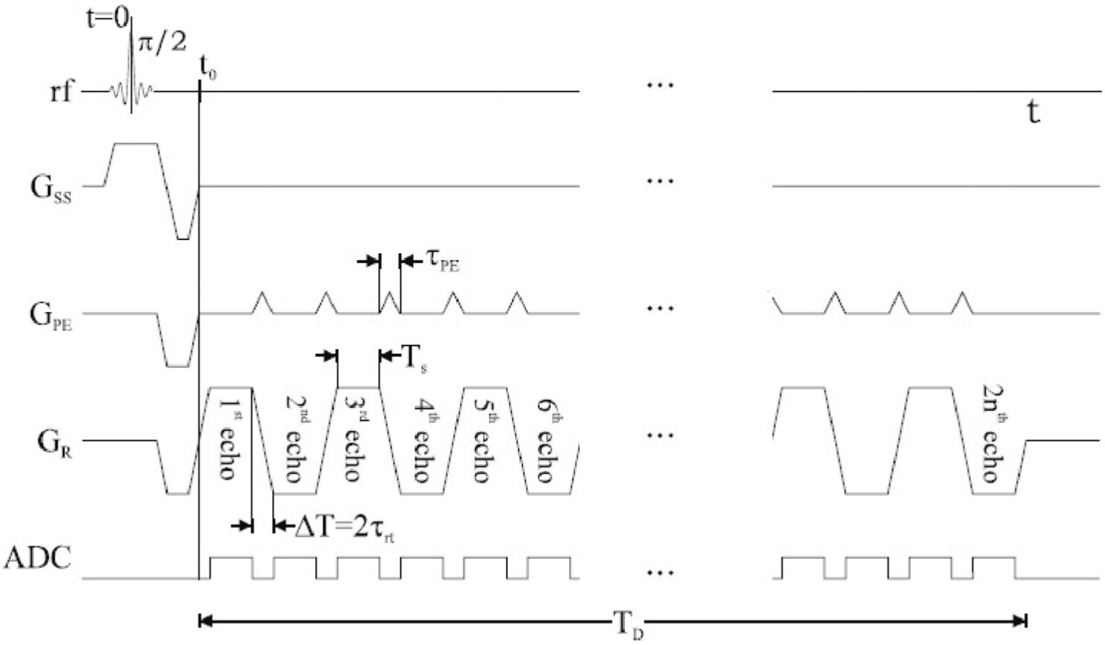


Figure 2.3: A pulse diagram for the EPI sequence. Figure taken from section 19.3 in [17], fig. 19.6.

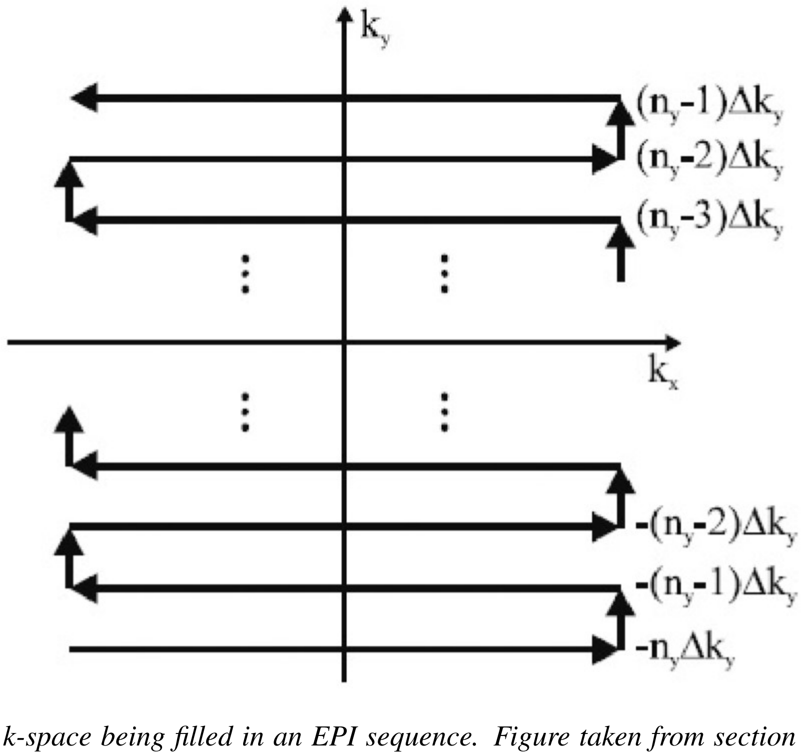


Figure 2.4: The k-space being filled in an EPI sequence. Figure taken from section 19.3 in [17], fig. 19.7.

2.1.6 Artifacts

Various artifacts can occur during MRI acquisition [18]. Artifacts can be divided into the three main categories: subject-related, hardware-related, and software-related. Typical hardware-related artifacts in EPI results from imperfections of the magnetic fields. Such imperfections can occur in both the strong main field and the gradient fields. This can lead to signal losses and distortions in the images, e.g. that a readout might not contain the full signal that it should have or that the signal has been moved to a neighbouring coordinate. Subject-related artifacts can be subject movement during the MRI acquisition. The movement can be either spatial or rotational. In addition, implants or other foreign bodies such as braces can create artifacts. Some artifacts are always present, such as those caused by fluid movement.

Within research using MRI, it is crucial to perform image processing. Some techniques are performed during scanning, while others are performed after image acquisition. There are many advanced preprocessing techniques that can be used to reduce the effects from certain artifacts on the images. Chapter 3.4 contains more information about preprocessing techniques that accounts for artifacts such as subject movement.

Artifacts in MRI images need to be identified to verify if the images are usable. If the artifacts severely affect the images, affected volumes can be removed, the subject can be removed from analysis, or the subject can be rescanned for inclusion in the study. In a data analysis context, images with artifacts that have not been removed from the dataset can potentially negatively affect the results. In addition, there is also a limit to motion correction and the images might have to be removed from the dataset if the subject movement is bigger than this limit.

2.1.7 fMRI

In order to do fMRI, the images have to be captured at a high rate. The imaging technique most commonly used is EPI because of the high temporal resolution of the resulting images. [18].

BOLD

Blood contains deoxyhemoglobin, which is paramagnetic [17]. By measuring the changes in image intensity that are hypothesised to originate from a change in deoxygenated blood, the cerebral blood usage can be examined. This principle is the basis for what is called Blood Oxygenation Level Dependant (BOLD) imaging, and is essential for fMRI [23]. A typical fMRI study is the finger-tapping experiment, where the subject is instructed to move the index finger while inside the MR-scanner. Using this paradigm, we will see activity in the primary motor cortex, since neuronal signals are being sent through the nervous system to initiate movement of the finger.

Neuronal activation in the brain is brief and fast. However, the BOLD response following the neuronal activation is delayed, and happens in the order of seconds after the activation [18]. Therefore, the signal being recorded in fMRI is not the neuronal activity itself but the delayed hemodynamic response to the neuronal activation, which can be recorded about 1 to 2 seconds later. Figure 2.5 shows the BOLD response to a short-duration stimulation. Note the short initial dip in BOLD signal right after the stimulus. Due to neuronal activity, the amount of oxygenated blood locally decreases, leading to a BOLD signal below the baseline. The neuronal activity increases the metabolic demands, increasing the amount of oxygenated blood to above baseline before reaching its peak. When returning to baseline, the BOLD signal returns to baseline, but decreases below it for a while. This effect is called the poststimulus undershoot. The signal being recorded in BOLD fMRI is the momentary loss in oxygenated blood over time, which is associated with the neuronal activity.

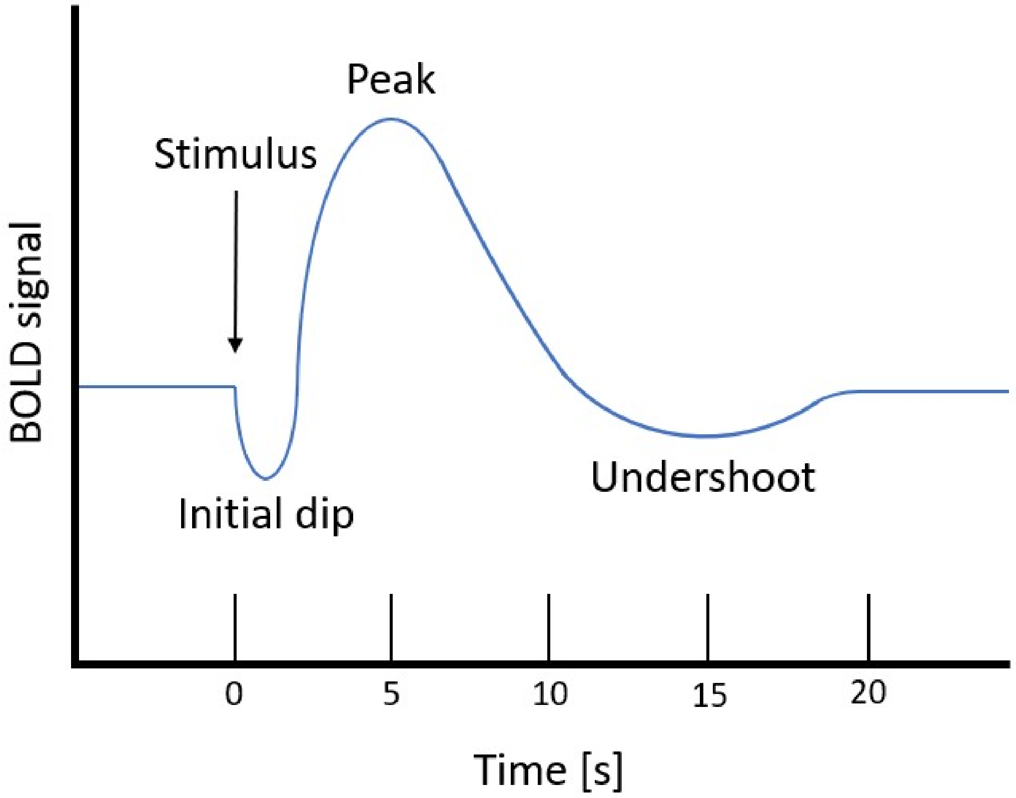


Figure 2.5: The BOLD signal response to a short-duration stimulus.

T_2^* contrast is the basis of BOLD-contrast imaging because T_2^*-weighted images are sensitive to paramagnetic changes in the tissue [18]. T_2^*-weighted BOLD contrast images have a long TR and an intermediate TE, typically 2000-3000 and 30 ms respectively at 3T [24]. The TR has to be long enough to let the longitudinal recovery to be almost complete and the T_1 contrast to be minimal, in addition to the fact that the whole k-space needs to be filled before the next excitation. If the TE time is too short, most of the transverse magnetization will still be present, no matter the T_2 time of the tissue. This results in no T_2 contrast. If the TE time is too long, all the transverse magnetization will be lost, also resulting in no T_2 contrast. The TE time also has to be set so that it is sensitive to the local field inhomogeneities.

2.1.8 Task-based and resting-state fMRI

The two main types of fMRI studies are task-based and resting-state studies. In task-based studies, the tasks being done by the subject are known as a paradigm, and can consist of a block design where every other block is a task block followed by a rest block where a fixation cross typically is presented on the screen. These blocks can be e.g. 30 seconds long and the duration and tasks vary based on the paradigm used for the study. For example, to study visual processing, a visual stimulus such as a flickering chequerboard, can be presented to the subject. To study auditory processing in the auditory cortex, a hearing task such as dichotic listening can be presented. Throughout the imaging session, volumes are acquired during task blocks and resting blocks. BOLD-EPI can be used to contrast brain activity during task and rest blocks. The goal is to associate the activations in the brain with the tasks that the subjects are performing. A common visual paradigm that have been used during fMRI is the Eriksen Flanker task [25], which can be used to assess cognitive control, i.e. the ability to ignore conflicting stimuli. In one version of the paradigm, five arrows are shown during the task block and the task is to decide which way the middle arrow is pointing. The subject has response grips in each hand during the fMRI session and is instructed to use the left-hand response grip when the middle arrow is pointing to the left and vice versa. The flanker stimuli consists of two conditions. During the congruent condition all arrows are pointing in the same direction, and in the incongruent condition the middle arrow points in a different direction than the surrounding flanker arrows.

In resting-state fMRI studies, the subject will not conduct any task [10]. Instead, the subject is instructed to rest, either with the eyes open or closed, but not asleep. This can be preferable in the study of psychiatric disorders, due to the technique not requiring any subjective input such as doing a task, making it easier to acquire. Resting-state fMRI can be used to study the functional connectivity in the brain, i.e. different anatomically separated brain regions being connected due to their simultaneous activations or fluctuations in resting-state [11]. The functional connectivity can be used to define brain networks, i.e. brain regions that are responsible for specific functions or related tasks [12]. An example of a brain network often studied with resting-state fMRI is the default mode network. Different techniques can be utilized for computing the functional connectivity. Seed-voxel based analyses rely on the choice of the initial seed in the brain and correlate the time series of this seed to the time series of every other voxel. Other techniques measure the whole brain functional connectivity based on data-driven or model-free methods, e.g. independent or principal component analysis. Functional connectivity studies can be used to study the brains of subjects with psychiatric disorders, and some studies have shown the correlation of dysconnectivity within the brain of subjects with SCZ [14–16, 26].

Task-based and resting-state fMRI can be used to investigate different aspects of brain activity, and both methods have their strengths and weaknesses. This study will focus on resting-state fMRI only to investigate the spontaneous fluctuations in brain activity which can be related to the characteristics of SCZ.

2.2 Machine Learning

ML is used when conventional computer programs do not suffice [27]. Problems where this could be the case is when recommending the next item to add to the shopping cart in an online store, when predicting the future prices of real estate, or when classifying disorders based on varying parameters. These kinds of ML models learn from experience, i.e. from previous examples. For the real estate example, a model could be trained on data from previously sold real estate, such as the price, area, location, and the year it was sold. This data would be called the training data and each instance would be called a data point. Each data point has a corresponding label, e.g. the price of the real estate, which is what the model is trying to predict. The variables that the model is using when training, e.g. the area, location, and year, are called features. The data which is used to test the performance of a model, where only the features are fed to the model and not the label, are called the test data.

For ML on fMRI images, the training data could consist of a portion of a group of subjects and the test data could be the remaining subjects. A data point could be the whole time series for a single subject, the label could be the sub-group of that subject and the features could be all the BOLD activations in the scan, i.e. the numerical voxel values in their coordinate position.

ML is the basis for artificial intelligence (AI), and there are many modern applications of this. OpenAI has released multiple of these. Given a text prompt, ChatGPT can create answers, provide information and hold conversation [28], DALL-E can create custom, realistic looking images [29] and Sora can generate videos up to a minute in length while maintaining high quality [30].

ML can be very useful on complex data, such as imaging data in medicine. Modern ML applications in medicine can e.g. help with diagnosis and patient treatment planning, or predict disorders and their development [31]. R. Fakoor et al. [32] have used ML to improve the accuracy in cancer classification problems by looking at gene expression data, Davatzikos et al. [33] have used ML to predict whether subjects lied or told the truth during an fMRI scan and A.H. Shoeb et al. [34] managed to use ML to detect the onset of epileptic seizures by looking at EEG data.

A ML algorithm works by saving parameter weights, which are used when creating a prediction on the test data [27]. These weights are changed and fine-tuned throughout training, and the more training data available, the more these weights can be tuned to potentially improve the performance of the model. For a classification problem, a ML algorithm will output the different classes that it has been trained on. A binary classification problem is when the labels are either/or, i.e. true or false, yes or no, 0 or 1. An example of this is a ML algorithm which will predict whether a disorder is present or not. A multilabel classification problem consists of multiple different labels, and a prediction will result in one of these labels, e.g. predicting youth, adult, or elderly from a set of brain MRI images. The real estate example is what is called a regression problem, where the prediction can be any continuous number and not belonging to any class. The example classification and regression problems are examples of supervised learning, which is when a model is trained on a dataset where the labels are known. Contrary to this is unsupervised learning, which is when the labels are not known and the ML model is asked to create groups or clusters of the inputs.

2.2.1 Linear regression

The most simple form of ML is linear regression, and a model could look like this:

$$\hat{y} = w_1 * x_1 + \ldots + w_n * x_n + b \tag{2.13}$$

where $\hat{y}$ is the prediction, x_1 - x_n are the features, w_1 - w_n are the corresponding weights for each feature and b is the bias. The weights determine how much impact each feature has on the result, and the bias is the result when all the features are zero. The performance of equation 2.13 as a ML model strongly depends on the correct choice of weights and bias. For a prediction with multiple data points, equation 2.13 can be rewritten as

$$\hat{\mathbf{y}} = X\mathbf{w} + b \tag{2.14}$$

where $\hat{\mathbf{y}}$ is the prediction vector, X is a two-dimensional matrix consisting of rows of data points with features as columns and $\mathbf{w}$ is a one-dimensional vector containing the weights of each feature.

In ML, the weights and biases will update themselves during training [27]. To do this, a performance measure has to be used to check if the model is doing good or not. This is usually done with a loss function, which calculates how much the model results are missing its targets. The squared error is a common loss function for linear regression and is as follows:

$$l^{(i)}(\mathbf{w}, b) = \frac{1}{2}(\hat{y}^{(i)} - y^{(i)})^2 \tag{2.15}$$

where $\hat{y}^{(i)}$ is the predicted value for data point i and $y^{(i)}$ is the true value for data point i. The performance of a model is then the average of all the losses on a dataset with n data points:

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(\mathbf{w}, b) \tag{2.16}$$

The model best fit for the training data is when $\mathbf{w}$ and b are chosen so that the loss is minimized when predicting on the training data [27]. This is often done by using stochastic gradient descent (SGD) or minibatch SGD, which is an optimization algorithm that finds the direction in which the parameters need to be updated in order to lower the loss function. Either SGD or a variant of it is used in most applications nowadays.

2.2.2 Stochastic gradient descent

Minibatch SGD works by finding the gradient of the loss function, i.e. the derivative of the loss function, to find which way to go to lower it [27]. Instead of taking the derivative of the whole loss function, or taking the derivative of every individual loss function for each data point, which can both take a lot of time, this algorithm takes the derivative of the loss function of a minibatch of samples at a time. The algorithm takes a minibatch B_t for each iteration t and computes the gradient, i.e. the derivative, of the average loss over the minibatch, before multiplying this average loss with the learning rate η. This is then subtracted from the current parameter values and results in an upgrade in parameters. The update looks like the following expression:

$$(\mathbf{w}, b) \leftarrow (\mathbf{w}, b) - \frac{\eta}{|B|} \sum_{i \in B_t} \partial_{(\mathbf{w},b)} l^{(i)}(\mathbf{w}, b) \tag{2.17}$$

where $|B|$ is the size of the minibatch, which is divided upon to normalize due to averaging. When optimizing with SGD, there might be local minima and/or saddle points, which might lead to the loss function not providing the desired parameter updates. This can be controlled by e.g. trying different learning rates, or changing the learning rate throughout training.

2.2.3 Machine learning in practice

A common misconception is that a ML algorithm and a ML model are the same thing, i.e. that the words are interchangeable. However, the difference is that every ML model has the structure of a ML algorithm. That means that a ML algorithm is the structure, i.e. the layers and their order. An instance of a certain ML algorithm is called a model. Therefore, a ML algorithm can not be trained but a ML model with the structure of a ML algorithm can be trained. Another couple of words that are often mixed or misunderstood are parameters and hyperparameters. A parameter is something that is saved within a model, e.g. a weight in a layer of a model. The parameters are what defines the model, and is the difference between two models with the same algorithm structure. A hyperparameter is the value that defines the ML algorithm, and is usually specified by the person who created the algorithm. Examples of hyperparameters are the learning rate and the layer size of a layer in the algorithm structure.

An important principle in ML is generalization, which means that the goal is not to recognize previously seen examples but to learn from patterns to be recognized in new, unseen data [27]. This means that models should not be overfit to the training data, which is when the models' parameters are fit too closely to the training data instead of the underlying patterns in the data. To combat this phenomenon, there exist methods called regularization methods. One of these methods is to increase the dataset size. Smaller datasets are more likely to cause overfitting, and introducing more data to learn from generally does not make a model perform worse. In addition, a bigger dataset opens the possibility for more complex models. A very complex model will likely overfit on a small dataset, but can potentially find patterns better in a larger dataset. When using small datasets, however, simpler models tend to outperform the overly complex ones.

Another way to combat overfitting is the introduction of validation data [27]. The validation data can be used as temporary test data so that the model can be validated throughout or in-between training to test the generalization of the model. This is called hold-out validation and is used in a process called model selection.

2.2.4 Model selection

Model selection is the process of selecting the best-performing model from a series of differently trained ML models [27]. They could vary in many ways, e.g. complexity, structure, hyperparameters, or level of preprocessing of the input. The choice of model depends on the chosen performance measure, e.g. the highest accuracy score or the lowest loss. In addition, the model selection procedure is not always done on the whole dataset, but on a subset in order to speed up the selection process due to large datasets, which is often the case in modern ML. However, the subset has to be representative of the whole dataset in order for the model selection to be justified. When the model selection is done, i.e. when the best hyperparameter combination has been chosen, a new model with the same hyperparameters can be trained on the whole training dataset.

The whole basis for ML is that there is an underlying assumption called the Independent and Identically Distributed (IID) assumption, which assumes that the training and test datasets are drawn independently from identical distributions [27]. Without this assumption, there would be no reason to believe that a model trained on training data would perform on an unseen test dataset, because the training data would not be representative of the test data. An example of where this could be a problem is if the training data consisted of only males and the test data consisted of only females. Based on the ML task, this could result in the IID assumption being invalid.

In addition to the IID, the dataset being used will have to be of a certain quality. The quality of the output will always reflect the quality of the input. Therefore, it is important to be aware of any outliers, errors, or loss in information in the dataset to address this. Examples of this could be a dataset containing numbers with too few decimals, or if the labelling of the dataset does not make sense.

Normal ML practice is to leave the test dataset completely unseen until used [27]. This is to not add any subjective bias, as the test dataset should represent any possible data point that can be used as input and shows the generalization of the model. Therefore, the test dataset should not be used or looked at before the model selection procedure has found place. The only thing to be in control of considering the test dataset is that it is actually a good representation of unseen data.

2.2.5 Weight decay

Back to the subject of regularization methods, weight decay is a method used to reduce the change of overfitting [27]. To introduce weight decay, first recall back to the loss function. The objective was to minimize the loss function when predicting on the training data. However, consider a new objective, to minimize the prediction loss in addition to a penalty term in the form of the size of the weight vector **w**, namely $||\mathbf{w}||^2$. The reason for using the square is for computational convenience, since the formula for calculating what is called the ℓ_2 *norm* is given by

$$||\mathbf{w}||_2 = \sqrt{\sum_{i=1}^{n} w_i^2} \tag{2.18}$$

The new objective, with both the loss function and the penalty term, is to minimize the following objective function J:

$$J = L(\mathbf{w}, b) + \frac{\lambda}{2}||w||^2 \tag{2.19}$$

where λ is the regularization constant, which controls the trade-off between minimizing the loss function and the penalty term. The ℓ_2 takes into account the large components of the weight vector **w**. This will make **w** less likely to be overfit to the training data, but instead make the weights more generalized. It will make the smaller weights bigger and the bigger weights smaller, so that more of the features will be used when predicting. This will also combat overfitting, because there is not just one single feature which will lead to the prediction result. Since more of the features are used, the generalization power of the model will be higher. Nevertheless, it's also important to think about the possibility that all the information that the ML model learns from lies within a small amount of features, i.e. if the prediction result only relies on a few single features. However, in fMRI image classification, where every voxel in an fMRI volume is a feature, the probability for this exact thing to happen is very small.

2.2.6 Classification

Classification in ML works similar to regression, but with a significant difference [27]. Instead of returning a single numerical value, a classification model returns a list containing the probabilities for each class. When the model predicts a certain class, it predicts the class with the highest of the probabilities in that list. This way, the model structure can stay almost the same but predict a class instead of a number. However, the numerical value that comes from the model is still important when calculating the loss.

Consider a classification problem where the input is a 2x2 grayscale image, and there are 2 classes to predict. The input image then consists of 4 numerical values x_1, x_2, x_3 and x_4. Every input value would have its corresponding weight, w_{nm}, and each output would have its own bias term, b_n. Using a similar model to 2.14, a classification model could look like this:

$$\begin{aligned} o_1 &= x_1w_{11} + x_2w_{12} + x_3w_{13} + x_4w_{14} + b_1 \\ o_2 &= x_1w_{21} + x_2w_{22} + x_3w_{23} + x_4w_{24} + b_2 \end{aligned} \tag{2.20}$$

Written more formally, in a notation more suited for code, the model can be written as

$$\mathbf{o} = \mathbf{W}x + \mathbf{b} \tag{2.21}$$

where **W** is a 2x4 matrix containing the weights and **b** is a 1-dimensional vector containing the two biases. **o** is the output vector. This model can also be shown as a neural network (NN) with an input layer of size 4 and an output layer of size 2, see figure 2.6.

Given that the outputs in **o** are probabilities, they should sum up to 1. Additionally, they should not be negative. To account for this, a softmax function can be applied. The softmax function takes the outputs **o** and converts them into a probability distribution $\hat{\mathbf{y}}$. The softmax function can be described as

$$\hat{\mathbf{y}} = softmax(o) \;\; where \;\; \hat{y}_i = \frac{exp(o_i)}{\sum_j exp(o_j)} \tag{2.22}$$

This ensures that the output list of the probabilities sums up to 1 and that none of them are negative, while still maintaining the relative difference between the probabilities.

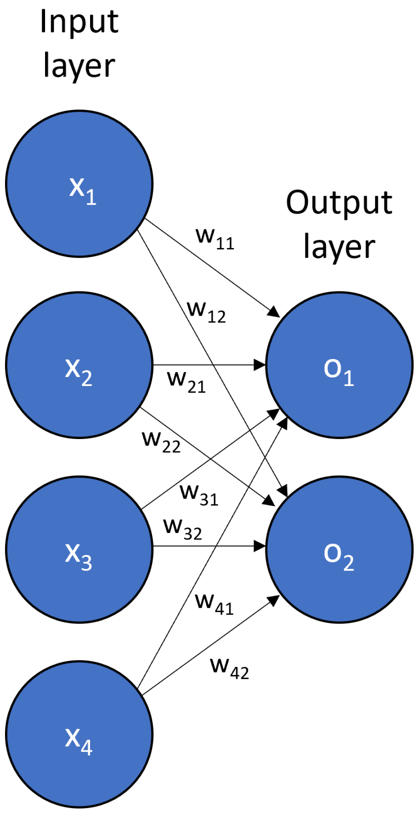


Figure 2.6: A simple classification model with 4 nodes as input and 2 nodes as output

2.2.7 Cross-entropy loss

For a classification task, the squared error is not usable. Maximum likelihood estimation is a good alternative for a classification task [27]. The probability for each outcome to be a certain class given the features are

$$P(\mathbf{Y} \mid \mathbf{X}) = \prod_{i=1}^{n} P(\mathbf{y^{(i)}} \mid \mathbf{x^{(i)}}) \tag{2.23}$$

where $\mathbf{Y}$ is the label vector and $\mathbf{X}$ is the feature matrix. Instead of maximizing equation 2.23, it is easier to minimize the negative log-likelihood, shown in the following equation.

$$-log\ P(\mathbf{Y} \mid \mathbf{X}) = \sum_{i=1}^{n} -log\ P(\mathbf{y^{(i)}} \mid \mathbf{x^{(i)}}) = \sum_{i=1}^{n} l(\mathbf{y^{(i)}}, \hat{\mathbf{y}}^{(\mathbf{i})}) \tag{2.24}$$

where $\hat{\mathbf{y}}^{(\mathbf{i})}$ is the model prediction, i.e. the class probability. Therefore, the loss function when predicting over q classes is then

$$l(\mathbf{y},\hat{\mathbf{y}}) = -\sum_{i=1}^{q} y_j \; log \; \hat{y}_j \tag{2.25}$$

which is called the cross entropy loss. This loss function is one of the most commonly used losses for classification problems. The reason for the name is that for each new data point, the model might miss the correct label and be "surprised" at the result, by a factor of $-log \; P(j)$ for each event j, after having assigned a probability $P(j)$ for the event. The "expected surprisal" is then the same as the definition of entropy for a distribution P:

$$H[P] = \sum_{j} -P(j) \; log \; P(j) \tag{2.26}$$

With P being a distribution of probabilities that the data is generated from, and Q being a distribution of the observer's subjective probabilities, the cross-entropy from P to Q is given by

$$H(P,Q) = \sum_{j} -P(j) \; log \; Q(j) \tag{2.27}$$

which is the same as the expected surprisal of the observer. Thus, it can be said that the cross-entropy loss is the same as minimizing the surprisal or maximizing the likelihood.

Weighted cross-entropy loss takes into account class imbalance in datasets. By providing class weights to the loss function, each weight will be multiplied to the loss function to increase the loss for the less represented classes and decrease the loss for the more represented classes, to punish the ML model for only predicting the most represented class.

2.3 Deep Learning

DL is a sub-genre of ML and is mainly the use of many-layered NNs [27]. Image recognition, AI and self-driving cars are examples that can utilize DL.

2.3.1 Neural Networks

A NN is a type of ML algorithm built up by neurons with weights between them [27]. They are in the form of layers (rows) of neurons beside each other, where every neuron in a layer is connected to every neuron in the layer before and after, but not to the other neurons in the same layer. For this reason, NNs are also known as fully connected layers. The neurons are connected by weights, just like the weights on the linear regression model. In fact, the simple example model could be seen as a one-layered NN, with all the inputs as one layer and the output prediction as the output layer.

Usually, NNs have more than just one input layer and one output layer [27]. In addition to these two layers, they can have hidden layers in between. A simple 6-layered example of a NN is shown on figure 2.7. The reason it is called 6-layered and not 7-layered is because the input layer does not require any calculations.

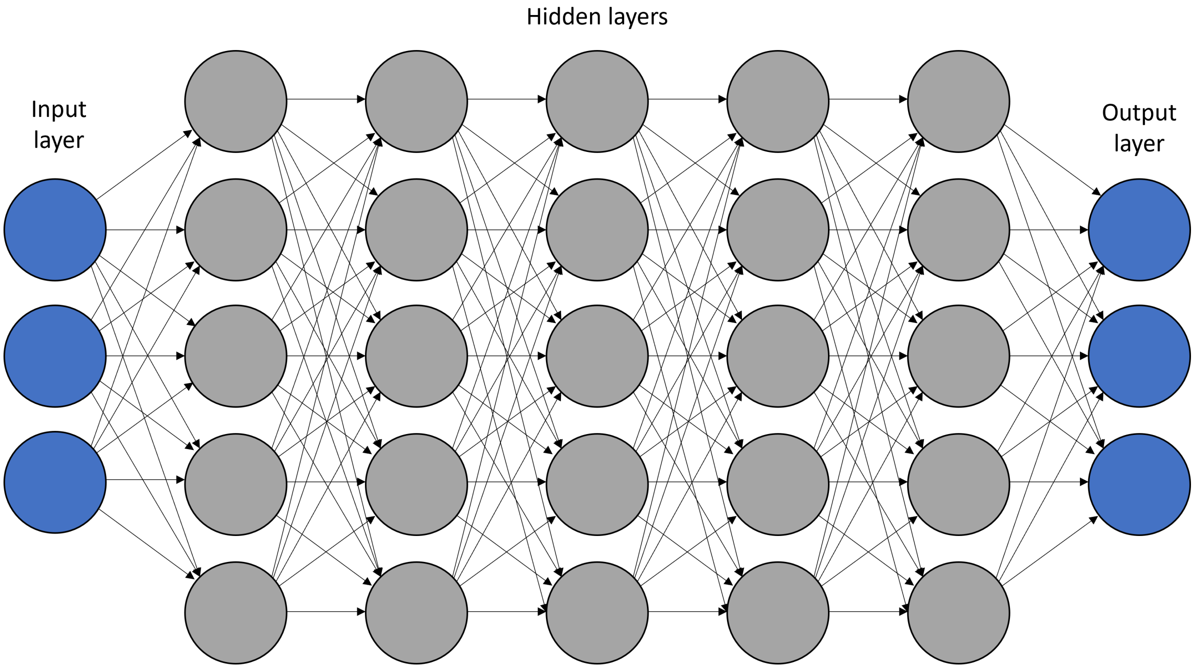


Figure 2.7: An example of a simple fully connected neural network with 5 hidden layers.

Given a 2-layered NN, i.e. with one hidden layer, the outputs of each layer can be written as

$$\begin{aligned}\mathbf{H} &= \mathbf{X}\mathbf{W}_1 + \mathbf{b}_1 \\ \mathbf{O} &= \mathbf{H}\mathbf{W}_2 + \mathbf{b}_2\end{aligned} \tag{2.28}$$

where **H** is the output of the hidden layer, **X** is the input, **W** are the weights, **b** are the biases and **O** is the output from the last layer [27]. A characteristic of NNs is that they have non-linear activation functions after each hidden layer. The activation function will operate on every value that comes out of a hidden layer. A common activation function is the rectified linear unit (ReLU) which is given by

$$\sigma(x) = max(0, x) \tag{2.29}$$

Adding this to equation 2.28 will change the NN to

$$\begin{aligned}\mathbf{H} &= \sigma(\mathbf{X}\mathbf{W}_1 + \mathbf{b}_1) \\ \mathbf{O} &= \mathbf{H}\mathbf{W}_2 + \mathbf{b}_2\end{aligned} \tag{2.30}$$

2.3.2 Forward propagation

When iterating through a NN, the inputs go through every layer, being updated by weights, biases, and activation functions before they become an output. This is what is called forward propagation, and is the process of calculating a prediction [27]. It is called forward because it orderly goes forward in the direction of the model structure, from input to output. One pass through a network is called a forward pass.

Computational graph

A computational graph shows the steps in the forward propagation [27]. For the two-layered network given in equation 2.30, a computational graph where the input size is 2, the hidden layer size is 4 and the output size is 2 could look like the one given in figure 2.8.

2.3.3 Backpropagation

As stated previously, the weights in ML models update during training by looking at which direction they need to be updated in order for the loss function to decrease, e.g. by minibatch SGD. However, with NNs this becomes a problem as SGD only calculates the gradient for a single variable. A NN is often many-layered, meaning that the total loss function would contain a lot of functions-within-functions. To account for this, gradient calculation for NNs are done using backwards propagation, or simply called backpropagation. The gradients are then provided to an optimization algorithm, e.g. minibatch SGD. It is called backpropagation because of how it moves in the opposite direction of the forward propagation, i.e. from the output to the input.

Backpropagation calculates the gradients of the NN parameters in a backwards fashion by using the chain rule, because of the nested characteristics of the functions in a NN. This way, the parameters can be updated by an optimizer based on the gradient that one

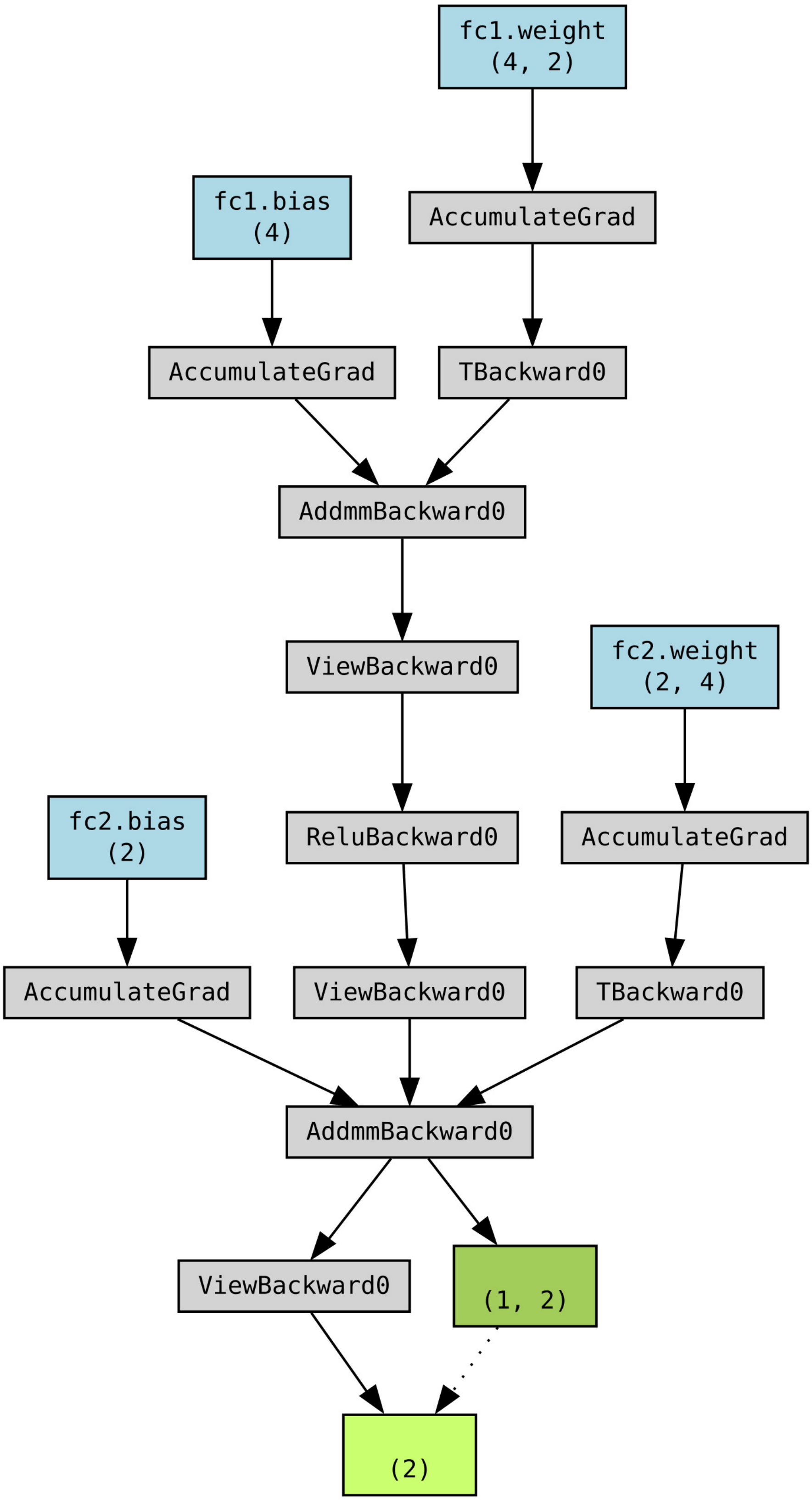


Figure 2.8: A computational graph of a simple example 2-layered NN, showing the dimensions of the input weights and biases for each layer along with every calculation that is being done in between, such as the gradient calculation and ReLU activation.

gets when going backwards through the network. The chain rule is given by

$$\frac{d}{dt}f(g(t)) = f'(g(t))g'(t) \tag{2.31}$$

or more simply written as

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx} \tag{2.32}$$

Given a two layered NN like the one in equation 2.30 with the following properties

$$\begin{aligned}
z_i &= \sum_j w_{ij}^{(1)} x_j + b_i^{(1)} \\
h_i &= \sigma(z_i) \\
y_k &= \sum_i w_{ki}^{(2)} h_i + b_k^{(2)} \\
L &= \frac{1}{2}\sum_k (\hat{y}_k - y_k)^2
\end{aligned} \tag{2.33}$$

and a computational graph like the one given in figure 2.8, the vectorized forms would look like this:

$$\begin{aligned}
\mathbf{z} &= \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)} \\
\mathbf{h} &= \sigma(\mathbf{z}) \\
\hat{\mathbf{y}} &= \mathbf{W}^{(2)}\mathbf{h} + \mathbf{b}^{(2)} \\
L &= \frac{1}{2}||\hat{\mathbf{y}} - \mathbf{y}||^2
\end{aligned} \tag{2.34}$$

where σ is the activation function and $\mathbf{y}$ is the true value. The parameters to be updated are $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, $\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$. The gradient to be computed is therefore

$$\nabla_{(\mathbf{W},\mathbf{b})}\, L(\mathbf{W},\mathbf{b}) = \left[\frac{\partial L}{\partial \mathbf{W}^{(1)}}, \frac{\partial L}{\partial \mathbf{b}^{(1)}}, \frac{\partial L}{\partial \mathbf{W}^{(2)}}, \frac{\partial L}{\partial \mathbf{b}^{(2)}}\right] \tag{2.35}$$

The backwards pass for this network would then look like this:

$$\frac{\partial L}{\partial \hat{\mathbf{y}}} = \hat{\mathbf{y}} - \mathbf{y}$$

$$\frac{\partial L}{\partial \mathbf{W}^{(2)}} = \frac{\partial L}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{W}^{(2)}} = \frac{\partial L}{\partial \hat{\mathbf{y}}} \mathbf{h}^\top$$

$$\frac{\partial L}{\partial \mathbf{b}^{(2)}} = \frac{\partial L}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{b}^{(2)}} = \frac{\partial L}{\partial \hat{\mathbf{y}}}$$

$$\frac{\partial L}{\partial \mathbf{h}} = \frac{\partial L}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{h}} = \mathbf{W}^{(2)\top} \frac{\partial L}{\partial \hat{\mathbf{y}}}$$

$$\frac{\partial L}{\partial \mathbf{z}} = \frac{\partial L}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} = \frac{\partial L}{\partial \mathbf{h}} * \sigma'(\mathbf{z})$$

$$\frac{\partial L}{\partial \mathbf{W}^{(1)}} = \frac{\partial L}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}} = \frac{\partial L}{\partial \mathbf{z}} \mathbf{x}^\top$$

$$\frac{\partial L}{\partial \mathbf{b}^{(1)}} = \frac{\partial L}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{b}^{(1)}} = \frac{\partial L}{\partial \mathbf{z}}$$

which shows the chain rule in action. It can be seen, then, that the gradient to be given to the optimizer is

$$\nabla_{(\mathbf{W},\mathbf{b})} L(\mathbf{W},\mathbf{b}) = \left[\frac{\partial L}{\partial \mathbf{z}} \mathbf{x}^\top, \; \frac{\partial L}{\partial \mathbf{z}}, \; \frac{\partial L}{\partial \hat{\mathbf{y}}} \mathbf{h}^\top, \; \frac{\partial L}{\partial \hat{\mathbf{y}}} \right] \tag{2.36}$$

When training a NN, it alternates between forwards propagation and backpropagation [27]. First, the parameter values are used in the forwards propagation to calculate the loss function. Second, the backpropagation algorithm calculates the gradients for the parameters. Lastly, the optimizer updates the parameter values in order to minimize the loss function. This repeats itself until the network has trained on the whole training dataset. In addition, it is common in DL to loop over the training dataset multiple times. Each iteration over the training dataset is called an epoch. The number of epochs is usually set high, to make sure that the models learns enough from the training dataset. This is the reason it takes a lot of time to train a deep NN, because it calculates gradients for each data point multiple times.

2.3.4 Dropout

A regularization technique often used in DL is dropout [27]. Where applied, this method drops out parts of the layer given a probability p, hence the name. In a NN, this can cause some neurons in a layer to not contribute to the prediction. The dropout is put after a layer, and can be used multiple times in a model. Dropout can help with generalization because in the case of the predictions being very dependent on a single neuron, the model is forced to make a prediction with the rest of them. Not one single neuron is then used for the prediction. Thus, dropout can help combat overfitting.

In addition, the remaining neurons, i.e. the neurons that were not dropped out, are normalized by the number of remaining neurons. Given that an activation in a neuron is h, the activations in the remaining layers change by

$$h' = \frac{h}{1-p} \tag{2.37}$$

which is also called the debiasing of a layer, which will keep the expectancy of that layer. Note that dropout is usually only used when training, not when testing. However, it can be claimed that a model is more confident in its predictions if it keeps performing the same after multiple uses of dropout.

2.3.5 Optimizers

As stated previously, the backpropagation algorithm is not what updated the parameters in a NN, but is rather the algorithm that computes the gradients for the parameters, e.g. the weights and biases. The gradients calculated from the backpropagation is fed to what is called an optimizer, which is what updates the parameters [27]. The minibatch SGD is an example of a common optimizer.

The goal of an optimizer is usually to minimize an objective, which is usually the loss function [27]. In other words, since what the optimizer ever sees is the training data, the goal becomes to minimize the training error. Therefore, the choice of optimizer can completely change the outcome of a model, due to how well it can also reduce the generalization error and not just the training error. It is also worth noting that when an optimizer finds the best value for a gradient, it lowers the loss the most on training data. This does not necessarily mean that this is the best value for generalization.

The optimizer goes through the objective function step-wise to find where the gradient is lowest, preferably zero. The step length is the hyperparameter called the learning rate, and can also change the outcome drastically.

Some things to consider when choosing the optimizer is how they behave in different situations [27]. In the case of local minima in the objective function, which is not uncommon in the objective functions in DL, the optimizer might settle on these as the gradient becomes close to zero instead of keeping the search up for the global minimum. In addition, the learning rate will also affect the optimizer's ability to get stuck in the local minima. Given a large enough learning rate, the optimizer might skip the local minimum altogether.

Other phenomena to keep in mind are saddle points, which is when the gradient can be close to zero without it being neither a local nor a global minimum, and vanishing gradients, which is when the gradient is close to zero for a lot of values, like in the tanh-function. It looks like a plateau with a downhill at the end, and the optimizer will have to take a lot of steps towards this downhill before the gradient gets any lower. However, finding local minima, saddle points and vanishing gradients might have a positive effect on the model as this might also contribute to combating overfitting. The global minima give, in fact, the parameter values that minimizes the loss on the training data.

2.3.6 Convolutional Neural Networks

Convolutional neural networks (CNN) are designed specifically to learn from image data [27] and are the most common form of DL. As the name states, they use convolutions to learn patterns in images. This approach makes it desirable because it works well and requires less computing power relative to conventional NNs on larger images.

Invariance

Convolutions use a principle called spacial invariance, which states that one piece of spacial information in an image should still be same no matter where in the image it is, i.e. it should not matter if the same thing in two different images containing the same information are on different coordinates [27].

Convolutions

A convolution is defined as

$$(f * g)(x) = \int f(z)g(x-z)dz \tag{2.38}$$

where f and g are two functions [27]. This is the overlap of the two functions as one of them is moved across the other. In DL, where inputs usually are discretized, the convolution will look like this:

$$(f * g)(i) = \sum_{a} f(a)g(i-a) \tag{2.39}$$

However, in 2D space, the convolutions become more tricky. In practice, what a convolution does is that it contains one or more filters, also called kernels, that slide over the input image. See figure 2.9 where the 3x3 input is on the left side of the operator and the 2x2 filter is on the right side of the operator. The result of the convolution is the 2x2 image on the far right side.

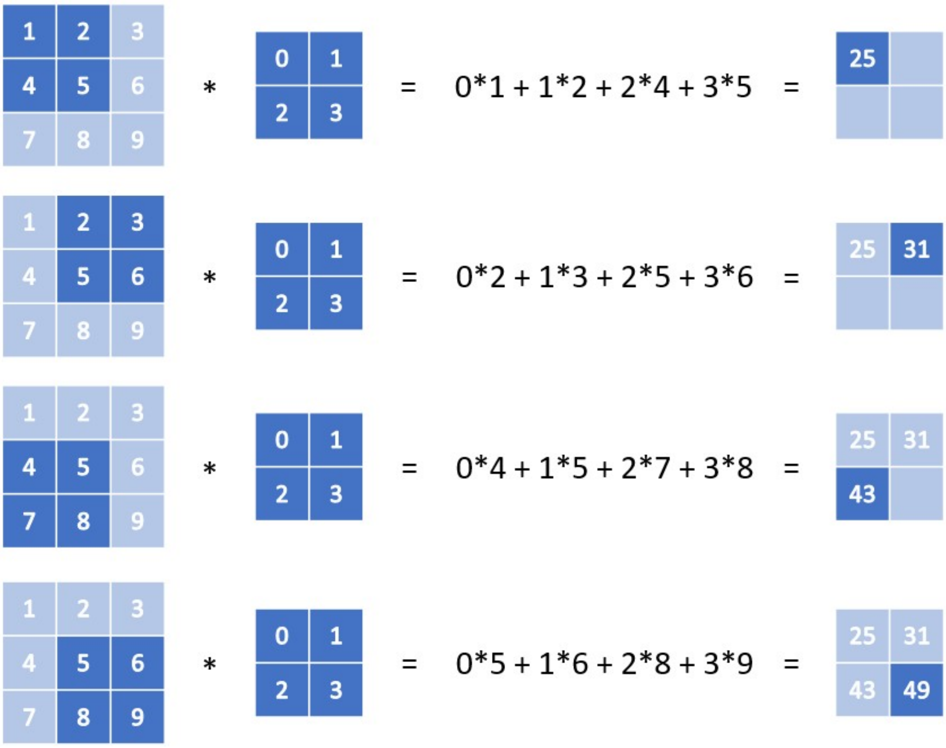


Figure 2.9: A step-wise example of a simple convolution with a 2x2 filter on a 3x3 input image. Modified figure from A. Zhang et al. [27].

The values in the filter is what changes during training. In 3D space, the convolution would be the same as in 2D, except that the input image and filter would have an extra dimension. The filter then moves row-wise in two dimensions, like in 2D space, before it takes a step in the third dimension and starts over again.

Padding and stride

In figure 2.9, the convolution has a stride of one and no padding, which means that the filter moves one pixel at a time and only within the borders of the input image. If the convolution had e.g. a symmetrical 0-padding of one or two, there would be added an outline of pixels with the value 0 around the input image before the convolution, see figure 2.10.

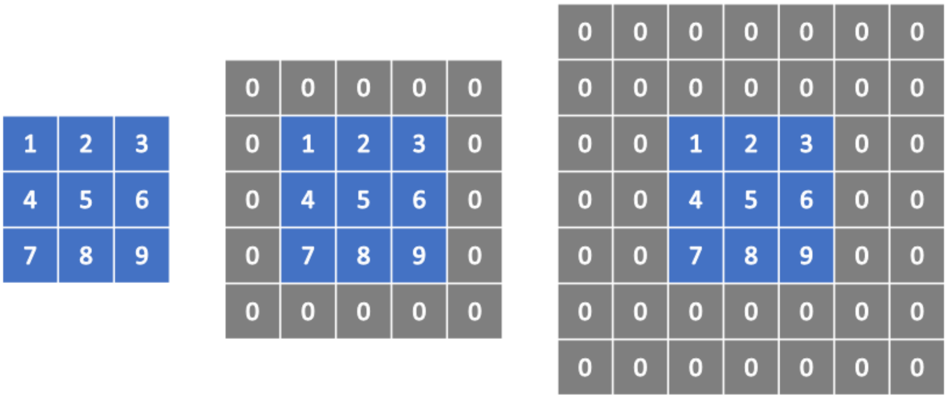


Figure 2.10: An example of how padding works in convolutions, with the images having a symmetrical 0-padding of 0, 1 and 2 from left to right. Modified figure from A. Zhang et al. [27].

Stride is the step length of the filter when it slides over the input image [27]. Figure 2.11 shows an example of a convolution where there is a 0-padding of 1 and a stride of 2. By choosing the padding, stride and size of the filter, the output dimension after the convolution can be controlled. Padding increases with output dimension, while stride and filter size decreases with the output dimension.

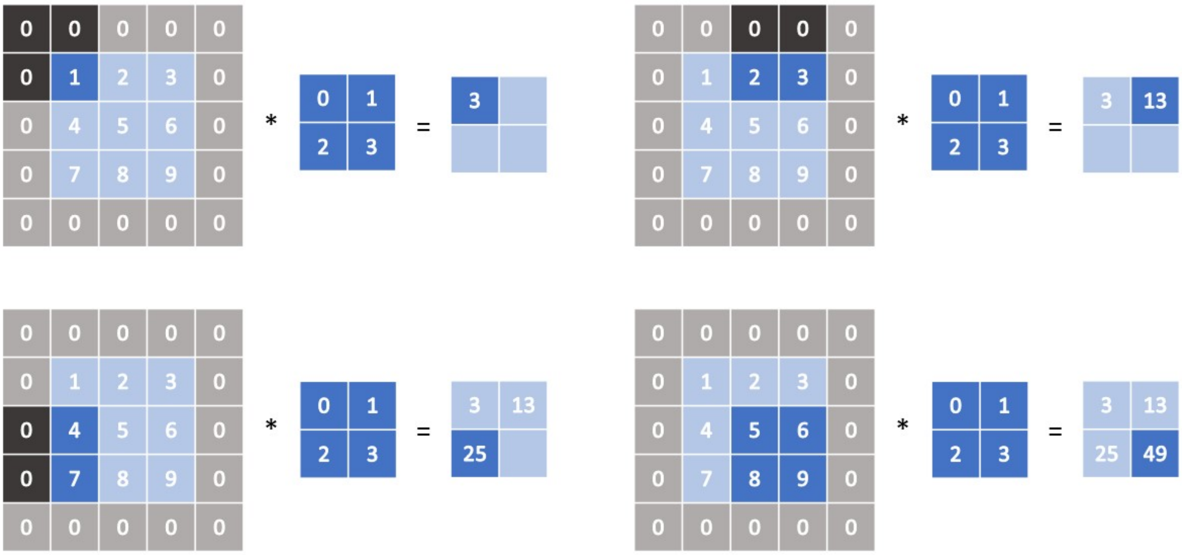


Figure 2.11: An example of how stride works in convolutions, with a symmetrical 0-padding of 1 and a stride of 2. Modified figure from A. Zhang et al. [27].

2.3.7 Pooling

Pooling takes into account the fact that the output of the final layer in a CNN should be sensitive to the entire input, i.e. that if the network were to look for something in the image, the final layer should "know" this no matter where that thing is in the image [27]. Like a convolutional filter, a pooling window slides over the input step-wise, but it doesn't contain any parameters. One often-used type of pooling is maximum pooling. Max-pooling computes the maximum value within the pooling window and assigns that value to the output. An example of a max-pooling operation can be seen on figure 2.12.

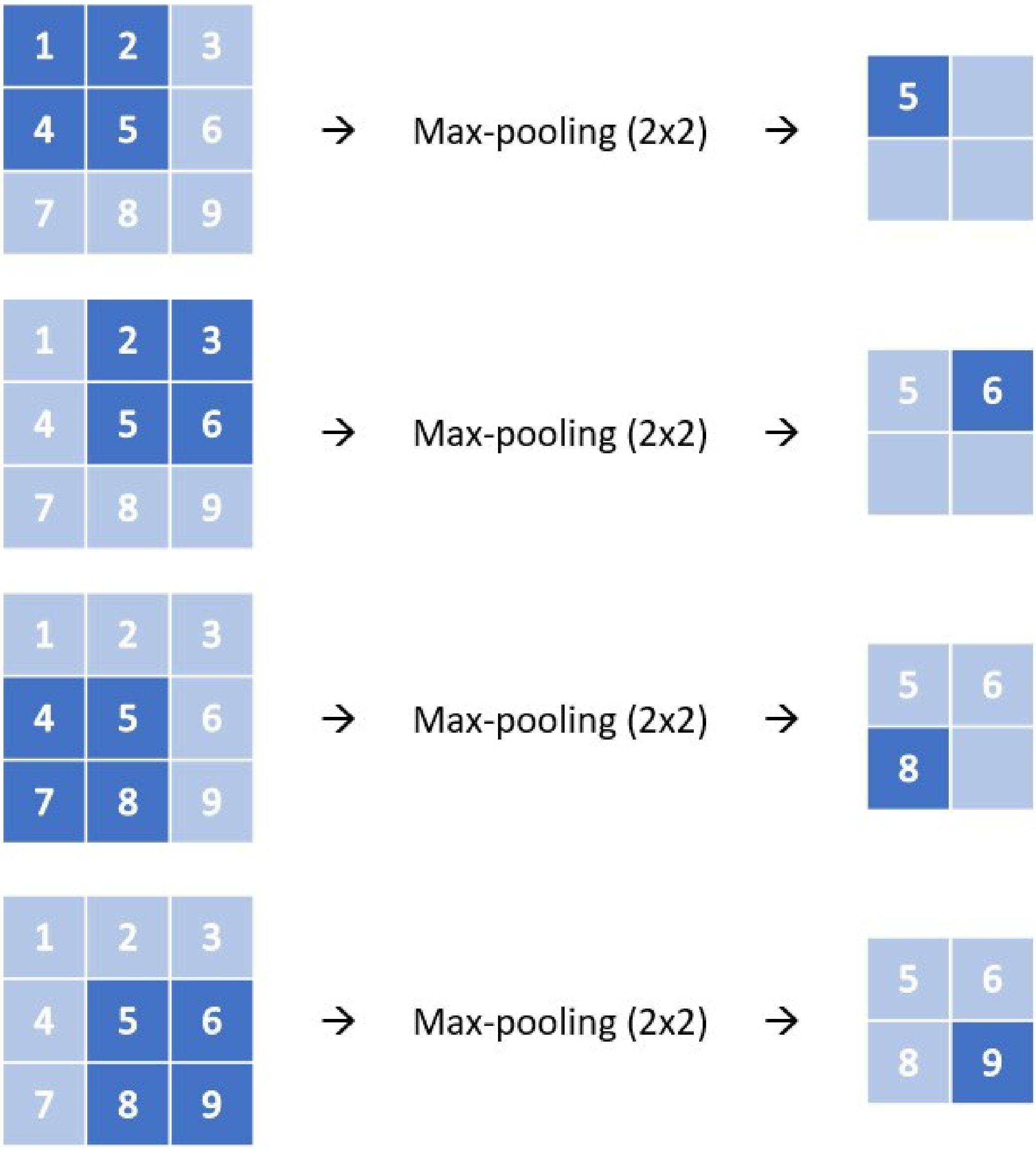


Figure 2.12: An example of how max-pooling works, with a 2x2 pooling window. Modified figure from A. Zhang et al. [27].

2.3.8 Channels

An input image can contain multiple channels [27]. The values in each pixel of the input is just a number, and does not contain any information on e.g. the colour of that pixel. Coloured images are therefore usually represented with 3 images, one for each of the colours red, green, and blue. This makes the input image consist of 3 channels. A channel can be seen as a dimension of the image. For example, each volume in an fMRI time series can be used as a channel. In a convolution, the number of filters depends on the number of input and output channels. With one output channel, each input channel has its own filter. However, for each output channel, there is a set of filters, with each set containing one filter for each input channel. Figure 2.13 shows how a convolution works with both multiple input and output channels.

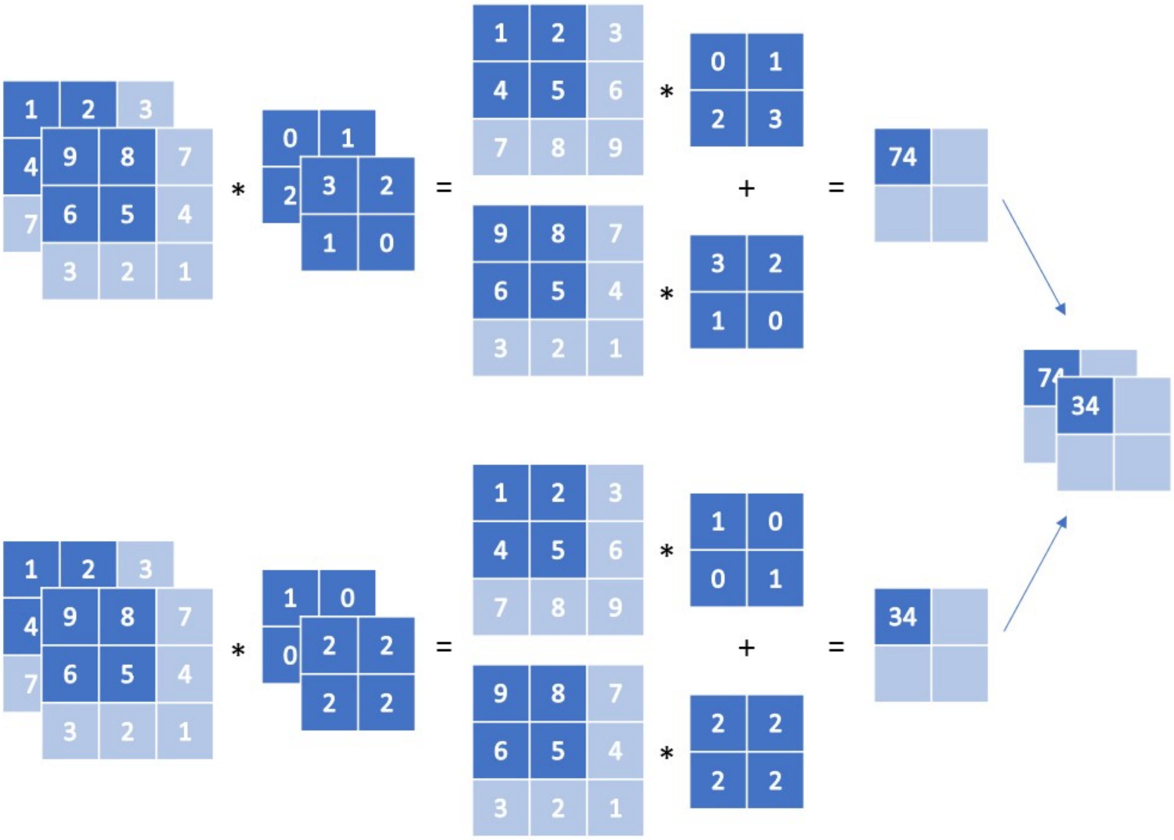


Figure 2.13: An example of how a convolution works when there are multiple input and output channels. Note that for every output channel, there is a set of filters with one unique filter per input channel. Modified figure from A. Zhang et al. [27].

2.4 Metrics

To assess and evaluate the performance of different ML models, the following metrics can be used. To define the metrics, different prediction outcomes are used. The true positive (TP) is when an actual positive value is predicted positive, the true negative (TN) is when an actual negative value is predicted negative, the false positive (FP) is when an actual negative value is predicted positive and the false negative (FN) is when an actual positive is predicted negative.

2.4.1 Accuracy

The accuracy score calculates the amount of correct predictions:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.40}$$

2.4.2 Precision

The precision score is defined as

$$Precision = \frac{TP}{TP+FP} \tag{2.41}$$

and is the number of correct positive predictions out of all the positive predictions. If precision is 1 there are no FP meaning that every predicted positive is an actual positive.

2.4.3 Recall

The recall score, also called sensitivity, is defined as

$$Recall = \frac{TP}{TP+FN} \tag{2.42}$$

Recall is the amount of correct positive predictions out of all the actual positives. It is therefore also called the True Positive Rate (TPR) and weights the amount of FN. If recall is 1 there are no FN meaning that every actual positive has been predicted positive.

2.4.4 Specificity

The specificity is defined as

$$Specificity = \frac{TN}{TN + FP} \tag{2.43}$$

The specificity is the amount of correct negative predictions out of all the actual negatives. It is therefore also called the True Negative Rate (TNR) and weights the amount of FP.

2.4.5 F1-score

The F1-score is defined as

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{2.44}$$

and is the harmonic mean of recall and precision.

2.4.6 Balanced Accuracy

The balanced accuracy score calculates the accuracy but takes into account the class imbalance. It is defined as

$$Balanced\ accuracy = \frac{sensitivity + specificity}{2} \tag{2.45}$$

but can also be thought of as the accuracy given weights to each class, i.e. if the positive class is twice as big as the negative, a correct prediction of the positive class will only count half as much towards the performance relative to a correct prediction of the negative class.

2.4.7 Statistical t-test

A statistical two-sample t-test compares the distribution between two populations and states if there is a statistically significant difference between the means of the populations [35]. The test assumes that the two distributions are normally distributed, have similar variances, and that they are independent of each other. The formula for Welch's t-test is as follows:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \qquad (2.46)$$

where $\bar{X}_1$ and $\bar{X}_2$ are the sample means of the two populations, s_1 and s_2 are their respective sample standard deviations and n_1 and n_2 are their respective sample sizes. A two-tailed test is to test whether the null-hypothesis, i.e. that the two populations have equal means, is true.

2.5 Schizophrenia

SCZ is a complex mental disorder which affects about 1 in 300 people worldwide [1]. Subjects with SCZ can experience psychosis in the form of hallucinations and delusions. The disorder is heavily taxing not just for the subject but also for their family and friends [2]. About 25% of subjects with SCZ experience one psychotic episode, and about the same percentage of subjects are diagnosed with a chronic variant of the disorder [5]. The remaining 50% have a combination of these, with at least one remission throughout their life with the disorder periodically being dormant. Out of all subjects with SCZ, approximately one third will be able to fully recover.

2.5.1 Symptoms and prevalence

The symptoms of SCZ are categorized into positive symptoms, negative symptoms and cognitive symptoms [5]. Positive symptoms mean that subjects experience the presence of perceptions that alter the way they experience reality and/or form their beliefs about the world. One example of a positive symptom is hallucinations, which can affect various senses such as auditory, olfactory, visual or tactile. The negative symptoms mean a reduction or absence of normal mental functioning. This can include loss or lack of emotion, attention, motivation, mild apathy, or antisocial behaviour. The cognitive symptoms affect the cognitive abilities of the subject. Examples can be problems with thinking and disorganized thinking.

2.5.2 Diagnostic criteria

According to ICD-11, to diagnose a subject with SCZ, two or more of the following symptoms must be present for a one-month period (one of the symptoms have to be from the first 4) [36]:

1) persistent delusions, e.g. grandiose delusions

2) persistent hallucinations, most commonly auditory

3) disorganized thinking, e.g. irrelevant speech and loose associations

4) experiences of influence, passivity or control, i.e. the experience that ones feelings, impulses, actions, or thoughts are not generated by oneself, are being placed in ones mind or withdrawn from ones mind by others, or that ones thoughts are being broadcast to others

5) negative symptoms as described above

6) grossly disorganized behaviour that impedes goal-directed activity, e.g. behaviour that appears bizarre or purposeless, unpredictable or inappropriate emotional responses that interferes with the ability to organize behaviour

7) psychomotor disturbances such as catatonic restlessness or agitation, posturing, waxy flexibility, negativism, mutism, or stupor

In addition, the symptoms can not be a manifestation of another medical condition and can not be due to the effects of a substance or medication on the central nervous system, including withdrawal effects.

2.5.3 Defining the disorder

Subjects with SCZ often do not have SCZ as their only diagnosis [37]. It is not rare to have multiple coexisting psychopathologies and/or comorbid medical conditions in addition to the main SCZ diagnosis. According to the U.S. Food and Drug Administration, comorbidity in SCZ has become the norm instead of the exception. This is connected to the fact that multiple pharmacotherapeutic agents that work effectively on SCZ can work on comorbid conditions as well, suggesting a biological overlap between the diagnoses. According to A.I. Green et al. [38] approximately 50% of subjects with SCZ suffer from at least one or more comorbid medical or psychiatric conditions, leading to a high mortality and morbidity rate.

The cause for SCZ is not completely understood, although it is thought to be a combination of genetic and environmental factors [39]. The biological effects can be found in the brain, and some of the genes that increase the risk for both SCZ, bipolar disorder, autism, and other neurological disorders are shared. Environmental factors that are thought to play a role in the onset of SCZ includes migration, cannabis abuse, old age in parents, childhood trauma and perinatal hypoxia, i.e. the lack of oxygen to the fetus during labour. It is difficult to define the exact onset of the disorder or predict the future development. As a result, personalized treatment is rare and the selection of treatment often follows a "trial and error" regime where a subject self-report of symptoms decides what medication is most appropriate [3]. Another contributing factor is the fact that the subjectiveness of both the subject and the clinician can play a role in the classification of the disorder. The subject might not be able to express their symptoms reliably, and the clinician might not be able to correctly comprehend what the subject is trying to express. Therefore, there is a need to move towards a more personalized approach [4].

There are discussions on whether there is need for a revision of the diagnostic criteria of the disorder [37], with Tandon et al. [39] questioning if SCZ is a "fundamentally flawed construct", claiming a lack of specificity in findings within SCZ research.

2.5.4 Sub-grouping of schizophrenia

SCZ is a genetically complex disorder and consists of a heterogeneous patient group. The disorder varies from subject to subject, and medication work on some subjects but not others [40]. Treatment of SCZ often follows a "trial and error" approach, where one drug is tested first, followed by a second drug if the first one does not work, etc. This keeps going until a potential working drug is found for the subject. However, this can take a considerable amount of time for some subjects. Correct classification of the mental disorder can aid in finding a working medication for the subject in question.

Subjects with SCZ have for a long time been separated into sub-groups [40]. Some argue that sub-grouping does not have clinical relevance since the patient group is so heterogeneous, while other practitioners may use sub-groups to categorize subjects based upon their symptoms. However, it is important to keep in mind that the mental disorder is complex and varies from subject to subject.

Paranoid Schizophrenia

"Paranoid schizophrenia, ICD-10 code F20.0, is dominated by relatively stable, often paranoid delusions, usually accompanied by hallucinations, particularly of the auditory variety, and perceptual disturbances. Disturbances of affect, volition and speech, and catatonic symptoms, are either absent or relatively inconspicuous." [41]. The sub-type is one of nine sub-types of SCZ in ICD-10. However, in the newer revision ICD-11, this sub-type along with the eight others have been removed because of their instability and invalidity [36]. Instead, the focus is more on the course of the disorder, i.e. if it is first-episode, multi-episode or continuous, in addition to if it is in remission or not. The sub-groups of SCZ used throughout this thesis are based on the sub-types from the ICD-10 classification.

2.5.5 Contribution from neuroimaging

Since the discovery of fMRI in the 90s, the method has been used to investigate brain activity in subjects with SCZ [5]. Many studies have used task-based fMRI to study the brain activity of patients where cognitive failure is a core feature. A recurring symptom of subjects with SCZ is loss or deficits in working memory. In addition, subjects with SCZ show alterations in executive functions such as planning and response inhibition, i.e. the ability to stop spontaneous actions. Assessing these abilities can help with diagnosis, but can also contribute to the understanding of the disorder in itself.

A 2023 study by K. Hugdahl et al. was performed while subjects experienced auditory hallucinations inside the MR-scanner [42]. They found a significant increase in activity in the ventromedial prefrontal cortex right before the reported start of the hallucination and a decrease right before the end of it. They concluded that this area of the brain could act as a switch when the subject experienced auditory hallucinations.

2.6 Previous contributions

2.6.1 COBRE dataset

The COBRE dataset [43–46] is an open and available dataset containing structural and resting-state fMRI on subjects with SCZ and healthy controls. This dataset has been used for binary classification on healthy controls and subjects with SCZ using DL. An article by R. Yu et al. [6] from 2023 used sparsity-guided multiple functional connectivity patterns, which takes into account the fact that the functional brain connectivity during resting-state can give information on neuropsychiatric disorders. Their method was to first create a sparse functional connectivity network, which shows the connectivity between different regions of interest (ROI), and then use CNNs on these network maps to learn patterns associated to SCZ. In addition, they included an analysis of regions that are potential biomarkers for SCZ. Using their model they had an accuracy score of 78.16%, a balanced accuracy score of 77.92% and an F1-score of 75.39%. Several regions in the brain showed consistent importance in several studies, such as the bilateral thalamus, right supplementary motor area, Lobule I- and II of vermis, right middle temporal gyrus, bilateral middle occipital gyrus, right lingual gyrus, right pallidum and left postcentral gyrus.

An article by J. Zheng et al. [7] used transfer learning to modify the VGG16 net [47], pretrained on the ImageNet dataset [48], to classify SCZ from the COBRE dataset using 2D convolutions on images preprocessed by binarization, standardization, and smoothing. They report an accuracy score of 87.85, a precision score of 87.11 and a recall score of 89.63, their model performing better than other popular ML algorithms and most of the state-of-the-art models on SCZ diagnosis. The accuracy score reported is 2.31 higher than that of the unmodified VGG16 model, and they claim that a more systematic and in-depth exploration is needed to progress on the subject.

2.6.2 Schizophrenia classification using structural MRI

By using the structural MRI images from scans, J. Oh et al. [8] in 2020 attempted to classify subjects with SCZ from healthy controls from 5 different datasets, one of them being the COBRE dataset. The methods consisted of preprocessing by normalizing before using 3D CNNs to classify. Similarly to R Yu et al. [6], the study by J. Oh et al. included an analysis on brain regions with the highest significance on the classification results. The article reports an accuracy score of 70.0 on a completely unseen, independent dataset. The subjects included in the test dataset had, however, both lower average disorder duration and lower average age, which could mean that the subjects in the test dataset has not progressed as far in the disorder. They found that the right temporal area and temporoparital areas informed their DL model about SCZ the most. They investigated the impact of the regions on accuracy of the DL models by excluding individual regions before prediction. The removal of the area roughly corresponding to the right temporal region gave the highest drop in accuracy. This indicates that the right temporal area changes the most during the development of the disorder.

One of the reasons behind the correct classification of SCZ subjects in DL models are the potential progressive changes structurally in the brain of subjects with SCZ, as studied by B. Oyabi et al. in 2011 [49]. This meta analysis included 27 studies and focused on 23 different ROIs. Altogether, 928 subjects with SCZ and 867 control subjects were studied over time, with the time between the initial and follow-up scans ranging from 1-10 years. The study showed that, on average, the change in certain ROIs in the brain either increased or decreased each year compared to healthy controls. The significant changes were ranging from the highest decrease of -0.59% for whole brain gray matter to the highest increase of +0.36% in the bilateral lateral ventricles, on average annually, relative to healthy controls. In addition, the study showed a decrease in the parietal white matter volume of -0.32%. They concluded that schizophrenic brains may have progressive structural abnormalities over time.

Another similar study from 2017 by S. Huhtaniska et al. [50] showed that the use of antipsychotic drugs had a high correlation with the decrease of parietal lobe volume and the increase of basal ganglia volume in subjects with SCZ. The results were analyzed on thirty-one studies and no other regions of the brain had statistically significant changes.

2.6.3 Resting-state and task-based fMRI

T. Mwansisya et al. [51] performed a systematic review in 2017 about functional changes in the brain in SCZ during resting-state and task-based fMRI. They included nineteen studies of first episode SCZ subjects compared to healthy controls. Among their findings, first episode SCZ subjects showed decreased signal or functional connectivity in different regions of the prefrontal cortex during resting-state fMRI. When combining the findings from the resting-state and task-based fMRI studies, the most prominent results were found in the temporal and frontal lobes. Specifically, hypo-activations were found in the dorsolateral prefrontal cortex, and abnormalities were found in the left superior temporal gyrus and the orbital frontal cortex. This supports their hypothesis that pathophysiological changes in the schizophrenic brain occurs in the frontotemporal pathway.

T. Mwansisya et al. [51] also suggest that the dysfunctional abnormalities in the dorsolateral prefrontal cortex and the orbital frontal cortex might correlate with functions that are diminished in subjects with SCZ, such as working memory, cognitive flexibility and social cognition. The hypo-activity in the dorsolateral prefrontal cortex has also been associated with the negative symptoms of SCZ. In addition, the superior temporal gyrus has often been associated with auditory hallucinations, which is a key symptom of SCZ. T. Mwansisya et al. suggest that the roots of SCZ are in the three areas mentioned, because of the correlations between these regions with brain abnormalities during fMRI scans, and that these regions plays a key role in the typical symptoms of SCZ.

Others have been using DL on resting-state fMRI to identify subjects on the autism spectrum amongst healthy controls. A.S. Heinsfeld et al. [52] did this in 2018 by collecting the mean time series activations from different ROIs for each subject, followed by calculating the functional connectivity between these regions. This resulted in a correlation matrix, which was flattened and used as input to a ML model, consisting of autoencoders and a fully connected NN. They reported an accuracy of 70%. M. Khosla et al. [53] in 2019 used the same dataset as Heinsfeld et al. with multiple channels of 3D images as input, where each channel consisted of the correlation between different ROIs. They used a 3D-CNN model to classify and reported an accuracy of approximately 72%. They mentioned that the choice of ROI can impact the performance when ML models are trained with functional connectivity.

A 2016 study on resting-state fMRI data from subjects with SCZ was done by J. Kim et al. [9]. In the same way as Heinsfeld et al. [52], they used a correlation map between each ROI from a template as input to the ML model and used a very similar classification pipeline using a NN. They reported an accuracy of 85.8% on the COBRE dataset, preprocessed using the SPM software using the same steps as reported later in this study, in addition to being smoothed with an 8 mm kernel followed by a functional connectivity extraction.

2.7 Thesis aim

The current classification of subjects with SCZ relies on a clinical assessment between psychiatrist and subject and reported symptoms. The aim of this thesis is to reveal if ML based on resting-state fMRI data (4D) can aid in sub-grouping with SCZ. Furthermore, it investigates the feasibility of this approach to distinguish subjects with SCZ from healthy controls. The following steps are proposed:

- To design, implement and run a deep learning (DL) pipeline that can handle 4D data.
- To optimize the ML model through hyperparameter fine-tuning.
- To investigate the effect of preprocessed imaging data on the performance of the ML model.
- To perform binary and multilabel classification of subjects with SCZ using DL models.
- To investigate and assess the performance of the models on an online dataset (N=148) and local datasets (N=316).

Chapter 3

Methods

3.1 The COBRE dataset

Preliminary training, validation, and testing has been done on the MRI scans from the COBRE-dataset from the Center for Biomedical Research Excellence [43–46], available online under a non-commercial creative commons licence. The dataset was collected and shared by the University of New Mexico and the scans were acquired on a 3T Siemens Trio scanner. The dataset contains anatomical T1-weighted images using a multi-echo MPRAGE (MEMPR) sequence and resting-state fMRI scans for 148 subjects in the age group 18 to 65, of which 75 were healthy controls. Before the original investigators started with the scanning, all the subjects were screened for any history of neurological disorders, mental retardation, head trauma, substance abuse or dependence. Table 3.1 shows the parameters for the anatomical scans, and table 3.2 shows the parameters for the resting state EPI-images.

TR [ms]	TE [ms]	Matrix size	Voxel size [mm^3]	Flip angle [°]	Slices
2530	1.64	256x256	1x1x1	7	176

Table 3.1: The imaging parameters for the anatomical images in the COBRE dataset.

TR [ms]	TE [ms]	Matrix size	Voxel size [mm^3]	Flip angle [°]	Slices	Volumes
2000	29	64x64	3x3x4	75	32	150

Table 3.2: The imaging parameters for the resting-state functional images in the COBRE dataset.

The dataset contained diagnostic information about the subjects, where subjects were diagnosed using the DSM-IV scale [54]. The distribution of the diagnoses in the patient group in the COBRE dataset can be seen on table 3.3. In the current study, 10 out of 148 subjects were excluded. Two were disenrolled, and eight subjects belonged to under-represented or non-relevant sub-groups. The number of subjects in the under-represented or non-relevant sub-groups were deemed too scarce to be used for ML. Therefore, the following sub-groups were excluded: 290.3, 295.2, 296.2, 296.4 and 311.

DSM-IV code	Main diagnosis	Frequency
290.3*	Senile dementia with delirium	1
295.1	Disorganized type schizophrenia	3
295.2*	Catatonic type schizophrenia	1
295.3	Paranoid type schizophrenia	41
295.6	Schizophrenic disorder, residual type	12
295.7	Schizoaffective disorder	7
295.9	Unspecified schizophrenia	6
296.2*	Major depressive affective disorder single episode	1
296.4*	Bipolar I disorder, most recent episode (or current) manic	1
311*	Depressive disorder, not elsewhere classified	1

*Table 3.3: The different diagnoses present in the COBRE dataset, along with their respective diagnosis codes and frequency. * excluded.*

In addition to the anatomical scan data and the functional resting-state scan data, the COBRE dataset also contains the age, gender, and handedness of each subject at the time of scanning.

Example images of a functional volume from the COBRE dataset are shown on figure 3.1 with the sagittal view on the left, the coronal view in the middle and the axial view on the right. The images are from the first time point in the time series. The corresponding T1-weighted anatomical images are shown on figure 3.2.

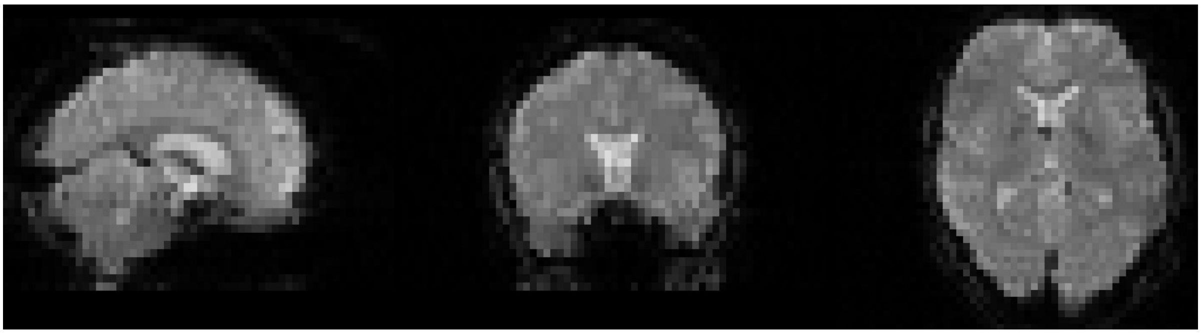


Figure 3.1: Example images of a raw functional volume from the COBRE dataset. Images cropped for visualization.

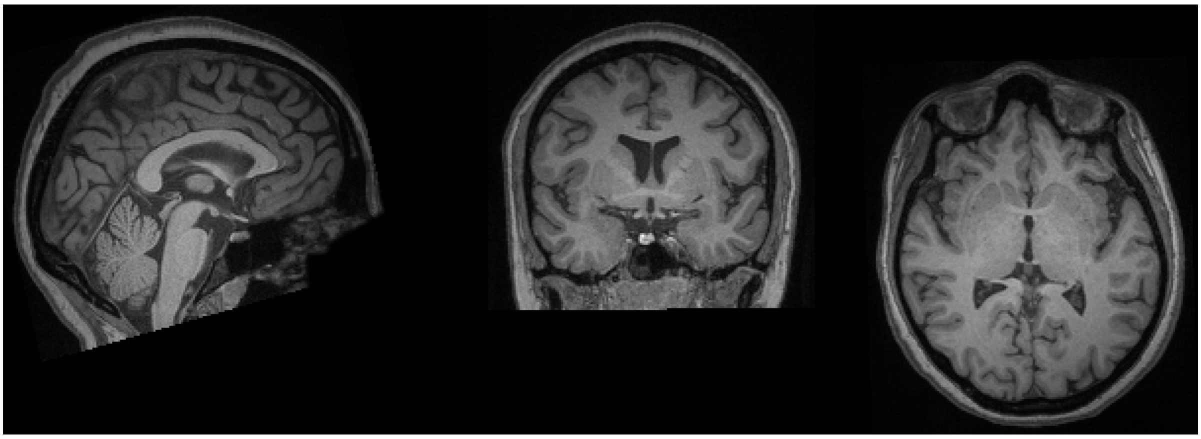


Figure 3.2: Example images of a raw anatomical T1-weighted volume from the COBRE dataset. Images cropped for visualization.

3.2 The ERC2 datasets

The ERC2-I and ERC2-II datasets were acquired as part of a larger project funded by two European Research Council Advanced Grants to Prof. Kenneth Hugdahl at the University of Bergen. The datasets are used in accordance with the Regional Ethics Committee (REK Vest #2016/800). All subjects in the ERC2 datasets were collected using the same GE 750 3T MR scanner at Haukeland University Hospital using an 8-channel head coil, apart from some subjects in the ERC2-I study who did not use this head coil. The subjects in the patient group had been diagnosed using the ICD-10 scale [41]. Similarly to the COBRE dataset, the ERC2 datasets contain information on the age, gender, and handedness of each subject. The datasets also include the score on the Positive and Negative Syndrome Scale (PANSS) for every patient. Before subjects were included for MR imaging in the ERC2-I and ERC2-II studies, all subjects were screened for any history of major head injuries, medical implanted devices, substance abuse, neurological- and medical illnesses.

3.2.1 The ERC2-I dataset

The ERC2-I dataset contains anatomical and functional scans for 70 subjects with mainly SCZ and 111 control subjects. The anatomical T1-weighted images were acquired using an SPGRE-sequence with parameters given in table 3.4 and the resting-state BOLD-fMRI images were acquired using EPI with parameters given in table 3.5.

TR [ms]	TE [ms]	Matrix size	Voxel size [mm^3]	Flip angle [°]	Slices
7700	2.96	256x256	1.0165x1.0156x1	14	188

Table 3.4: The imaging parameters for the anatomical images in the ERC2-I dataset.

TR [ms]	TE [ms]	Matrix size	Voxel size [mm^3]	Flip angle [°]	Slices	Volumes
2000	30	128x128	1.72x1.72x3	90	30	160

Table 3.5: The imaging parameters for the resting-state functional images in the ERC2-I dataset.

The distribution of the diagnoses in the patient group in the ERC2-I dataset can be seen in table 3.6 and the age distribution of the subjects in the ERC2-I dataset is shown on figure 3.3. Example images of a functional volume from the ERC2-I dataset are shown on figure 3.4 with the sagittal view on the left, the coronal view in the middle and the axial view on the right. The images are from the first time point in the time series. The corresponding T1-weighted anatomical images are shown on figure 3.5.

ICD-10 code	Main diagnosis	Frequency
F12*	Mental and behavioural disorders due to use of cannabinoids	2
F15*	Mental and behavioural disorders due to use of other stimulants	5
F19*	Mental and behavioural disorders due to multiple drug use	1
F20.0	Paranoid schizophrenia	28
F20.3	Undifferentiated schizophrenia	9
F20.4	Post-schizophrenic depression	1
F20.9	Schizophrenia, unspecified	1
F21	Schizotypal disorder	1
F22	Persistent delusional disorders	13
F23	Acute and transient psychotic disorders	13
F25	Schizoaffective disorders	4
F28	Other nonorganic psychotic disorders	1
F29	Unspecified nonorganic psychosis	6
F30.2*	Mania with psychotic symptoms	1
F31.7*	Bipolar affective disorder, currently in remission	1
F32.9*	Depressive episode, unspecified	1

*Table 3.6: The different diagnoses present in the ERC2-I dataset, along with their respective diagnosis codes and frequency. * excluded.*

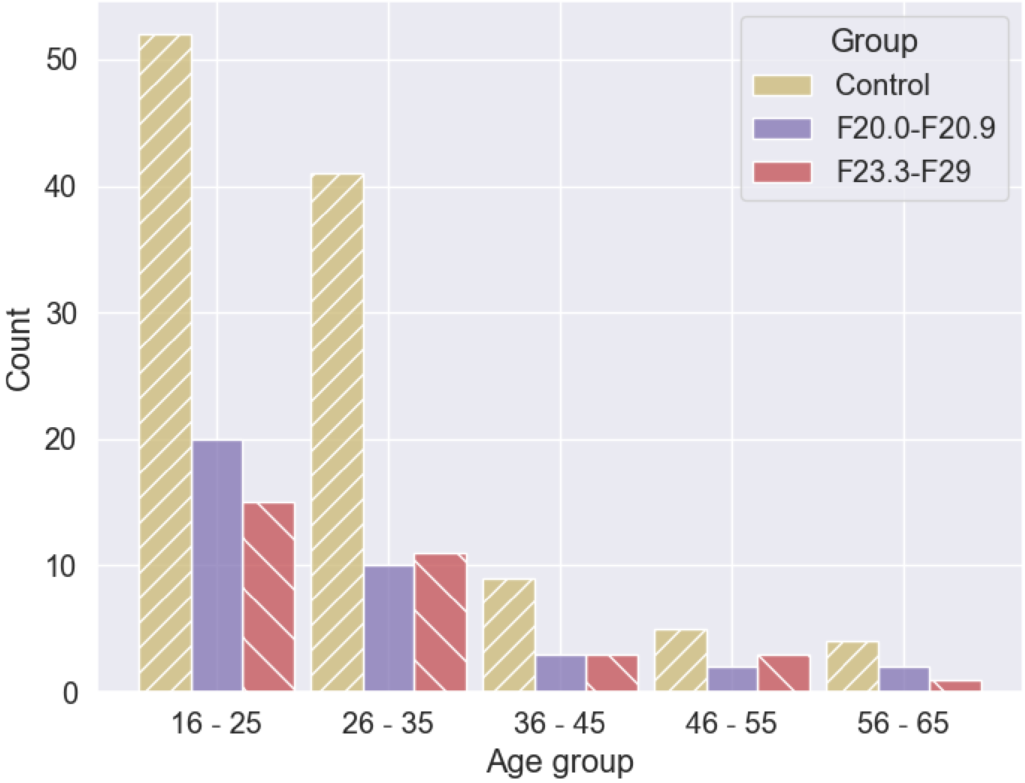


Figure 3.3: The age distribution of the subjects in the ERC2-I dataset.

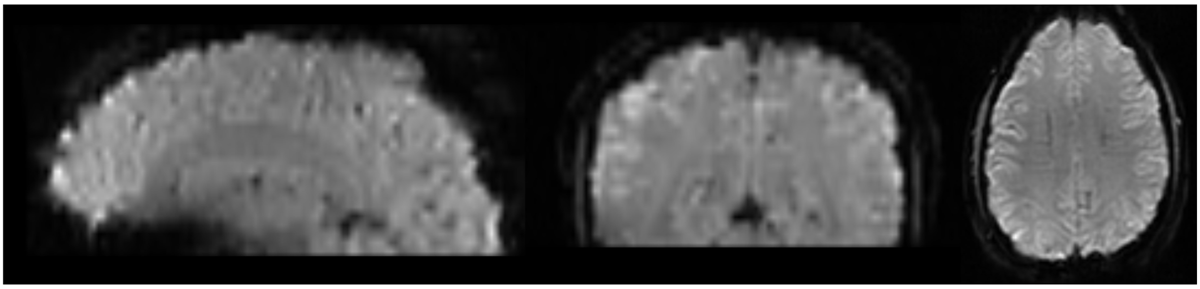


Figure 3.4: Example images of a raw functional volume from the ERC2-I dataset. Images cropped for visualization.

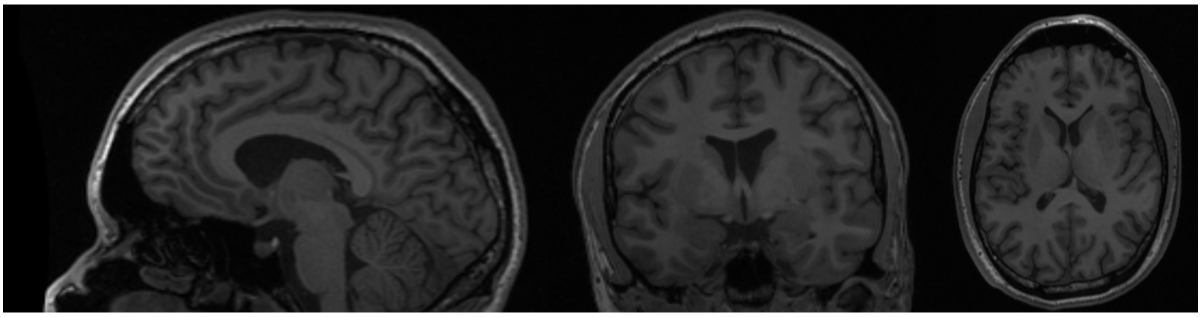


Figure 3.5: Example images of a raw anatomical T1-weighted volume from the ERC2-I dataset. Images cropped for visualization.

3.2.2 The ERC2-II dataset

The ERC2-II dataset contains anatomical and functional scans for 54 subjects with auditory verbal hallucinations and 81 control subjects, where 54 of these control subjects were age and gender matched with the patient group. To be included in the original study, subjects had to experience auditory verbal hallucinations, therefore not all subjects in the patient group had SCZ. The anatomical T1-weighted images were acquired using a FSPGR-sequence with parameters given in table 3.7 and the resting-state BOLD-fMRI images were acquired using EPI with the same parameters as for the ERC2-I dataset, see figure 3.5. The distribution of the diagnoses in the patient group in the ERC2-II dataset can be seen in table 3.8.

TR [ms]	TE [ms]	Matrix size	Voxel size [mm^3]	Flip angle [°]	slices
6800	2.98	256x256	1x1x1	12	188

Table 3.7: The imaging parameters for the anatomical images in the ERC2-II dataset.

ICD-10 code	Main diagnosis	Frequency
F06.0*	Organic hallucinosis	1
F12.5*	Mental and behavioural disorders due to use of cannabinoids	1
F19*	Other psychoactive substance related disorders	4
F20.0	Paranoid schizophrenia	26
F20.3	Undifferentiated schizophrenia	5
F23.3	Other acute predominantly delusional psychotic disorders	1
F25	Schizoaffective disorder	4
F29	Unspecified nonorganic psychosis	3
F31*	Bipolar disorder	1
F32.3*	Severe depressive episode with psychotic symptoms	1
F60*	Personality disorder	2
F61*	Mixed and other personality disorders	1
F62.8*	Other enduring personality changes	1
F90*	Attention-deficit hyperactivity disorders	1

*Table 3.8: The different diagnoses present in the ERC2-II dataset, along with their respective diagnosis codes and frequency. * excluded.*

Example images of a functional volume from the ERC2-II dataset are shown on figure 3.6 with the sagittal view on the left, the coronal view in the middle and the axial view on the right. The images are from the first time point in the time series. The corresponding T1-weighted anatomical images are shown on figure 3.7.

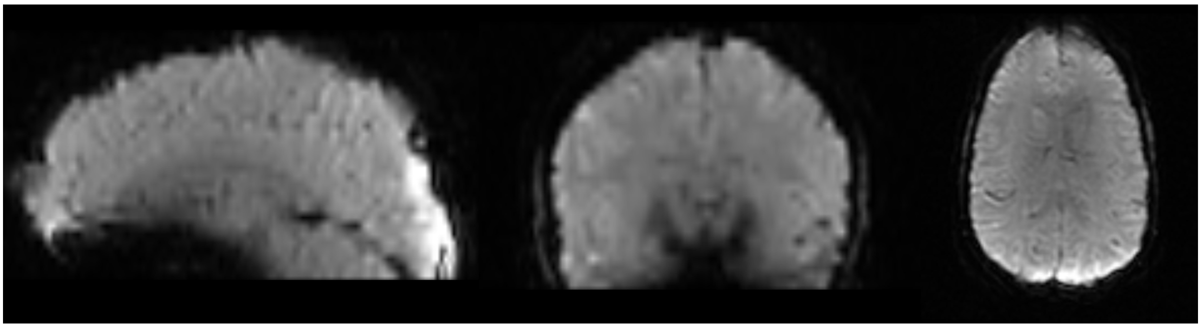


Figure 3.6: Example images of a raw functional volume from the ERC2-II dataset. Images cropped for visualization.

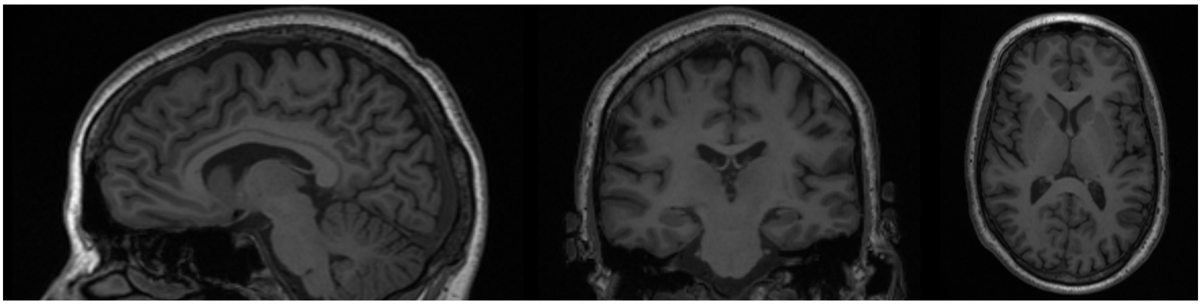


Figure 3.7: Example images of a raw anatomical T1-weighted volume from the ERC2-II dataset. Images cropped for visualization.

3.2.3 ERC2 subject selection process

For the analyses in the current study, a selection process was performed for subjects that would be included for ML. Since the ERC2-II dataset collection focused on subjects with auditory hallucinations, there were several non-schizophrenic patients present in the dataset, e.g. bipolar disorder, major depression and personality disorders. For both the ERC2-I and the ERC2-II datasets, all healthy controls were included in the analysis and patients with the ICD-10 codes F20-F29 as their main diagnosis were included. Subjects with complications due to drugs, such as psychosis due to drug abuse, were excluded from analysis. The selection process was conducted in close cooperation with a psychiatrist to ensure that only subjects with relevant diagnoses for the planned analysis were included.

3.3 Dataset comparison

A comparison between the datasets was performed to investigate dataset variation. The differences in age, gender, and handedness distribution were investigated along with the frequency of the different diagnoses present in the datasets. A statistical t-test was performed to address the age differences between the datasets.

3.4 Preprocessing

Functional data from all subjects were preprocessed using the SPM12 package (statistical parametric mapping, UCL, London, UK, https://www.fil.ion.ucl.ac.uk/spm/) in MATLAB R2022b 2020a (the MathWorks, Inc., Massachusetts, United States). Preprocessing is crucial before doing analysis on fMRI as the raw images can show effects of head movement, physiological cycles and inhomogeneities in the magnetic field [11]. The preprocessing of fMRI volumes usually consists of correction of movement, transformations into standard space, signal normalization and spatial smoothing [55]. Most of these preprocessing steps rely on setting the correct parameters in order for the preprocessing to achieve the wanted results. This is why the standard SPM12 preprocessing pipeline is used in the current study. Smoothing is a preprocessing step where the parameters can differ a lot between studies, and it has both advantages and disadvantages. It can increase the signal-to-noise ratio (SNR) but can also remove important information in the images. It is therefore important to choose appropriate smoothing parameters when preprocessing fMRI images.

3.4.1 Realigning and unwarping

The realignment step translates and rotates all the 3D images in the time series to match the first 3D image of that subject [56]. This is done using a least squares approach [57]. The goal of this process is to realign the images to account for movement. It outputs all the rigid body transformations, namely the translation in the xyz-plane and rotation in the roll-pitch-yaw-plane. It also creates a mean image based on the whole time-series to use in co-registration.

The unwarping step corrects for susceptibility-by-movement interaction, which is the fact that an EPI-image taken in one position is not identical to an image taken in another position, which causes deformations in the image [57]. It uses estimations on how the movement has effected the images to correct for these deformations.

3.4.2 Co-registration

The co-registration and normalization steps account for the differences in subject brain sizes and shapes [56]. These steps ensure that the same voxels between subject volumes are the same voxel in the brain, over the entire time series. This is done by first registering the anatomical volume to a standard MNI template and then performing the same transformations to each functional volume. The standard template is shown in figure 3.8. This procedure is done on every subject, so that every volume is the same shape and size as the template. The co-registration step only does the first part of this, which is moving the anatomical image to a reference image of the functional images. Since all the functional images are realigned, the mean of these images is a good reference.

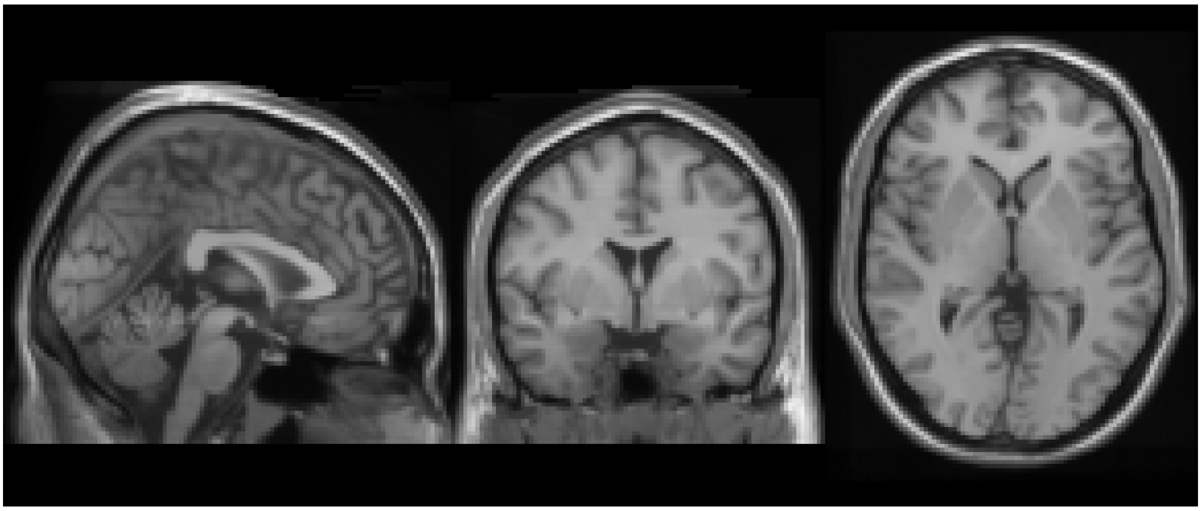


Figure 3.8: The standard T1 MNI-template brain for use in the co-registration step in the preprocessing pipeline. The brain is part of the SPM12 package.

3.4.3 Segmentation

The segmentation step is performed to best match the anatomical image to a template [56]. The templates used for every subject are the default SPM segmentation templates, and are shown in figure 3.9. The outputs from this step are the different segmentations of the anatomical image, as well as the affine transformations done to the anatomical image to match it to the template. Affine transformations are the combination of the rigid body transformations, zooms, and shears.

The option to "Save Bias Corrected" was chosen to aid in automated processing of the images [57]. It removes an artifact that can provide changes in the intensity of the image.

3.4.4 Normalizing

The normalization step applies the transformations done to the anatomical image in the last few steps to all the functional images in the time series [57], see subsection 3.4.2.

3.4.5 Smoothing

The smoothing step convolves the functional volumes with a 3D Gaussian kernel [57] to remove or reduce noise and magnify the fMRI signal [56]. To assess the effect of

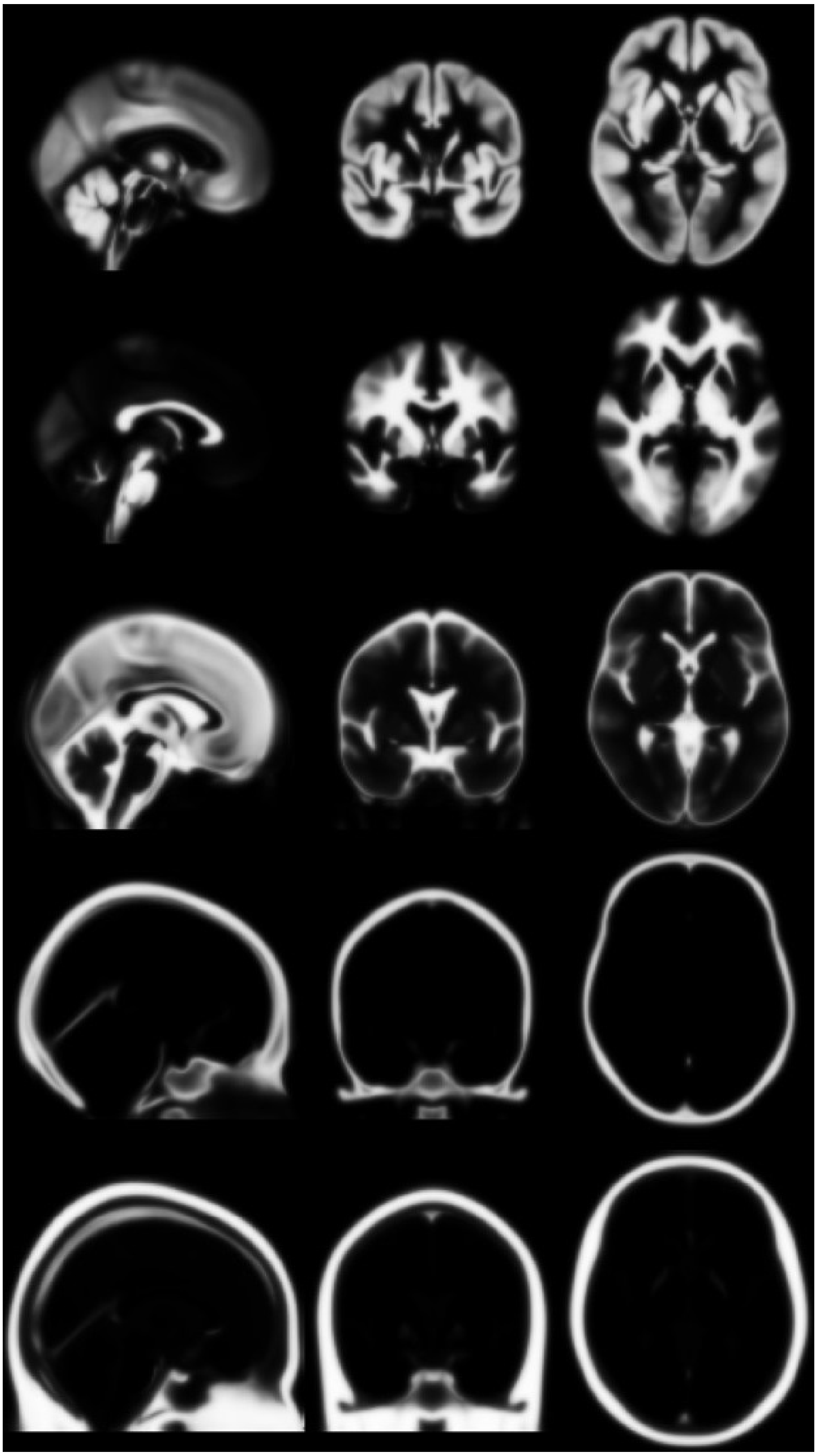


Figure 3.9: The MNI segmentation template used in the preprossessing, where the rows correspond to the gray matter, white matter, cerebrospinal fluid, soft tissue and bone structure of the brain, respectively. Images cropped for visualization.

smoothing, two different kernels of 4 mm and 8 mm were used. Thus, resulting in three versions for each subject: non-smoothed volumes, volumes with a 4 mm smoothing kernel and volumes with an 8 mm smoothing kernel.

3.5 Environment and setup

The ML for this study was conducted using different machines and versions of python and the ML library PyTorch [58].

When using the COBRE dataset only, i.e. for algorithm engineering and tasks A-B, the training and testing was performed on an NVIDIA GeForce RTX 3060 Laptop GPU with a GPU memory of 6 GB and computer memory of 16 GB. Python version 3.11.5 and PyTorch version 2.1.0 were used. However, for hyperparameter tuning, python version 3.9.18 and PyTorch 2.2.0 were used due to hyperparameter tuning library requirements.

When using the ERC2-I and ERC2-II datasets, i.e. for tasks C-E, the training and testing was performed on an Intel Xeon E5-2630 processor unit with 10 cores and 2.2 GHz base frequency and computer memory of 110 GB, through remote desktop. Python version 3.9.2 and PyTorch version 2.2.2 were used. The different versions of programming languages and libraries should not have had any effect on the ML results.

3.6 Algorithm Engineering

A variety of approaches were explored for creating algorithm structures for the study. The first algorithms were created to classify between subjects with SCZ and control subjects, and were trained and validated using the COBRE-dataset.

The first ML algorithm tested was a CNN inspired by the AlexNet algorithm [59], using 3 convolutional layers going up and down in number of channels, each with a ReLU activation function and a max-pooling layer following it. The vector was then flattened and put into two fully connected layers before providing an output with a probability for each of the two classes. The second algorithm was UNET-inspired [60]. It followed the same structure as the first half of the UNET algorithm, see figure 3.10. The proposed algorithm would increase the number of channels between each max pooling layer until the limit of the computer system was met. It would then squeeze the input until it became a one dimensional array which was used as input to a series of fully connected layers, resulting in a prediction. However, this solution required too much memory due to the large input size of the data in the current study.

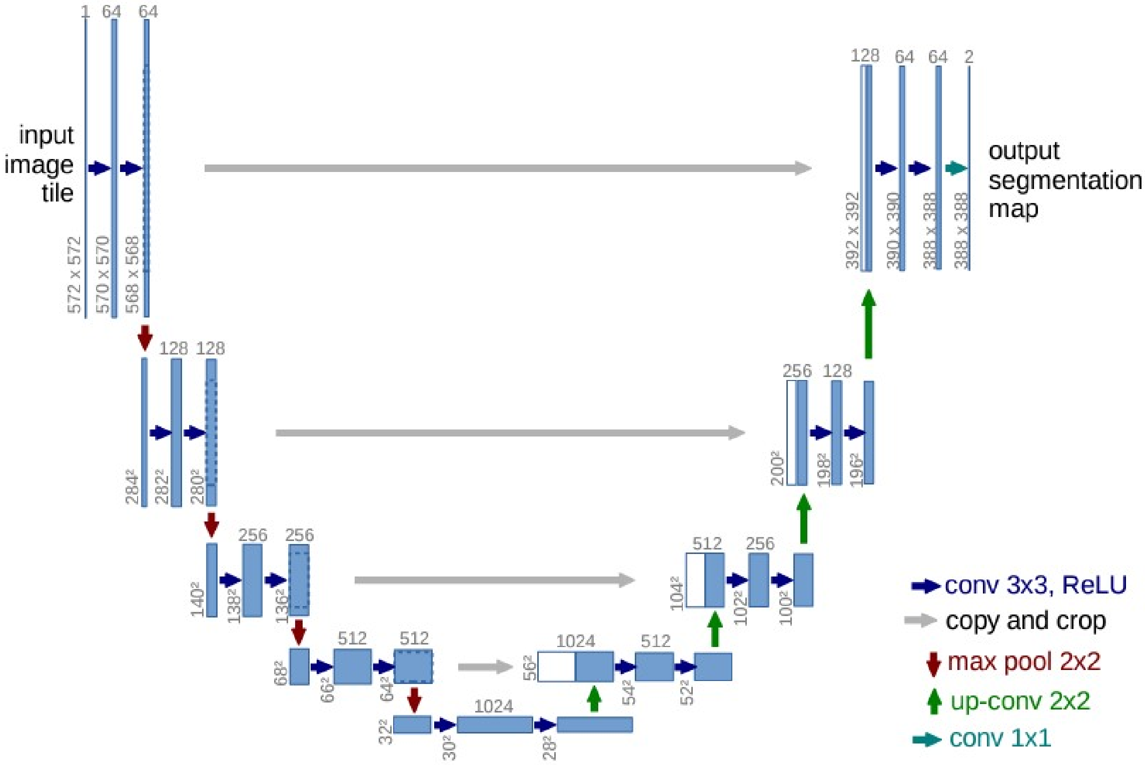


Figure 3.10: The UNET algorithm structure by O. Ronneberger et al. Figure taken from [60].

The final solution was a network that gathered information in the fourth dimension, carried that information into the third dimension, etc. until the inputs had been transformed into a straight line in one dimension. This was then used as input to a series of fully connected layers, which produced the classification probabilities. This way, the GPU memory problem would be resolved by going down in the number of channels for each convolution. A visualization of the algorithm structure can be seen on figure 3.11.

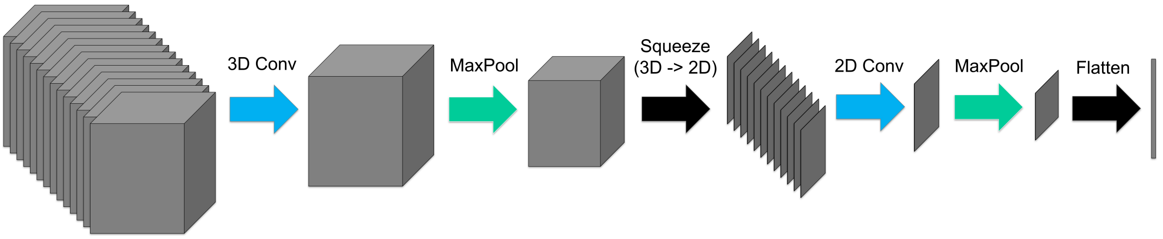


Figure 3.11: The structure of the proposed algorithm, showing the layers of a model created with this algorithm.

The algorithm consisted of three units, each one virtually "removing" a dimension from the input. The first unit had 150 3D images as input, convolved them first into 10 channels in one layer and then into 1 channel in the next layer, resulting in a single 3D volume. Both of these layers had a ReLU activation function following them. In unit 2, this volume was squeezed into slices, with one layer convolving them into a 1 channel

2D image, followed by a ReLU activation function. Both of the first 2 units ends with a dropout layer and a max-pooling layer with kernel size 2^n where n is the current dimension. In the last unit, the single 2D slice was flattened into an array which was used as input to the first fully connected layer with 16 output neurons, followed by a ReLU activation function. The second fully connected layer had output neurons equalling the number of classes. This algorithm structure, along with cross entropy loss as the loss function and Adam [61] as the optimizer was the basis for all algorithm structures used in this study. Adam is a popular, robust and effective optimization algorithm often used in DL [27]. The name is derived from adaptive moment estimation and updates the learning rate individually for each parameter. It also modifies the learning rate based on the direction of the previous parameter update.

Before training the ML models, the dataset was randomly split so that the ratio between the classes was the same, i.e. so that the ratio between subjects with SCZ and healthy controls was the same in the training, validation, and test datasets. In addition, a fixed seed was used to ensure that the same split was used for every new model created so that the models could be compared to each other, i.e. so that the dataset split itself would not benefit one model over the other. The training dataset size was set to 55% of the total dataset size, the validation dataset was set to 25% and the test dataset was set to 20%. The splits were chosen because it was beneficial to include most of the low-represented classes in the training dataset, and have some representatives of each class in the validation and test datasets. This split was used for all analyses throughout the current study, along with the matched ratio between classes within the training, validation and test datasets for both the binary and multilabel analyses.

3.7 Task A: Establishing ML pipeline using online data

This task consisted of preliminary algorithm creation and validation. The objective was to learn about ML practices, algorithm engineering and image learning. The COBRE dataset was used for training and validation. In addition, the first 10 volumes were removed from each subject to correct for movement in the beginning of the recording. This was, however, only performed in this section.

3.7.1 Binary

Binary classification was performed, meaning that a subject was predicted to either be a subject with SCZ or a healthy control. The learning rate was set to 0.001 in agreement with other studies [6, 8, 53]. Dropout rate was set to 0.4 by bootstrapping and batch size was set to 5 due to GPU memory limitations. Number of epochs was set to 100. The three smoothing versions of the functional images were used as input to three different models with the same structure. The analyses were first carried out using the default cross entropy loss function from PyTorch, but were later ran through with weights for each of the classes. These weights were calculated based on the number of representatives in each class present in the training dataset. For the binary classification, all subjects from the COBRE dataset with a SCZ diagnosis were used, i.e. the selection process from section 3.1 was ignored due to the sub-groups not being relevant. This was only the case for the binary classification.

3.7.2 Multilabel

The mutltilabel classification was carried out using the different sub-groups as classes, i.e. the different sub-groups given in table 3.3 apart from the ones that were excluded. The different classes were 295.3 - Paranoid type SCZ, 295.6 - Schizophrenic disorder residual type, 295.7 - Schizoaffective disorder, 295.9 - Unspecified SCZ, along with the healthy controls. Also here both unweighted and weighted cross loss entropy were used as loss functions. All other hyperparameters were the same as for the binary classification.

3.8 Task B: Sub-grouping using the online dataset

Several sub-groups of SCZ diagnoses were represented in the datasets. To have all patients on the same diagnosis scale, the diagnosis of patients in the COBRE dataset was translated into ICD-10. Table 3.9 shows the sub-grouping codes in DSM-IV and ICD-10.

Diagnosis	DSM-IV	ICD-10
Disorganized schizophrenia	295.1	F20.1
Catatonic schizophrenia	295.2	F20.2
Paranoid schizophrenia	295.3	F20.0
Residual schizophrenia	295.6	F20.5
Schizoaffective disorder	295.7	F25.9
Unspecified schizophrenia	295.9	F20.9

Table 3.9: All sub-groups from the COBRE dataset used for classification, showing both the ICD-10 and DSM-IV diagnosis codes.

All analyses in task B were performed on the COBRE dataset only. Two patient groups were defined based on the sub-groups present in the COBRE and ERC2-II datasets. This was done to resolve the problem of few representatives in some sub-groups. The patient groups were created based on cooperation with a clinical psychiatrist. The reason that the sub-groups from the ERC2-II dataset were used to define the two new patient groups was to prepare for analysis on the ERC2-II dataset at a later time. Table 3.10 shows the two new groups of patients from the COBRE dataset, the main SCZ and other SCZ groups, along with the healthy control (HC) group. The main SCZ group consisted of subjects with SCZ with a diagnosis of F20.0-F20.9 and the other SCZ group consisted of subjects on the SCZ spectrum such as schizoaffective disorder F25.9.

New patient group	Inclusions	Number of subjects
HC	Control	72
Main SCZ	F20.0-F20.9	62
Other SCZ	F25.9	7

Table 3.10: The groups to be classified from the COBRE dataset.

In addition to the changed grouping of the subjects, two hyperparameter tuning runs were conducted on non-smoothed images to find the hyperparameters that provided the lowest loss and/or highest accuracy. A weight decay of 0.025 was also added to the Adam optimizer to attempt to increase performance and combat overfitting.

3.8.1 Hyperparameter tuning

Hyperparameter tuning, and thus model selection, was done by using Ray Tune [62]. Tune is a python library for hyperparameter tuning. Using this library, the Tune functions are merged into the existing ML code, and it wraps the train function for Tune to access it. A search space of hyperparameters was defined, as well as an objective function. Two search spaces were used and are shown in tables 3.11 and 3.12. The objective function was to lower the loss function. Tune will loop over a search space by creating all possible combinations of the hyperparameters, and test one model for each combination. For the first run, the hyperparameter combinations ran through 20 epochs before training stopped. This was to first rule out the hyperparameters that did not significantly improve the outcome of the algorithm, before conducting a bigger tuning run, where the hyperparameter combinations went through 200 epochs before training stopped. Hyperparameter tuning was only conducted for the multilabel analysis.

Layer size	5, 10
Learning rate	0.01, 0.001
Dropout rate	0.3, 0.5, 0.7

Table 3.11: The hyperparameter values tuned in the first hyperparameter tuning run.

Layer size	5
Learning rate	0.01, 0.001
Dropout rate	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8

Table 3.12: The hyperparameter values tuned in the second hyperparameter tuning run.

3.9 Task C: Sub-grouping using the ERC2-II dataset

This task consisted of training and classification on the preprocessed non-smoothed functional scans from the ERC2-II dataset. The models trained in this task had the structures of the two best performing models from the hyperparameter tuning in task B. This was to investigate if the best performing hyperparameter combination on one dataset would be reliable on another dataset, collected at a different scanner. The computational graph for the models is shown in figure 3.12.

In the same way as in task B, this classification was conducted using the three classification groups HC, main SCZ and other SCZ. The diagnoses present in each of the groups from the ERC2-II dataset are shown in table 3.13, along with the size of each group. There was also conducted a binary analysis on control vs patient, using both the main SCZ and the other SCZ group as the patient group.

Supergroup/New group	Sub-group	Nr. of subjects
HC	Control	81
Main SCZ	F20.0-F20.9	30
Other SCZ	F23.3-F29	7

Table 3.13: The groups to be classified from the ERC2-II dataset.

3.10 Task D: Data merging

Task D consisted of classification on a combined dataset of the COBRE, ERC2-I and ERC2-II datasets. The combined dataset was split into train, validation and test datasets. The trained model had the same structure as the models used in task B and C, see figure 3.12. There were two objectives in this task. The first was to use a combined dataset to investigate how increasing the data size would affect the performance of the models. The second objective was to check if the hyperparameter combination from task B (using only the COBRE dataset) could be used on images from a combination of datasets that were collected at a different scanners and from different parts of the world.

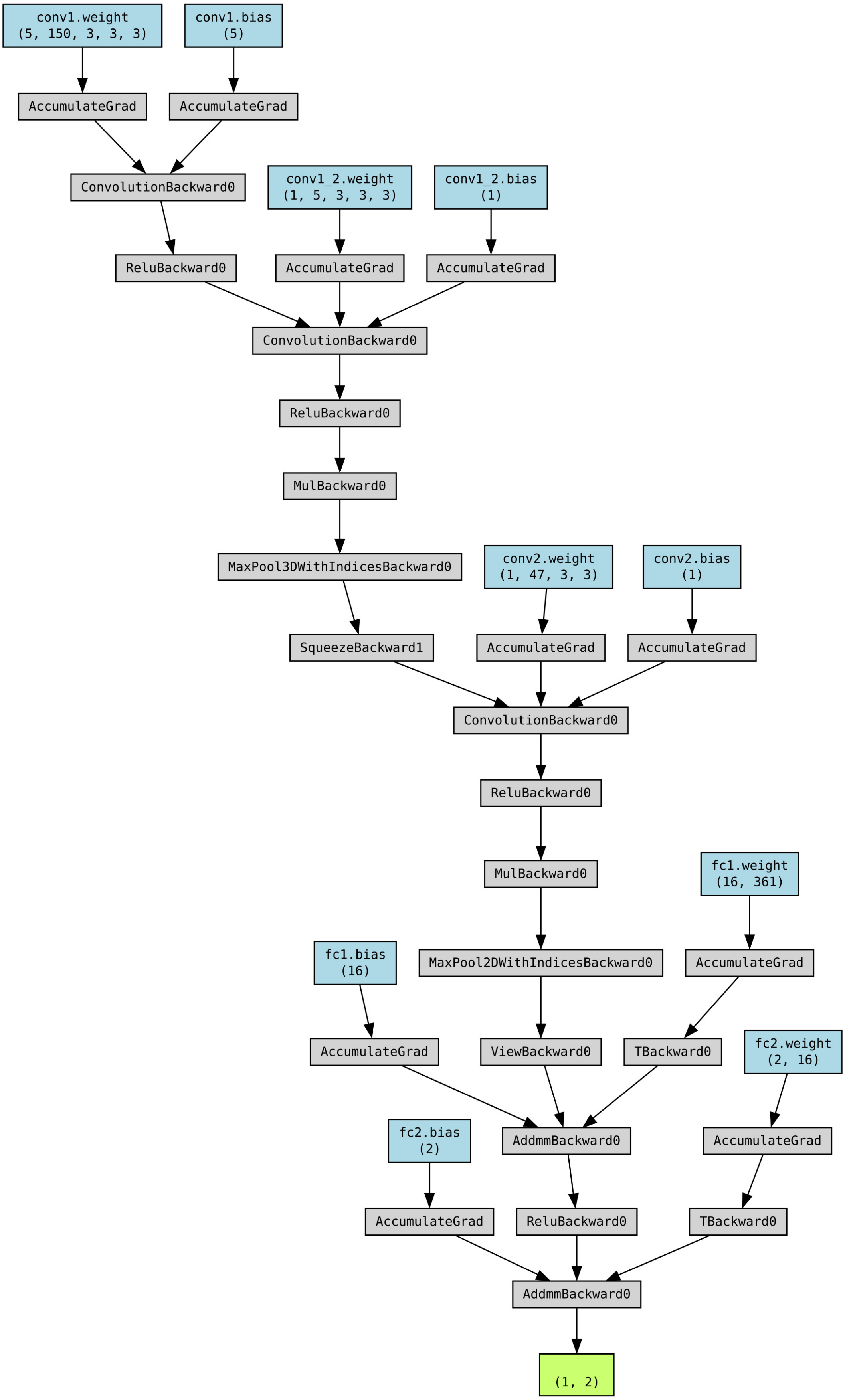


Figure 3.12: The computational graph of the model structure used throughout the current study, after hyperparameter tuning.

3.11 Task E: Classification of unseen test data

To measure the generalization and reliability of the ML models, the final analyses assessed the performance of the models on unseen data, performing both binary and multilabel classification. The test data from the two ERC2 datasets had not yet been seen by any model, so these test datasets were representative of unseen data drawn from the same distribution. The test datasets contained the same ratio between the different subject groups (HC, main SCZ and other SCZ) as the training datasets. For this section, multiple different model structures were used for model selection before deciding on two final models to use on test data, one for binary classification and one for multilabel classification. The models were trained, validated and tested on the ERC2-I and ERC2-II datasets.

Chapter 4

Results

4.1 Dataset comparison

The comparison of age, gender, handedness and sub-group frequency in the COBRE, ERC2-I and ERC2-II datasets can be seen in figures 4.1-4.4. The mean and standard deviation of the ages in the datasets can be seen in table 4.1. A statistical two-sample t-test was performed to address age differences between the datasets, based on section 2.4.7. The results from this can be seen in table 4.2.

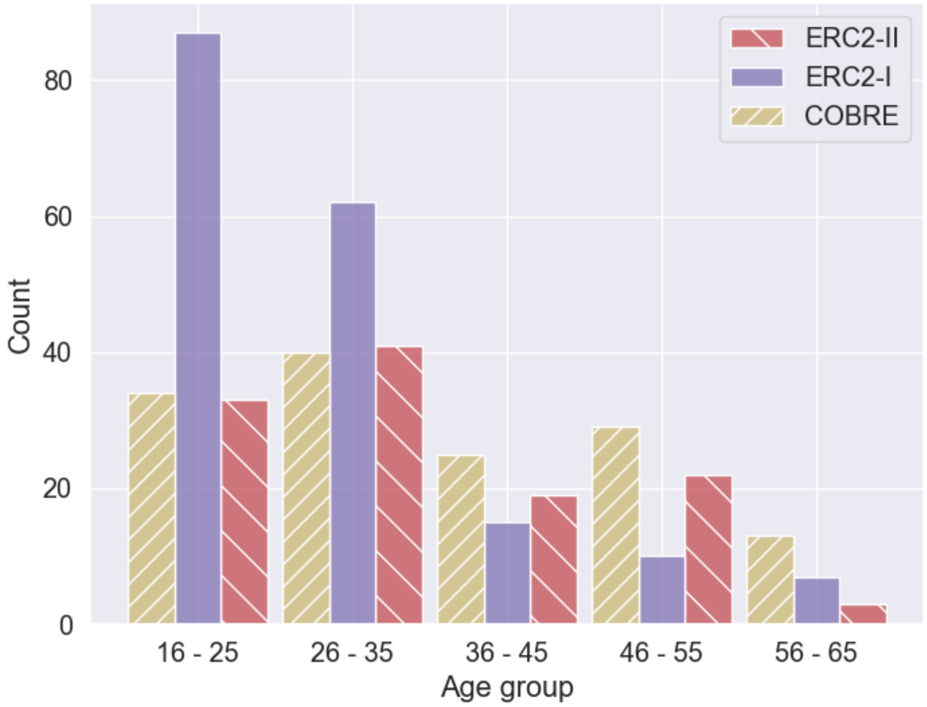


Figure 4.1: The age distributions for the subjects in the COBRE, ERC2-II and ERC2-I datasets.

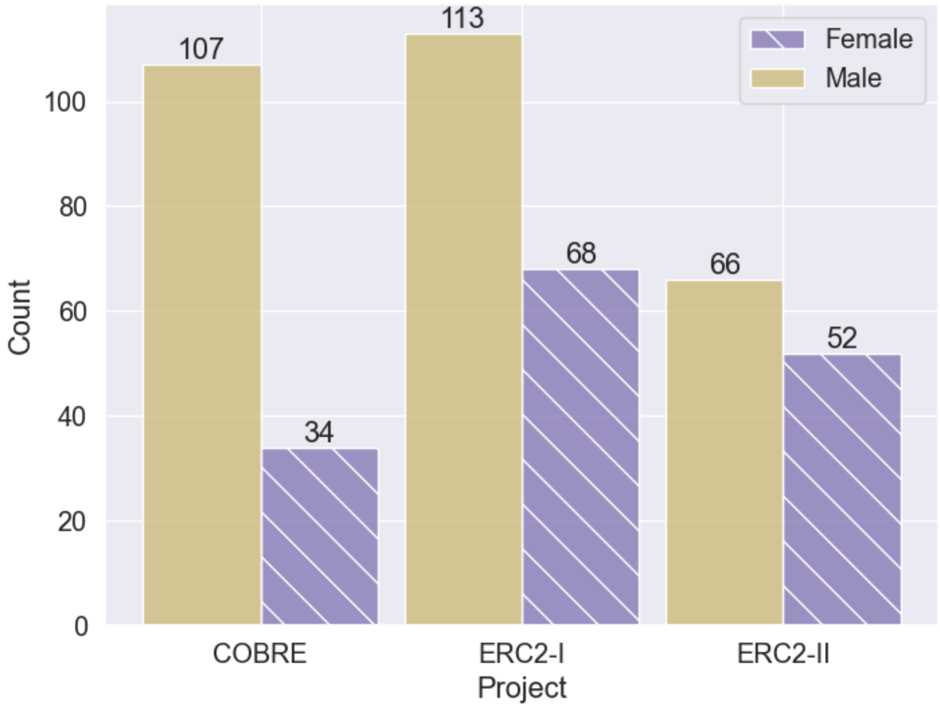


Figure 4.2: The gender distribution for the subjects in the COBRE, ERC2-II and ERC2-I datasets.

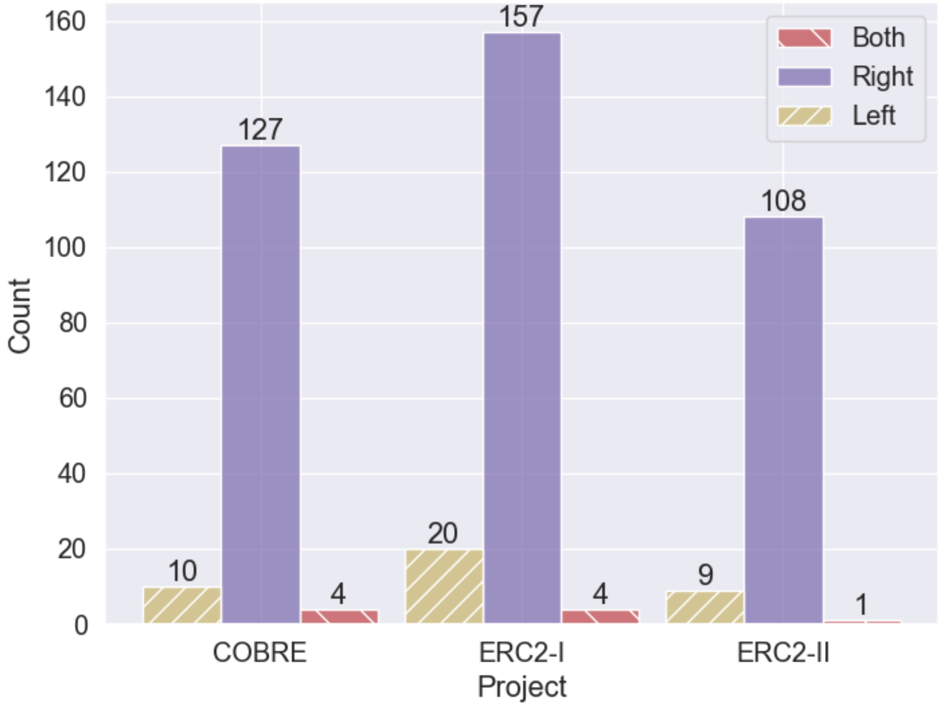


Figure 4.3: The distribution of the handedness of the subjects in the COBRE, ERC2-II and ERC2-I datasets.

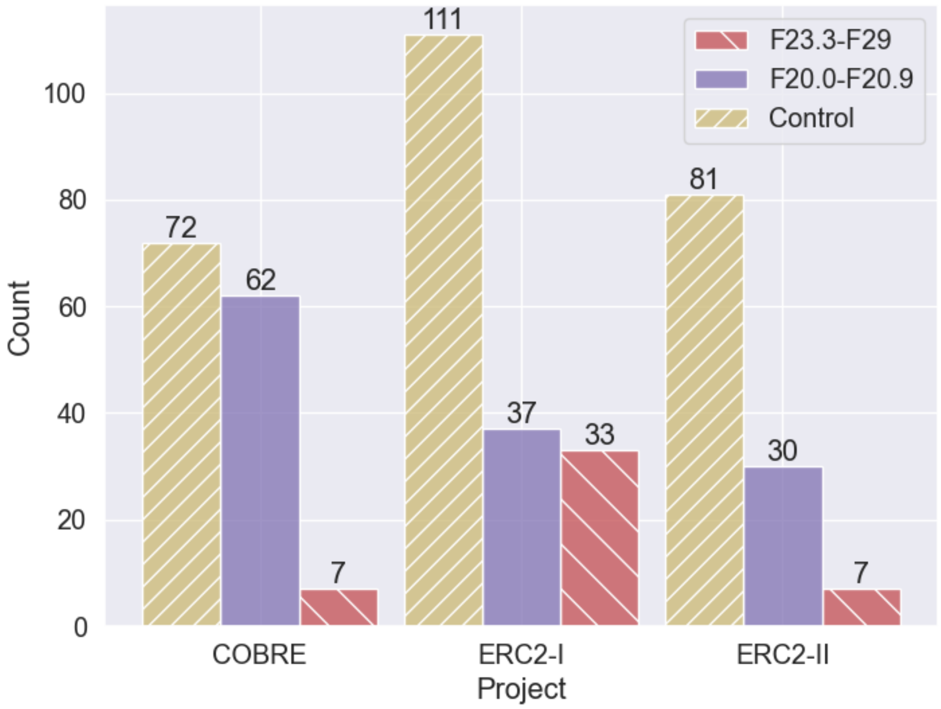


Figure 4.4: The frequency of the different sub-groups in the COBRE, ERC2-I and ERC2-II datasets.

Datasets	COBRE	ERC2-I	ERC2-II
Mean	36.8	29.2	34.1
Standard deviation	12.9	10.1	10.6

Table 4.1: The age mean and standard deviation of the three datasets COBRE, ERC2-I and ERC2-II.

Datasets	COBRE & ERC2-II	COBRE & ERC2-I	ERC2-II & ERC2-I
t-statistic	-1.86	5.74	-3.93
P-value	0.0631	2.62e-08	0.000109
5% significance	No	Yes	Yes

Table 4.2: Results from a two-sample statistical t-test on the ages of the three datasets COBRE, ERC2-I and ERC2-II.

The subjects in the ERC2-I dataset are significantly younger than the subjects in the COBRE and ERC2-II datasets (figure 4.1 and table 4.2). There is also an imbalance in gender distribution between the datasets, with the majority of the subjects in the COBRE and ERC2-I datasets being male. The ERC2-II dataset has an approximate equal split between males and females. The handedness distribution of the subjects between the three datasets are quite similar. The COBRE and ERC2-II datasets have an equal amount of subjects in the other SCZ group, while the ERC2-I dataset has almost 5 times as many subjects in this group. The ERC2 datasets have approximately the same amount of subjects in the main SCZ group, while the COBRE dataset has close to twice as many subjects in this group. In every dataset, the HC group is bigger than the patient groups.

4.2 Preprocessing

The results of the preprocessing on the same slice of the example functional COBRE images (figure 3.1) are shown in figure 4.5. A non-smoothed functional image of a random example subject from each of the COBRE, ERC2-I and ERC2-II datasets are shown in figure 4.6. It took around 10 minutes to fully preprocess one subject.

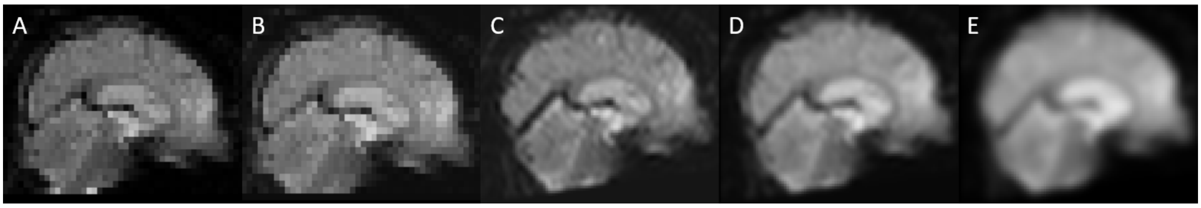


Figure 4.5: Results of the different preprocessing steps, shown on the same slice. A) Raw image. B) After realignment and unwarping. C) After normalization. D) After smoothing with 4 mm kernel. E) After smoothing with 8 mm kernel. Images cropped for visualization. fMRI image from the COBRE dataset.

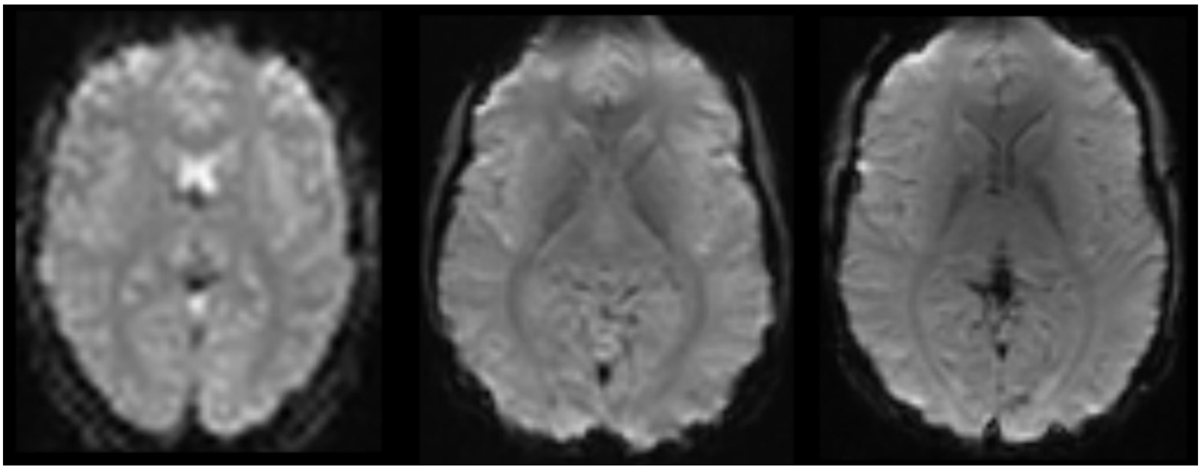


Figure 4.6: An example non-smoothed functional image from each of the three datasets COBRE, ERC2-I and ERC2-II. Images cropped for visualization.

4.2.1 Realigning and unwarping

The result from the realigning and unwarping steps from subsection 3.4.1 on the example image from the COBRE dataset (figure 3.1) is shown on figure 4.7.

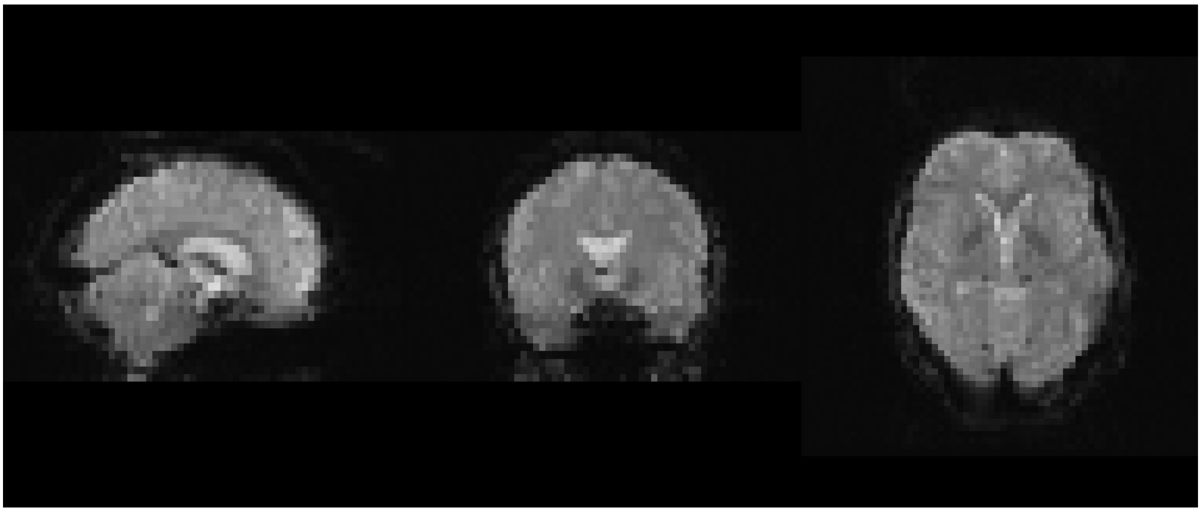


Figure 4.7: The example functional image from the COBRE dataset after realignment and unwarping. Images cropped for visualization.

4.2.2 Co-registration

The result from the co-registration step from subsection 3.4.2 on the example image from the COBRE dataset (figure 3.2) is shown on figure 4.8.

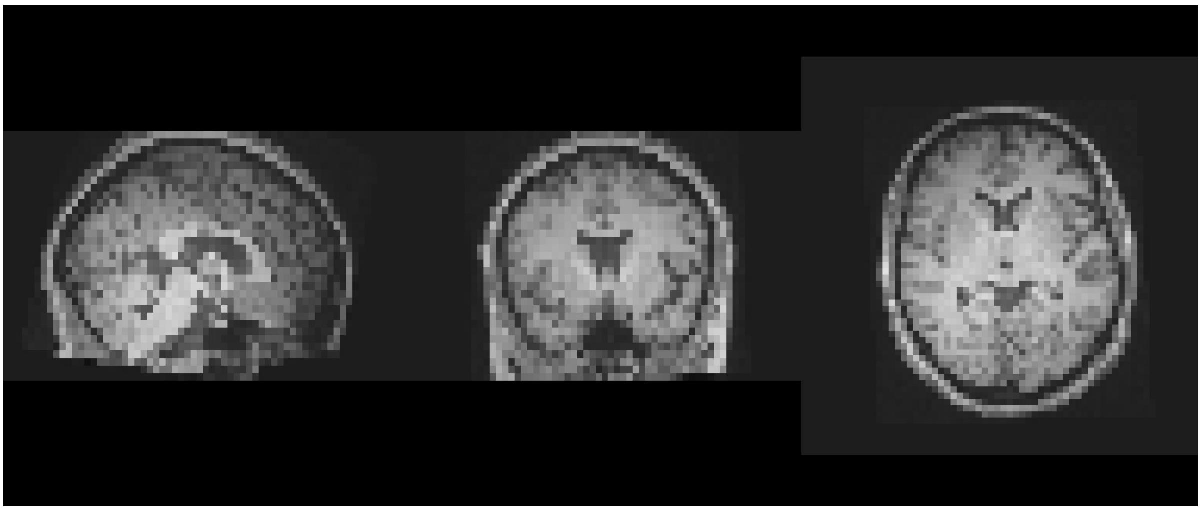


Figure 4.8: The example anatomical image from the COBRE dataset after co-registration. Images cropped for visualization.

4.2.3 Segmentation

The results from the segmentation step from subsection 3.4.3 on the example anatomical COBRE image (figure 3.2) are shown on figure 4.9.

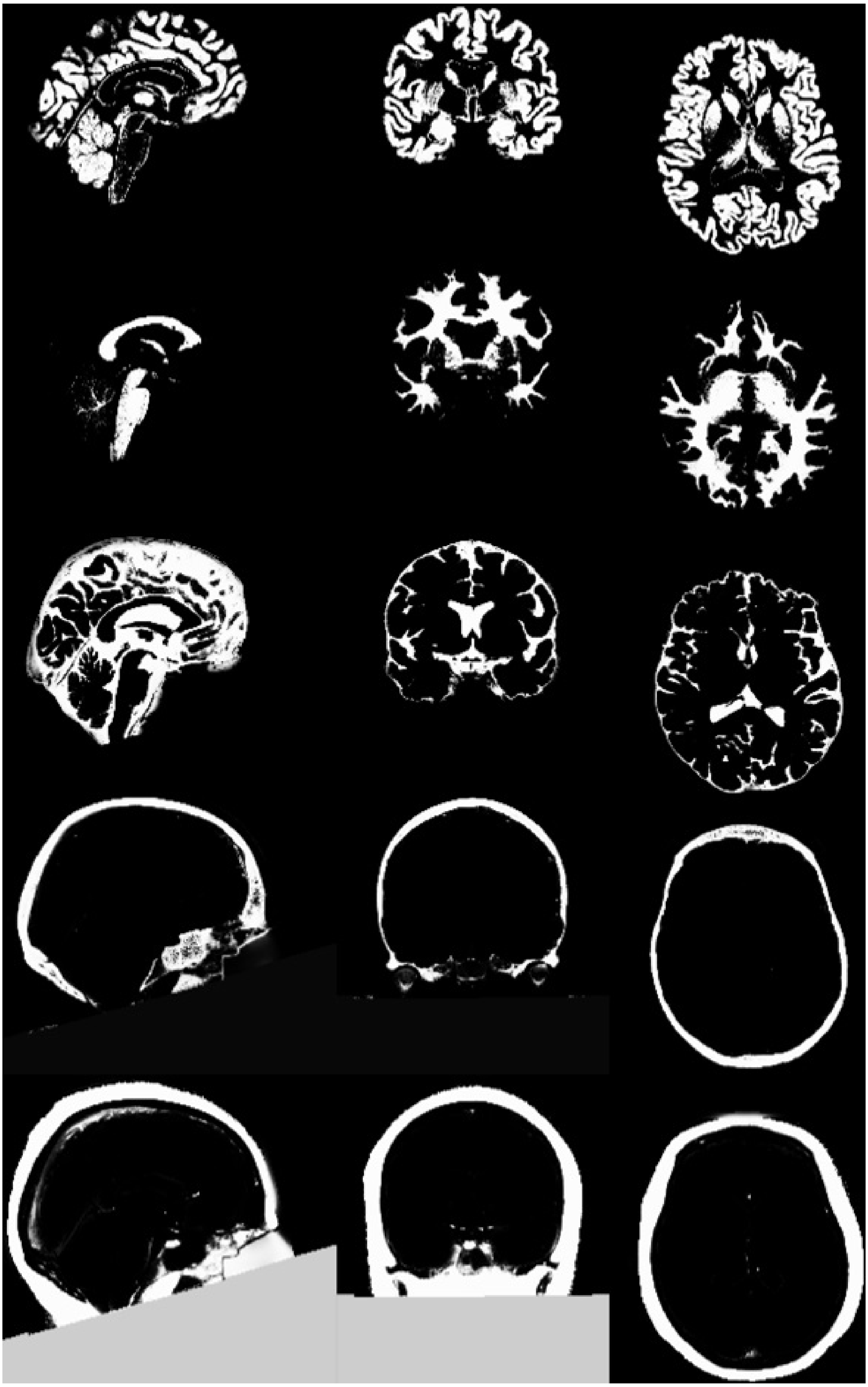


Figure 4.9: The segmentation after the segmentation step on the example anatomical COBRE image, where the rows correspond to the gray matter, white matter, cerebrospinal fluid, soft tissue and bone structure of the brain, respectively. Images cropped for visualization.

4.2.4 Normalizing

The result from the normalizing step from subsection 3.4.4 on the example functional COBRE image (figure 3.1) is shown on figure 4.10.

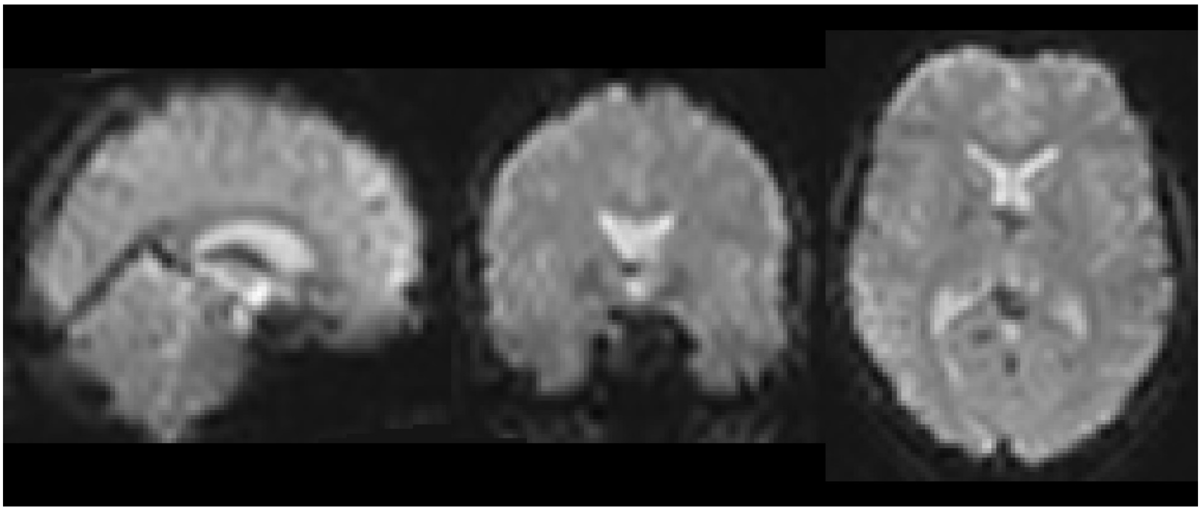


Figure 4.10: The example functional COBRE image after the normalization step of the preprocessing pipeline. Images cropped for visualization.

4.2.5 Smoothing

The results from the smoothing step from subsection 3.4.5 on the example functional COBRE image (figure 3.1) are shown on figures 4.11 and 4.12 for the 4 mm kernel and 8 mm kernel, respectively.

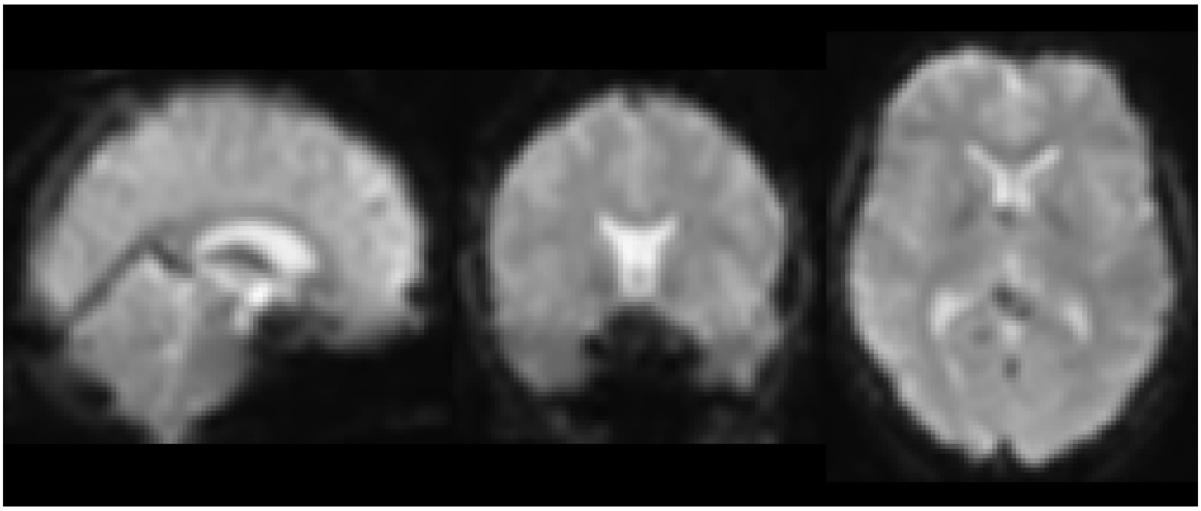


Figure 4.11: The example functional COBRE image after the smoothing step of the preprocessing pipeline, smoothed with a 4 mm kernel. Images cropped for visualization.

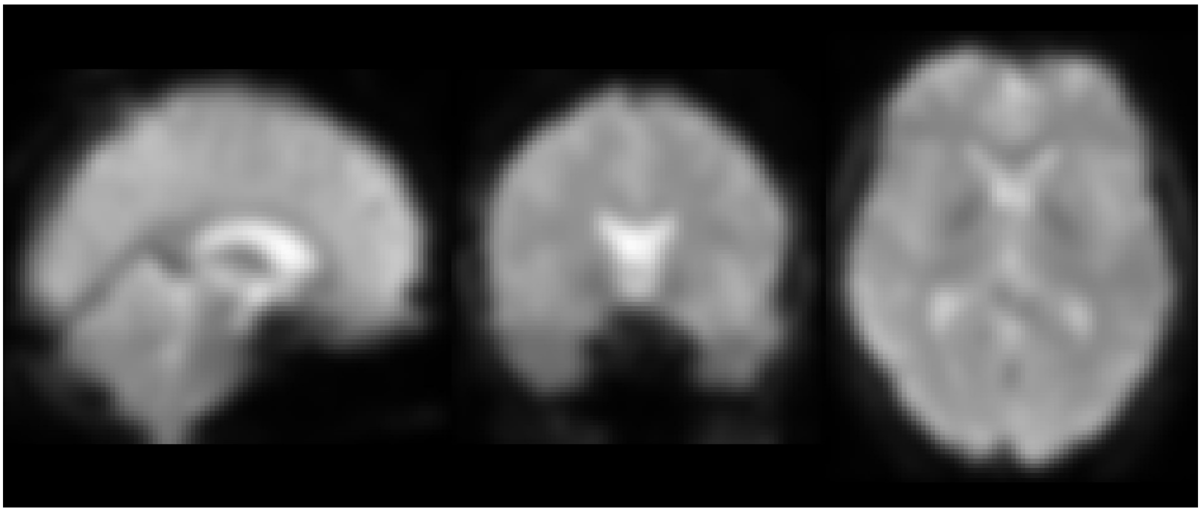


Figure 4.12: The example functional COBRE image after the smoothing step of the preprocessing pipeline, smoothed with a 8 mm kernel. Images cropped for visualization.

4.3 Task A: Establishing ML pipeline using online data

4.3.1 Binary classification on COBRE

The results from the binary classification on the COBRE dataset, i.e. classification of control or patient, are shown in tables 4.3 and 4.4 for the non-smoothed, 4 mm kernel smoothed, and 8 mm kernel smoothed images for unweighted and weighted cross entropy loss functions, respectively. The corresponding losses during training with weighted cross entropy loss are given in figures 4.13-4.15 for the non-smoothed, 4 mm kernel smoothed, and 8 mm kernel smoothed volumes, respectively. The accuracies during training are given in figures 4.16-4.18, respectively. Every model was trained for 100 epochs and the training time was 137±4 minutes.

As seen in table 4.3, the three models trained on the different volumes performed similarly when using an unweighted loss function. The model trained on the volumes smoothed with an 8 mm kernel achieved slightly higher balanced accuracy than the other models, but had the lowest F1-score. However, the models trained with a weighted loss function performed better, see table 4.4. The model trained on the non-smoothed images performed the best, with an accuracy of 0.8 and a balanced accuracy of 0.788.

The plots for the loss during training for the different models using a weighted loss function, figures 4.13, 4.14 and 4.15, show that the validation loss was increasing before training stopped. The plots for the accuracy during training, figures 4.16, 4.17 and 4.18, show that the validation accuracy during training had reached its maximum and that the models were overfitting at the end of training.

	Accuracy	Balanced accuracy	Recall	Precision	F1-score
Non-smoothed	0.579	0.589	0.400	0.667	0.565
4 mm smoothed	0.579	0.583	0.500	0.625	0.577
8 mm smoothed	0.579	0.594	0.300	0.750	0.541

Table 4.3: The binary classification validation results on the COBRE scans using the unweighted cross entropy loss.

	Accuracy	Balanced accuracy	Recall	Precision	F1-score
Non-smoothed	0.800	0.788	0.909	0.769	0.796
4 mm smoothed	0.650	0.662	0.545	0.750	0.647
8 mm smoothed	0.550	0.591	0.182	1.000	0.469

Table 4.4: The binary classification validation results on the COBRE scans using the weighted cross entropy loss.

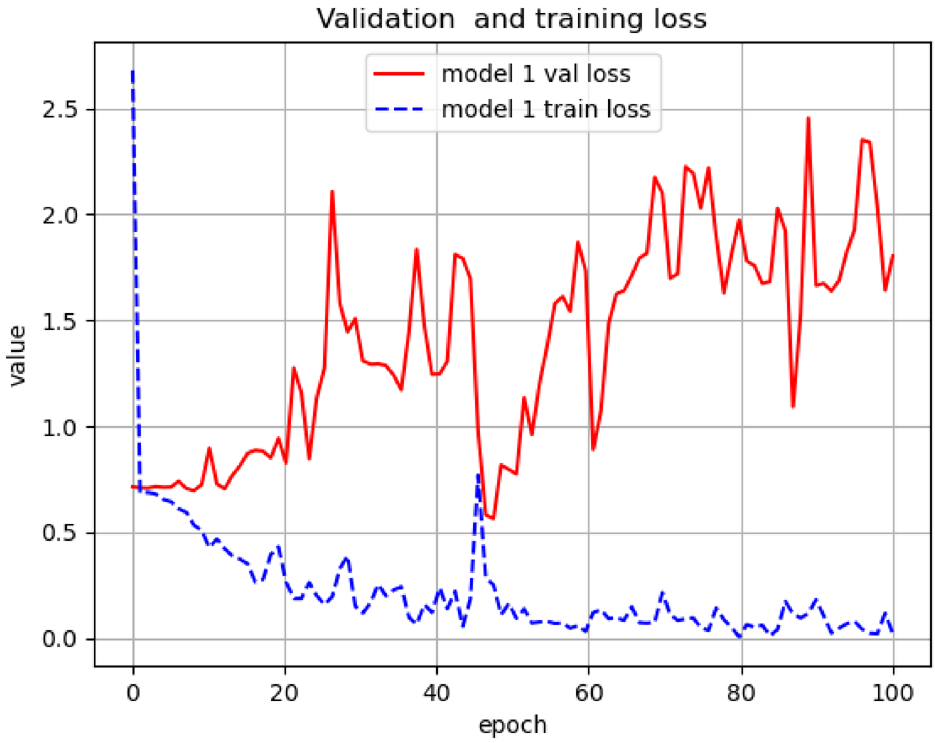


Figure 4.13: The training and validation loss on the binary classification on the non-smoothed images from the COBRE dataset with weighted cross entropy loss.

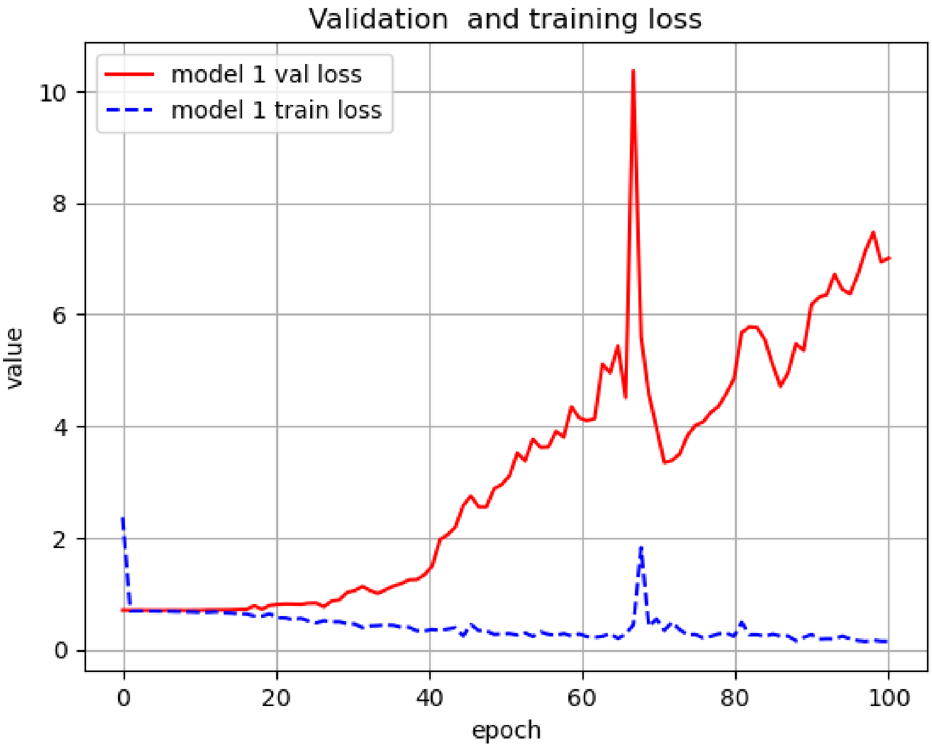


Figure 4.14: The training and validation loss on the binary classification on the 4 mm kernel smoothed images from the COBRE dataset with weighted cross entropy loss.

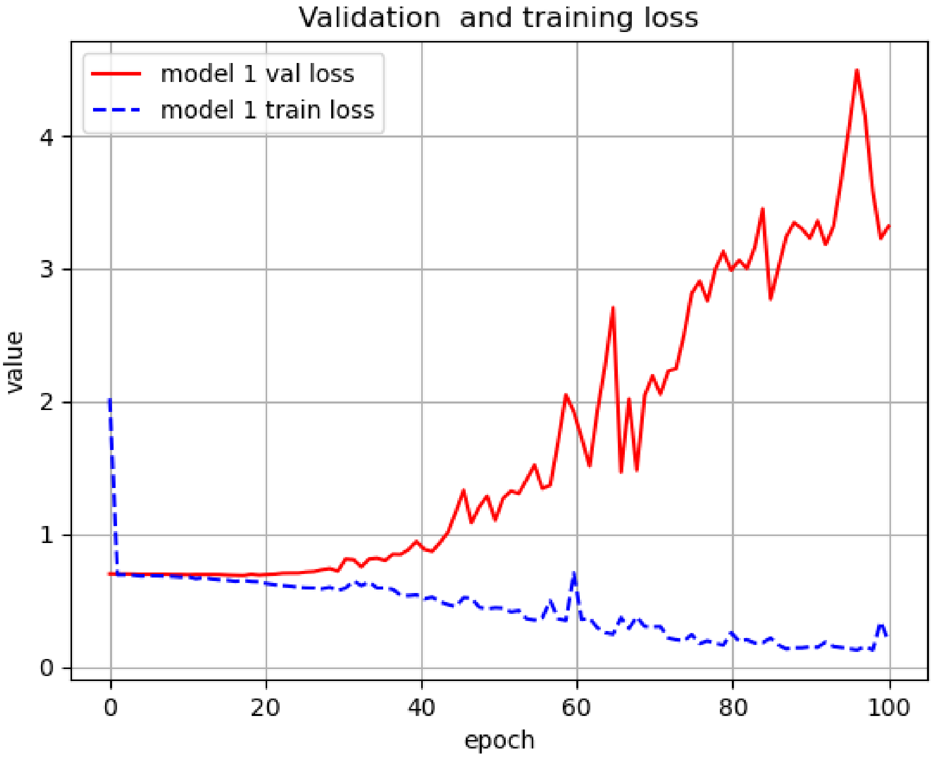


Figure 4.15: The training and validation loss on the binary classification on the 8 mm kernel smoothed images from the COBRE dataset with weighted cross entropy loss.

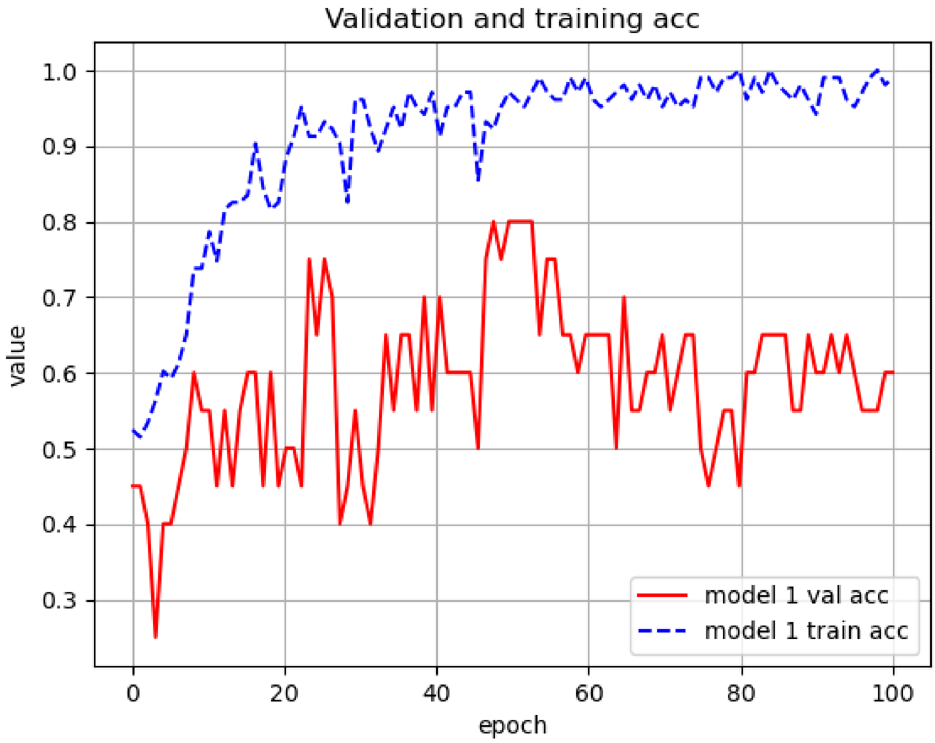


Figure 4.16: The training and validation accuracies on the binary classification on the non-smoothed images from the COBRE dataset with weighted cross entropy loss.

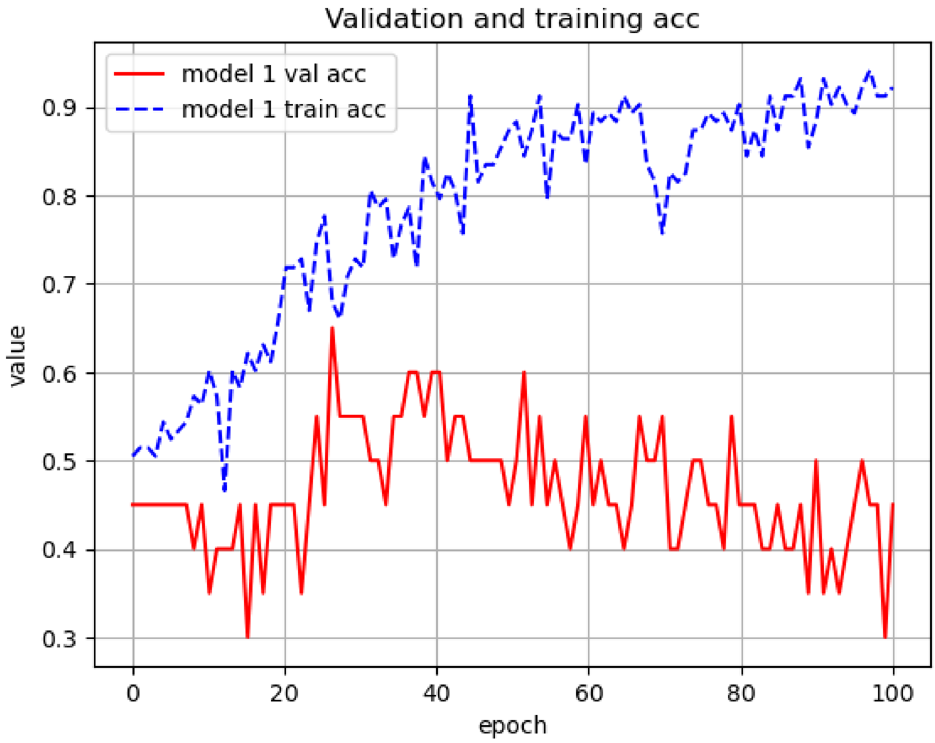


Figure 4.17: The training and validation accuracies on the binary classification on the 4 mm kernel smoothed images from the COBRE dataset with weighted cross entropy loss.

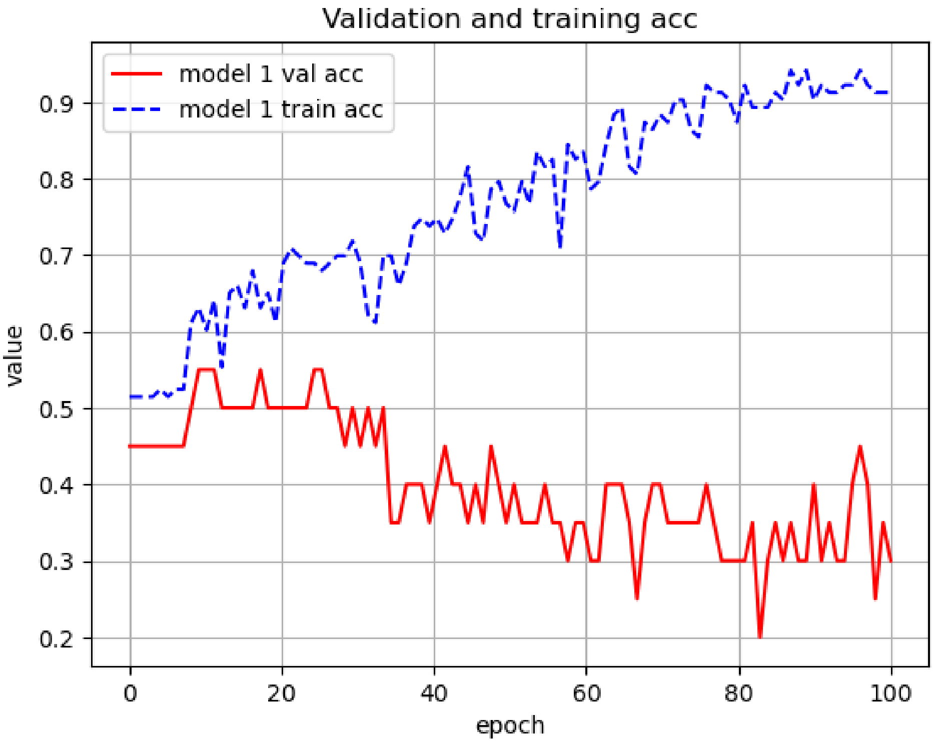


Figure 4.18: The training and validation accuracies on the binary classification on the 8 mm kernel smoothed images from the COBRE dataset with weighted cross entropy loss.

4.3.2 Multilabel classification on COBRE

The results of the multilabel classification on the COBRE dataset are shown in tables 4.5 and 4.6 for the non-smoothed, 4 mm kernel smoothed and 8 mm kernel smoothed images for both the unweighted and weighted cross entropy loss functions, respectively. Every model was trained for 100 epochs and the training time was 129±2 minutes.

Tables 4.5 and 4.6 show that the models trained on the non-smoothed images had the highest score on every metric in the multilabel case with and without weights in the loss function.

	Accuracy	Balanced accuracy	Recall	Precision	F1-score
Non-smoothed	0.579	0.278	0.579	0.460	0.502
4 mm smoothed	0.474	0.200	0.474	0.224	0.305
8 mm smoothed	0.474	0.200	0.474	0.224	0.305

Table 4.5: The multilabel classification validation results on the COBRE scans using the unweighted cross entropy loss.

	Accuracy	Balanced accuracy	Recall	Precision	F1-score
Non-smoothed	0.737	0.456	0.737	0.674	0.690
4 mm smoothed	0.526	0.267	0.526	0.416	0.465
8 mm smoothed	0.526	0.233	0.526	0.553	0.406

Table 4.6: The binary classification validation results on the COBRE scans using the weighted cross entropy loss.

4.4 Task B: Sub-grouping using the online dataset

The results from the first hyperparameter tuning run, as described in section 3.8.1, are shown in table 4.7. The model with a layer size of 5, a learning rate of 0.001 and a dropout rate of 0.3 achieved the highest accuracy, while the model with the same layer size, a learning rate of 0.01 and a dropout rate of 0.5 achieved the lowest loss, after 20 epochs. The two best performing hyperparameter combinations from the second, bigger hyperparameter tuning run are shown in table 4.8, where the model with a dropout rate of 0.2 achieved the highest accuracy and the model with a dropout rate of 0.4 achieved the lowest loss after 200 epochs. The total time for the hyperparameter tuning was 17 hours and 44 minutes for the first run and 2 days, 12 hours and 19 minutes for the second run.

Layer size	Learning rate	Dropout rate	Best val accuracy	Best val loss
5	0.01	0.3	0.514	0.921
5	0.001	0.3	**0.600**	1.294
5	0.01	0.5	0.514	**0.916**
5	0.001	0.5	0.514	1.057
5	0.01	0.7	0.486	0.930
5	0.001	0.7	0.514	1.000
10	0.01	0.3	0.514	0.982
10	0.001	0.3	0.514	1.028
10	0.01	0.5	0.514	0.937
10	0.001	0.5	0.514	1.010
10	0.01	0.7	0.429	0.959
10	0.001	0.7	0.429	1.025

Table 4.7: The results from the first hyperparameter tuning run.

Dropout	Best model val accuracy	Best model val loss
0.2	**0.657**	0.961
0.4	0.571	**0.808**

Table 4.8: The results from the second hyperparameter tuning. Both models had layer size of 5 and a learning rate of 0.01.

The results from the multilabel analysis on the COBRE-dataset using the two best model structures from the hyperparameter tuning are shown in table 4.9. Note here that the model with the highest balanced accuracy during training was chosen. The corresponding balanced accuracy, accuracy, and loss during training are shown in figures 4.19-4.21 for the model with a dropout rate of 0.2 and 4.22-4.24 for the model with a dropout rate of 0.4, respectively. The best performing model was chosen right before epoch nr. 30 for the model with a dropout rate of 0.2 and right after epoch nr. 15 for the model with a dropout rate of 0.4. The loss plateaued around epoch nr. 40 for both models.

Dropout	Accuracy	Balanced accuracy	Recall	Precision	F1-score
0.2	0.600	0.574	0.600	0.659	0.616
0.4	0.571	0.533	0.571	0.600	0.535

Table 4.9: The classification results on validation data using the two best models from the hyperparameter tuning.

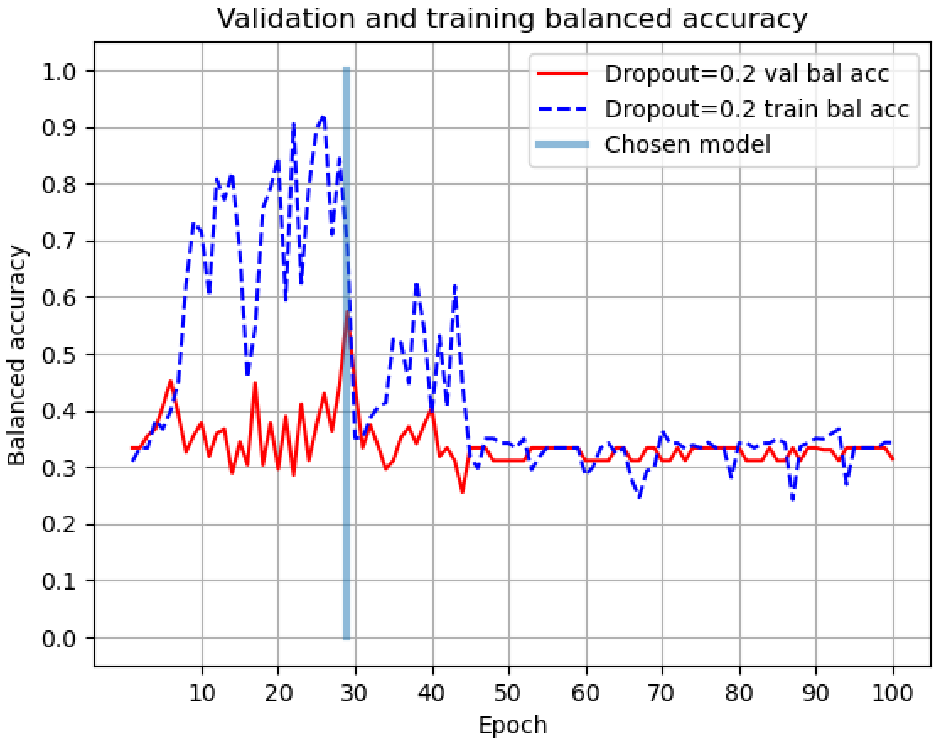


Figure 4.19: The balanced accuracy score on validation data during training for the model with a dropout rate of 0.2. The chosen model from the training is shown in a blue, vertical line.

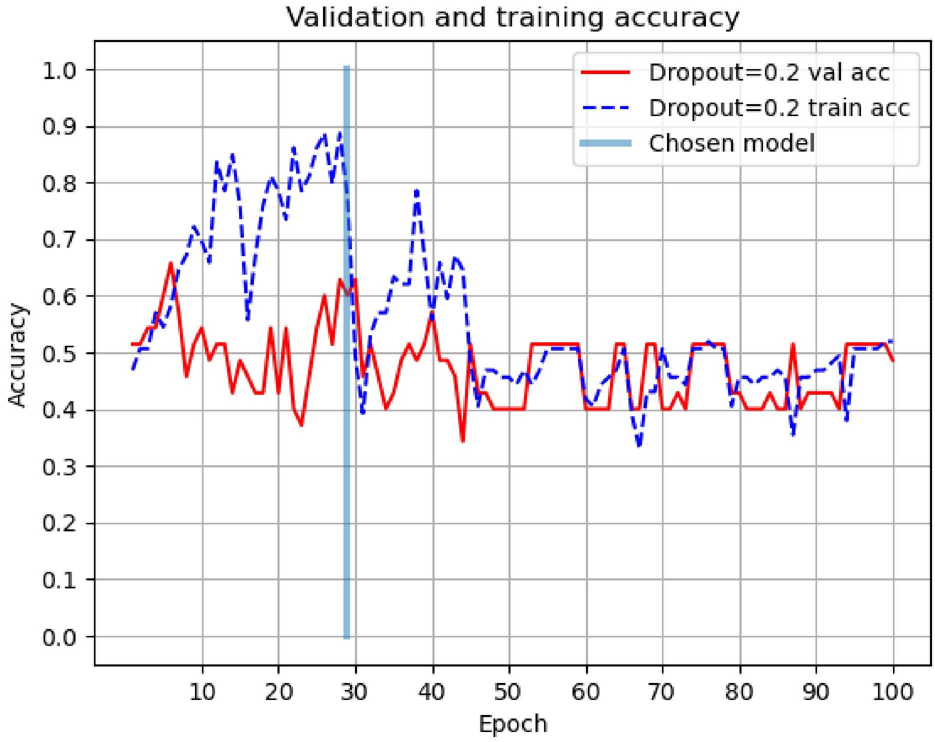


Figure 4.20: The accuracy score on validation data during training for the model with a dropout rate of 0.2. The chosen model from the training is shown in a blue, vertical line.

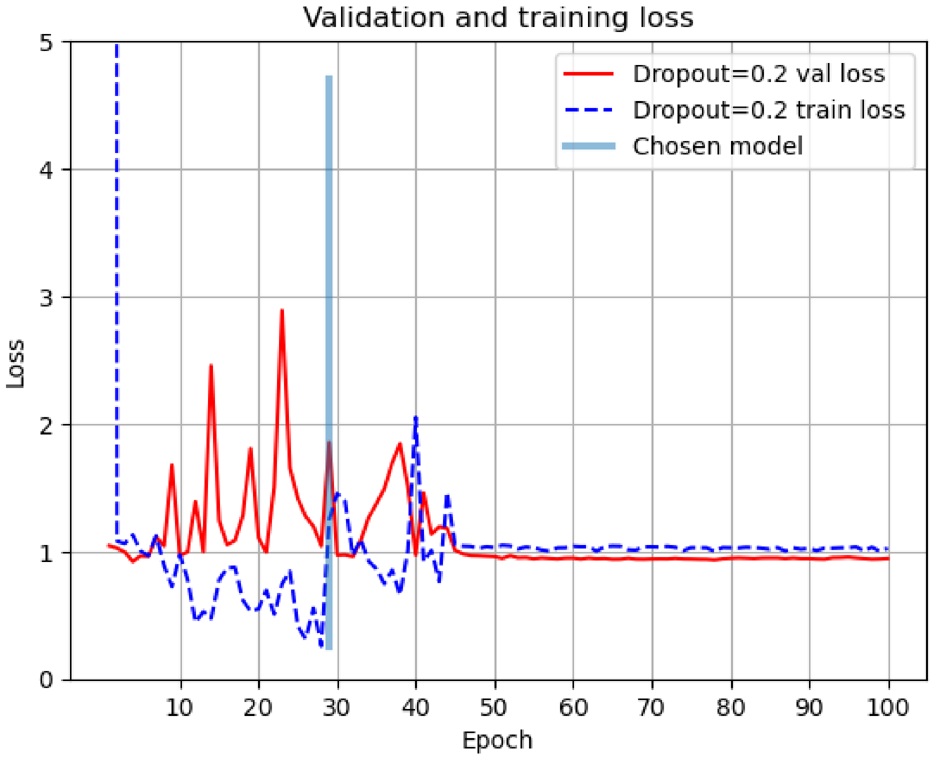


Figure 4.21: The loss on validation data during training for the model with a dropout rate of 0.2. The chosen model from the training is shown in a blue, vertical line.

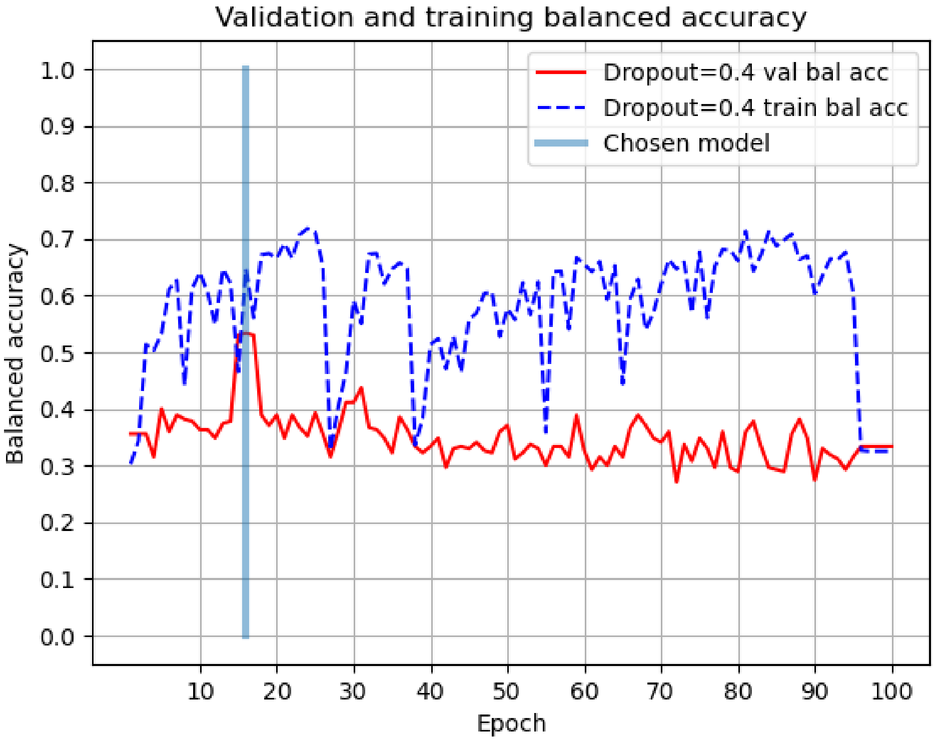


Figure 4.22: The balanced accuracy score on validation data during training for the model with a dropout rate of 0.4. The chosen model from the training is shown in a blue, vertical line.

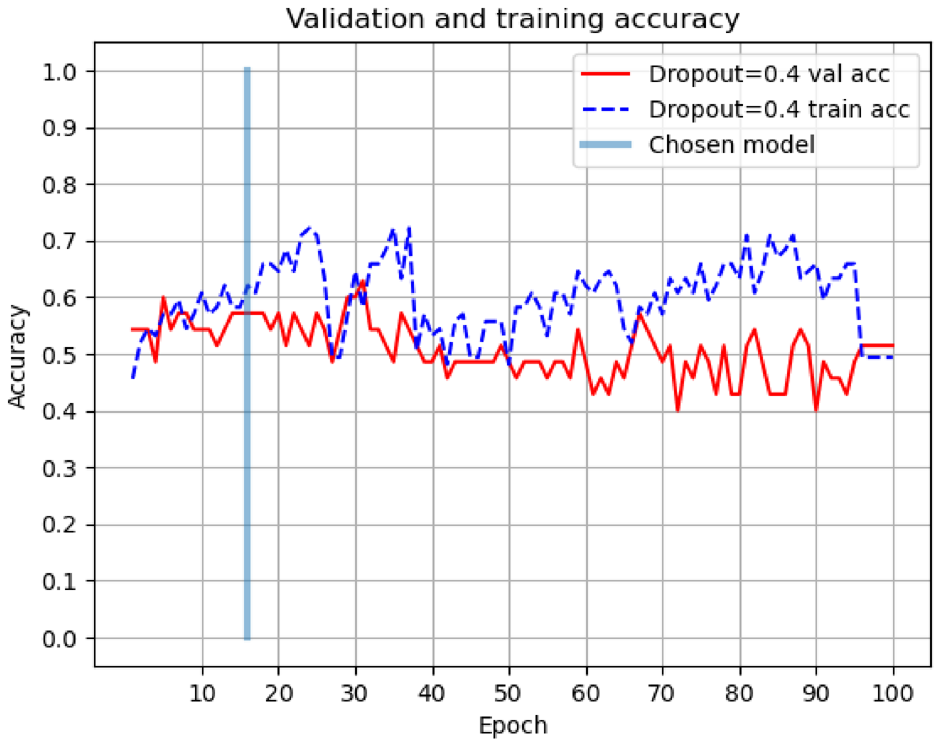


Figure 4.23: The accuracy score on validation data during training for the model with a dropout rate of 0.4. The chosen model from the training is shown in a blue, vertical line.

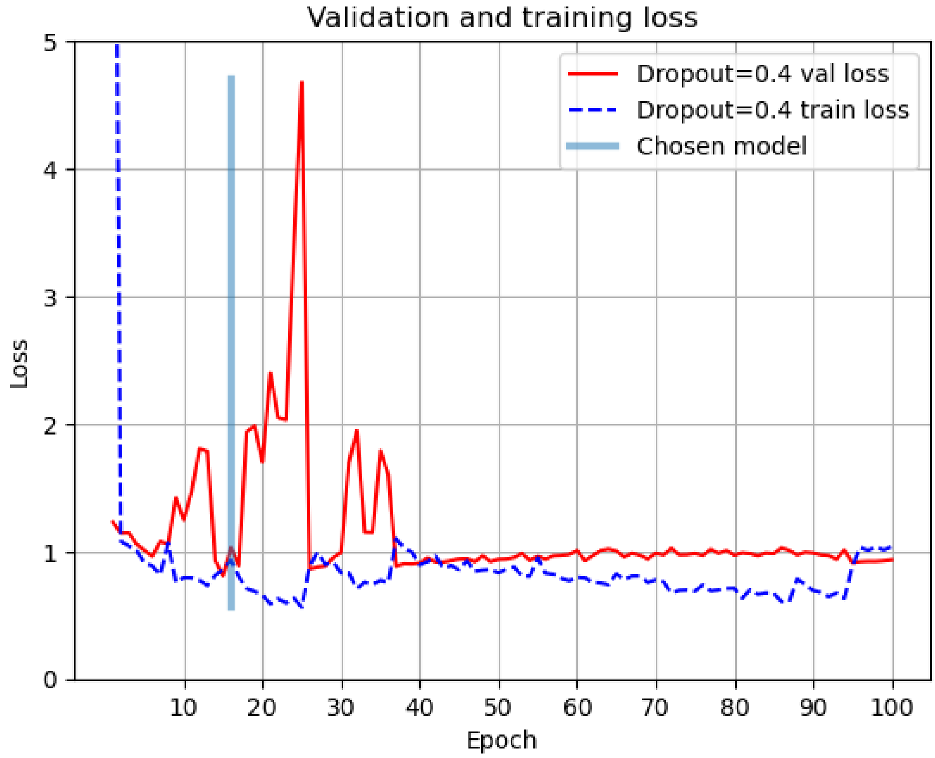


Figure 4.24: The loss on validation data during training for the model with a dropout rate of 0.4. The chosen model from the training is shown in a blue, vertical line.

4.5 Task C: Sub-grouping using the ERC2-II dataset

The results from the analysis on the ERC2-II dataset using the algorithm structures from task B are shown in tables 4.10 and 4.11 for the binary and multilabel classification. For the binary classification, the accuracy score, balanced accuracy score and loss during training are shown in figures 4.25-4.27 for the model structure with a dropout rate of 0.2 and figures 4.28-4.30 for the model structure with a dropout rate of 0.4. For the multilabel classification, the accuracy score, balanced accuracy score and loss during training are shown in figures 4.31-4.33 for the model structure with a dropout rate of 0.2 and figures 4.34-4.36 for the model structure with a dropout rate of 0.4.

Tables 4.10 and 4.11 show that the models with a dropout rate of 0.4 are performing equally or better than the models with a dropout rate of 0.2 when classifying on the validation dataset of the ERC2-II dataset, for both the binary and multilabel classification. The plots for the binary classification model with a dropout rate of 0.2 during training show that the accuracy, balanced accuracy and loss follow a similar trend, showing that the models performs approximately equal on training and validation data throughout the whole training (figures 4.25-4.27). This suggests that the model is failing to learn from the training dataset. The same can be said for the plots for the binary classification model with a dropout rate of 0.4, shown on figures 4.28-4.30.

For the multilabel case, the plots with the accuracy, balanced accuracy and loss during training for the model with a dropout rate of 0.2 show that the model attempts to learn from the training data but hits a plateau at around epoch 30 (figures 4.31-4.33). From this point onward, the model performs equally good on both validation and test data. The same can be said for the model with a dropout rate of 0.4, shown on figures 4.34-4.36, where the model hits a plateau at around epoch 70.

Dropout	Accuracy	Balanced accuracy	Recall	Precision	F1-score
0.2	0.667	0.500	0.667	0.444	0.533
0.4	0.700	0.550	0.700	0.793	0.605

Table 4.10: The binary classification results on the ERC2-II validation data using the two different algorithm structures from task B.

Dropout	Accuracy	Balanced accuracy	Recall	Precision	F1-score
0.2	0.700	0.375	0.700	0.726	0.603
0.4	0.700	0.400	0.700	0.736	0.646

Table 4.11: The multilabel classification results on the ERC2-II validation data using the two different algorithm structures from task B.

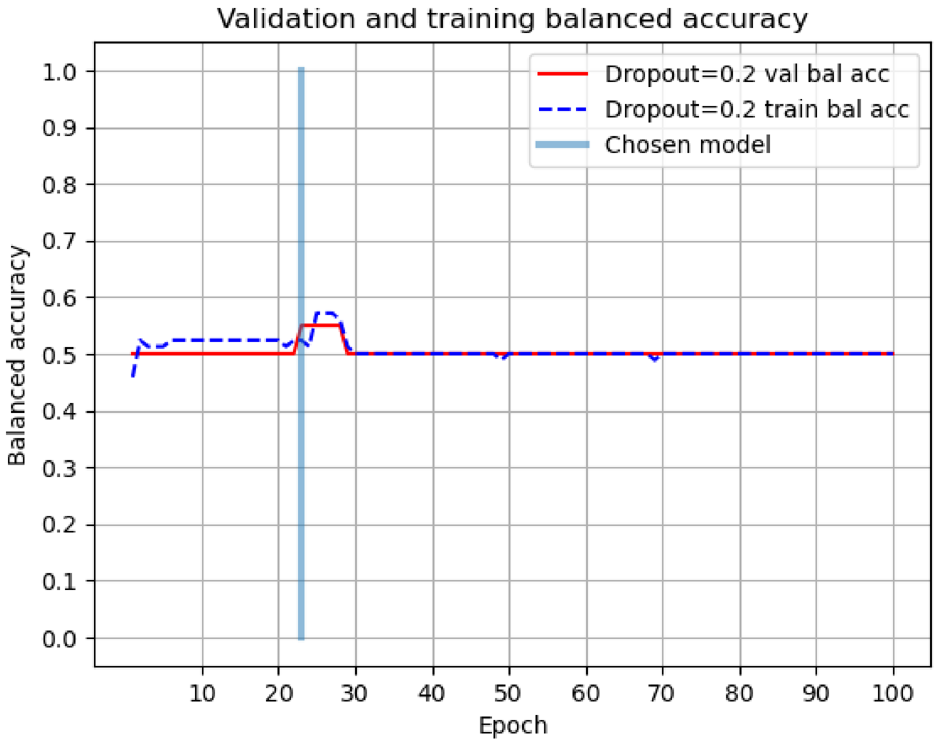


Figure 4.25: The balanced accuracy score on the ERC2-II validation data during binary training for the model with a dropout rate of 0.2. The chosen model from the training is shown in a blue, vertical line.

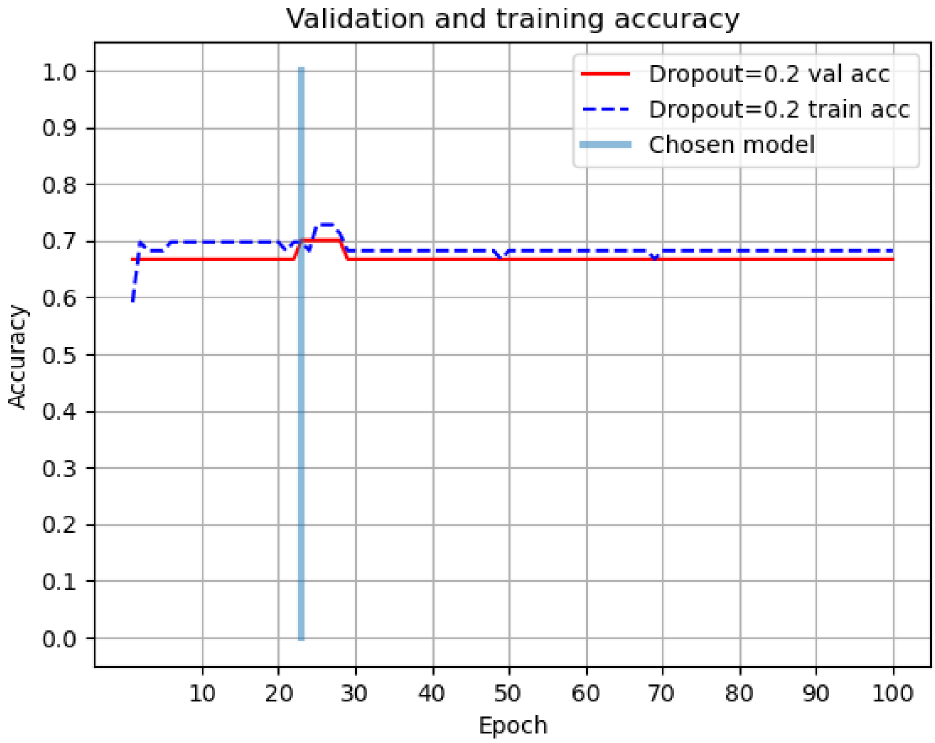


Figure 4.26: The accuracy score on the ERC2-II validation data during binary training for the model with a dropout rate of 0.2. The chosen model from the training is shown in a blue, vertical line.

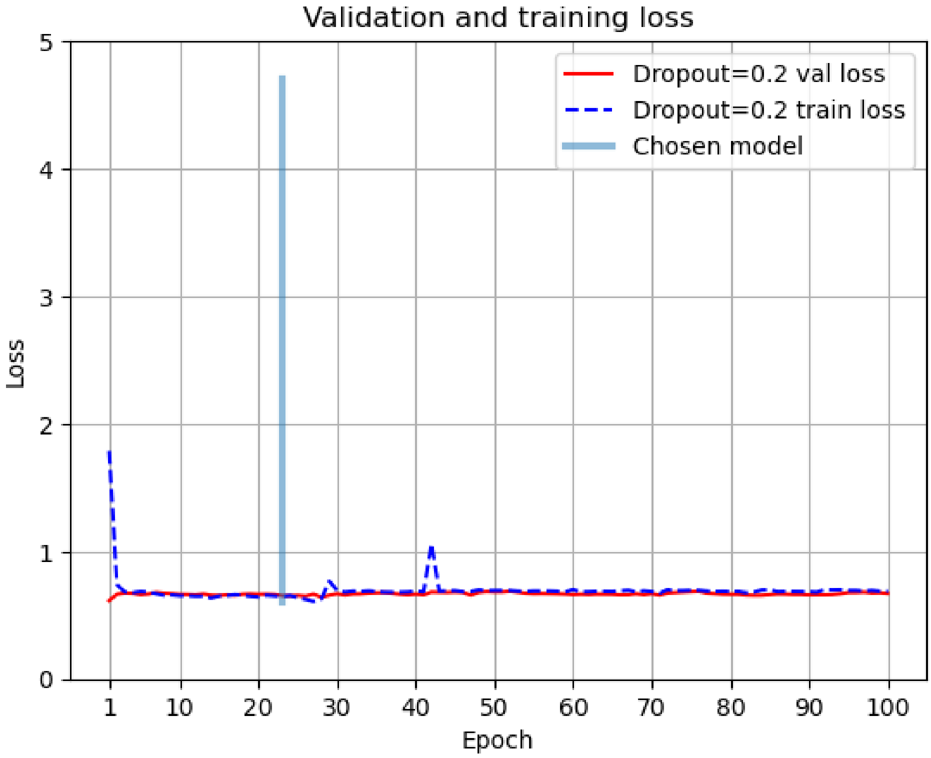


Figure 4.27: The loss on the ERC2-II validation data during binary training for the model with a dropout rate of 0.2. The chosen model from the training is shown in a blue, vertical line.

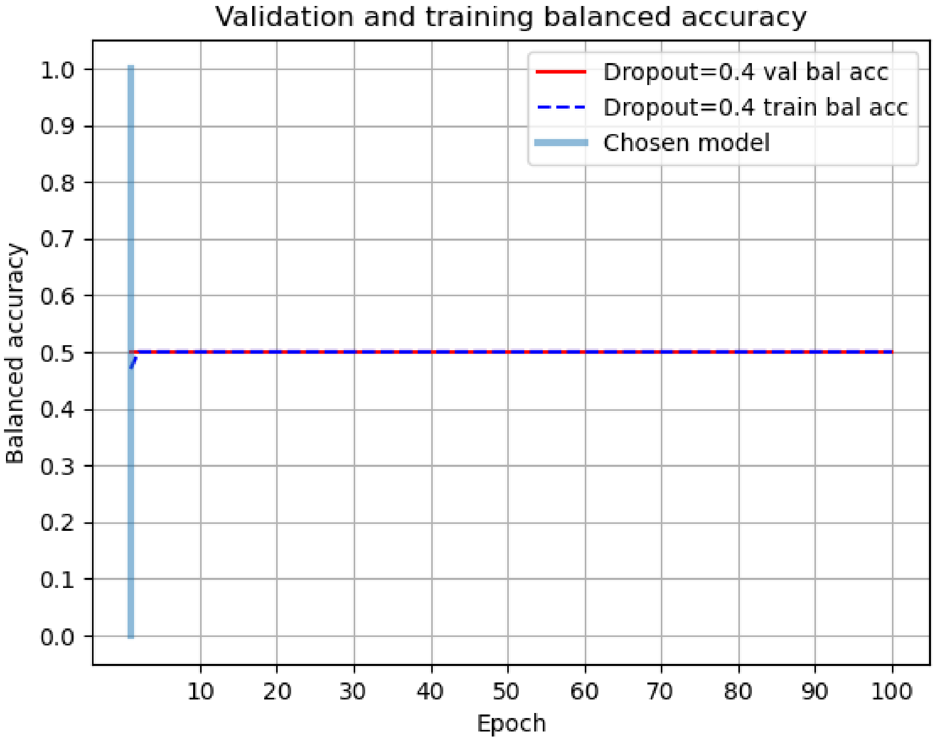


Figure 4.28: The balanced accuracy score on the ERC2-II validation data during binary training for the model with a dropout rate of 0.4. The chosen model from the training is shown in a blue, vertical line.

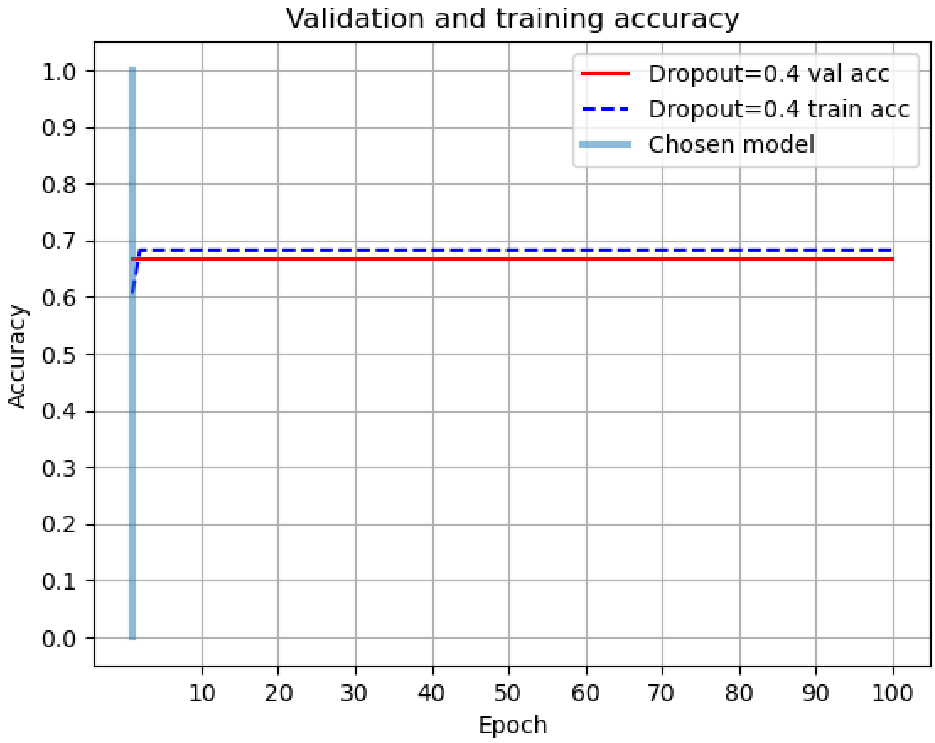


Figure 4.29: The accuracy score on the ERC2-II validation data during binary training for the model with a dropout rate of 0.4. The chosen model from the training is shown in a blue, vertical line.

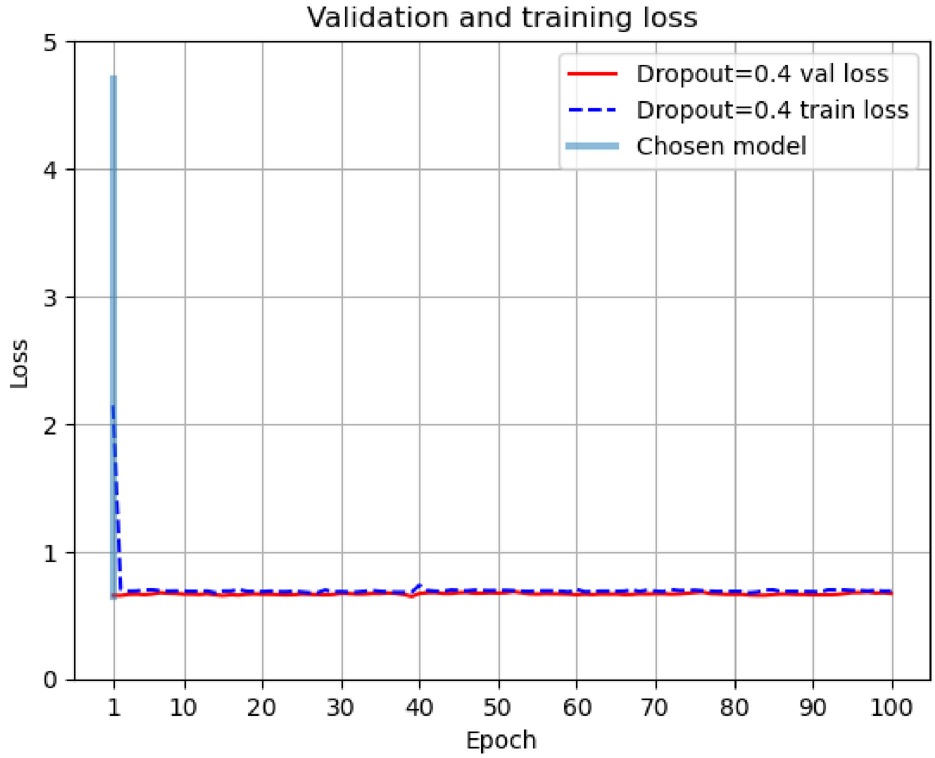


Figure 4.30: The loss on the ERC2-II validation data during binary training for the model with a dropout rate of 0.4. The chosen model from the training is shown in a blue, vertical line.

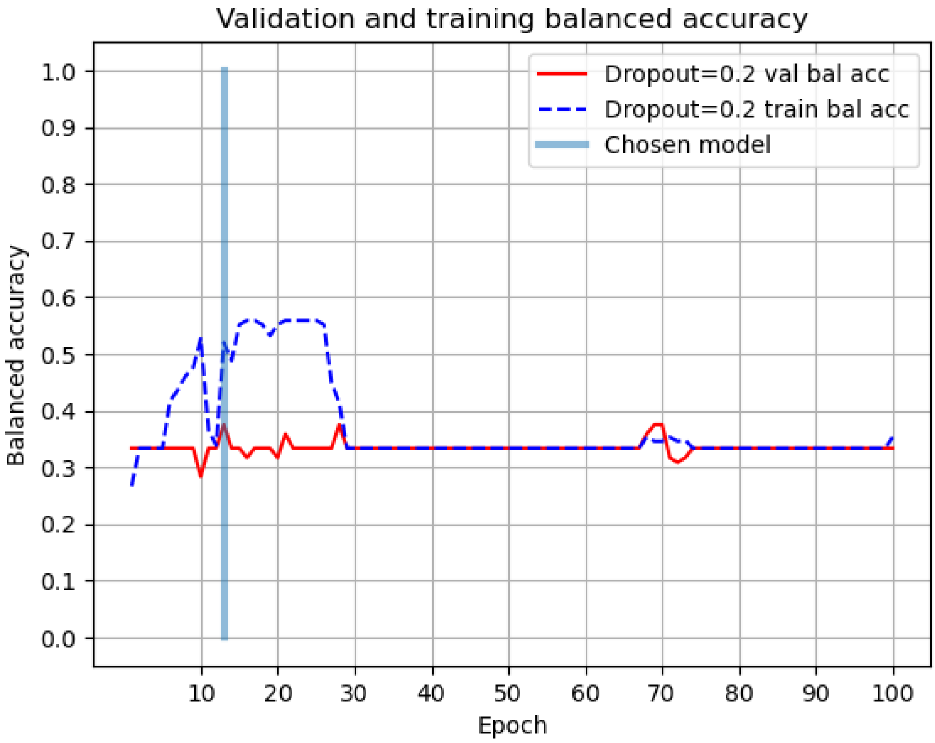


Figure 4.31: The balanced accuracy score on the ERC2-II validation data during multilabel training for the model with a dropout rate of 0.2. The chosen model from the training is shown in a blue, vertical line.

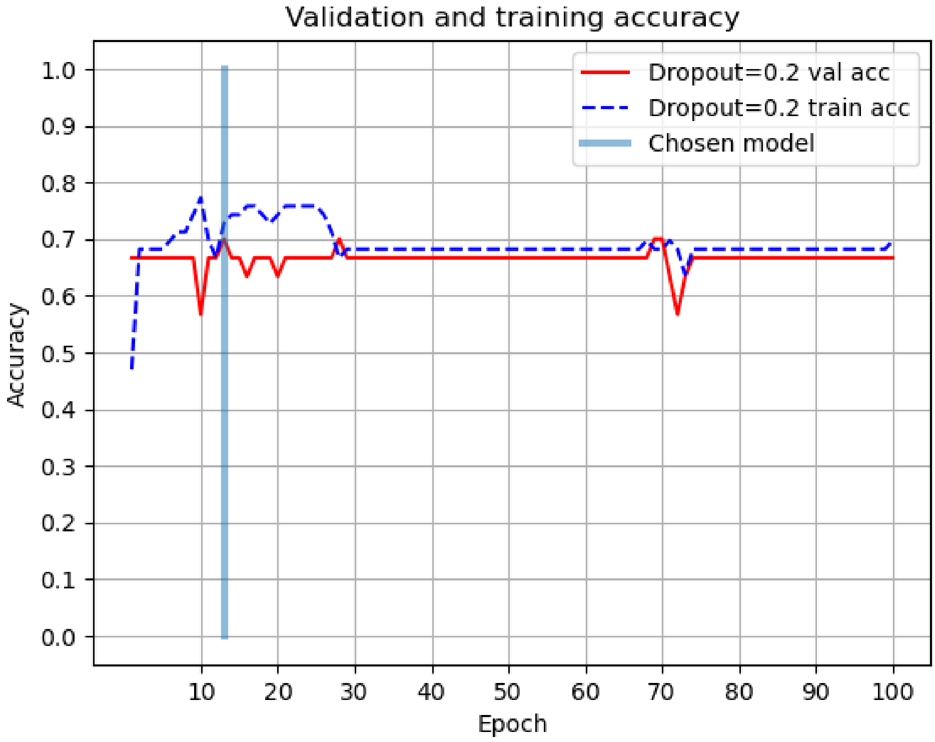


Figure 4.32: The accuracy score on the ERC2-II validation data during multilabel training for the model with a dropout rate of 0.2. The chosen model from the training is shown in a blue, vertical line.

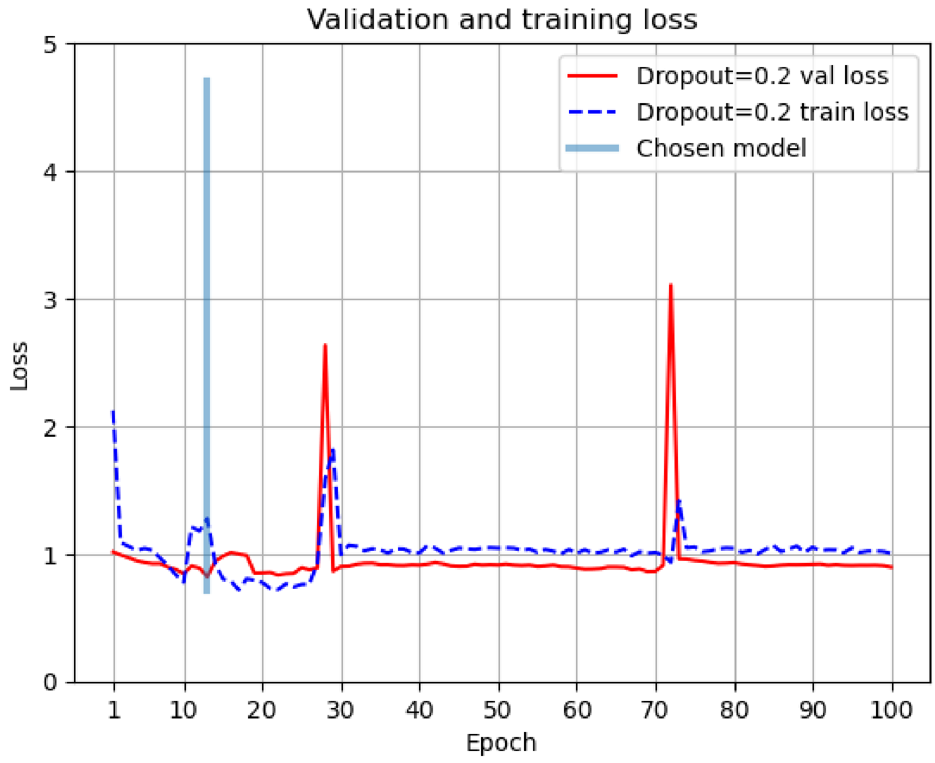


Figure 4.33: The loss on the ERC2-II validation data during multilabel training for the model with a dropout rate of 0.2. The chosen model from the training is shown in a blue, vertical line.

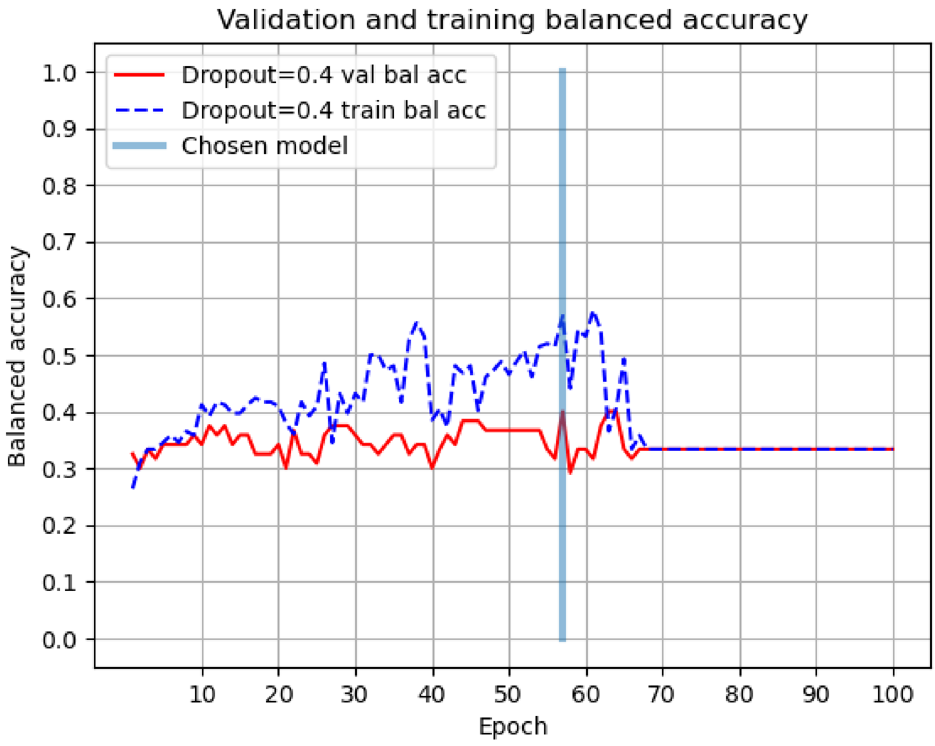


Figure 4.34: The balanced accuracy score on the ERC2-II validation data during multilabel training for the model with a dropout rate of 0.4. The chosen model from the training is shown in a blue, vertical line.

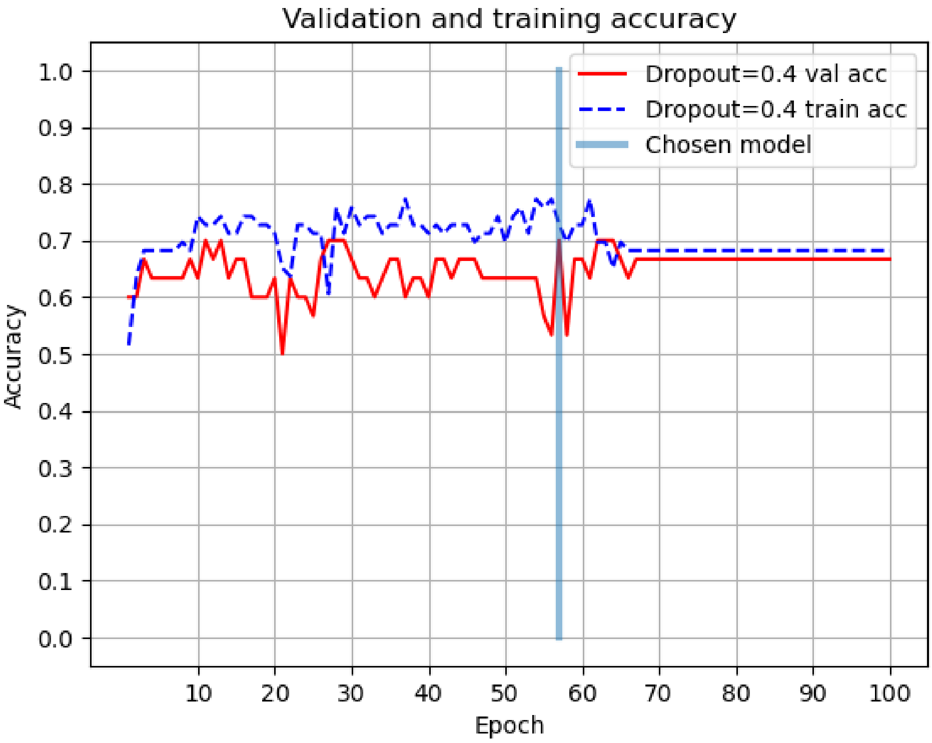


Figure 4.35: The accuracy score on the ERC2-II validation data during multilabel training for the model with a dropout rate of 0.4. The chosen model from the training is shown in a blue, vertical line.

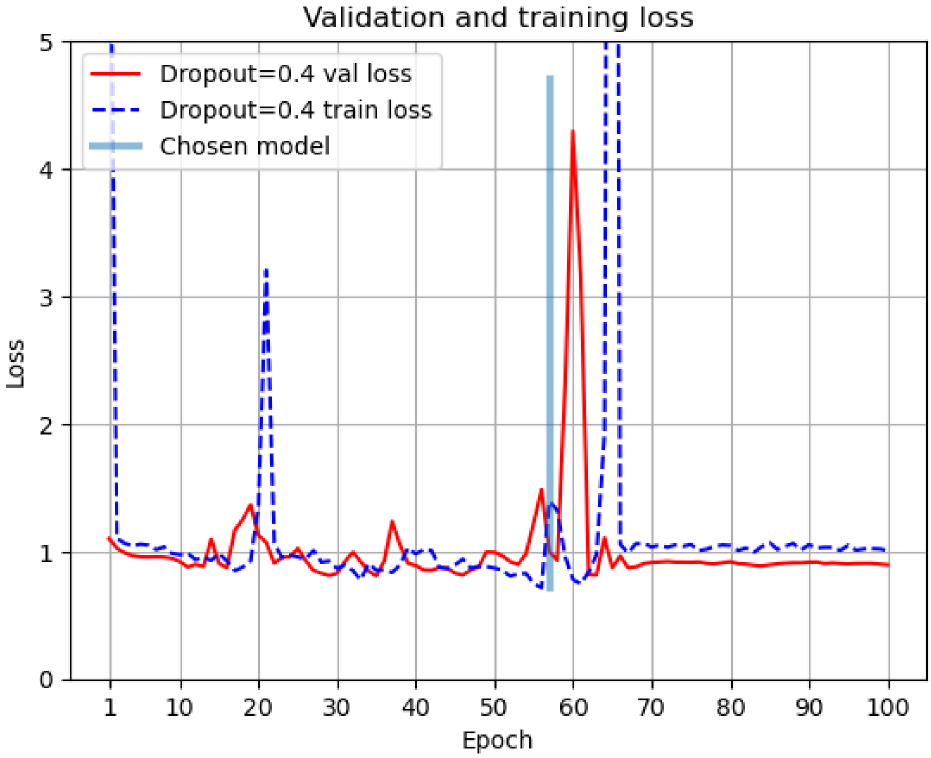


Figure 4.36: The loss on the ERC2-II validation data during multilabel training for the model with a dropout rate of 0.4. The chosen model from the training is shown in a blue, vertical line.

4.6 Task D: Data merging

The data merging task consisted of re-training the two model structures from task B on the training data from the combined dataset of the COBRE, ERC2-I and ERC2-II datasets, before validating on the validation dataset. The results are shown in tables 4.12 and 4.13 for the binary and multilabel classification, respectively.

Dropout	Accuracy	Balanced accuracy	Recall	Precision	F1-score
0.2	0.596	0.500	0.596	0.356	0.446
0.4	0.596	0.500	0.596	0.356	0.446

Table 4.12: The binary classification results on the validation data of the combined dataset of COBRE, ERC2-II and ERC2-I using the two different algorithm structures from task B.

Dropout	Accuracy	Balanced accuracy	Recall	Precision	F1-score
0.2	0.596	0.333	0.596	0.356	0.446
0.4	0.596	0.333	0.596	0.356	0.446

Table 4.13: The multilabel classification results on the validation data of the combined dataset of COBRE, ERC2-II and ERC2-I using the two different algorithm structures from task B.

Table 4.12 shows that both of the binary models perform equally with a balanced accuracy of 0.5, suggesting that every prediction belonged to the same class. The results from the multilabel classification in table 4.13 follow a similar trend, where both models perform equally with a balanced accuracy of 0.33, suggesting that every prediction was the same and therefore that the model fails to distinguish between groups.

4.7 Task E: Classification of unseen test data

The binary and multilabel model selection analyses on the ERC2 datasets are shown in tables 4.14 and 4.15, respectively. The training ran for 200 epochs for each model structure and the training time was 27±2 hours. The models with a layer size of 10, a learning rate of 0.0001 and a dropout rate of 0.4 performed best on validation data for both the binary and multilabel analyses. These two models were used on unseen test data and the results are shown in table 4.16. Both models perform substantially worse on test data than on validation data.

Layer size	Learning rate	Dropout rate	Accuracy	Balanced accuracy
5	0.001	0.2	0.662	0.632
5	0.001	0.4	0.649	0.668
10	0.001	0.2	0.743	0.727
10	0.001	0.4	0.730	0.740
10	0.0001	0.2	0.716	0.658
10	**0.0001**	**0.4**	**0.784**	**0.743**

Table 4.14: The binary classification results on the validation data of the ERC2 datasets.

Layer size	Learning rate	Dropout rate	Accuracy	Balanced accuracy
5	0.001	0.2	0.596	0.440
5	0.001	0.4	0.635	0.436
10	0.001	0.2	0.581	0.510
10	0.001	0.4	0.622	0.480
10	0.0001	0.2	0.662	0.449
10	**0.0001**	**0.4**	**0.716**	**0.517**

Table 4.15: The multilabel classification results on the validation data of the ERC2 datasets.

Analysis	Accuracy	Balanced accuracy	Recall	Precision	F1-score
Binary	0.576	0.554	0.576	0.590	0.582
Multilabel	0.508	0.348	0.508	0.477	0.492

Table 4.16: The classification of unseen test data using the best models on validation data, both binary and multilabel classification.

Chapter 5

Discussion

The problem addressed in this thesis is difficult and probably an ill-posed problem with more variability than what would be ideal for sub-grouping in SCZ. Therefore, the overall performance of the implemented ML models are dependent on several factors discussed in the following sections.

5.1 Sub-groups - does it work?

The performance of the ML models that were developed in the current study heavily relies on the level of detail in the patient grouping done in task B. The patient grouping was performed because it was challenging for the ML models to classify all the different sub-groups from the ICD-10 and DSM-IV classification, considering how few representatives there were in some of the sub-groups in the COBRE dataset. As previously described, the sub-grouping was performed from a clinical perspective in close conversation with a psychiatrist to make sure that the new patient grouping was clinically relevant. The idea was to group every patient with ICD-10 codes beginning in F20, i.e. F20.0 to F20.9, as the main SCZ group and the other subjects on the SCZ spectrum as the other SCZ group. The grouping of the patients was performed to create groups with enough representatives to be used for ML.

Other ideas were to sub-group based on the severity of the disorder, given by the PANSS score. This score is being used clinically, and is estimated using a questionnaire filled by an expert in dialogue with the subject to score positive and negative symptoms in psychotic disorders. Based upon the scores of various items of the PANSS, subjects could be divided based on degree of hallucinatory behaviour (e.g. mild and severe) or the scores from the entire scale could be used for a regression task. However, this was not possible as PANSS scores were not present in the COBRE dataset. A future direction would be to include the PANSS score data in ML. Had the study started with the ERC2 datasets, this would have been a viable direction that would also be clinically relevant. However, it was important to establish the model on the online data as the approach was implemented from scratch.

Results from this study may suggest that there are several limitations when classifying sub-groups based on resting state fMRI alone. It should however be noted that the sub-grouping from a clinical perspective also has limitations and is not necessarily a reliable gold standard. This is a known issue addressed by the ICD. The sub-groups have been phased out in the new ICD-11, partly because of the heterogeneous patient group. Therefore, future work should be open to investigate other grouping criteria. An example of a different grouping could be to use each of the larger groups of the ICD-10 classification, e.g. F20 as one group and F21 as another etc.

5.2 Effect of preprocessing

Preprocessing is crucial when analysing fMRI [11], in particular subject motion which needs to be corrected for. The preprocessing steps made sure that all the scans were aligned, which resulted in brain regions/voxels being in the same coordinate(s) from subject to subject. The main difference between the scans are thus the activations rather than the structural differences and/or subject movement, and could be preferable for the ML models since the activations may contain information on the disorder.

The preprocessing steps performed in the current study are the baseline preprocessing pipeline for fMRI images in the SPM package. This is to make sure that all subjects were in the same reference space, the MNI space. The MNI template is widely used and makes the current study more comparable to other studies. The different smoothing kernels in the current study were used to address the fact that the choice of smoothing kernel can heavily impact the results of the preprocessing [55] and hence the input to the ML algorithm.

5.3 Datasets - are they comparable?

The basis for the whole study is the IID assumption, as explained in section 2.2.4. This assumes that the COBRE dataset is expected to be drawn from the same distribution as the ERC2 datasets, as they both consist of mainly subjects with SCZ with resting-state fMRI data. However, the following paragraphs suggest some differences that challenge the IID assumption.

Figure 4.1 shows that the age distribution between the three datasets differ. Using two-sample t-tests, it was found a significant difference in age between the datasets. This shows that the COBRE and ERC2-II datasets have a similar age distribution, and that the subjects from the ERC2-I dataset are significantly younger. This may impact the results as younger brains are different from older brains (e.g. volume size and ventricle size) and a ML model might learn from these characteristics instead of the resting state BOLD signal, which may include critical information about the disorder. Furthermore, figure 3.3 shows that the patient groups have similar age distributions, while the HC group consists of younger participants. This could potentially influence the ML model to learn that younger brain features are more likely to belong to a control subject.

In task C, when using the ERC2-II dataset only, the idea was to continue using the hyperparameter combinations which performed best on the COBRE dataset. These two algorithm structures had a layer size of 5, learning rate of 0.01 and a dropout rate of 0.2 or 0.4. However, one issue was that the datasets were acquired using different scanners. The COBRE dataset was acquired on a Siemens scanner, while the ERC2 datasets were acquired on a GE scanner. In addition, they were collected with different aims and different protocols. While the COBRE dataset consists of mainly subjects with SCZ, the ERC2-II dataset consists of subjects with auditory hallucinations, where many of them are diagnosed with SCZ.

These differences could have an effect on the ML, mainly the difference in the scanner. A study by R. Kushol et al. [63] found that using images acquired from different scanners can impact ML. They investigated how training on MRI scans from different scanners can impact classification on images from other scanners. The performance of their models drops drastically when classification was performed on images from a different scanner than the one used for training data. They used a trained model to classify images from a different dataset and did not re-train like what was done in the current study. The argument that a hyperparameter combination that performed well on a dataset from one scanner did not perform well when used on a dataset from another scanner, still makes sense due to the IID assumption not being completely valid.

The COBRE dataset contained no information on PANSS score or hallucinations, which can make it hard to both determine if the dataset is representative of the ERC2 datasets and to determine how well the dataset is for hyperparameter tuning.

In addition, the resolution of the raw images differ between the datasets, with the images from the ERC2 datasets having 4 times as many pixels per slice compared to the COBRE dataset. Although the spatial resolution after the preprocessing is the same for all images no matter the original resolution, this might play a part in the ML. By looking at the example images from each of the datasets after preprocessing, figure 4.6 shows a difference in the images nonetheless. The two example images from the ERC2 dataset are much more similar to each other than to the example image from the COBRE dataset. Given more time, the effect of the different smoothing levels would have been tested for the ERC2 datasets in the same way as it was done for the COBRE dataset. The higher resolution in the raw images of the ERC2 datasets could have led to another level of smoothing making the models perform better, i.e. the level of smoothing based on the COBRE dataset is not necessarily the best choice for the higher resolution ERC2 images.

5.4 Feature extraction vs raw data

Some previous studies on the same area have used feature extraction before applying DL on MRI scans of subjects with SCZ [6], [7], [8], [9] or on subjects on the autism spectre [52], [53]. They have achieved significant results, and this might be because of the feature extraction that was performed. However, in this study, no feature extraction was performed on the fMRI images before DL was applied. This is because of the complex structure of DL algorithms. In theory, they should be able to learn from complex patterns in the data without any prior feature extraction, given enough data [64]. A DL algorithm can, in a way, do the feature extraction itself on the data, as the algorithm learns from the features that provide the most information. In addition, if the DL algorithm finds these complex patterns, there is less subjective bias in the predictions because there is no choice between different feature extraction methods. This choice in itself could contribute to overfitting. Furthermore, feature extraction could also potentially remove crucial information from the data.

The sizes of the datasets used in this study are small in the context of DL. Popular datasets used in DL are the ImageNet dataset [48], consisting of 3.2 million images, and the RadImageNet [65], consisting of 1.35 million annotated medical images. Dataset sizes like these are one of the reasons why DL has been so successful. The argument that a DL algorithm can learn any complex pattern in a dataset is only valid if there is enough data. The idea in the current study was that the high complexity of the four dimensional data would compensate for the dataset size. However, the low number of subjects could be one of the reasons why the methods used in this study, i.e. with no feature extraction, do not outperform previous contributions in the area. However, compared to other studies on resting-state fMRI or SCZ who have used only the COBRE or the ERC2-II datasets [6, 7, 9, 42], the total dataset size in the current study is larger. This could potentially increase the performance or generalization of the results, but from a DL perspective, the dataset sizes are not big enough to confidently confirm or deny that the proposed methods in the current study are reliable.

5.5 Metrics

Most articles classifying with ML are using the accuracy score as their chosen performance measure, even if the dataset is balanced or not. In this study, the balanced accuracy score is used because of the very imbalanced dataset after the patient grouping. The thought was that using the balanced accuracy would make the performance more trustworthy, and in addition increase the chance of choosing a model that predicted one of the two patient groups. This was preferable because of the tendency of the model to predict that all subjects were HC. Furthermore, it allowed the use of all the healthy controls that were not age and gender matched with a subject, which increased the total data size. Even though the dataset was then imbalanced, it potentially created a model that was more certain when predicting healthy controls. In addition, this made it comparable to studies with a balanced dataset.

The inclusion of accuracy in all the results was to make it comparable to other studies, as accuracy frequently is the chosen performance measure. In addition, the recall and precision was used to investigate the case where the models predicted only one class. The F1-score was used in the same way as the balanced accuracy score, as this measure also takes into account the class imbalance. For the binary classification, the recall is the relative number of correct positive predictions amongst all the actual positives and the precision score is the same as the relative number of correct positive predictions amongst all positive predictions. For the multilabel classification, the recall and precision becomes the weighted average of the recall and precision for every class, respectively. This means that for every multilabel classification task, the recall was the same as the accuracy score. The precision score weights the amount of FP but ignores any FN, meaning that using precision in model selection in this case would result in a model that ignores every time a positive is predicted wrong, e.g. every time a subject with SCZ is predicted to be either a healthy control or a subject on the SCZ spectrum apart from SCZ itself. This was important in the current study and cannot be ignored, and precision was therefore used as a way of showing information rather than being used for model selection.

5.6 Algorithm engineering choices

The algorithm structure used in the current study was limited by available resources of the computer environment, especially the GPU memory of the system. This limits the batch size, the depth of the network and the layer sizes. Given better computing power and memory, the algorithm could be deeper and more complex, potentially providing better results. The reason that the system was a limitation was due to the four dimensional nature of the input data, which made the ML models calculate parameters in one or two dimensions more than what have been done in previous studies on the topic [6–9]. The exponential increase in number of parameter calculations for each dimension in the input data might be one of the reasons that four dimensional data as input has not been used much previously.

5.7 Task A: Establishing ML pipeline using online data

The reason behind using three different levels of preprocessing was to investigate whether it had any effect on the ML algorithm. For the binary classification in task A, i.e. when only using the COBRE dataset, models trained on the three image types performed similarly with unweighted cross entropy loss, but differently with the weighted cross entropy loss, see figures 4.3 and 4.4. The model trained on the non-smoothed images achieved an accuracy of 80% and outperformed the other two models on every metric except for precision. However, this was due to models trained on the 8 mm smoothed images predicting the same class excessively, resulting in a high precision.

For the multilabel classification in task A, the model trained on the non-smoothed images outperformed the other two models on every given metric, both with and without weights in the loss function. Based on these results, the analyses from that point forward were performed with non-smoothed images exclusively, for both binary and multilabel classification. A reason for this result could be that the models trained on non-smoothed images had more information, since smoothing lowers the contrast in the images. The non-smoothed images contains more information from the raw images, with reduced effects of head movement, physiological cycles and inhomogeneities in the magnetic field, which potentially can be seen in the raw images as described in section 3.4 [11].

As seen in tables 4.3-4.6, the addition of weights to the cross entropy loss function resulted in a significant increase in performance in all cases except for the binary classification on images smoothed with the 8 mm kernel. Due to these results, weighted cross entropy loss became the standard loss function to be used for the rest of this study.

A reason for this could be that the function punishes the majority class, in this case the HC group, and rewards the loss when predicting the less represented classes, which for task A was the sub-groups present in the COBRE dataset.

The Adam optimizer was chosen because it is computationally efficient, requires little memory allocation and works well with a large amount of parameters [61]. The addition of the weight decay was to help reduce the chance of overfitting. The value of 0.025 was inspired by I. Loshchilov et al. [66] who attempted to find the optimal value for weight decay for the SGD and Adam optimizers.

5.8 Task B: Sub-grouping using the online data

5.8.1 Hyperparameter tuning

The results from the first hyperparameter tuning run from table 4.7 show that using a layer size of 5 gave the best results on both accuracy and loss. Therefore, the search space for the second run only included a layer size of 5 but kept the two different learning rates. This was because using a learning rate of 0.01 gave the lowest loss, while using 0.001 gave the highest accuracy. In addition, the number of dropout rates was increased to 7 to expand the search space for the bigger tuning run. The second hyperparameter tuning run provided two different model structures, one performing better on accuracy and one performing better on loss. Therefore, due to the difficulty of choosing one of the model structures, both were carried on to the next section of the current study.

The two models were re-trained to look at other metrics such as F1-score and balanced accuracy score and to identify which classes were predicted wrongly, to determine which of the two that was to be used on the combined dataset in the following section. This was because the model with the highest balanced accuracy during training was to be chosen instead of the one with the highest accuracy. However, the hypothesis was that this would be easier with the combination of the COBRE- and the ERC2 datasets, since the smallest patient group (other SCZ) would be doubled in size after the combination.

The highest accuracy and balanced accuracy from the multilabel classification on the sub-groups in the COBRE dataset, i.e. from task A, were 0.737 and 0.456, respectively (table 4.6). When comparing this to the best model from task B (from table 4.9), where the accuracy and balanced accuracy were 0.600 and 0.574, the model from task B achieved a higher balanced accuracy but a lower accuracy than the model from task A. This could be because the previous model was better at predicting the classes with most representatives but worse at predicting the others, i.e. better at predicting control but worse at predicting the two patient groups. However, since the balanced accuracy was the chosen metric for the remainder of the study, the hyperparameter tuning along with the patient grouping was a step in the right direction.

The hyperparameters used for tuning were chosen because it was thought that they would affect the predictions the most, and because it was challenging to initially set these hyperparameters to an optimal value. The learning rate is often the most important hyperparameter to tune, and it makes more sense to scale the learning rate in the log-domain than linearly [67]. Lower learning rates were not considered in task B due to the substantial increase in training time. The layer size was tuned because it was thought that a more complex network could potentially learn more from the training data. The dropout rate was tuned because setting it too low could result in overfitting, but setting it too high could result in the models not learning from the data.

Given enough time and computing power, the batch size would be tuned because it heavily impacts the performance of a ML model [68]. The batch size could be set to 1 to enable online learning [27] or it could be set to a higher value than in the current study to calculate the gradients for more data points at a time. Additionally, one could add more layers to allow the network to learn more complex patterns from each data point. However, it would not necessarily perform better than the models used in the current study due to the higher chance of overfitting with a more complex algorithm structure.

5.9 Task C: Sub-grouping using the ERC2-II dataset

The classification results from task C can be seen in table 4.11 for the multilabel classification and table 4.10 for the binary classification. The model with a dropout rate of 0.4 slightly outperformed the model with a dropout rate of 0.2, although both models appear to have underperformed compared to the results from task B. The balanced accuracy, accuracy, and loss during training (figures 4.25-4.36) all converge early and show that the models do not perform well on the validation data. Unlike the plots for loss in task B, i.e. on the COBRE dataset, these plots show that the training loss does not go down. This could mean that the models do not learn from the training data, and it is therefore an overreach to expect that the models would perform well on the validation data. A possible reason for this is that the hyperparameter combinations that performed well on the COBRE dataset were overfit to this type of data, as discussed in section 5.3.

5.10 Task D: Data merging

The results in table 4.12 show that both of the binary models achieve a balanced accuracy of 0.5 which is the same as the baseline prediction balanced accuracy for a binary classification problem. The same can be said for the multilabel classification results from table 4.13 where the balanced accuracy is 0.33. This was not as predicted and can indicate that it was challenging for the models to learn from the dataset.

It was thought that the combination of the COBRE, ERC2-I and ERC2-II datasets would perform better than on the ERC2-II dataset alone due to the substantial increase in dataset size. In addition, the increase in dataset size would increase the generalizability since the datasets were acquired on different scanners and from different parts of the world. Furthermore, the combination of the datasets was suitable due to the similar age and sub-group distributions present in the datasets, in addition to the similar functional scanning parameters. However, judging by balanced accuracy, the models trained on the combined dataset performed worse than the models trained on the ERC2-II dataset alone, both in the binary and multilabel case. This could be due to the differences in the datasets. It is possible that parameter changes that decrease the loss on scans from the ERC2-II dataset could increase the loss on scans from the COBRE dataset and vice versa, due to the differences described in section 5.3.

5.11 Task E: Classification of unseen test data

The classification in task E was conducted using different algorithm structures than in previous tasks. This was due to the poor performance of the algorithms. Therefore, different layer sizes were tested because the model could potentially learn more complex patterns, and various learning rates were tested because the parameter updates could potentially be better, further decreasing the loss. The two dropout rates of 0.2 and 0.4 were included to investigate which dropout rate would increase the generalization the most. This was possible due to the training being run on a fairly powerful remote desktop with more available memory than in tasks A and B.

The models performed worse on the test data than on the validation data. A reason for this could be that the models were performing well on the validation data due to randomness and not because it learned from the patterns in the dataset. It could also have been because of the training, validation and test datasets not being big enough to represent the whole dataset by themselves, i.e. not big enough to give a representation of the generalization of the model. Both of these issues could potentially have been resolved with a bigger dataset size, but also with a different model structure that addresses the low dataset size. There could also be an underlying issue in the patient grouping, meaning that there possibly were no patterns for the model to learn. However, the models in this task performed better on validation data than the models from task C and D, suggesting that the combination of the two ERC2 datasets was advantageous.

One reason for only using the ERC2 datasets and not the COBRE dataset in task E was that the ERC2 images had higher resolution than the COBRE images. This does not necessarily mean that the ERC2 images are better, but the difference in resolution might have an effect on the learning of the ML model. Another reason was that the two ERC2 datasets were acquired on the same scanner. Using images from different scanners have previously been shown to provide worse results than if all images were collected on the same scanner [63], as described in section 5.3. By using only the ERC2 datasets, the models trained in this section become GE-scanner specific ML models.

Chapter 6

Conclusions

The current study investigated the possibility of sub-grouping schizophrenia spectrum disorders using deep learning on four dimensional resting-state fMRI volumes. The main findings include that using deep learning on four dimensional data is feasible and that a binary classification on four dimensional data could be reliable given enough computing power and a complex algorithm structure. In addition, the results show that the least amount of preprocessing gives the most information on classification with the grouping performed on the subjects in this study. Furthermore, a hyperparameter tuning was conducted to explore which hyperparameters would have a positive effect on the ML algorithm structure. However, the findings show that the proposed methods do not reliably classify sub-groups of SCZ based on resting-state fMRI volumes alone. In addition, the current study shows that a hyperparameter combination that works well on one dataset not necessarily will work well on another dataset, possibly due to the scanner differences and/or resolution differences in the images.

The results show that increasing the dataset size improves the performance of the ML algorithm, albeit with a different algorithm structure. This confirms the theory that a larger dataset size positively influences the training of a ML model. In addition to this, the models trained on the combined dataset also provides a better generalization performance due to the datasets being acquired at scanners from different manufacturers. Future development for this area of research could be to address the task from a different perspective. One way could be to use PANSS scores to create a regression model or to use medication information from the patients to assess how they affect their brain structurally and/or functionally. To the knowledge of the author, deep learning on four dimensional fMRI volumes had not been accomplished before, making this a novel study. The work is therefore foundational for future exploration of this area of research.

Appendix A

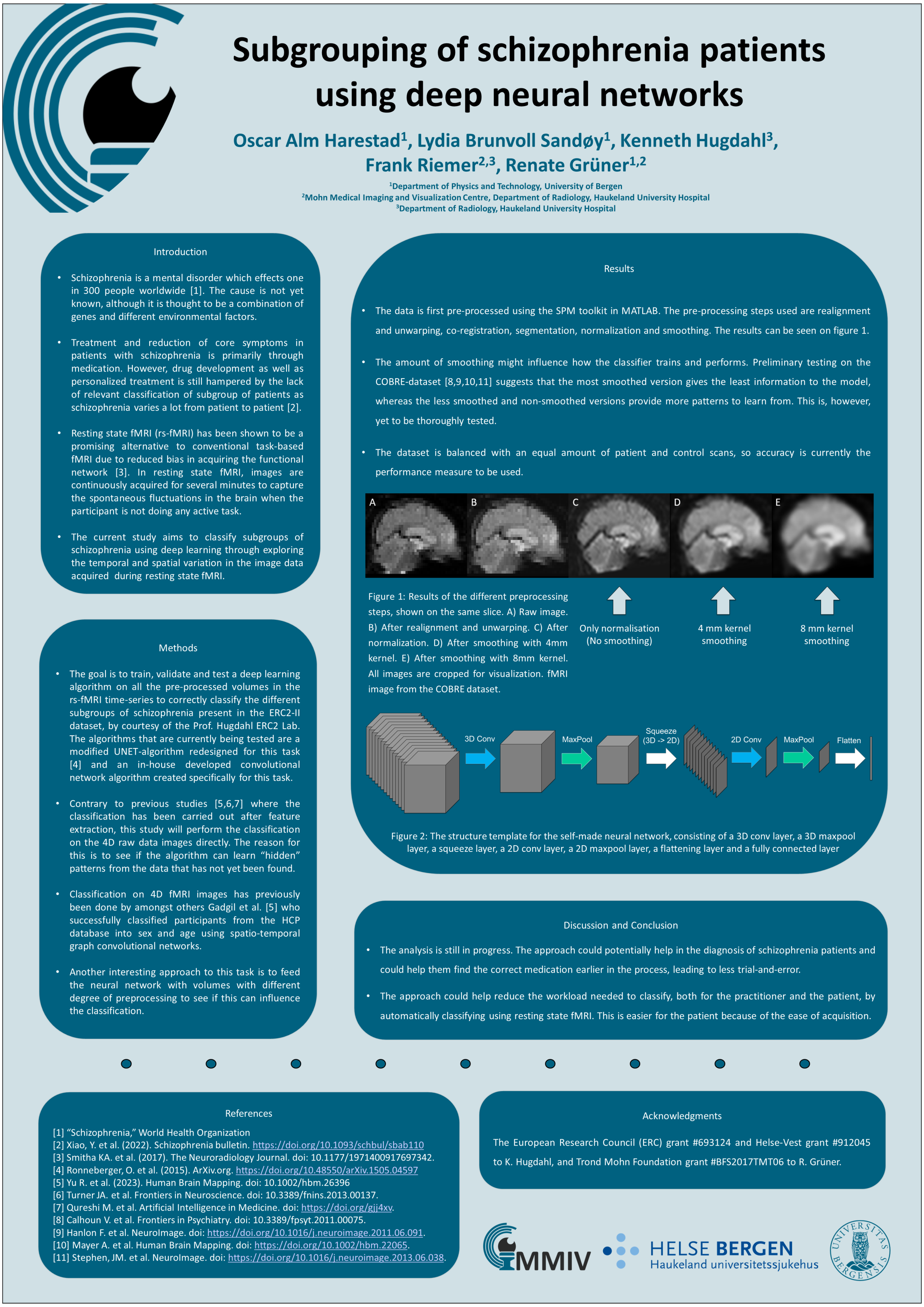
Code

A GitHub repository containing code used in this study can be found on https://github.com/harestado/MasterThesis. The repository includes scripts for preprocessing of images in MATLAB, along with scripts for training, validation, and testing for all datasets in python.

Appendix B

MMIV conference - Poster

The poster used at the 2023 MMIV conference is shown on figure B.1. The poster contains information about the current study, such as a small introduction, summary of proposed methods, and preliminary results.



Subgrouping of schizophrenia patients using deep neural networks

Oscar Alm Harestad[1], Lydia Brunvoll Sandøy[1], Kenneth Hugdahl[3], Frank Riemer[2,3], Renate Grüner[1,2]

[1]Department of Physics and Technology, University of Bergen
[2]Mohn Medical Imaging and Visualization Centre, Department of Radiology, Haukeland University Hospital
[3]Department of Radiology, Haukeland University Hospital

Introduction

- Schizophrenia is a mental disorder which effects one in 300 people worldwide [1]. The cause is not yet known, although it is thought to be a combination of genes and different environmental factors.
- Treatment and reduction of core symptoms in patients with schizophrenia is primarily through medication. However, drug development as well as personalized treatment is still hampered by the lack of relevant classification of subgroup of patients as schizophrenia varies a lot from patient to patient [2].
- Resting state fMRI (rs-fMRI) has been shown to be a promising alternative to conventional task-based fMRI due to reduced bias in acquiring the functional network [3]. In resting state fMRI, images are continuously acquired for several minutes to capture the spontaneous fluctuations in the brain when the participant is not doing any active task.
- The current study aims to classify subgroups of schizophrenia using deep learning through exploring the temporal and spatial variation in the image data acquired during resting state fMRI.

Methods

- The goal is to train, validate and test a deep learning algorithm on all the pre-processed volumes in the rs-fMRI time-series to correctly classify the different subgroups of schizophrenia present in the ERC2-II dataset, by courtesy of the Prof. Hugdahl ERC2 Lab. The algorithms that are currently being tested are a modified UNET-algorithm redesigned for this task [4] and an in-house developed convolutional network algorithm created specifically for this task.
- Contrary to previous studies [5,6,7] where the classification has been carried out after feature extraction, this study will perform the classification on the 4D raw data images directly. The reason for this is to see if the algorithm can learn “hidden” patterns from the data that has not yet been found.
- Classification on 4D fMRI images has previously been done by amongst others Gadgil et al. [5] who successfully classified participants from the HCP database into sex and age using spatio-temporal graph convolutional networks.
- Another interesting approach to this task is to feed the neural network with volumes with different degree of preprocessing to see if this can influence the classification.

Results

- The data is first pre-processed using the SPM toolkit in MATLAB. The pre-processing steps used are realignment and unwarping, co-registration, segmentation, normalization and smoothing. The results can be seen on figure 1.
- The amount of smoothing might influence how the classifier trains and performs. Preliminary testing on the COBRE-dataset [8,9,10,11] suggests that the most smoothed version gives the least information to the model, whereas the less smoothed and non-smoothed versions provide more patterns to learn from. This is, however, yet to be thoroughly tested.
- The dataset is balanced with an equal amount of patient and control scans, so accuracy is currently the performance measure to be used.

Figure 1: Results of the different preprocessing steps, shown on the same slice. A) Raw image. B) After realignment and unwarping. C) After normalization. D) After smoothing with 4mm kernel. E) After smoothing with 8mm kernel. All images are cropped for visualization. fMRI image from the COBRE dataset.

Figure 2: The structure template for the self-made neural network, consisting of a 3D conv layer, a 3D maxpool layer, a squeeze layer, a 2D conv layer, a 2D maxpool layer, a flattening layer and a fully connected layer

Discussion and Conclusion

- The analysis is still in progress. The approach could potentially help in the diagnosis of schizophrenia patients and could help them find the correct medication earlier in the process, leading to less trial-and-error.
- The approach could help reduce the workload needed to classify, both for the practitioner and the patient, by automatically classifying using resting state fMRI. This is easier for the patient because of the ease of acquisition.

References

[1] “Schizophrenia,” World Health Organization
[2] Xiao, Y. et al. (2022). Schizophrenia bulletin. https://doi.org/10.1093/schbul/sbab110
[3] Smitha KA. et al. (2017). The Neuroradiology Journal. doi: 10.1177/1971400917697342.
[4] Ronneberger, O. et al. (2015). ArXiv.org. https://doi.org/10.48550/arXiv.1505.04597
[5] Yu R. et al. (2023). Human Brain Mapping. doi: 10.1002/hbm.26396
[6] Turner JA. et al. Frontiers in Neuroscience. doi: 10.3389/fnins.2013.00137.
[7] Qureshi M. et al. Artificial Intelligence in Medicine. doi: https://doi.org/gjj4xv.
[8] Calhoun V. et al. Frontiers in Psychiatry. doi: 10.3389/fpsyt.2011.00075.
[9] Hanlon F. et al. NeuroImage. doi: https://doi.org/10.1016/j.neuroimage.2011.06.091.
[10] Mayer A. et al. Human Brain Mapping. doi: https://doi.org/10.1002/hbm.22065.
[11] Stephen, JM. et al. NeuroImage. doi: https://doi.org/10.1016/j.neuroimage.2013.06.038.

Acknowledgments

The European Research Council (ERC) grant #693124 and Helse-Vest grant #912045 to K. Hugdahl, and Trond Mohn Foundation grant #BFS2017TMT06 to R. Grüner.

Figure B.1: The poster used for the MMIV conference, autumn 2023.

Bibliography

[1] WHO, “Schizophrenia,” Jan 2022. (document), 1, 2.5

[2] W. Rössler, H. J. Salize, J. van Os, and A. Riecher-Rössler, “Size of burden of schizophrenia and psychotic disorders,” *Eur. Neuropsychopharmacol.*, vol. 15, pp. 399–409, Aug. 2005. (document), 1, 2.5

[3] R. Alisauskiene, E. Johnsen, R. Gjestad, R. A. Kroken, E. Kjelby, I. Sinkeviciute, F. Fathian, I. Joa, S. K. Reitan, M. Rettenbacher, and E.-M. Løberg, “Does drug use affect the efficacy of amisulpride, aripiprazole and olanzapine in patients with schizophrenia spectrum disorders? results from a pragmatic, randomised study,” *General Hospital Psychiatry*, vol. 83, pp. 185–193, 2023. (document), 1, 2.5.3

[4] A. J. Nashwan and B. Elawfi, “Beyond one-size-fits-all: The rise of personalized treatment in schizophrenia,” *Pers. Med. Psychiatry*, vol. 43-44, p. 100118, Mar. 2024. (document), 1, 2.5.3

[5] B. R. Rund, *Schizofreni*. Hertervig forlag, 2005. (document), 1, 2.5, 2.5.1, 2.5.5

[6] R. Yu, C. Pan, L. Bian, X. Fei, and M. Chen, “Sparsity-guided multiple functional connectivity patterns for classification of schizophrenia via convolutional network,” *Human brain mapping*, vol. 44, 06 2023. (document), 1, 2.6.1, 2.6.2, 3.7.1, 5.4, 5.6

[7] J. Zheng, X. Wei, J. Wang, H. Lin, H. Pan, and Y. Shi, “Diagnosis of schizophrenia based on deep learning using fmri,” *Computational and Mathematical Methods in Medicine*, vol. 2021, p. 17, Nov 2021. (document), 1, 2.6.1, 5.4, 5.6

[8] J. Oh, B.-L. Oh, K.-U. Lee, J.-H. Chae, and K. Yun, “Identifying schizophrenia using structural mri with a deep learning algorithm,” *Frontiers in Psychiatry*, vol. 11, Feb 2020. (document), 1, 2.6.2, 3.7.1, 5.4, 5.6

[9] J. Kim, V. D. Calhoun, E. Shim, and J.-H. Lee, “Deep neural network with weight sparsity control and pre-training extracts hierarchical features and enhances clas-

sification performance: Evidence from whole-brain resting-state functional connectivity patterns of schizophrenia," *NeuroImage*, vol. 124, pp. 127–146, 2016. (document), 1, 2.6.3, 5.4, 5.6

[10] B. Biswal, Z. Yetkin, V. Haughton, and J. Hyde, "Functional connectivity in the motor cortex of resting human brain using echo-planar mri," *Magnetic Resonance in Medicine*, vol. 34, pp. 537–541, 10 1995. 1, 2.1.8

[11] J. E. Chen and G. H. Glover, "Functional magnetic resonance imaging methods," *Neuropsychol. Rev.*, vol. 25, pp. 289–313, Sept. 2015. 1, 2.1.8, 3.4, 5.2, 5.7

[12] S. E. Petersen and O. Sporns, "Brain networks and cognitive architectures," *Neuron*, vol. 88, pp. 207–219, Oct. 2015. 1, 2.1.8

[13] M. E. Raichle, "The brain's default mode network," *Annu. Rev. Neurosci.*, vol. 38, pp. 433–447, July 2015. 1

[14] D. Dong, Y. Wang, X. Chang, C. Luo, and D. Yao, "Dysfunction of Large-Scale Brain Networks in Schizophrenia: A Meta-analysis of Resting-State Functional Connectivity," *Schizophrenia Bulletin*, vol. 44, pp. 168–181, 03 2017. 1, 2.1.8

[15] N. D. Woodward and C. J. Cascio, "Resting-state functional connectivity in psychiatric disorders," *JAMA Psychiatry*, vol. 72, pp. 743–744, Aug. 2015. 1, 2.1.8

[16] J. T. Baker, A. J. Holmes, G. A. Masters, B. T. T. Yeo, F. Krienen, R. L. Buckner, and D. Öngür, "Disruption of cortical association networks in schizophrenia and psychotic bipolar disorder," *JAMA Psychiatry*, vol. 71, pp. 109–118, Feb. 2014. 1, 2.1.8

[17] R. W. Brown, Y.-C. N. Cheng, E. M. Haacke, M. R. Thompson, and R. Venkatesan, *Magnetic Resonance Imaging: Physical principles and sequence design*. Wiley Blackwell, 2014. 2.1.1, 2.1.3, 2.1.3, 2.1.5, 2.3, 2.4, 2.1.7

[18] S. A. Huettel, A. W. Song, and G. McCarthy, *Functional Magnetic Resonance Imaging*. Sinauer Associates, 2004. 2.1.1, 2.1.4, 2.1.5, 2.1.6, 2.1.7, 2.1.7, 2.1.7

[19] C. Westbrook, C. K. Roth, and J. Talbot, *MRI in Practice*. Wiley-Blackwell, fourth ed., 2011. 2.1.2, 2.1.2, 2.1.2, 2.1.3

[20] E. R. Grüner, "Compendium phys212 medical physics and technology," 2012. 2.1.2, 2.1.3

[21] D. W. MacRobbie, E. A. Moore, M. J. Graves, and M. R. Prince, *MRI from Picture to Proton*. Cambridge University Press, 3 ed., 2017. 2.1.3, 2.1.5, 2.1.5, 2.1.5

[22] A. Bjørnerud, “The physics of magnetic resonance imaging,” 2006. 2.1.4

[23] S. Ogawa, T.-M. Lee, A. S. Nayak, and P. Glynn, “Oxygenation-sensitive contrast in magnetic resonance image of rodent brain at high magnetic fields,” *Magnetic Resonance in Medicine*, vol. 14, no. 1, pp. 68–78, 1990. 2.1.7

[24] J. M. Soares, R. Magalhães, P. S. Moreira, A. Sousa, E. Ganz, A. Sampaio, V. Alves, P. Marques, and N. Sousa, “A hitchhiker’s guide to functional magnetic resonance imaging,” *Front. Neurosci.*, vol. 10, p. 515, Nov. 2016. 2.1.7

[25] G. Baghdadi, F. Towhidkhah, and M. Rajabi, “Chapter 7 - assessment methods,” in *Neurocognitive Mechanisms of Attention* (G. Baghdadi, F. Towhidkhah, and M. Rajabi, eds.), pp. 203–250, Academic Press, 2021. 2.1.8

[26] G. Collin and M. P. van den Heuvel, “Anatomical and functional brain network architecture in schizophrenia,” in *The Neurobiology of Schizophrenia*, pp. 313–336, Elsevier, 2016. 2.1.8

[27] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. Cambridge University Press, 2023. `https://D2L.ai`. 2.2, 2.2.1, 2.2.1, 2.2.2, 2.2.3, 2.2.4, 2.2.5, 2.2.6, 2.2.7, 2.3, 2.3.1, 2.3.1, 2.3.2, 2.3.2, 2.3.3, 2.3.4, 2.3.5, 2.3.6, 2.3.6, 2.3.6, 2.9, 2.10, 2.3.6, 2.11, 2.3.7, 2.12, 2.3.8, 2.13, 3.6, 5.8.1

[28] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” 2020. 2.2

[29] J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, Y. Guo, W. Manassra, P. Dhariwal, C. Chu, Y. Jiao, and A. Ramesh, “Improving image generation with better captions,” 2023. 2.2

[30] Y. Liu, K. Zhang, Y. Li, Z. Yan, C. Gao, R. Chen, Z. Yuan, Y. Huang, H. Sun, J. Gao, L. He, and L. Sun, “Sora: A review on background, technology, limitations, and opportunities of large vision models,” 2024. 2.2

[31] M. Shehab, L. Abualigah, Q. Shambour, M. A. Abu-Hashem, M. K. Y. Shambour, A. I. Alsalibi, and A. H. Gandomi, “Machine learning in medical applications: A review of state-of-the-art methods,” *Computers in Biology and Medicine*, vol. 145, p. 105458, 2022. 2.2

[32] R. Fakoor, F. Ladhak, A. Nazi, and M. Huber, “Using deep learning to enhance cancer diagnosis and classification,” in *Proceedings of the international conference on machine learning*, vol. 28, pp. 3937–3949, 2013. 2.2

[33] C. Davatzikos, K. Ruparel, Y. Fan, D. Shen, M. Acharyya, J. W. Loughead, R. C. Gur, and D. D. Langleben, “Classifying spatial patterns of brain activity with machine learning methods: application to lie detection,” *Neuroimage*, vol. 28, no. 3, pp. 663–668, 2005. 2.2

[34] A. H. Shoeb and J. V. Guttag, “Application of machine learning to epileptic seizure detection,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 975–982, 2010. 2.2

[35] J. L. Devore, K. N. Berk, and M. A. Carlton, *Modern mathematical statistics with applications*. Springer texts in statistics, Cham, Switzerland: Springer Nature, 3 ed., Apr. 2021. 2.4.7

[36] WHO, *International Classification of Diseases, Eleventh Revision (ICD-11)*. World Health Organization, 2022. 2.5.2, 2.5.4

[37] H. M. Abdullah, H. Azeb Shahul, M. Y. Hwang, and S. Ferrando, “Comorbidity in schizophrenia: Conceptual issues and clinical management,” *FOCUS*, vol. 18, no. 4, pp. 386–390, 2020. 2.5.3

[38] A. I. Green, C. M. Canuso, M. J. Brenner, and J. D. Wojcik, “Detection and management of comorbidity in patients with schizophrenia,” *Psychiatric Clinics of North America*, vol. 26, no. 1, pp. 115–139, 2003. 2.5.3

[39] R. Tandon, H. Nasrallah, S. Akbarian, W. T. Carpenter, L. E. DeLisi, W. Gaebel, M. F. Green, R. E. Gur, S. Heckers, J. M. Kane, D. Malaspina, A. Meyer-Lindenberg, R. Murray, M. Owen, J. W. Smoller, W. Yassin, and M. Keshavan, “The schizophrenia syndrome, circa 2024: What we know and how that informs its nature,” *Schizophrenia Research*, vol. 264, pp. 1–28, 2024. 2.5.3

[40] Y. Xiao, W. Liao, Z. Long, B. Tao, Q. Zhao, C. Luo, C. A. Tamminga, M. S. Keshavan, G. D. Pearlson, B. A. Clementz, E. S. Gershon, E. I. Ivleva, S. K. Keedy, B. B. Biswal, A. Mechelli, R. Lencer, J. A. Sweeney, S. Lui, and Q. Gong, “Subtyping Schizophrenia Patients Based on Patterns of Structural Brain Alterations,” *Schizophrenia Bulletin*, vol. 48, pp. 241–250, 09 2021. 2.5.4

[41] WHO, *International Classification of Diseases, Tenth Revision (ICD-10)*. World Health Organization, 2016. 2.5.4, 3.2

[42] K. Hugdahl, A. R. Craven, E. Johnsen, L. Ersland, D. Stoyanov, S. Kandilarova, L. Brunvoll Sandøy, R. A. Kroken, E.-M. Løberg, and I. E. C. Sommer, "Neural Activation in the Ventromedial Prefrontal Cortex Precedes Conscious Experience of Being in or out of a Transient Hallucinatory State," *Schizophrenia Bulletin*, vol. 49, pp. S58–S67, 05 2022. 2.5.5, 5.4

[43] V. Calhoun, J. Sui, K. Kiehl, J. Turner, E. Allen, and G. Pearlson, "Exploring the psychosis functional connectome: Aberrant intrinsic networks in schizophrenia and bipolar disorder," *Frontiers in Psychiatry*, vol. 2, 2012. 2.6.1, 3.1

[44] F. M. Hanlon, J. M. Houck, C. J. Pyeatt, S. L. Lundy, M. J. Euler, M. P. Weisend, R. J. Thoma, J. R. Bustillo, G. A. Miller, and C. D. Tesche, "Bilateral hippocampal dysfunction in schizophrenia," *NeuroImage*, vol. 58, no. 4, pp. 1158–1168, 2011. 2.6.1, 3.1

[45] A. R. Mayer, D. Ruhl, F. Merideth, J. Ling, F. M. Hanlon, J. Bustillo, and J. Cañive, "Functional imaging of the hemodynamic sensory gating response in schizophrenia," *Human Brain Mapping*, vol. 34, no. 9, pp. 2302–2312, 2013. 2.6.1, 3.1

[46] J. Stephen, B. Coffman, R. Jung, J. Bustillo, C. Aine, and V. Calhoun, "Using joint ica to link function and structure using meg and dti in schizophrenia," *NeuroImage*, vol. 83, pp. 418–430, 2013. 2.6.1, 3.1

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015. 2.6.1

[48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. 2.6.1, 5.4

[49] B. Olabi, I. Ellison-Wright, A. M. McIntosh, S. J. Wood, E. Bullmore, and S. M. Lawrie, "Are there progressive brain changes in schizophrenia? a meta-analysis of structural magnetic resonance imaging studies," *Biological Psychiatry*, vol. 70, p. 8896, Jul 2011. 2.6.2

[50] S. Huhtaniska, E. Jääskeläinen, N. Hirvonen, J. Remes, G. K. Murray, J. Veijola, M. Isohanni, and J. Miettunen, "Long-term antipsychotic use and brain changes in schizophrenia - a systematic review and meta-analysis," *Human Psychopharmacology: Clinical and Experimental*, vol. 32, Mar 2017. 2.6.2

[51] T. E. Mwansisya, A. Hu, Y. Li, X. Chen, G. Wu, X. Huang, D. Lv, Z. Li, C. Liu, Z. Xue, and et al., "Task and resting-state fmri studies in first-episode schizophrenia: A systematic review," *Schizophrenia Research*, vol. 189, p. 918, Nov 2017. 2.6.3

[52] A. S. Heinsfeld, A. R. Franco, R. C. Craddock, A. Buchweitz, and F. Meneguzzi, "Identification of autism spectrum disorder using deep learning and the abide dataset," *NeuroImage: Clinical*, vol. 17, pp. 16–23, 2018. 2.6.3, 5.4

[53] M. Khosla, K. Jamison, A. Kuceyeski, and M. R. Sabuncu, "Ensemble learning with 3d convolutional neural networks for functional connectome-based prediction," *NeuroImage*, vol. 199, pp. 651–662, 2019. 2.6.3, 3.7.1, 5.4

[54] American Psychiatric Association, *Diagnostic and statistical manual of mental disorders*. Arlington, TX: American Psychiatric Press, 4 ed., May 1994. 3.1

[55] M. Mikl, R. Marecek, P. Hlustík, M. Pavlicová, A. Drastich, P. Chlebus, M. Brázdil, and P. Krupa, "Effects of spatial smoothing on fMRI group inferences," *Magn. Reson. Imaging*, vol. 26, pp. 490–503, May 2008. 3.4, 5.2

[56] A. Jahn, D. Levitas, E. Holscher, J. T. Johnson, A. Sayal, jstaph, JohannesWiesner, J. Clucas, T. M. Tapera, and justbennet, *andrewjahn/AndysBrainBook:*. Zenodo, jan 2022. 3.4.1, 3.4.2, 3.4.3, 3.4.5

[57] J. Ashburner, G. Barnes, C.-C. Chen, J. Daunizeau, G. Flandin, K. Friston, D. Gitelman, V. Glauche, R. Henson, C. Hutton, A. Jafarian, S. Kiebel, J. Kilner, V. Litvak, J. Mattout, R. Moran, W. Penny, C. Phillips, A. Razi, K. Stephan, S. Tak, A. Tyrer, and P. Zeidman, "Spm12 manual," oct 2021. 3.4.1, 3.4.3, 3.4.4, 3.4.5

[58] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017. 3.5

[59] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, pp. 84–90, May 2017. 3.6

[60] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. 3.6, 3.10

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. 3.6, 5.7

[62] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica, "Tune: A research platform for distributed model selection and training," *arXiv preprint arXiv:1807.05118*, 2018. 3.8.1

[63] R. Kushol, P. Parnianpour, A. H. Wilman, S. Kalra, and Y.-H. Yang, "Effects of MRI scanner manufacturers in classification tasks with deep learning models," *Sci. Rep.*, vol. 13, p. 16791, Oct. 2023. 5.3, 5.11

[64] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989. 5.4

[65] X. Mei, Z. Liu, P. M. Robson, B. Marinelli, M. Huang, A. Doshi, A. Jacobi, C. Cao, K. E. Link, T. Yang, Y. Wang, H. Greenspan, T. Deyer, Z. A. Fayad, and Y. Yang, "Radimagenet: An open radiologic deep learning research dataset for effective transfer learning," *Radiology: Artificial Intelligence*, vol. 0, no. ja, p. e210315, 2022. 5.4

[66] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019. 5.7

[67] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," 2012. 5.8.1

[68] J. E. Cejudo Grano de Oro, P. J. Koch, J. Krois, A. Garcia Cantu Ros, J. Patel, H. Meyer-Lueckel, and F. Schwendicke, "Hyperparameter tuning and automatic image augmentation for deep learning-based angle classification on intraoral photographs-a retrospective study," *Diagnostics (Basel)*, vol. 12, p. 1526, June 2022. 5.8.1