UNIVERSITY OF CALGARY

Using Machine Learning Towards Decision Support for Refractory Epilepsy Cases

by

Bijan Farhoudi

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN COMPUTER SCIENCE

CALGARY, ALBERTA

JANUARY, 2023

Abstract

Between 0.5% to 1.0% of people in North America suffer from epilepsy, and around 30% of patients are drug-resistant. Some drug-resistant patients are candidates for surgery and up to 60% to 70% of patients who undergo surgery become seizure-free. Finding a magnetic resonance imaging (MRI) abnormality on pre-operative imaging increases the chance of surgical success. However, up to 30% to 40% of pre-operative MRIs have no clear lesion in people with drug-resistant epilepsy, and only up to 40% to 50% of non-lesional MRI cases become seizure-free after surgery. The focus of this work was to design decision support tools to help clinicians evaluate patients for surgery. As the first step, we investigated the possibility of segregating MRIs with abnormality from MRIs without any abnormality using Deep Learning models. Such models would help clinicians when they examine MRIs to find an abnormality. Considering the value of predicting surgery results, in our next step, we explored the possibility of predicting the outcome of surgery using MRI and Deep Learning. Our results indicate that both lesional and non-lesional MRIs of patients with epilepsy contain signals that Deep Learning models can harness to predict the operative success., Finally, we explored the possibility of finding an abnormality in MRIs that were reported by radiologists as non-lesional by using Deep Learning.

Preface

This thesis is an original work by Bijan Farhoudi. No part of this thesis has been previously published.

Acknowledgements

I would like to express my deepest appreciation to my supervisors, Dr. Frank Maurer and Dr. Samuel Wiebe, for guiding and mentoring me throughout my Ph.D. studies. I am extremely grateful that I was their student. I also would like to thank Dr. Colin Josephson and Dr. Paolo Federico, who were on my supervisory committee.

Besides my supervisory committee, I would like to thank Anirudh Koul, Dr. Guillermo Delgado Garcia, Dr. Victoria Mosher, and Dr. Joseph Peedicail, who have helped me through the research projects.

I also must thank Dr. Hadi Kharrazi and Dr. Emma Towlson, who are members of my examination committee.

Dedication

I dedicate this thesis to my parents, Enayatollah and Safoura, who have supported me throughout my studies.

Table of Contents

Introduction

Around 0.5 to 1.0% of the world population suffers from epilepsy (Fiest et al., 2017; Theodore et al., 2006; Wiebe et al., 1999). Of this number, approximately 30% to 40% are drug-resistant (Kwan & Brodie, 2000) and may be candidates for evaluation for surgery. Randomized controlled trials of epilepsy surgery have shown that 58% to 73% of people with drug-resistant temporal lobe epilepsy became seizure-free after surgery, compared to 0% to 8% who continued to receive non-invasive management (Engel et al., 2012; Wiebe et al., 2001; Willard et al., 2022). In children, up to 37% with drug-resistant epilepsy became seizure-free after surgery compared to 0% with medications (Dwivedi et al., 2017). A systematic review and meta-analysis has indicated that long-term follow-up for seizure freedom after temporal lobe surgery is similar to those reported in short-term studies (Téllez-Zenteno et al., 2005). It is also worth mentioning that the success rate is lower for patients with extratemporal epilepsy (De Tisi et al., 2011).

The primary purpose of this thesis is to investigate the possibility of using machine learning and magnetic resonance imaging (MRI) to help physicians when dealing with drug-resistant patients who may be candidates for surgery. The goal is to provide tools to improve the result of surgery and increase the chances of seizure freedom after surgery.

As the quality of MRI is improving, experts have better means to identify the location of the abnormality (Bernasconi et al., 2011). MRI of 44% of patients with epilepsy have no sign of abnormality (MRI-negative), and usually, the number is higher for extratemporal epilepsy cases (Téllez-Zenteno et al., 2010). This is important because one of the main factors in the success of epilepsy surgery is the presence of abnormality or lesion on MRI (Bien et al., 2009; Lascano et al., 2016). A meta-analysis study by Téllez-Zenteno et al. has identified that patients with identifiable abnormality in MRI or histopathology are 40% more likely to become seizure-free after surgery than those without identifiable abnormality (Téllez-Zenteno et al., 2010). As a result, creating tools that would help clinicians deal with non-lesional MRI cases would be very valuable.

Clinical data and information gathered from diagnostic tools help clinicians define the risks and benefits of surgery for each case, and as a result, there are studies which have focused on identifying predictors of the success of the surgery (Cossu et al., 2008; Jeha et al., 2007). There are research studies which have designed binary classification tools to predict seizure freedom after surgery using Machine Learning algorithms such as K-Nearest Neighbours, Support Vector Machines, Neural Networks and Deep Learning. The mentioned algorithms have used features from MRI, electroencephalogram (EEG), Positron Emission Tomography (PET), intracranial-EEG, clinical data, or neuropsychological data as input (Senders et al., 2018). The related work chapter provides an overview of these types of studies.

MRI has been used in two different ways in the mentioned systems. Either the interpretation of MRI scans from experts has been fed to the systems (Arle et al., 1999), or features have been extracted from MRI and then provided as input to the algorithms (Feis et al., 2013). Research has shown that the structure of connectome networks extracted from diffusion-weighted MRI differs in brains with epilepsy compared to healthy brains (DeSalvo et al., 2014; Liu et al., 2014). In a study, Deep Learning and connectome networks have been used to predict the outcome of the surgery (Gleichgerrcht et al., 2018). Deep Learning algorithms are advanced versions of non-linear neural networks and were used for a medical case for the first time in the 1990s (Lo et al., 1995). There was not much attention in research communities on Deep Learning in the 2000s, but since 2012 advances in hardware and an abundance of data have revived Deep Learning algorithms. The medical imaging community has taken notice of Deep Learning algorithms since 2015 and started to use them for different cases (Litjens et al., 2017).

There have been different studies designed to predict the outcome of surgery for non-lesional MRI cases using clinical information or tools such as EEG (Grigsby et al., 1998; Zakaria et al., 2012). For example, a study identified that magnetoencephalography (MEG) could be used to identify the best candidates for surgery (Ramachandrannair et al., 2007), or in another study, clinical factors such as seizure frequency, age of onset of seizures, and a number of failed antiseizure medications (ASMs) have been identified as primary factors to predict the result for pediatric cases (Arya et al., 2016).

MRIs which have been interpreted as normal by experts may still have information to predict the outcome of surgery. There is a possibility that they would contain features indicative of the outcome. Assuming that would be the case, the mentioned features can be provided to a classifier to predict the outcome. Convolutional Neural Networks (CNNs) have two main components that can help with the prediction. First, they automatically select the most relevant features, and then they use the selected features to predict the outcome (I. Goodfellow et al., 2016; Rosebrock, 2017). Given the mentioned characteristics of both the problem and CNNs, they can be a good candidate to be used for predicting the outcome of surgery for refractory epilepsy with non-lesional MRI.

There are three main research topics that we have worked on:

1. To investigate if it is possible to differentiate between normal and abnormal MRIs using Deep Learning models.

   A model which can mark an MRI as lesional or non-lesional can help physicians when identifying candidates for surgery or when planning the surgery. For example, when a physician does not find an abnormality in an MRI, while the model suggests that there should be one, the physician may re-examine the MRI and find the abnormality.

2. To investigate if it is possible to predict the result of surgery using MRI.

   Predicting the result of surgery for both lesional and non-lesional cases can significantly impact surgery outcomes.

3. To investigate the possibility of helping physicians find abnormality in MRIs that are reported to be non-lesional by comparing MRIs of healthy brains and non-lesional brains using Deep Learning.

   Finding abnormality in MRIs is one of the indicative factors for successful surgery, and as a result, helping physicians to find abnormality can be very useful.

The introductory chapters of this document contain information about epilepsy, machine learning, Deep Learning, and computer vision, along with a related work chapter which presents an overview of other published research related to the focus of the thesis work. After providing the introductory information, the three mentioned main projects are presented in detail.

1 Background

Given the interdisciplinary nature of this work, the background chapter has two sections. The first section targets readers without medical background and provides general information about epilepsy. The second section provides an overview on machine learning, computer vision and Deep Learning algorithms. It also discusses the application of Deep Learning to analyze medical images.

1.1 Epilepsy

Epilepsy is the fourth most common neurological disease after migraine, stroke and Alzheimer's and affects between 0.5 to 1% of the global population (Ngugi et al., 2010). The prevalence of epilepsy is higher in low-income and middle-income countries compared to high-income countries. This could be due to higher rates of traffic accidents, birth injuries and neuroinfectious disorders (Singh & Trevick, 2016). Around 75% of epileptic seizures start during childhood which indicates the susceptibility of the developing brain to seizures (Stafstrom & Carmant, 2015). Reported incidences also increase in older age groups (more than 50-60 years old). Epilepsy reported in older adults could be caused by other neurological problems such as tumours or stroke. Although, in general, the prevalence of epilepsy is the same in both male and female patients, in some regions, the rate of male patients is higher, which could be due to underreporting by women considering the fact that in those regions, the women could be regarded as unmarriable if the diagnosis would be known (Fiest et al., 2017).

1.1.1 Definition

Epilepsy is a result of abnormal electrical brain activity, also known as seizure. The following is the clinical definition of epilepsy provided by the International League Against Epilepsy (ILAE):

1. "At least two unprovoked (or reflex) seizure occurring more than 24 hours apart;
2. One unprovoked (or reflex) seizer and a probability of future seizures similar to the general recurrence risk (at least 60%) after two unprovoked seizures, occurring over the next 10 years;
3. Diagnosis of an epilepsy syndrome.

Epilepsy is considered resolved for age-dependent epilepsy syndromes in those past the applicable age or those who have remained seizure-free for the last 10 years, with no seizure medicines for the last 5 years." (*Definition of Epilepsy 2014 // International League Against Epilepsy*, n.d.)

Provoked seizures are a type of seizure which are caused by head injury, fever or other reversable insults, while unprovoked seizures mentioned in the above definition do not have an immediate cause.

1.1.2 Symptoms

Patients experience different conditions depending on the type of seizures and their location. Symptoms of epilepsy include (Devinsky et al., 2018):

- Loss of consciousness or awareness
- Stiffing of the muscles
- Jerking of legs and arms
- Sensation that rises from the abdomen to the chest

- Experience of déjà vu

1.1.3 Seizure Classification

ILAE classifies seizures into three main groups: focal onset, generalized onset, and unknown onset. Figure 2.1 shows both basic and expanded versions of the classification. The basic version is meant to be used by practitioners who are not specialized in epilepsy, and the extended version is meant for physicians who are specialized in epilepsy (R. S. Fisher et al., 2017).

Focal Onset

Focal seizures originate in neuronal networks in only one hemisphere of the brain. The manifestation of focal seizures on the body depends on where the seizure starts in the brain. For example, seizures starting in the occipital lobe may result in visual phenomena. These types of seizures are divided into two main categories.

1. **Aware (without loss of consciousness).** This type of seizure does not cause loss of consciousness but may alter other senses, feelings, or emotions. It also can cause involuntary movements of some parts of the body, such as arms and legs.
2. **Impaired awareness.** This type of seizure results in change or loss of awareness or consciousness. During the seizure, the person my stare into space and not respond to the surrounding environment. The patient may also perform repetitive movements.

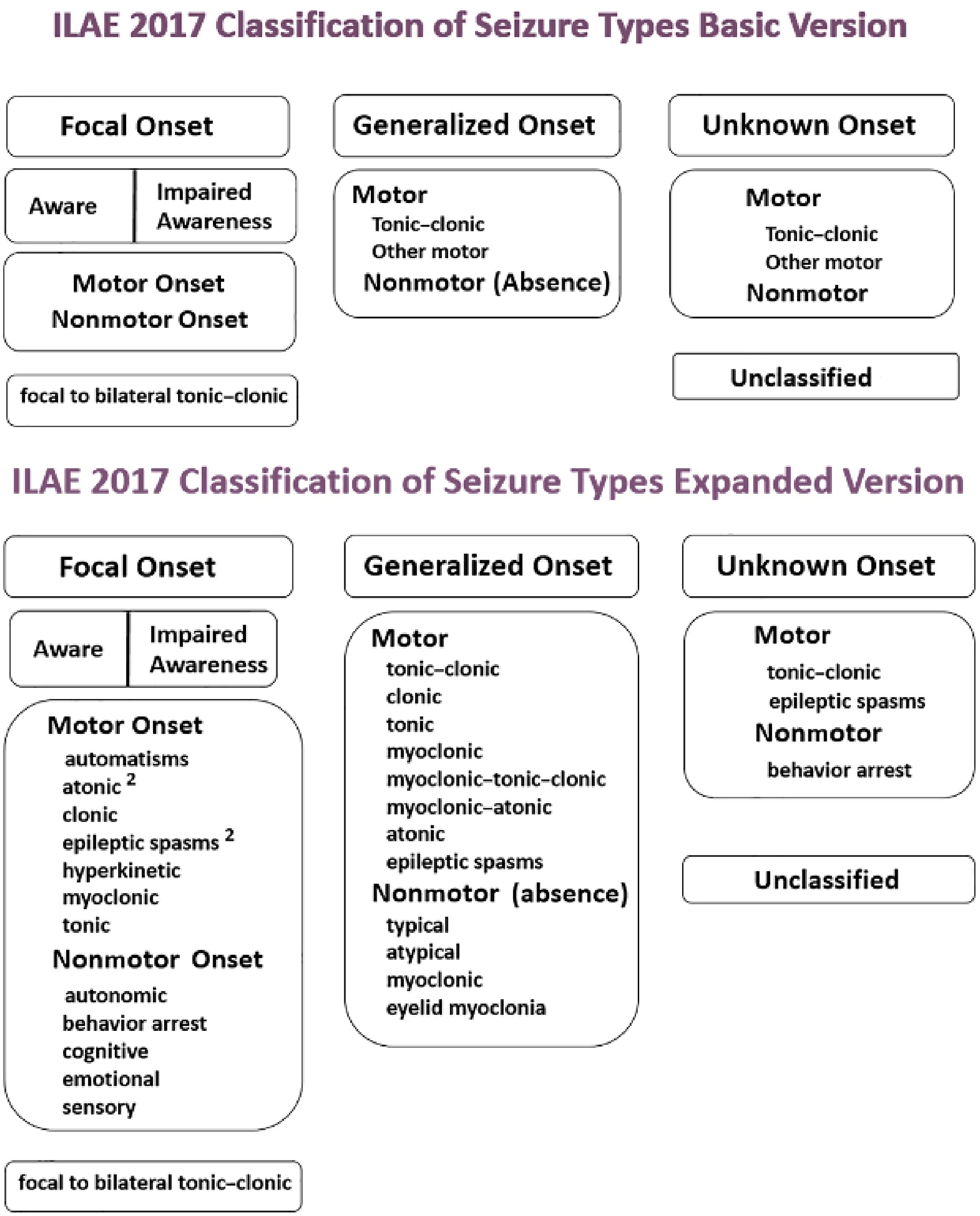


Figure 2.1: International League Against Epilepsy's definition of different types of seizures

Generalized Onset

This type of seizure rapidly engages bilaterally distributed networks. Therefore, some seizures may start focally and change to generalized onset.

1. **Motor**
    a. **Tonic-clonic.** They may result in a sudden loss of consciousness, body stiffening (tonic), and then paroxysmal body shaking/convulsions (clonic). They are considered to be the most dramatic type of seizure. They can sometimes result in biting the tongue or loss of bladder control.
    b. **Other Motor.** These types of seizures are classified to seven types of seizures listed in figure 2.1. They each affect the body in different ways. For example, Tonic seizures usually affect muscles in the back, arms, and legs, whereas Clonic seizures affect the neck, face, and arms.
2. **Non-motors (absence).** This type of seizure may cause a short loss of awareness and mainly occur in children. There may be subtle body movements during the seizure.

1.1.4 Causes

The cause of epilepsy cannot be identified in all people who have epilepsy, but the following is a list of factors which can be considered as the root of the illness (Devinsky et al., 2018).

- Genetic
- Structural (example: stroke and brain tumours)
- Infectious (example: viral brain infections)
- Metabolic (example: facilitated glucose transporter member 1 deficiency)
- Immune (example: multiple sclerosis and autoimmune encephalitis)

It is also noteworthy to mention that there are factors which may increase the risk of epilepsy. The following are three of the mentioned factors (Pitkänen et al., 2016):

- Brain injuries
- Brain tumours and stroke
- Central Nervous System infection

Up to 10% of people in the world will experience seizure at some point in their lives (Hauser & Beghi, 2008), but not all seizures fall under the definition of an epileptic seizure. History and clinical examinations along with the context and situation in which the seizure happened, help to diagnose epilepsy. Diagnostic tools such as EEG and neuroimaging also help clinicians in diagnosing, treating, and prognosticating epilepsy (Stafstrom & Carmant, 2015).

1.1.5 Comorbidities, Mortality, and Complications

Mental health disorders are among the main epilepsy comorbidities. For example, depression is one of the major comorbidities of epilepsy, and around 30% of patients with epilepsy suffer from depression. This number is more than 50% for treatment-resistant patients. Depression is also associated with a reduction in adherence to medication and the highest seizure frequency (Boylan et al., 2004; Fazel et al., 2013; Fiest et al., 2013). Stress and anxiety are two other known mental health disorders in patients with epilepsy. People who have epilepsy report higher levels of stress compared to the rest of the population (McKee & Privitera, 2017). Sexual dysfunction is also common in patients with epilepsy and could have pharmacological or illness specific reasons behind it (Atif et al., 2016; Y. Yang & Wang, 2016). It is also reported that nonpsychiatric issues such as cardiovascular and respiratory disorders, diabetes, and obesity have higher rates in patients with epilepsy compared to the normal population (Strine et al., 2005).

Epilepsy can lead to dangerous circumstances if the seizure happens at certain times. For example, it can lead to falling, drowning or car accidents. Two of the rare but life-threatening complications are status epilepticus and sudden unexpected death in epilepsy (SUDEP). Status epilepticus is a state of continuous seizure that lasts more than five minutes or a state of frequent recurrent seizures (Devinsky et al., 2018).

1.1.6 Temporal Lobe Epilepsy

The human brain has two hemispheres, and each hemisphere contains four lobes (*Brain Lobes - Mayo Clinic*, n.d.) (Figure 2.2):

- **Frontal lobe:** This part of the brain is responsible for cognitive functions, emotional regulation, planning, reasoning, problem-solving and control of voluntary movements.
- **Parietal lobe:** This part of the brain is responsible for integrating sensory information.
- **Temporal lobe:** This part of the brain is responsible for processing information about memory, receiving auditory information from the ears and analyzing complex visual information. The hippocampus is located in the middle of the temporal lobe and is responsible for learning and memory.
- **Occipital lobe:** This part of the brain is mainly responsible for vision-related functions.

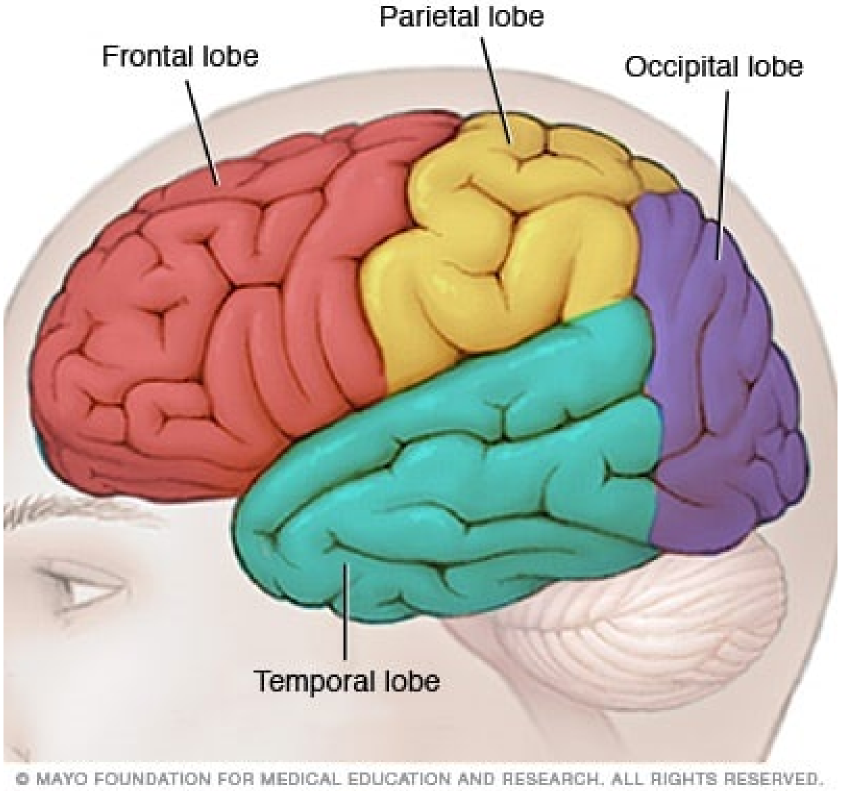

Frontal lobe
Parietal lobe
Occipital lobe
Temporal lobe
© MAYO FOUNDATION FOR MEDICAL EDUCATION AND RESEARCH. ALL RIGHTS RESERVED.

Figure 2.2: Brain lobes and their location. The image is borrowed from Mayo Clinic's website (*Brain Lobes - Mayo Clinic*, n.d.).

Temporal lobe epilepsy is the most common type of epilepsy when considering focal seizures. Seizures originating in the middle of the temporal lobe cause changes in behaviour, memory, and responsiveness. They also usually negatively impact cognition. Seizures starting in the temporal lobe may discharge beyond the hippocampus (Stafstrom & Carmant, 2015). Most of the conducted surgeries are on temporal lobe epilepsy cases.

1.1.7 Treatments

There are at least more than 20 Anti-seizure drugs (ASDs), also known as Antiepileptic Drugs (AEDs) or Antiseizure Medications (ASMs). The type and dosage of medication are decided by the physicians based on age, type of epilepsy, history, and other clinical findings related to the patient. Usually, the medication course starts with lower dosages, and if needed, the dosage is increased. Around 80% of patients experience adverse effects of the ASDs, and in 30% to 40% of the cases, the implications are more severe and may affect adherence to the medication (Marson et al., 2005). Approximately 50% of patients get to the state of seizure control after using the first drug. The second drug would help another 13% to achieve seizure control, and less than 4% will achieve seizure control after using the third drug (Kwan & Brodie, 2000). Usually, if the patients would not improve after using two types of medication, they would be considered for surgery—typically, up to 60-70% of patients who have surgery become seizure-free, depending on the type and site of operation.

To find the sections of the brain where the seizure starts, the following diagnostic tools are being used:

- Routine and sleep-deprived EEG studies
- Inpatient video-EEG and Invasive EEG (with surgically placed electrodes)
- MRI
- Positron emission tomography (PET). PET is a type of imaging.
- Single-photon emission computerized tomography (SPECT). This procedure measures the blood flow in different parts of the brain. Usually, parts which cause seizures have a higher blood flow.
- Magnetoencephalography or MEG scan is a medical test which measures magnetic fields produced in the brain.

Depending on the identified location and the area to be removed, there may be a need for an additional procedure such as fMRI and Wada test. Wada test helps with identifying which side of the brain controls language. It also identifies which side is more responsible for memory-related functions. Neuropsychological tests may also be recommended to measure learning skills and establish the baseline for cognitive functions.

Considering different factors, the best candidates would be selected for surgery by physicians. Surgery is mainly an option for focal epilepsy cases and rarely is considered for generalized epilepsy. Depending on the location of the start of the seizure in the brain, different types of surgery would be considered. The following are the types of surgeries:

a. Resective surgery: In this type of surgery, a small portion of the brain is removed, typically with the aim of rendering the patient completely seizure-free.
b. Laser interstitial thermal therapy (LITT): In this type of surgery, a laser is used to pinpoint and destroy a small section of the brain, again with the goal of complete seizure freedom.
c. Deep brain stimulation: In this type of surgery, a device is implanted in the brain to release electrical signals to prevent abnormal signals. This is usually considered to be a palliative procedure, with the goal of reducing the frequency of seizures.
d. Corpus Callosotomy: Corpus callosum connects the two brain hemispheres. In some cases, disrupting this connection would prevent the spread of seizures from one side of the brain to the other side.
e. Hemispherectomy. In this type of surgery, one side of the cerebral cortex is removed.
f. Functional hemispherectomy. In this type of surgery, connection nerves are removed, and it is mainly considered for children (*Epilepsy Surgery - Mayo Clinic*, n.d.)

The status of seizures is measured based on two scales; Engel Outcome Scale (Engel, 1993) and International League Against Epilepsy (ILAE) Outcome Scale (Wieser et al., 2001). The Engel Scale has three classes with subclasses, and ILAE Scale has six classes without subclasses. Using these classes and subclasses, the outcome of surgery can be described. For example, Engel class IA would be an indication of being completely seizure free since surgery, and ILAE class 1 is an indication of being seizure-free. Table 2.1 lists all the classes of both scales.

There are guidelines developed for managing patients with epilepsy, and they have specific sections for drug-resistant patients. For example, Figure 2.3 reflects a section of Provincial Guidelines for Epilepsy Surgery Referrals in Ontario (*Original Guidelines - Ontario Epilepsy Guidelines*, n.d.). As the flowchart indicates, once it is identified that the second ASM is not effective, the case is marked as Medically Refractory Epilepsy (MRE), and the patient is referred to District Epilepsy Centre DEC. In DEC, the patient would be admitted to an Epilepsy Monitoring Unit and based on all the gathered information, the team decides on the possibility of surgery and sends the patient to a Regional Epilepsy Surgery Centre (RESC). Regional Epilepsy Surgery Centres have a multidisciplinary team which includes:

- epileptologist
- technologist
- nurse
- neuropsychologist
- medical social worker
- clinical psychologist/psychiatrist
- community epilepsy liaison

- neurosurgeons

This team makes the final decision on surgery and proceeds based on patient consent. The composition of the team reflects the importance and complexity of the decisions that are made by the team.

Engel Outcome Scale	**ILAE Outcome Scale**
Class I: Free of disabling seizures • IA: Completely seizure-free since surgery • IB: Non disabling simple partial seizures only since surgery • IC: Some disabling seizures after surgery, but free of disabling seizures for at least 2 years • ID: Generalized convulsions with antiepileptic drug withdrawal only Class II: Rare disabling seizures ("almost seizure-free") • IIA: Initially free of disabling seizures but has rare seizures now • IIB: Rare disabling seizures since surgery • IIC: More than rare disabling seizures after surgery, but rare seizures for at least 2 years • IID: Nocturnal seizures only Class III: Worthwhile improvement • IIIA: Worthwhile seizure reduction • IIIB: Prolonged seizure-free intervals amounting to greater than half the follow-up period, but not less than 2 years 1. Class IV: No worthwhile improvement o IVA: Significant seizure reduction o IVB: No appreciable change o IVC: Seizures worse	• Class 1: Completely seizure free; no auras • Class 2: Only auras; no other seizures • Class 3: 1 to 3 seizure days per year; ± auras • Class 4: 4 seizure days per year to 50% reduction of baseline seizure days; ± auras • Class 5: Less than 50% reduction of baseline seizure days; ± auras • Class 6: More than 100% increase of baseline seizure days; ± auras

Table 2.1: Engle and ILAE Seizure Status Scales (Engel Surgical Outcome Scale – MGH Epilepsy Service, n.d.)

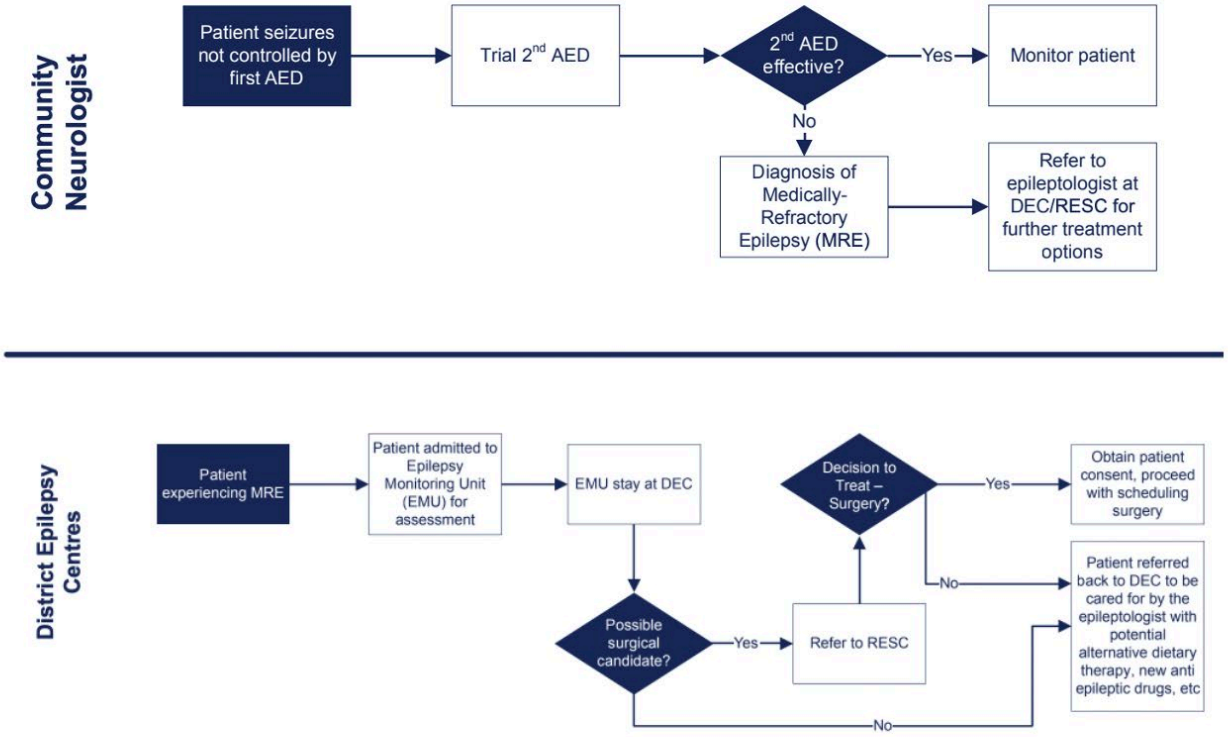

Community Neurologist
Patient seizures not controlled by first AED
Trial 2nd AED
2nd AED effective?
Yes
Monitor patient
No
Diagnosis of Medically-Refractory Epilepsy (MRE)
Refer to epileptologist at DEC/RESC for further treatment options
District Epilepsy Centres
Patient experiencing MRE
Patient admitted to Epilepsy Monitoring Unit (EMU) for assessment
EMU stay at DEC
Decision to Treat – Surgery?
Yes
Obtain patient consent, proceed with scheduling surgery
No
Patient referred back to DEC to be cared for by the epileptologist with potential alternative dietary therapy, new anti epileptic drugs, etc
Possible surgical candidate?
Yes
Refer to RESC
No

Figure 2.3: Ontario's guideline to manage drug resistance patients (Original Guidelines - Ontario Epilepsy Guidelines, n.d.).

One of the options considered for drug-resistant patients who are not candidates for resective surgery is neurostimulation. The Vagus Nerve Stimulator (VNS) is a neurostimulation device which is implanted in the body, and it has been shown that it reduces seizure frequency. As Figure 2.4 shows, in this method, a signal generator gets connected to the vagus nerve. More recent methods for neurostimulation have also shown promising results (R. Fisher et al., 2010; Milby et al., 2009; Ryvlin et al., 2018). Dietary therapies are also another treatment option. Different types of diets have shown seizure reduction or even, in some cases, seizure freedom (Cervenka et al., 2016; Lefevre & Aronson, 2000). Adherence to the diet can be challenging as it may have adverse effects such as constipation. The diets also may cause long-term complications such as nutritional deficiency or growth complications in children (Kang et al., 2004). Precision medicine is another avenue which is currently being explored and shows promising results.

Having targeted solutions based on genetic dysfunction can become a very effective choice in the future (Kearney et al., 2019).

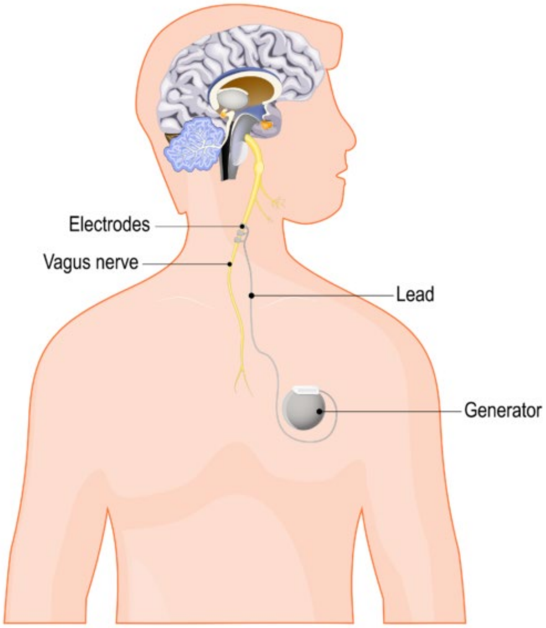


Figure 2.4: Vagus Nerve Stimulation (VSN) therapy. The signal generator sends signals to the vagus never. The image is borrowed from Epilepsy Queensland website (Vagus Nerve Stimulation (VNS) Therapy - Epilepsy Queensland, n.d.).

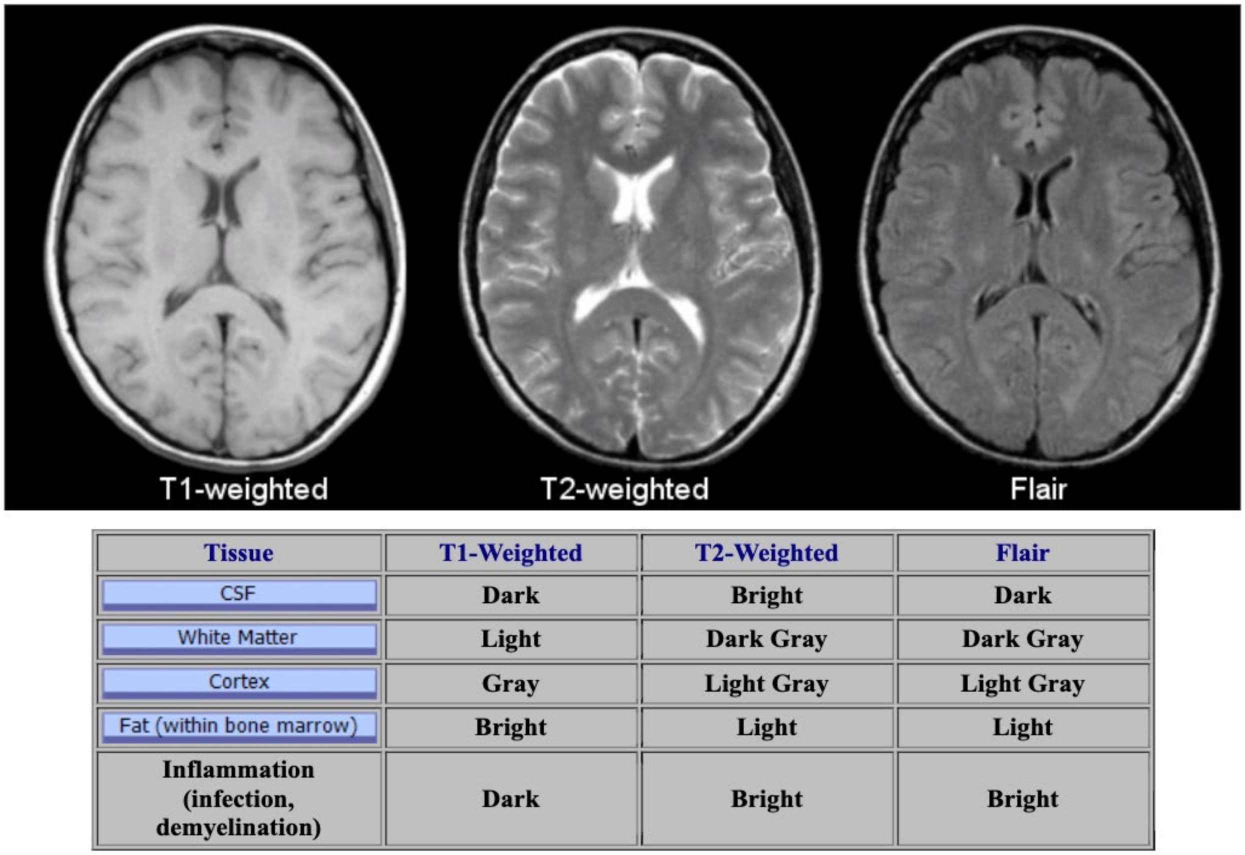
1.1.8 MRI and Epilepsy

Magnetic Resonance Imaging is an imaging technique which uses magnetic fields and radio frequency energy (electromagnetic waves) to produce images of different organs of the body. For epilepsy patients, the brain MRI of the patient would be investigated to find structural abnormalities.

Changing parameters related to the magnetic field and the radio frequency energy during the image acquisition process would result in different types of images called sequences. Three major types of sequences are T1-weighted, T2-weighted and Fluid Attenuated Inversion Recovery (FLAIR). Different tissues in the human body have different brightness in each MRI sequence (Figure 2.5). One of the main parameters in MRI is the strength of the magnetic field, and the unit of its measurement is Tesla. Usually, the MRI scans used for epilepsy are either 1.5T or 3T. The higher magnetic field results in a higher quality image which could be beneficial in finding abnormalities in MRIs.

The International League Against Epilepsy Neuroimaging Task Force consensus report published in 2019 recommends the following MRI formats to be examined when providing care for patients with epilepsy.

- Isotropic, millimetric 3D T1 images
- Isotropic, millimetric 3D FLAIR images
- High-resolution 2D submillimeter T2 images (Bernasconi et al., 2019)



Tissue	T1-Weighted	T2-Weighted	Flair
CSF	Dark	Bright	Dark
White Matter	Light	Dark Gray	Dark Gray
Cortex	Gray	Light Gray	Light Gray
Fat (within bone marrow)	Bright	Light	Light
Inflammation (infection, demyelination)	Dark	Bright	Bright

Figure 2.5: MRI of the brain in three different sequences. As it is listed in the table, different tissues look differently in each sequence. The image is borrowed from the MRI Basics website (MRI Basics, n.d.).

An example of the presence of lesional area displayed in slices of a FLAIR image is presented in Figure 2.6.

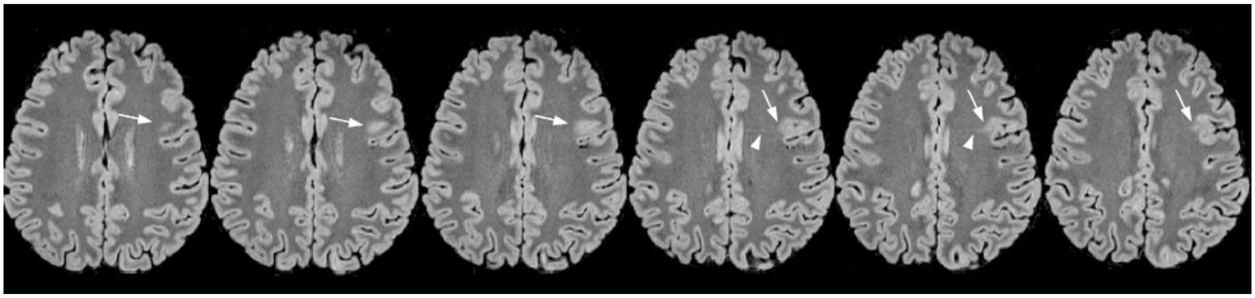


Figure 2.6: An example of the manifestation of epilepsy related lesion shown by the tip of arrows in six slices of a FLAIR image (Bernasconi et al., 2019).

MR images are usually stored in DICOM format, but scientists in the neuroimaging community use the Nifti format to analyze the images. The DICOM format consists of a set of 2D images (or files) which together can render 3D format, whereas the Nifti format is a single file which renders a 3D image.

1.2 Artificial Intelligence, Machine Learning and Deep Learning

In this section, an overview of technical concepts related to the projects of the thesis is provided. The main purpose is to cover basic concepts related to machine learning (ML) and Deep Learning (DL). Before starting to explore these concepts, it would be a good idea to mention how AI, ML and DL are related to each other. Conceptually, AI can be described as attempts to automate intellectual tasks performed by humans (Chollet, 2021). The field encompasses different methods to achieve such states. One example would be using rule-based methods to control traffic lights. ML is another set of tools in the realm of AI, and DL is a type of ML algorithm (Figure 2.7).

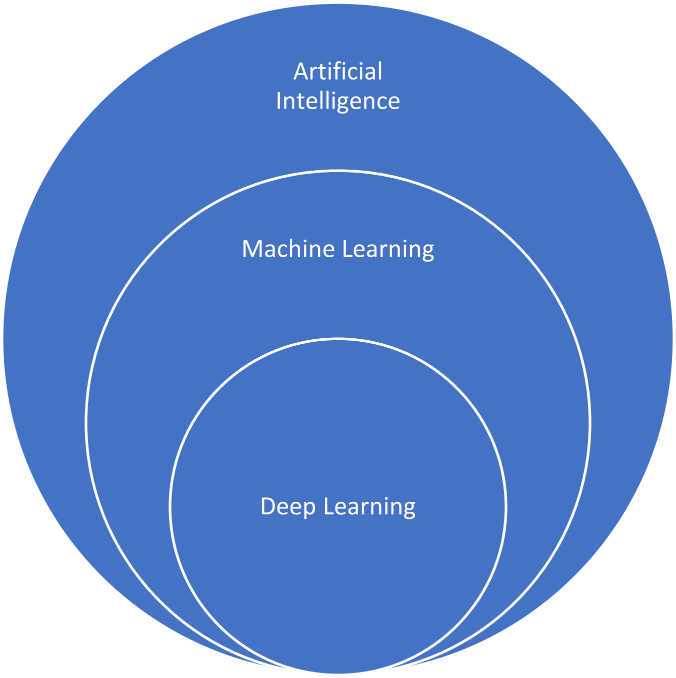


Figure 2.7: Machine learning is a set of tools that can be used when developing Artificial Intelligence solutions. Deep Learning is a subset of machine learning algorithms.

1.2.1 Machine Learning

Machine learning is a set of algorithms which analyzes data and desired outcomes to infer rules which can help to find answers for new data. For example, if you want to have a system which automatically tags images of dogs and cats, first you need to present many images of dogs and cats which are tagged by humans to a machine learning algorithm. The algorithm will find/learn patterns which it will use to automatically tag new pictures of dogs and cats. The learned rules are called a model. In the process of learning, the machine learning algorithm learns how to transform the data into a more useful representation of data which would result in the expected output. In other words, "all machine-learning algorithms consist of automatically finding such transformations that turn data into more useful representation for a given task" (Rosebrock, 2017).

Classical Machine Learning Algorithms

Excluding Deep Learning, machine learning algorithms generally can be grouped into four categories:

- Probabilistic modeling
- Early artificial neural networks
- Kernel methods
- Decision trees, random forests, and gradient boosting machines

Probabilistic Modeling

Probabilistic models use statistical principles for machine learning type tasks. They are the first generation of machine learning algorithms. An example of this type of model is Naïve Bayes classifier which is based on Bayes' theorem. Naïve Bayes has been used to analyze data even before computers. Another famous example of this type of model is logistic regression. Logistic regression is an algorithm which is vastly being used for classification and is usually one of the first choices to be tested for classification tasks.

Early Artificial Neural Networks

Early forms of artificial neural networks were developed in the 1950s and 1960s. Artificial Neural Networks are mainly designed for pattern recognition and classification. They were originally inspired by neurons in the brain and are building blocks of Deep Learning algorithms (Rosebrock, 2017). In the 1950s, Perceptron Networks, which are a famous type of neural networks, were introduced by Rosenblatt (Rosenblatt, 1958). The basic structure of the early models started with nodes, which resemble neurons (Figure 2.8). Inputs to a unit were added up and passed to a function to produce the output. The mentioned function is called the activation function, and in the early artificial neural networks, they were step functions. In the 1960s, it was shown that if the activation function is a linear function, the artificial neural network can only solve linear problems, and in order to solve nonlinear problems, the activation function needs to be a nonlinear function (Minsky & Papert, 1969). Further research in the following decades improved the structure of neural networks and training algorithms. These advances enabled the neural networks to be used as non-linear classifiers (Rumelhart et al., 1988; Werbos, 1974). A major breakthrough happened when in the mid-1980s backpropagation was introduced as a method to train the networks. This enabled the researchers to automatically train multilayer feedforward neural networks (Figure 2.9). Other types of artificial neural networks such as point-attractor networks were also started to be invented and researched in the 1970s and 1980s. For example, Hopfield networks in which all the nodes are connected to each other were introduced in that time (Hopfield, 1982) Figure (2.10).

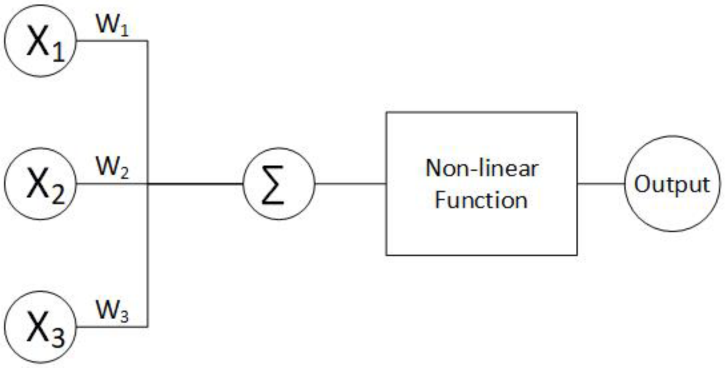


Figure 2.8. The simplest type of nodes (neurons) in artificial neural networks. The inputs are multiplied by a weight, and then all the inputs are added up and passed to a non-linear function.

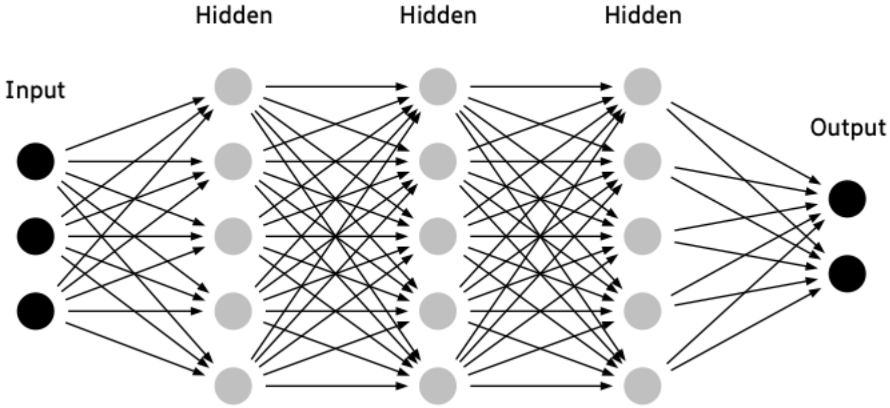


Figure 2.9: An example of a feedforward network. The image is produced by Dotnet program (GitHub - Martisak/Dotnets: Create Simple Drawings of Neural Networks Using Graphviz, n.d.*).*

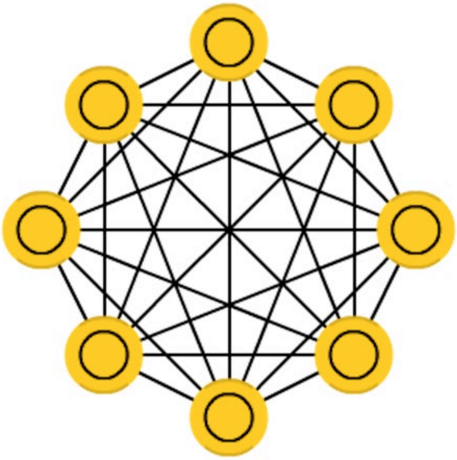


Figure 2.10: Presentation of a Hopfield network with eight nodes. All the nodes are connected to all the other nodes. The picture is from primo.ai (*Hopfield Network (HN) - PRIMO.Ai*, n.d.)

Kernel Methods

Transforming data points to their presentation in a higher dimension may make the classification task easier (Figure 2.11). This transformation is not an easy task, and kernel methods were introduced to find alternative solutions. These type of methods omit the calculation of the coordinates of the data points in the higher dimension and finds the hyperplanes as decision boundaries based on the distance of the data points in the new space. Support Vector Machine (SVM) is the most famous algorithm of Kernel methods, and its evolved version was developed in the 1990s at Bell Labs (Cortes & Vapnik, 1995).

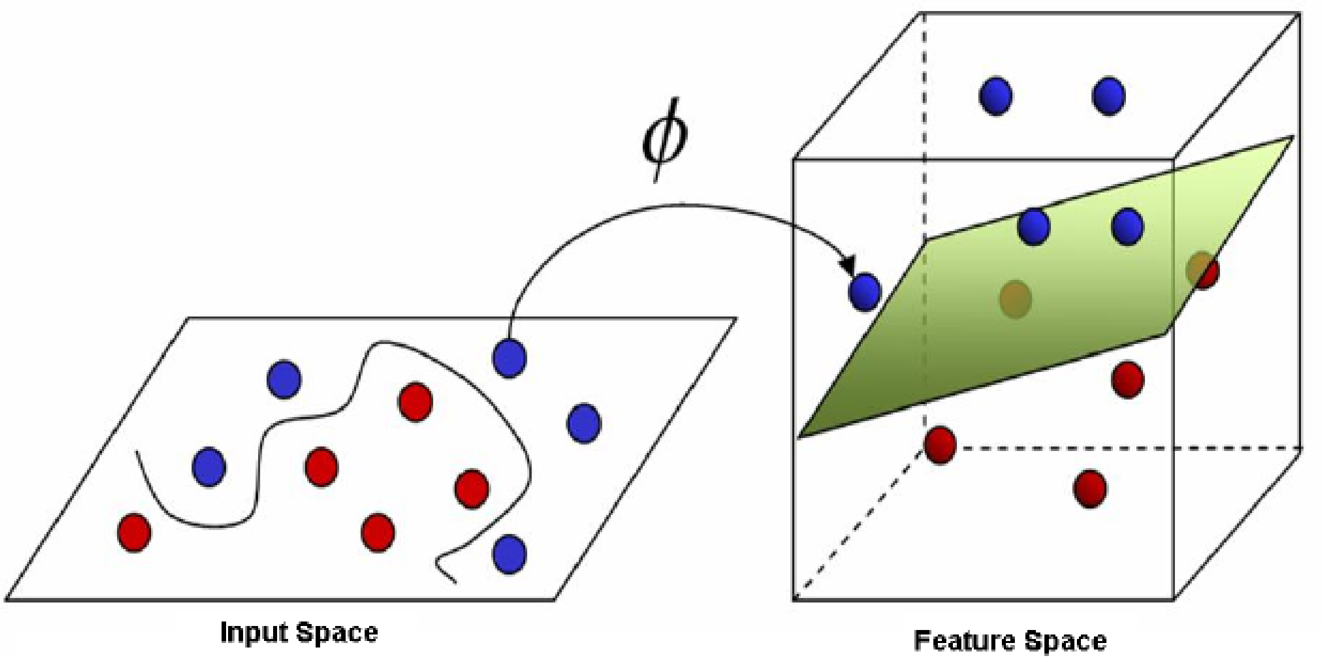


Figure 2.11: Presentation of data points in a higher dimension may make it easier to find decision boundaries. The image is borrowed from a Medium blog by Kundo (*Finding Non-Linear Decision Boundary in SVM | by Sourodip Kundu | Medium*, n.d.).

Decision trees, random forests, and gradient boosting machines

Decision trees are tree-like models. Each internal decision point presents a conditional statement which is applied to the input of that node to decide which branch the data belongs to. They are used to classify data or to predict based on input data (Figure 2.12).

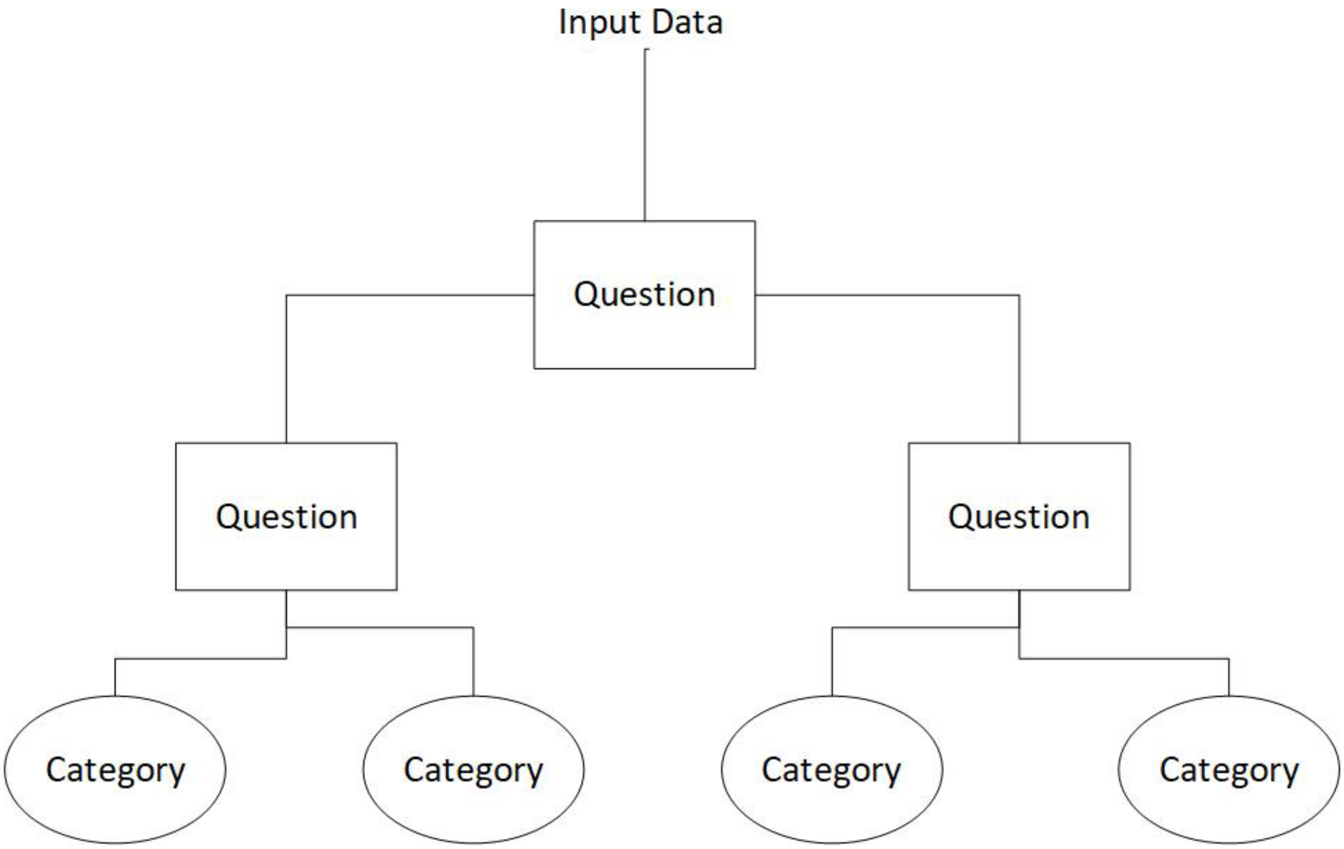


Figure 2.12: A basic presentation of a decision tree algorithm.

Random forest algorithms work by creating many specialized decision trees and ensembling their results. Gradient boosting machine is another group of machine learning algorithms, and similar to random forests, it ensembles the results of models. In these types of models, gradient boosting, which is a method to improve the result of learning models, is applied to weak models, generally decision trees, to improve their results. Basically, gradient boosting is an iterative algorithm which works as the following. First, it starts by fitting the first model to the data. Then the second model is fitted with the purpose of improving the performance of the first model. On the third iteration, another model is fitted with the purpose of improving the performance of the ensemble version of the first two models. This process is repeated in a fashion that each time the new model improves the performance of an ensemble of models of the previous iterations.

Finally, the ensemble of all the models is the outcome of the algorithm. In most cases, gradient boosting machines outperform decision trees and random forests.

Deep Learning has proven to be a very powerful tool for many machine learning related problems, specifically with perceptual data such as images, videos, and sounds. The reason that it has attracted attention is not only the fact that it outperforms other models in many cases but also its ability to automatically perform feature engineering and select the most appropriate features for the task at hand (Chollet, 2021). In the next section, Deep Learning for medical images is explored as it pertains to the main concepts used in this work.

1.2.2 Deep Learning and Medical Images

Background

Rule-based image processing systems were among the first artificial intelligence algorithms which have been used to analyze medical images (Haugeland, 1985). Applying computer vision algorithms which learn patterns was the next step in analyzing medical images. In these types of algorithms, the system is trained to recognize patterns based on handcrafted features from a chosen dataset. One of the ways to improve the performance of these types of algorithms is to conduct various methods to find the appropriate features. Given the challenges of finding relevant features, having a system which can identify the right features and learn them automatically would be very useful. As mentioned before, finding features which optimally represent the data is one of the main attributes of Deep Learning (DL) algorithms. There are different types of DL algorithms, such as CNNs, Recurrent Neural Networks (RNNs)(Graves, 2013) and Generative Adversarial Networks (GANs) (I. J. Goodfellow et al., 2014).

Convolutional Neural Network (CNN) is one of the major types of DL algorithms used extensively for analyzing medical images. CNNs have two sections. In the first part, the input goes through layers of convolutional filters. These filters extract appropriate features and pass them to the second section. In the second section, a neural network performs the machine learning task based on extracted features (Figure 2.13) (Rosebrock, 2017). CNNs consist of layers and each layer contains units which are called neurons. These units take the output from neurons of the previous layer, apply a mathematical function to them and then pass the results to the next layer. During the training phase, the mentioned functions would be defined. (Chollet, 2021).

In the lower convolutional filters of the first section (convolutional layers), simple patterns such as lines and edges are recognized, and as the number of filters grows, more sophisticated patterns such as complete shapes are recognized by the filters. The output of each convolution layer is called a feature map. The last feature map of the convolution layer has fewer data points compared to the number of pixels of the image. In Figure 2.14, the activations of convolution layers for corresponding images are visualized (Zeiler & Fergus, 2014). Based on the mentioned information, in a CNN which classifies cats and dogs images, the very first layers probably extract lines, corners and circles, whereas the few last layers of the convolution extract patterns like the shape of the face of the cat or dog. Another interesting point about the convolution layers is that it does not matter where the learned pattern would be expressed in the image. After learning a pattern in the middle of the image, the convolution layer can extract it from anywhere on the image.

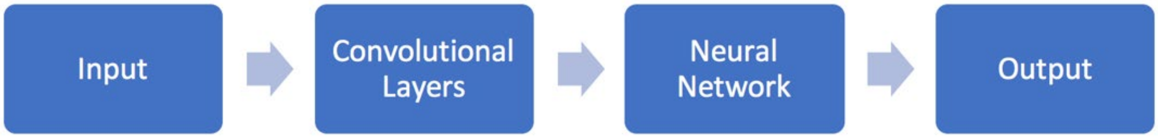


Figure 2.13: Input, output and two main sections of a CNN

Different algorithms have been designed to train the networks, and one of the famous and effective algorithms is stochastic gradient descent (Rosebrock, 2017). The architecture of a network is defined by different factors such as the number of layers, types of layers and how the neurons of each layer are connected to the next layer. In the last decade, the architecture of CNNs has evolved, improved, and changed to solve specific problems. Three types of CNNs architectures are listed in Table 2.2.

Research on CNNs started in the late seventies (Fukushima, 1980), and they were used to analyze medical images in the nineties (Lo et al., 1995). A seminal paper on CNNs was published at the end of the last century when they were used for classifying handwritten digits (LeCun et al., 1998). After that, for more than a decade, there were not many research works published on the subject. The CNNs introduced in the nineties needed powerful computational systems to calculate their parameters. Advances in core computational powers and methods which improve the algorithm itself have made CNNs more practical in the recent decade. In fact, the power of CNNs was not revealed until 2012, when technological advances and the use of Graphical Processing Units (GPUs) instead of Central Processing Units (CPUs) for DL algorithms allowed researchers to apply them to a large amount of data (Rosebrock, 2017). The tipping point was when Krizhevsky and his colleagues used DL to win the ImageNet competition in 2012 (Krizhevsky et al., 2012). After that, extensive work has been done to improve the architecture of DL algorithms. The network introduced by Krizhevsky is called AlexNet and is a stepping stone for the variety of algorithms which were designed in the last decade. The name of the network is based on the first name of Krizhevsky. Researchers who work on medical image analysis noticed the power of Deep Learning algorithms in 2013. Related work started appearing in workshops and conferences, followed by journal papers in 2013 and 2014. The number of publications related to applying CNNs to medical images grew rapidly in 2015 and 2016 (Litjens et al., 2017).

Type	Notes
Conventional Architecture	These networks have two main sections. One which selects the features and one which classifies (Krizhevsky et al., 2012).
Multi-stream Architecture	These networks are designed to accept multi sources of data and merge them at some point (Farabet et al., 2013).
Segmentation Architecture	The architecture of CNN has been changed to improve the efficiency for segmentation tasks (Ronneberger et al., 2015).

Table 2.3: Three types of architectures of CNNs.

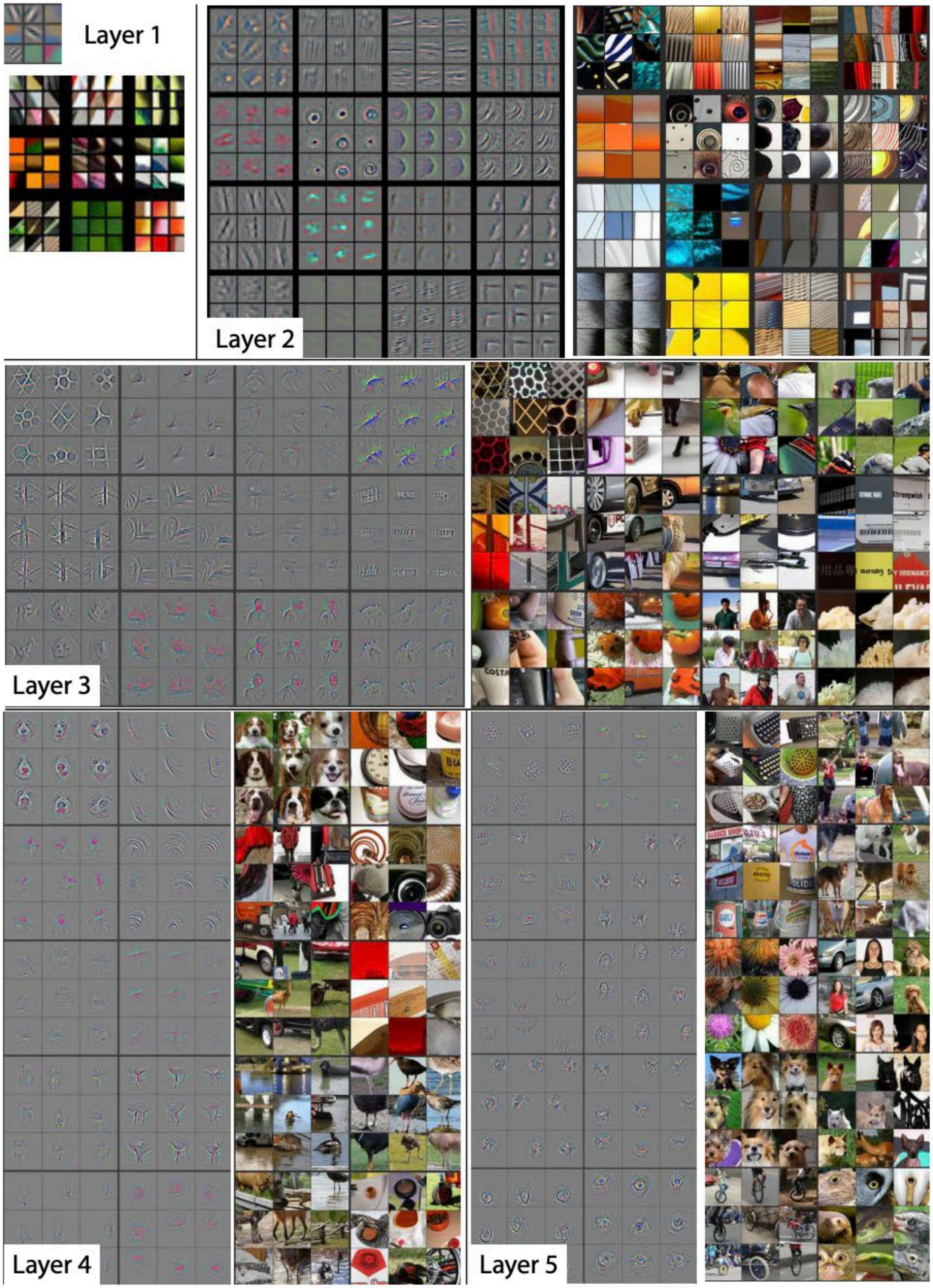


Figure 2.14: Visualization of features in convolution layers of a trained network (Zeiler & Fergus, 2014)

Applying DL on Medical Images

Using Clinical Decision Support Systems (CDSS) can improve the quality of care in a variety of ways. For example, they can bring standardization to healthcare which results in receiving the same quality of care regardless of the geographical location, or they can help with providing care with the most recent evidence given that the CDSSs knowledge can be based on the most recent evidence. They can address a lack of resources in rural areas or developing countries. DL algorithms can help design CDSSs, and there have been many related works published on this subject. For example, they have been used for skin lesion classification (Esteva, et al., 2017). DL algorithms have been used to analyze images of different anatomical areas of the body produced by different modalities. For example, they have been used to analyze images of the brain (Ghafoorian et al., 2016), retina (Zilly et al., 2017), chest (Rajpurkar et al., 2017), breast (Samala et al., 2016) and heart (C. Chen et al., 2019). A review conducted in 2017 on PubMed and ArXive databases has identified that 58 major projects which have used DL algorithms to analyze brain MRI have been published until 2017. The DL algorithms have been used for different purposes, such as classification, localization, segmentation, and registration (Table 2.3) (Litjens et al., 2017).

Task	Types
Classification	a. Image or exam classification – Helping to identify if a certain disease exists or it does not. Example: detecting Alzheimer's disease (Ebrahimighahnavieh et al., 2020) b. Object or lesion classification – Helping to classify different objects in the image. Example: Classification of skin cancer (Esteva et al., 2017)
Localization	a. Localization – Helping to identify the location of an organ or other objects. Localization can be used as a pre-processing step for other tasks, such as segmentation. For example, identifying the volume of interest (Wolterink et al., 2016). b. Object or lesion detection – Helping to find a lesion in an image. Example: Automatic detection of cerebral microbleeds from MRI (Dou et al., 2016)
Segmentation	a. Organ segmentation – providing information for quantitative analysis of the volume and shape of the organs. Example: Subcortical segmentation in MRI (Dolz et al., 2016) b. Lesion segmentation – Helping with both detecting the lesion and segmentation of the structures. Example: Brain tumor detection and segmentation (Dong et al., 2017)
Registration	a. Registration – Helping with the spatial alignment of different images – Example: Using Deep Learning for registering images from different modalities (Simonovsky et al., 2016)

Other	a. Survival prediction – Example: Survival Time prediction of brain tumor patients (Nie et al., 2016) b. Image enhancement – Example: Suppression of bony structures in chest radiographs (W. Yang et al., 2017). c. Etc.

Table 2.3: Deep Learning applications for medical images from a variety of modalities

Seminal Networks

Considering the focus of this work, it would be useful to mention the seminal networks and their background. These networks are considered to be the cornerstones of the current momentum in Deep Learning and classification research. Large image databases also have been instrumental in the advancements in Deep Learning. An example of the mentioned databases is ImageNet, with 14 million images. Since 2010 ImageNet has run an annual contest for image classification, which is called the Large Scale Visual Recognition Challenge (LSVRC) (Russakovsky et al., 2015). ImageNet challenges are usually the place of birth for the mentioned networks. AlexNet, GoogLeNet, VGG, and ResNet are among the few major networks which are cornerstones of many other networks' architecture. For the last few years, the winning algorithms of LSVRC have performed classification as well as humans do.

AlexNet, which was designed at the University of Toronto, won first place in LSVRC in 2012 and is considered to be the reason that the computer vision community started to focus on CNNs. AlexNet has eight layers; five layers in a convolutional section and three layers of a fully connected neural network for the classification (Krizhevsky et al., 2012). GoogLeNet acquired the first place in the 2014 ImageNet competition and had 22 layers. It was introduced by Szgedy et al. in 2014 (Szegedy et al., 2015) and resulted in a class of networks called inception models. VGG acquired second place in the 2014 ImageNet competition, and it was designed by the Visual Geometry Group from Oxford. It has two variations; a 16-layer network (VGG-16) and a 19-layer network (VGG-19) (Simonyan & Zisserman, 2015). Training is a resource-intensive function and usually needs many data points. As mentioned before, a method called transfer learning uses an already trained network for a specific task and fine-tunes it for another task with fewer datapoints (Chollet, 2021). Although VGG was originally trained for natural images, it has been deployed for medical images using transfer learning (Amit et al., 2017).

In 2014 and 2015, research groups were adding to the layers of the networks to increase the accuracy, but it was identified that training becomes challenging as the number of layers grows due to a problem called vanishing gradients. In order to resolve this problem, He et al. redesigned the classical architecture of CNNs and introduced residual blocks (He et al., 2016). Their network is named ResNet, and its variations have been used in many studies to analyze MRI (Farooq et al., 2017; Fulton et al., 2019; Korolev et al., 2017).

When reviewing the role that DL algorithms play in medical image research, it is worth noting the required and available hardware and software tools. As mentioned before, training DL algorithms are very resource-intensive task, and the lack of required computational power was one of the main reasons that there was no major activity regarding DL algorithms more than ten years after the DL works were introduced in the nineties. Training DL algorithms involves matrix-related calculations, and given that Graphical Processing Units (GPUs) are designed to deal with matrices, they can be better computational units for DL compared to Central Processing Units (CPUs). Since 2012 GPUs have been used for DL algorithms, and programming libraries have been developed to bridge between high-level programming languages and GPUs (Litjens et al., 2017; Rosebrock, 2017). For example, Figure 2.15 shows how Keras, which is a python library that provides a high-level interface for creating neural networks, communicates to an NVIDIA GPU. Keras itself uses either Tensorflow or Theano, which are both python libraries, to create and run neural networks. The two mentioned libraries use the NVIDIA CUDA Deep Neural Network library (cuDNN) to run the functions on GPUs. cuDNN is a library designed to implement neural network related calculations on GPUs.

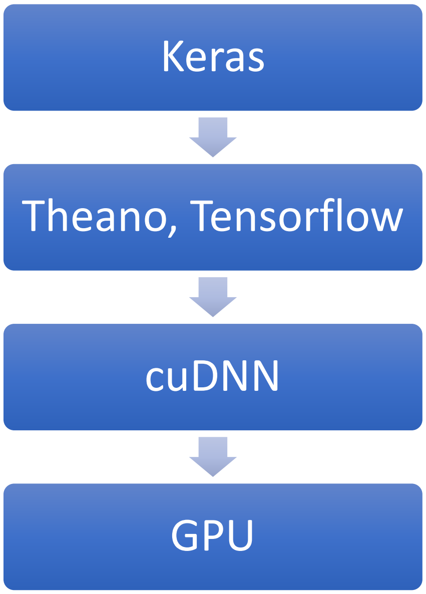


Figure 2.15: This picture shows how Keras communicates to GPU to run neural network related functions.

Different types of layers in the convolution section of the CNNs

Convolution Layers

There are different types of layers in the convolutional layers section of CNNs. The main type uses convolution filters (or kernels) to extra features. To gain a better understanding of these types of layers, it is worth visiting the convolution filter concept.

Convolution Filters

Convolution is a technique which can help with extracting specific features from an image. For example, in Figure 2.16, the result of applying different filters on two images has been depicted. In these examples, vertical edges and horizontal edges are extracted from the images. Any two-dimensional image is basically a matrix of pixel values. By applying a specific matrix to the image using the method which will be explained here, we can extract specific components of an image. Using this technique, we can also sharpen or smoothen the original image using the proper matrix. The matrix which is applied to the original matrix is called Kernel.

Applying Kernel

If we have a black-and-white image, the image can be presented as a two-dimensional matrix. For example, let us assume that the matrix shown in Figure 2.17a presents an image, and we want to apply the kernel presented in Figure 2.17b to be applied on it. The center point of the kernel is called the anchor, and in this example, the anchor point is 5. To apply the kernel to the image, we put the kernel on top of the image and calculate the value of the pixel, which overlaps with the kernel value in the fashion shown in Figure 2.17c. We multiply the values of a kernel which overlap with pixels and add them. The calculated value is the new value of the pixel, which overlaps with the anchor. We keep sliding the kernel over the image to calculate all the new values of pixels (Figure 2.17d). The process starts from the top-left corner, and the kernel slides; therefore, in the first step, the anchor point overlaps with the top-left pixel. When calculating the values of pixels which are located on the edges and corners of the image, some of the kernel elements do not overlap with the pixels, and that creates a challenge. There are a few methods to overcome this problem. In one of the methods, which is illustrated in figure 2.17e, we can just ignore the elements which are located outside of the image. CNN learns the values of the kernel matrix and other values related to this section during the training. The person who designs the network still needs to specify the number of filters and the size of the kernel. The design also dictates the sliding steps of the filter on the image. For 3D images, such as MRI, these filters are 3D as opposed to 2D (such as the example discussed above). There are also other types of convolutions layers, such 1D convolution layer, which its main purpose is to reduce the dimension of the data.

Pooling Layer

Although the convolution layers by themselves reduce the dimensionality, there are other types of layers which are used to downsample the number of datapoint in each layer. One of the mentioned types does not learn how to reduce the dimension using linear transformation but has a hardcoded method. For example, Max Pooling, which is a very common type of these kinds of layers, extracts the maximum number of fixed blocks of numbers and passes it to the next layer. These blocks usually consist of four numbers (Figure 2.18).

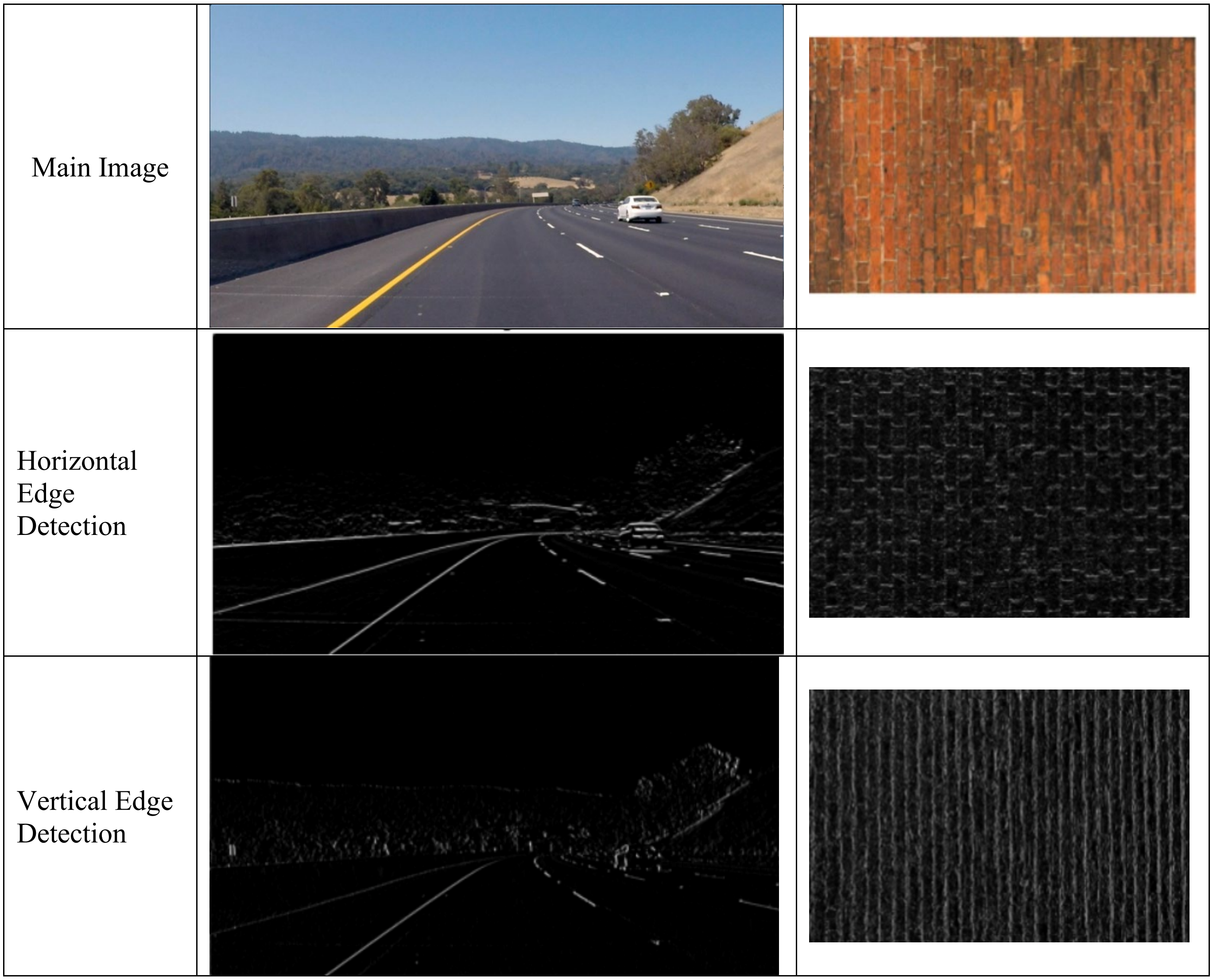


Figure 2.16: Applying filters to two images to extract horizontal and vertical edges. Source: (*Filters in Convolutional Neural Networks*, n.d.)

1	0	2	2	4	3
1	1	0	3	2	2
3	0	2	0	3	1
1	1	0	3	4	2
5	2	3	2	1	1

(a)

0	-1	0
-1	5	-1
0	-1	0

(b)

1*0	0*-1	2*0	2	4	3
1*-1	1*5	0*-1	3	2	2
3*0	0*-1	2*0	0	3	1
1	1	0	3	4	2
5	2	3	2	1	1

(1*0)+(0*-1)+(2*0)+(1*-1)+(1*5)+(0*-1)+(3*0)+(0*-1)+(2*0)=4

(c)

1	0*0	2*-1	2*0	4	3
1	1*-1	0*5	3*-1	2	2
3	0*0	2*-1	0*0	3	1
1	1	0	3	4	2
5	2	3	2	1	1

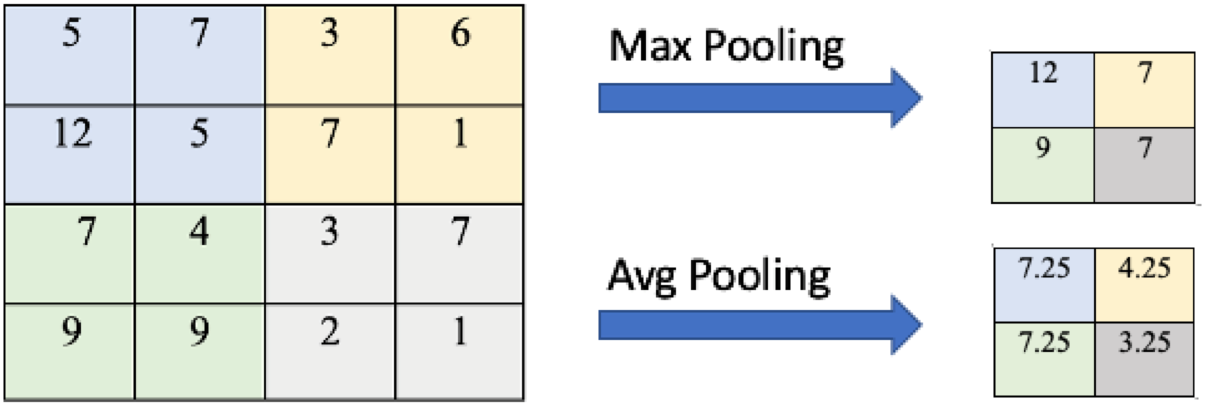
(d)

(1*5)+(1*0)+(1*-1)+(1*0)=4

0	-1	0				
-1	1*5	0*-1	2	2	4	3
0	1*-1	1*0	0	3	2	2
	3	0	2	0	3	1
	1	1	0	3	4	2
	5	2	3	2	1	1

(e)

Figure 2.17: showing how to apply a convolution filter on a matrix.



5	7	3	6
12	5	7	1
7	4	3	7
9	9	2	1

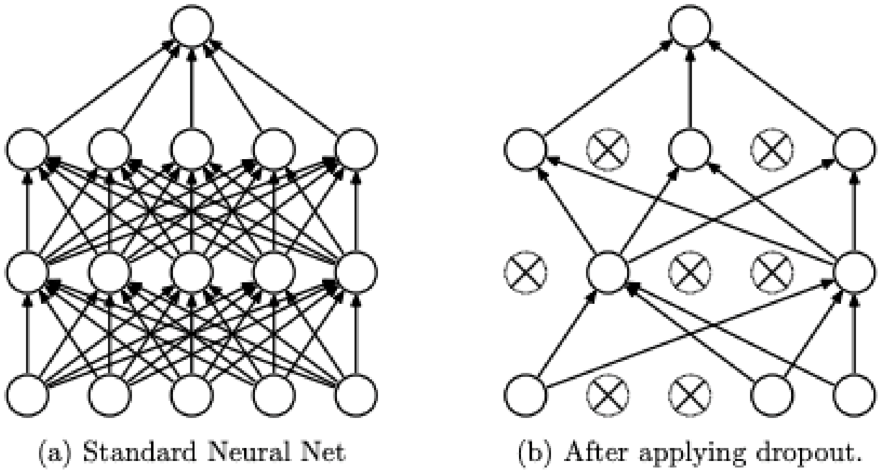
12	7
9	7

7.25	4.25
7.25	3.25

Figure 2.18: Applying Max Pooling and Average Pooling on a matrix.

Preventing Overfitting – Dropout

There are a few different methods to prevent overfitting in neural networks. Regularization is a well-known method in machine learning to prevent overfitting. Another method which is designed for the neural network is called dropout. What dropout does is to ignore a portion of the neurons of a layer during the learning phase. In other words, it randomly chooses some of the outputs of each layer and sets them to zero (Figure 2.19).



(a) Standard Neural Net

(b) After applying dropout.

Figure 2.19: Applying dropout to a neural network. Source: (Srivastava et al., 2014)

Training and Many Images

Generally, many images are required to train Deep Learning algorithms and not having access to a desirable number of data points is a well-known issue when training Deep Learning algorithms. Researchers are working on different methods to overcome this problem. A well-known method is using data augmentation techniques, such as introducing noise to existing images or flipping them. The mentioned techniques are usually used during most CNN training processes. Creating synthetic data or using transfer learning are the two other methods which can be deployed depending on the nature of the task at hand.

Generating Synthetic Data

Creating synthetic data for healthcare is an active area of research. Different algorithms are being designed and used to create synthetic medical data. There are research groups that are working on creating synthetic medical images using computer algorithms. The following are two of the algorithms being used to create synthetic medical data.

1. Generative Adversarial Networks

Generative Adversarial Networks (GANs) are a type of Deep Learning algorithm that can create new images which look like the training set that was provided. Using this algorithm, there are research works which have used GANs to create new data points to train CNNs (Sixt et al., 2016). GANs also have been used for medical image synthesis in order to increase the accuracy of networks which have been trained for a classification task (Yi et al., 2018). Figure 2.20 shows medical images generated with different GAN methods along with the original data. Using

GANs to create medical images may help researchers in the medical image community in the future.

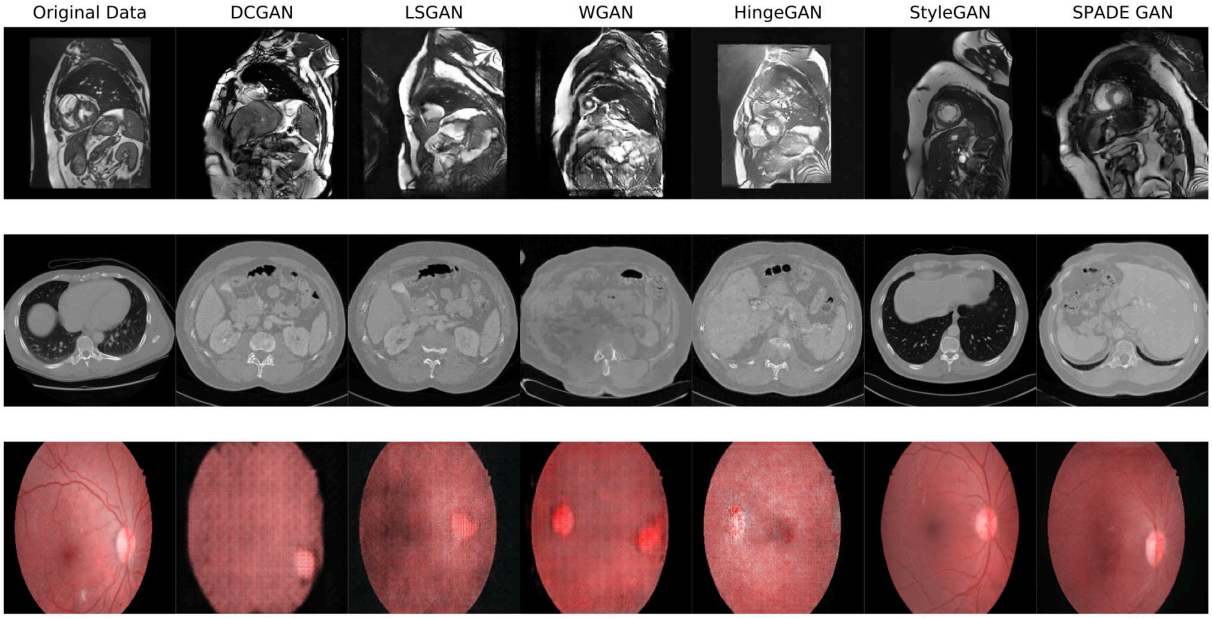


Figure 2.20: Examples of generated medical images using different types of GANs. The first column is real data. Source: (Skandarani et al., 2021)

2. Diffusion Models

Recently, generating natural images using text-to-image models such as DALL-E 2 has attracted attention. In these models, the computer generates an image based on described text (Figure 2.21). The emergence of such models shows that the future of image and video content will be affected by materials created by computer algorithms. DALLE-2 project used a diffusion model to render the images. Recently a research group has used a diffusion model to generate brain MRIs and created an openly available synthetic database with 100000 brain images (Figure 2.22) (Pinaya et al., 2022). This type of effort will enhance the available data for future research in medical image analysis.

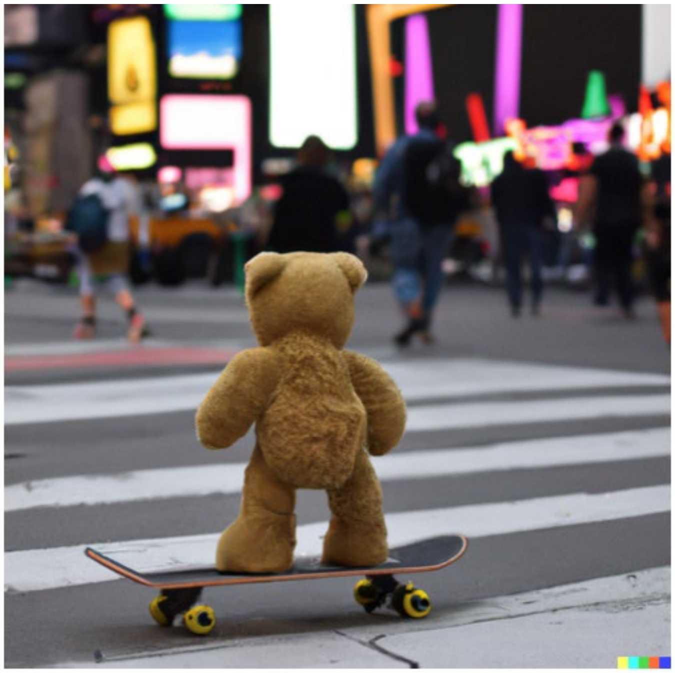


Figure 2.21: An image generated by a text-to-image model (DALLE 2) based on the following text: "a teddy bear on a skateboard in times square". Source: (Ramesh et al., 2022)

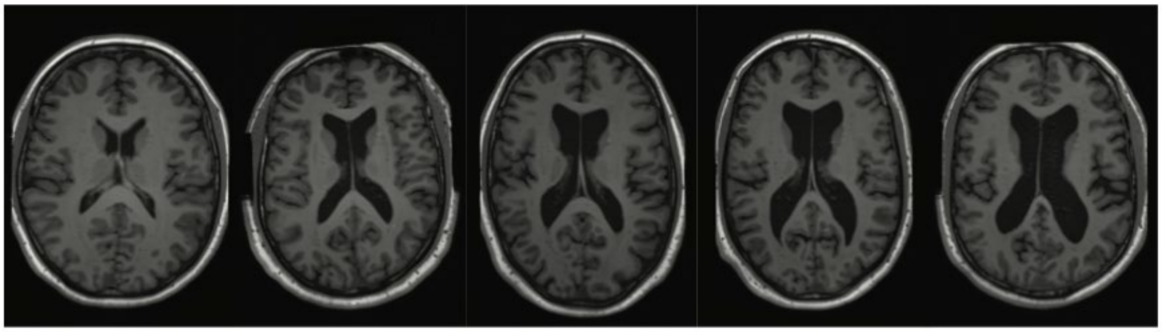


Figure 2.22: Synthetic MRI created by a Latent Diffusion Model. Source: (Pinaya et al., 2022)

Transfer Learning

A method called transfer learning uses an already trained network and fine-tunes it for another task (Chollet, 2021). This method has two phases. In the first phase, a network is trained on a large dataset, and then in the second phase, the trained network is retrained on a specific dataset for the target task. The first phase of training helps the network to learn general and less specific features that can be used on the target task. Transfer learning can result in requiring a lower number of images for the target task. One of the popular datasets that is being used for transfer learning is ImageNet. As it was mentioned before, ImageNet is an image database with 14 million images (Russakovsky et al., 2015). There are studies which have used transfer learning and ImageNet for medical images (Gulshan et al., 2016; Majkowska et al., 2020). Recent research published by Google has investigated works which have used networks trained on ImageNet and transfer learning for classifying medical images. It has been identified that transfer learning does not improve performance on medical image classification tasks with a large number of images, but it would shorten the convergence time. They also identified that the lower layers of the network (layers closer to input) have little change during the second phase of training (training for target task). This means that features learned in lower layers during the first phase of training are reused in the second phase (Raghu et al., 2019). The finding matches the conventional understanding of the patterns learned in lower layers. As the work of Zeiler and Fergus indicated, the lower levels learn simpler patterns, such as edges, while the higher layers learn more sophisticated patterns (Zeiler & Fergus, 2014). If the nature of images in the first phase and second phase are the same, more layers probably keep the learned patterns and reuse them in the second phase of training. This would reduce the number of images needed to train in the second phase. For example, with transfer learning and using a network which has been trained on ImageNet, the network can reach to around 90% accuracy with only three images. Figure 2.23 depicts the effect of the number of images used during transfer learning for the cat vs dog task on accuracy. It is important to note that one of the reasons for such a low number is that there are many cats and dogs images in the ImageNet dataset, but still, the example shows that transfer learning reduces the required number of data points. It also can be speculated that there may be an inverse correlation between relative heterogeneity between target images and the number of images required to learn to discern between them.

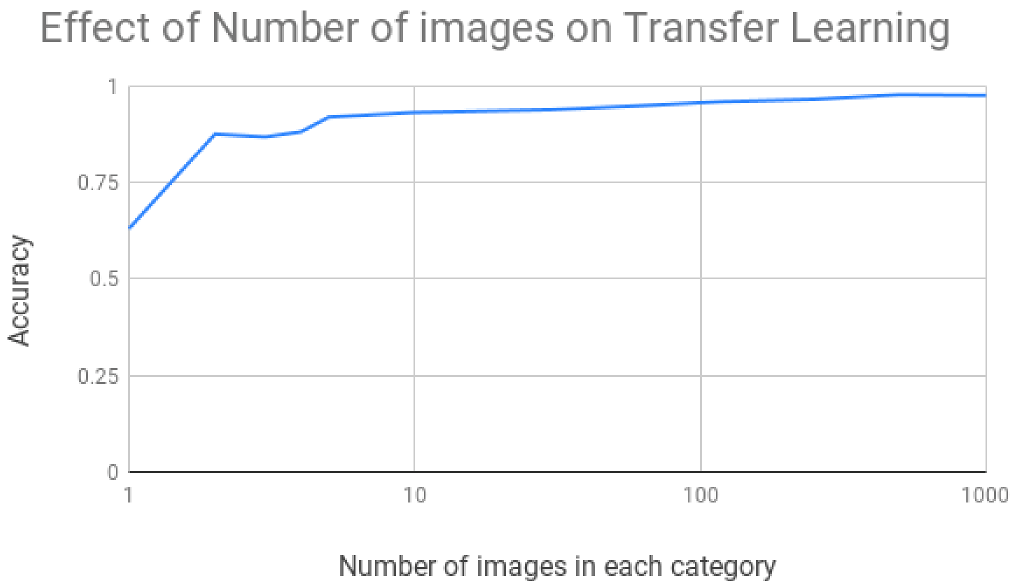


Figure 2.23 Effect of data size on accuracy in transfer learning (Koul et al., 2019)

2 Related Work

Machine Learning algorithms have been used to predict the outcome of neurosurgical surgeries. These algorithms use medical data or data from different diagnostic tools, such as MRI, to classify or predict. During the training phase, if the algorithm is provided with labelled data, it is considered to be a supervised algorithm. If the training data is not labelled, the algorithm is considered to be an unsupervised algorithm (Chollet, 2021). There are studies which have used supervised learning methods and features such as Clinical, Neuropsychological, EEG, PET, Pathology and MRI to predict the outcome of surgery for refractory epilepsy cases (Senders et al., 2018).

Naïve Bayes, Logistic Regression, K-Nearest Neighbours (K-NN) and Random Forest are four well-known classification algorithms which have been used in many different fields, including predicting neurosurgical outcomes. For example, in a study, clinical and neuropsychological features have been used to predict the outcome of temporal lobe surgery for patients with epilepsy, using Naïve Bayes, Logistic Regression and K-NN classifiers. The number of patients included in this study was 23, and leave-one-out cross-validation was used to evaluate the performance of each algorithm. 90% accuracy has been reported for each of the used algorithms (Armañanzas et al., 2013). In another study with 16 patients, Random Forest and FMZ images were used to predict the outcome with 87% accuracy (Yankam Njiwa et al., 2015).

Support Vector Machine (SVM) was introduced in the 1990s by Corinna Cortes and Vladimir Vapnik (Cortes & Vapnik, 1995) and since then has become very popular in solving different classification and regression problems. In a retrospective study of 20 patients who were diagnosed with mesial temporal epilepsy, SVM was used to predict the outcome of surgery with 95% accuracy. Features extracted from MRI, EEG, clinical data and demographics were used as input to the algorithm (Memarian et al., 2015). There are other studies which have used SVM and extracted features from MRI to predict the outcome of the surgery. For example, Munsell et al. used SVM and extracted features from MRI to predict the outcome of surgery with 70% accuracy (Munsell et al., 2015). In another study, Taylor et al. used SVM and reported 79% accuracy and 65% specificity when predicting the seizure outcome (Taylor et al., 2018).

Artificial Neural Networks (ANNs) are another class of Machine Learning algorithms that can be used for pattern recognition and classification. Neural Networks have been used to predict the outcome of surgery for patients with epilepsy. In a study, neural networks predicted the outcome for Engel class 1 with 81.8% accuracy and the outcome for Engel class 2 with 95.4% accuracy (Grigsby et al., 1998). In another similar study, features such as MRI evaluation, EEG evaluation, neuropsychological evaluation and seizure frequency were fed to the neural networks. The study has reported 98% accuracy (Arle et al., 1999).

2.1 MRI and Machine Learning for Predicting the Outcome

As it was mentioned before, MRI data has been used in two different ways to provide information for the machine learning classifiers in order to predict the outcome of surgery. In the first method, the interpretations of MRI by experts have been encoded and passed to the classifier (Arle et al., 1999), and in the second method, features from the images have been extracted from the MRI and provided to the classifier.

2.1.1 Morphological and Volumetric Features

Morphological and volumetric features of different regions of the brain extracted from the MRI of patients have been used by different research groups to predict the outcome of the surgery. For example, Bernhardt et al. have used the surface morphology of the hippocampus, amygdala, and entorhinal cortex to predict the outcome of surgery for temporal lobe epilepsy. To evaluate their classifier, they used an independent cohort of 27 patients from another MRI scanner and achieved 96% accuracy (Bernhardt et al., 2015). In another study, the white matter (WM) areas of the brain in the MRI of 49 patients with epilepsy who underwent surgery were segmented. Five patients within the group had non-lesional MRIs. After preprocessing and segmentation, the voxels values of WM were fed to an SVM to predict the outcome of surgery using leave-one-out cross-validation. In their study, they achieved 94% accuracy (Feis et al., 2013).

2.1.2 Connectome

The Connectome map of the brain can be produced by processing diffusion-weighted MRI and fMRI. The first one would identify the network connectivity of different regions of the brain, and the second one would show the network activities. DeSalvo et al. have used diffusion-weighted MRI and identified that there is a difference in the network connectivity of patients with left lobe temporal epilepsy and the healthy control group's network connectivity. Using graph theory analysis, they have found that distant connectivity between some regions of the brain of patients is increased compared to the control group, while there are regions in which their short connectivity has increased compared to healthy brains (DeSalvo et al., 2014). In another study by Liu et al., white matter, structural networks of mesial temporal sclerosis (mTLE) were compared to healthy controls, and it was identified that there are differences between the network connectivity of healthy brains and patient's brains (Liu et al., 2014). Considering different brain connectivity studies, it can be concluded that epilepsy is a disorder in a brain's network connectivity. More specifically, medication refractory focal epilepsy can be considered to be a result of focal lesional problems and network disorders (Tavakol et al., 2019).

The structure of the network connectivity can play a role in the success of surgery for drug-resistant epilepsy cases. A study by Bonilha et al. compared the connectome of three groups. The first group was patients with mesial temporal epilepsy who went under surgery and were seizure-free at least one year after the operation. The second group had surgery but were not seizure-free for over the first post-operative year, and the third group was a healthy control group. Their comparison identified that all three groups have different patterns in their connectivity of the networks (Bonilha et al., 2013). This work was followed by another study, which was led by Bonilha, and identified that the degree of difference between the network structure of the brain of patients with TLE and healthy brains could predict the outcome of anterior lobe lobectomy with 83% accuracy. Furthermore, by adding clinical data, the group has improved the accuracy to 88% (Bonilha et al., 2015). Taylor and colleagues have studied the effects of surgery on the structure of networks. They used T1-weighted MRI to identify preoperative and postoperative connectome networks. Analyzing the networks, they identified fifteen connections which would change during the surgery and can be used to predict the outcome of temporal lube surgery. Eight of these fifteen were connected to the ipsilateral temporal lobe (Taylor et al., 2018). In a research work by Gleichgerrcht and colleagues, Deep Learning has been applied to the connectome of the whole brain to predict the outcome of surgery of patients with mesial temporal lobe epilepsy. They retrospectively studied 50 patients, of which 36 became seizure-free, and 14 did not become seizure-free. Their model showed positive predictive value of 88 ± 7% and mean negative predictive value of 79 ± 8%. They also have identified that predicting only based on clinical data would yield an accuracy of less than 50% (Gleichgerrcht et al., 2018).

2.2 Non-lesional MRI

Advancements in scanning technologies for acquiring MRI have improved clinicians' ability to detect epileptogenic regions, which would help identify the area of surgery (Bernasconi et al., 2011). However, around 44% of patients' MRIs are considered non-lesional. Following an operation, histopathology applied to the resected tissue can be used to identify MRI-occult lesions, and the number of non-lesional cases drops to approximately 16% (Téllez-Zenteno et al., 2010). In many cases, retrospectively reviewing the MRI after gathering information by histopathology, the area of the lesion is detectable (Bien et al., 2009). When reviewing the studies which have worked on non-lesional MRI cases, the definition of non-lesional in that study should be noted. For example, in a study by Arya et al., MRI was marked as non-lesional if it was "completely normal or if there were only non-specific, non-localising findings that could not direct a surgical plan" (Arya et al., 2016).

Although surgery can help reduce the frequency and number of seizures in medication refractory non-lesional epilepsy, the chance of success is significantly lower compared to MRI-lesional cases (Chapman et al., 2005; Jayakar et al., 2008; Jeha et al., 2007; Téllez-Zenteno et al., 2010). The rate of success of surgery for non-lesional MRI cases has been investigated by different research groups. For example, in a study by Bien et al., 38% of 29 patients were reported to become seizure-free after one year (Bien et al., 2009). Another study showed that 44% of 101 children were seizure-free after two years (Jayakar et al., 2008). Zakaria and colleagues have reported 49% success for extratemporal cases (n=36), while another study has reported 42% of 43 patients with extratemporal epilepsy have become seizure-free after two years (See et al., 2013; Zakaria et al., 2012). It is worth noting that there are studies which showed much better results. For example, in a retrospective study in South Korea, 27 children (mean age of 7.8) with MRI-negative were followed up at least two years after surgery, with a range of 2.2-9 years and a mean of 4.3 years of follow-up. The results showed that 67% of surgeries had Engle Class 1 outcome (Seo et al., 2009). A systematic review and meta-analysis published in 2010 has identified that studies which have grouped both temporal and extratemporal in one category have reported 46% of success rate. The reported number for studies which focused only on temporal lobe cases was 51%, and the number of extratemporal was 35% (Téllez-Zenteno et al., 2010).

Given the importance of finding the best candidates for surgery in non-lesional MRI cases, identifying factors which would help predict the surgery can be very helpful. A study conducted by Arya et al. focused on identifying predictors for the surgical outcome of pediatric MRI-negative, drug-resistant cases. In this work, they have analyzed clinical data such as medication history and information from diagnostic tools to determine which factors have the most predictive value. They have concluded that the seizure frequency, age of onset of seizures, and the number of failed anti-epileptic drugs are the most predicting factors (Arya et al., 2016).

The presence of specific patterns in ictal onsets data produced by scalp EEG and intracranial EEG (iEEG) in MRI-negative extratemporal epilepsy cases have been indicated as predictors of favourable outcome of surgery (Noe et al., 2013; Park et al., 2002; Shi et al., 2017; Zakaria et al., 2012). Using intraoperative electrocorticography (ECoG) can also improve the chance of success for surgery (Grewal et al., 2019).

Data from Magnetoencephalography (MEG) also has been shown to be a predictor of the result of surgery. Ramachandrannair et al. have conducted a retrospective research study on 27 non-lesional MRIs of children who went under surgery. 36% of the patients became seizure-free after surgery, and all the patients in the mentioned group had a MEG cluster in the final resection area (Ramachandrannair et al., 2007).

Positron Emission Tomography (PET) is another diagnostic tool that can be helpful for finding appropriate candidates for surgery in non-lesional cases. In a study conducted by LoPinto-Khoury et al., two groups of patients with lesional MRI and non-lesional MRI who underwent anterior temporal lobectomy (ATL) were analyzed. The result of their work has shown that outcome of surgery for PET-positive/MRI-negative patients is similar to the result of surgery for cases with lesional MRI (Lopinto-Khoury et al., 2012).

In our work, we mainly have investigated the possibility of automatically extracting features from MRIs and using them to design tools for dealing with patients with epilepsy who are drug-resistant. Identifying the possibility of using such features to improve the outcome of surgery for non-lesional cases was a major part of our work.

3 Classifying an MRI as Normal or Abnormal Using CNNs (Project 1)

3.1 Introduction

Recent advances in machine learning and computer vision have enabled computers to classify images with near human performance or, in some cases, better than human performance (He et al., 2015, 2016; Hu et al., 2018). Deep Learning algorithms, specifically Convolutional Neural Networks (CNNs), have played a key role in enabling computers to perform such tasks. The medical image analysis community took notice of such advances around 2015, and since then, the mentioned algorithms have been used to analyze medical images. Using Deep Learning algorithms to analyze and classify MRIs of patients with epilepsy may help physicians when planning surgical treatment and may improve patient outcomes.

MRIs of patients with epilepsy can be marked as normal or abnormal depending on the reading from the radiologist and neurologist. If the clinicians find abnormalities related to epilepsy on the MRI, it would be marked as abnormal, and if no lesion related to epilepsy is found on the MRI, it would be marked as normal. This first step in classifying MRIs may help focus investigations and surgical options. A decision support tool which can mark an MRI as normal or abnormal (lesional and non-lesional) can prevent physicians from missing an abnormality. It also can help physicians in training and improve the quality of care in areas where access to resources is limited.

Considering the potential benefits of such a tool, we decided to train a CNN which distinguishes between normal and abnormal MRIs. The prerequisites of designing and building the engine, which can be the core of the decision support tool, are the following: access to marked MRI, proper software and hardware tools, and appropriate algorithms. We had access to the marked MRIs through the Calgary Comprehensive Epilepsy Centre. Training CNNs require GPUs, and we were granted access to Nvidia GPUs through the University of Calgary's Advanced Research Computing cluster. In the following, we have provided a quick review of the evolution of different CNN architectures and the architect of CNN that I used.

Convolutional Neural Networks Architectures:

A variety of CNN's architectures have been designed and researched. Each architecture has its own design variables. These variables consist of the number of layers, the type of layers, how they are connected to each other, how they are configured, and the order of the layers. Choices made for these variables impact the performance of the networks. Figure 4.1 shows the architecture of the VGG-16 network, which was referred to in chapter 2. In the figure, you will notice three terms which were not explained before in this document. ReLU is the activation function being used for each layer, and a fully connected network is referred to as a traditional feedforward neural network which is used for classification purposes after the convolutional layers have passed the features to it. The last layer, which is a Softmax layer, helps with identifying the probability that the network assigns to each class regarding the presented image.

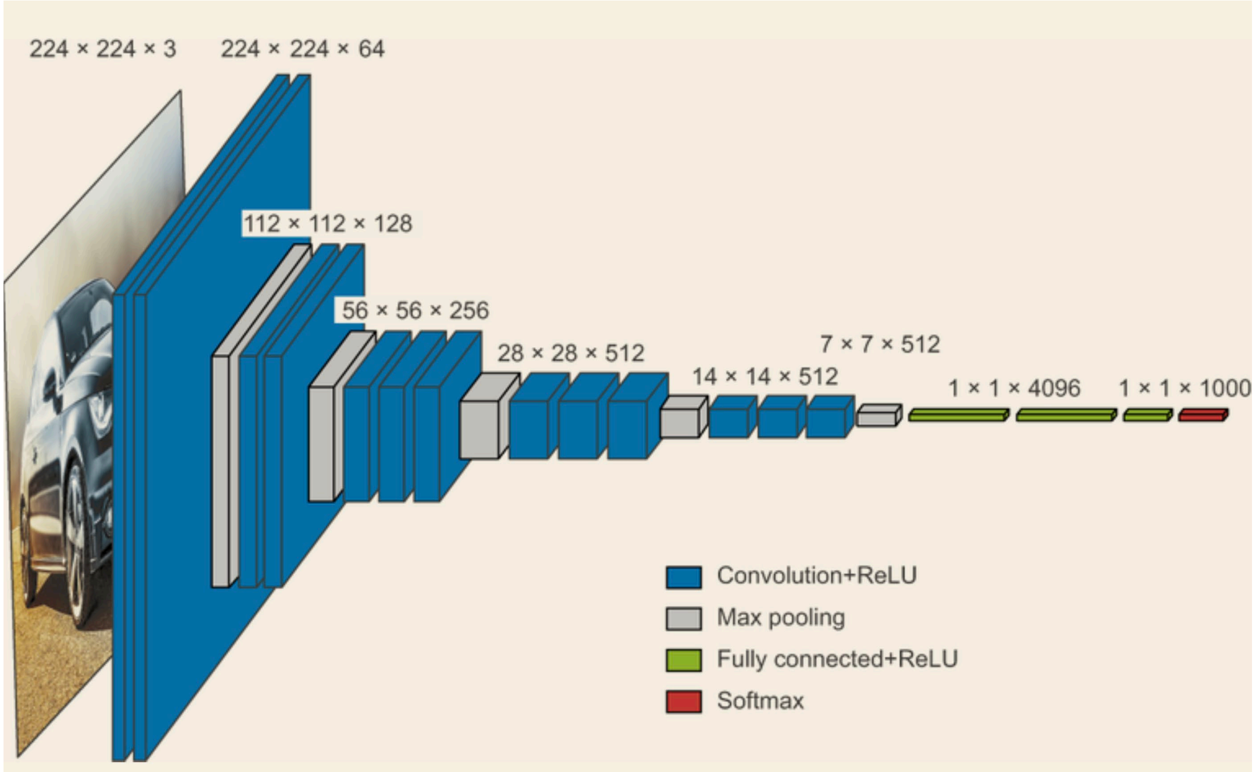
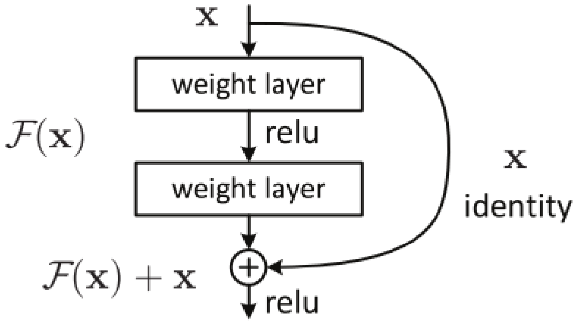


Figure 4.1: Architecture of VGG-16, trained to classify images to 1000 classes. Source: (Chollet, 2021)

In general, increasing the number of layers results in better classification models, as the last layers can extract more abstract concepts. However, as researchers added to the layers, it was identified that there is a limit to how deep a network can be. As the number of layers increases, the network encounters the vanishing gradient problem during the training phase. This problem stems from how the backpropagation algorithm works. When the number of layers passes a certain limit, the successive gradient procedure on each layer results in disrupting the main goal of reducing the error of prediction or classification. Basically, during this process, the main signal gets lost as the number of layers increases. In order to solve this problem, the residual block was introduced by researchers at Microsoft (He et al., 2016). The residual block adds the input to a layer or a few layers to the output of the block, and this results in resolving the vanishing gradient problem. Figure 4.2 shows the residual block as it was presented in the mentioned seminal paper.



x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
x
weight layer
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 4.2: The design of the residual block, which solves the vanishing gradient problem. Source: (He et al., 2016)

In the mentioned seminal paper, they implemented and compared a few different networks on the same database. The results show that an 18-layer network outperforms a 34-layer network due to the mentioned vanishing gradient problem, but when they restructured the networks using the residual block, the results changed. The 34-layer network with residual blocks outperformed all the other networks (plain 18-layer, 18-layer with residual blocks, and plain 34-layer). Figure 4.3 shows the architecture of a plain 34-layer network and a 34-layer network with residual blocks. In the paper, they also implemented a network with 152 layers and showed that it outperformed everything else. They have named this structure ResNet and added the number of layers as a suffix (For example, ResNet-34).

Following the success of ResNet architecture, Dense Convolutional Neural Networks (DenseNet) were introduced. In this architecture, the feature map of each layer is passed to the input of all the other layers ahead of it. Figure 4.4 shows an abstract depiction of such a network architecture. The DensNet structure eliminates the vanishing gradient problem while reinforcing feature propagation and reuse. It reduces the number of parameters which are used during training (Huang et al., 2017). At the time that this work was published, DensNet showed improvement in performance compared to other available architectures.

3.2 Methodology and Results

As was mentioned before, the goal of this project was to train a network which differentiates between lesional and non-lesional MRIs. Initially, we used the ResNet architecture for the network. A pipeline which included loading the MRIs, training the network, and reporting the performance against validation and test set was implemented using TensorFlow. TensorFlow is a library which is developed by Google. After that, we decided to test DenseNet architecture performance and, this time, used the PyTorch library to implement the mentioned pipeline (instead of using TensorFlow). PyTorch is developed by Meta (Facebook) and is an open source Deep Learning framework. Noticing that the DensNet structure performs better, we chose DensNet to run experiments and find the best solution for this problem. Eventually, we used a DenseNet-121 structure as the main network structure. It has been shown that DensNet uses fewer number of parameters while outperforming ResNet (Huang et al., 2017). Appendix A gives more information about the convolution layers of the network that we used.

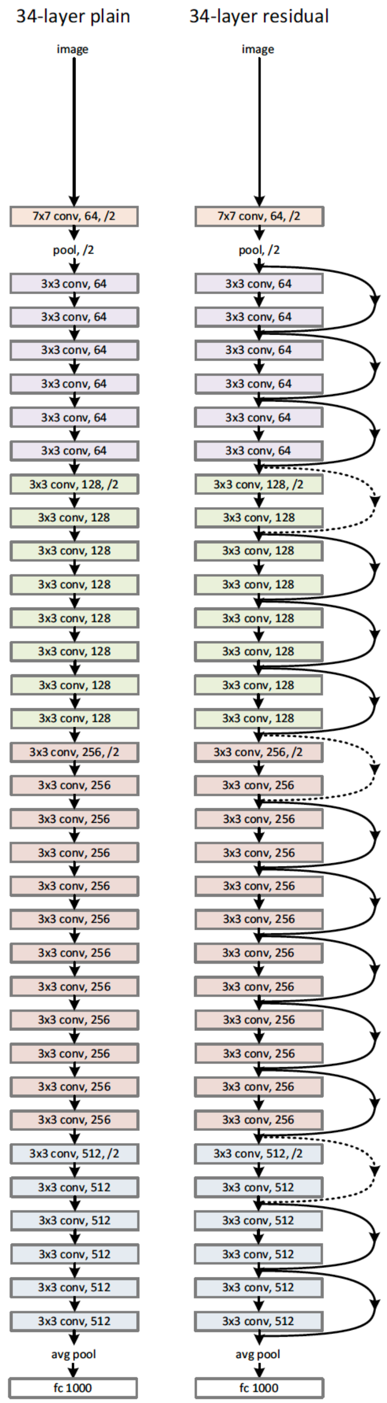


Figure 4.3: Architecture of a plain 34-layer network compared to the ResNet 34-layer. Source: (He et al., 2016)

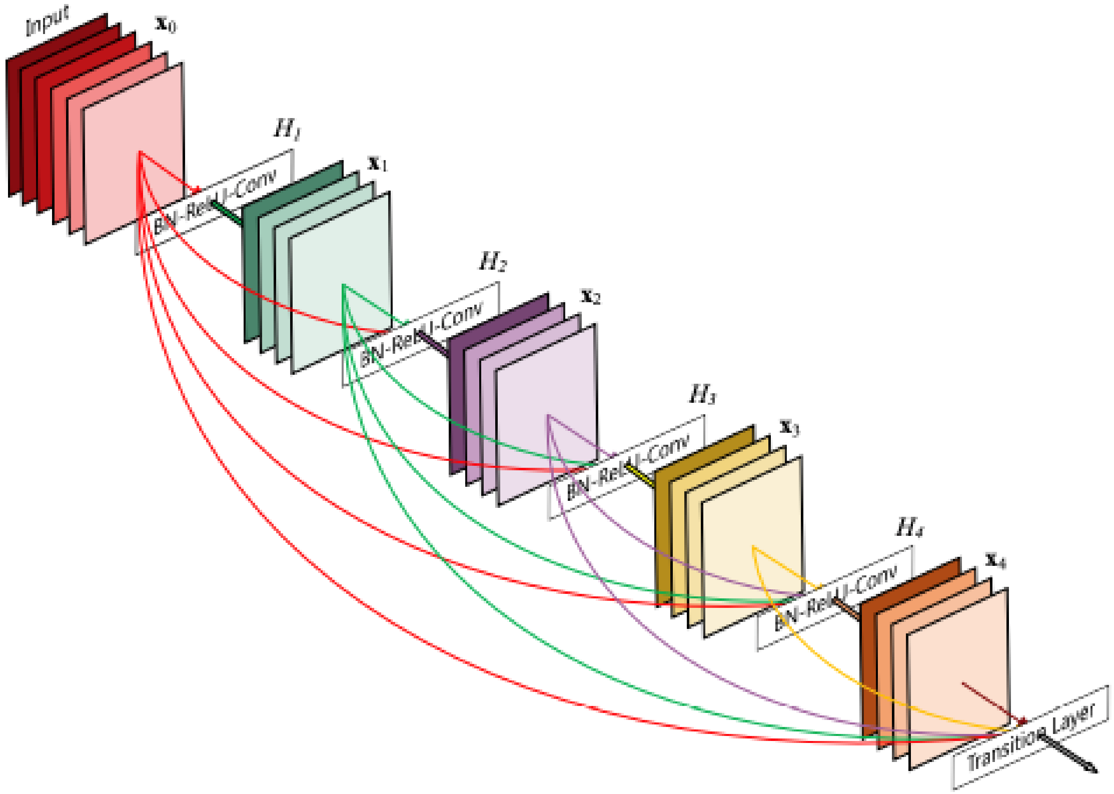


Figure 4.4: An example of a Densely Connected Convolutional Network (DensNet). Source: (Huang et al., 2017)

We had access to 399 MRIs of patients with epilepsy. 168 MRI had epilepsy-related abnormalities reported for them, and the rest were non-lesional (Figure 4.5). We obtained the images from Calgary Comprehensive Epilepsy Program and, more specifically, from “Federico Lab” (*Federico Epilepsy Neuroimaging | Home | Cumming School of Medicine | University of Calgary*, n.d.). The average age of recruited patients was 41.3 ranging from 16 to 88. Table 4.1 and Figure 4.6 depict more information about the demographics of each group. We divided the data into three groups of training, validating and test sets. We allocated 79 images for the test set, and it included 34 lesional cases and 45 non-lesional MRIs. The validation set had 40 MRIs which included 20 for each class. That would leave 115 non-lesional MRIs and 165 lesional MRIs for the training set. Deep Learning networks are trained in batches, and due to hardware limitations and the size of MRIs, each batch could only have 2 MRIs in them. In order for the network to see both classes in each round, each batch contained one item of both classes. To have this setting, we had to balance the training data. We balanced the training set by duplicating 50 of the lesional MRIs. These images were chosen randomly.

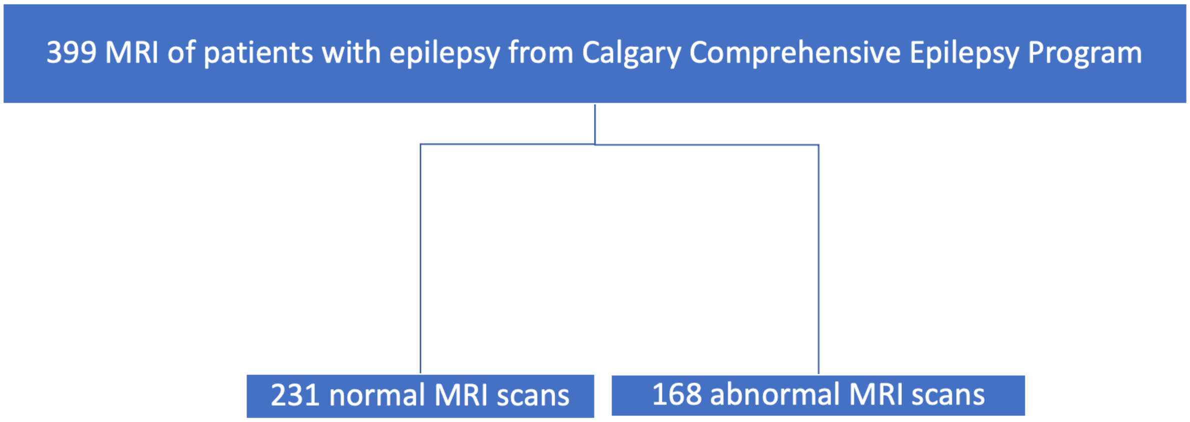

399 MRI of patients with epilepsy from Calgary Comprehensive Epilepsy Program
231 normal MRI scans
168 abnormal MRI scans

Figure 4.5: Information about the data

	Age	**Sex**

Non-lesional	Avg = 40.6
	Median = 37
	IQR = 17.75
		57% female

Lesional	Avg = 42.3
	Median = 39
	IQR = 20.75
		51% female

Table 4.1: Demographics of the cases.

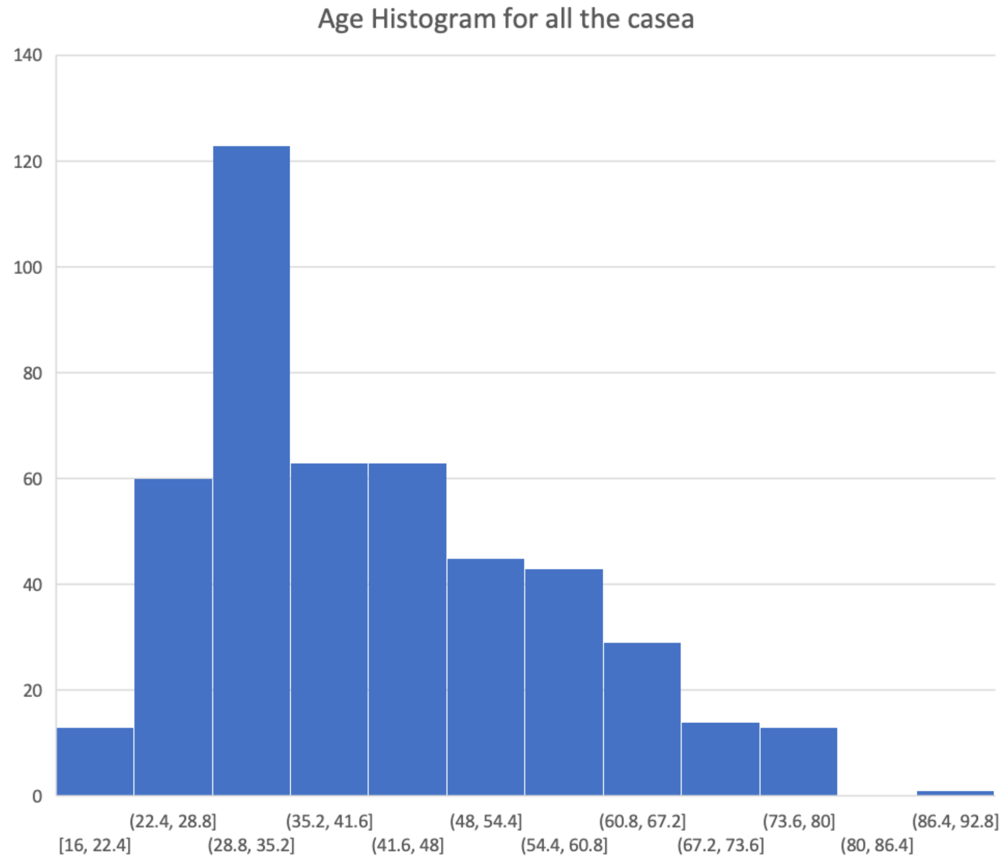


Figure 4.6: Age histogram for all the cases.

Data preparation:

MRI data is 3D in nature and can be presented in a three-dimensional tensor. To train the network and test it, all the data should have the same dimensions. We had access to T1-weighted, T2-weighted and FLAIR MRIs, and testing identified that only the networks which are trained on FLAIR would work on this task. MRIs were in DICOM format and were converted to NIFTI format. The main reason for converting DICOM to NIFTI was that the libraries that we used in our research work with NIFTI format. The dimension of the FLAIR MRIs was mainly 512, 512, 48 (for x, y, and z, respectively). All the MRIs had 512 as the x and y dimensions, but the z dimension was not constant for all the samples. We had two choices to create a fixed size for all the MRIs:

1. Downsample all the MRIs to the lowest resolution
2. Interpolate all the samples to the most common z dimension

Option 1 results in losing data, and as a result, we chose option number 2. We also normalized the scaled intensity of all the data points. Figure 4.7 shows the steps of data preparation. The learning rate for the network was set to $1*10^{-5}$, and the network was trained for 200 epochs. The architecture of the network is depicted in Appendix A, and the chosen loss function was the Cross-Entropy loss function. For the training, we used Nvidia's Tesla V100-PCIE GPU with 16 GB of GPU RAM. With this specific hardware, training the network with the mentioned hyperparameters took around 24 hours. After training, when we ran the model against the test set, it achieved F1 score of 0.68 (95% CI, 0.58-0.78) and accuracy of 0.71(95% CI, 0.61-0.81). It is worth mentioning that the Mathews Correlation Coefficient (MCC) was 0.40. Details of the results are presented in Table 4.2. The mentioned table shows the results based on precision and recall, which are equal to Positive Predictive Value (PPV) and sensitivity, respectively. In the following, a review of how to calculate F1 and MCC is provided.

The F1 score is calculated using the following formula:

2 * (precision * recall)/(precision + recall)

Basically, the F1 score is the harmonic mean of precision and recall.

MCC is defined based on the following:

((TP * TN) – (FP * FN)) / (sqrt((TP + FP)(TP + FN)(TN + FP)(TN + FN)) where

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

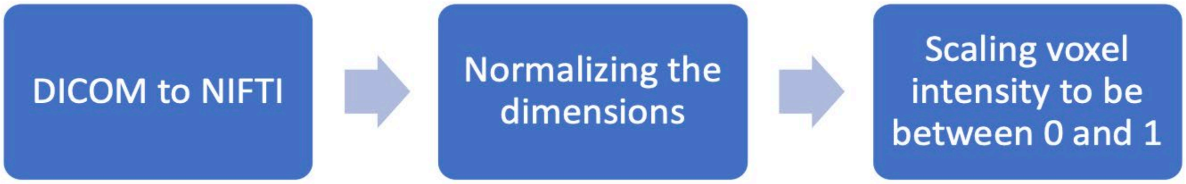

DICOM to NIFTI
Normalizing the dimensions
Scaling voxel intensity to be between 0 and 1

Figure 4.7: The steps to prepare the MRI data.

	Precision or Positive Predictive Value (PPV)	Recall or Sensitivity	F1-score
Lesional (n = 34)	0.76	0.47	0.58
Non-lesional (n=45)	0.69	0.89	0.78
Avg	0.73	0.68	0.68

Accuracy: 0.71(95% CI, 0.61-0.81)

Table 4.2: Result of DenseNet-121 network for classifying lesional vs non-lesional MRIs.

3.3 Conclusion

Different diagnostic tools are being used in the process of examining patients with epilepsy. These tools are used to plan the course of treatment. One of the main diagnostic tools is MRI and clinicians use it to find epilepsy-related abnormalities in the brain for focal epilepsy cases. Finding these abnormalities is not a trivial task, and there are recommendations developed by ILAE on how to find the abnormalities and which sequences of MRI to look at. Having access to clinical decision support tools which identify if there is any abnormality found by the machine may prevent clinicians from missing finding an abnormality. These types of tools, if trained with data tagged by world-class clinicians, would be helpful in training settings and in areas which deal with a lack of resources.

In our experiments, we have trained a Deep Learning algorithm to differentiate between MRIs with abnormality and MRIs without any abnormality. Several architectures and different hyperparameters were tested to achieve a good solution. Our experiments indicate that the networks can extract the signal in these MRIs. It has to be mentioned that given the existence of different types of epilepsy cases in the dataset, dividing the dataset into the groups of different types of epilepsy cases and then training the network for each group probably would result in better models, but our main challenge was lack of enough data to conduct the mentioned experiments. In general, it has to be stated that we had a very limited number of images for such a hard classification task, and this is the most important limitation of our work. Initially, we started the work with even fewer data and considered using space registration (MNI 305) so that voxel locations between images correspond to each other. But as the work progressed, we acquired more data and progressed, as it is explained in the chapter.

Another practical, exciting project that could be considered as follow-up work in the future would be training a network which discriminates between MRIs of patients who have abnormality reported in their histopathology report and MRIs in which no abnormality is reported in their MRI or histopathology report. In a study, physicians were asked to re-examine MRIs which were initially reported as non-lesional after the histopathology report indicated that there was a lesion in the brain (Bien et al., 2009). When they reevaluation the MRIs, physicians found an abnormality in some of the MRIs. This suggests that after not finding any abnormality in MRI, if a model indicates that there is a lesion in the brain, it may encourage the clinician to reevaluate the MRI.

This type of work can help when designing and developing other decision support systems related to epilepsy. For example, it can be helpful when designing a system which segments MRI for Focal Cortical Dysplasia (FCD). Another example is to use it for systems which can identify the type of abnormality. The type of work presented in this chapter can filter the MRIs which have abnormality and then send them to models that identify the type of abnormality. The improved and enhanced version of the model may help clinicians to choose which sequence of MRI to look at to find the abnormality.

The performance of classification models such as the one presented in this chapter is generally improved with more data, and as a result, we speculate that training a model with more data would probably improve the performance of the model. On the other hand, it seems the model, which was trained in this chapter, has a preponderance to over-call MRIs as non-lesional. It can be speculated that this is the result of more non-lesional cases in the training set. Although we have tried to balance the dataset by randomly duplicating some of the lesional cases, the model may be overfitting in favor of non-lesional cases. More data could lead to a more balanced dataset which would help reduce this propensity in the future. Another step which could bring a more rigorous evaluation would be using leave-one-out for training and evaluation. Given that training these types of models are very time-consuming, we did not test the leave-one-out method, but further research can perform such tests.

Data from different epilepsy centres should be considered before a general model, which would be a candidate to be used in a clinical setting, is designed and developed. One option would be to combine the data from different centres and consequently train and test the model. Another option is to train models in each centre and then test the models against the test set data from where the train set was used. For centres with fewer data, transfer learning concepts might be deployed. Federated learning is also an important option that must be considered.

Data from other sources, such as EEG and clinical notes, can be used to improve the results of the models discussed in this chapter. All the mentioned data can be fed to one classification model, or different models, which use each of these data separately, can be trained and then the results can be ensembled.

4 Predicting Result of Surgery (Project 2)

4.1 Introduction

Around 0.5 to 1.0% of the world population suffers from epilepsy (Fiest et al., 2017; Theodore et al., 2006; Wiebe et al., 1999), of whom approximately 30% to 40% are drug-resistant (Kwan & Brodie, 2000) and may be candidates for surgery. Randomized controlled trials of epilepsy surgery have shown sustained seizure freedom rates following temporal lobe operations in adults range from 58% to 73% with surgery, compared to 0% to 8% with medications alone (Engel et al., 2012; Wiebe et al., 2001; Willard et al., 2022) and in children 37% with surgery and 0% with medications (Dwivedi et al., 2017) A systematic review and meta-analysis of long-term outcomes indicates that seizure freedom for various types of surgery is maintained, although it declines over time (Téllez-Zenteno et al., 2005). Around 44% of patient's MRIs are considered non-lesional, and in such cases, the success of surgery (variously defined as seizure freedom, Engel 1 or Engel 1a) is reported to be between 35% to 51% (Bien et al., 2009; Chapman et al., 2005; Jayakar et al., 2008; Jeha et al., 2007; Téllez-Zenteno et al., 2010; Zakaria et al., 2012).

Predicting the surgical outcome is of paramount importance. Traditionally, predictive models have applied statistical models to clinical, EEG and imaging data to identify predictors of surgical success (Cossu et al., 2008; Jeha et al., 2007; Tonini et al., 2004). However, with the advent of machine learning, novel analytic approaches have used a variety of algorithms, such as K-Nearest Neighbours, Support Vector Machines, Neural Networks and Deep Learning. These algorithms have used features from Magnetic Resonance Image (MRI), electroencephalogram (EEG), intracranial-EEG, clinical data, or neuropsychological data as input (Arle et al., 1999;

Armañanzas et al., 2013; Grigsby et al., 1998; Memarian et al., 2015; Munsell et al., 2015; Senders et al., 2018; Taylor et al., 2018; Yankam Njiwa et al., 2015).

Clinical MRI data have been used in two different ways in machine learning to predict surgical outcome. One method involves encoding the clinical interpretation of the MRI provided by experts, which is then passed on to the classifier (Arle et al., 1999). The other method involves using features extracted from the MRI and providing these to the classifier. Typically, the features involve morphological, volumetric (Bernhardt et al., 2015; Feis et al., 2013) and connectome data (Gleichgerrcht et al., 2018). Deciding on which MRI features optimally represent the data is challenging, and here Deep Learning (DL) algorithms, such as Convolutional Neural Networks (CNNs), can offer significant advantages. Our work explores the application of CNNs to structural MRI data to predict surgical outcomes without manually extracting morphological, volumetric or connectome related features. Specifically, we hypothesize that CNNs can extract features which would help to find signals to predict the result of surgery.

4.2 Methods

Participants

We studied consecutive patients from the Calgary Comprehensive Epilepsy Program with focal drug-resistant epilepsy who underwent surgery, agreed to participate in the study, and we had access to their T1-weighted MRI imaging before surgery. We acquired the images from Federico Lab (*Federico Epilepsy Neuroimaging | Home | Cumming School of Medicine | University of Calgary*, n.d.), and the scan dates were from 2011 to 2019. Clinical and demographic characteristics were obtained, and surgical outcome at the last follow-up was dichotomized as

Engel class I and class II to IV (Engel, 1993). GE Discovery MR 750 - 3 Tesla, 3D T1-weighted MRIs were used for analysis. Considering all the mentioned factors, we could include seventy-five patients in our study. The clinical and demographic characteristics of the patients are presented in Table 5.1. Engel I outcome was achieved in 34 patients during a median follow-up of 23 months (IQR = 36). MRI abnormalities were reported in 56 patients, while 19 were reported as non-lesional. Among lesional cases, 22 were reported to have Engel I outcome during median follow-up of 24 months (IQR = 34.25). Among non-lesional cases, the latter 12 achieved Engel Class I outcomes after a median follow-up of 27.5 months (IQR = 39.5). In addition, we had access to 3D T1-weighted MRIs of 387 patients with epilepsy who did not have surgery but agreed to participate in the study. These MRIs had the exact same specifications as the 75 MRIs of the operated patients and were acquired using the same imaging equipment. Figure 5.1 summarizes the information about the patients and the outcome of surgeries.

4.3 Deep Learning and Image Classification

CNNs are one of the major DL algorithms which have been used extensively for medical image classifications (Litjens et al., 2017). As it was mentioned in the previous chapters, CNNs have two sections. In the first part, the input goes through layers of convolutional filters, and these filters extract appropriate features and pass them to the second section. In the second section, there is a neural network which learns the features and performs the classification task (Rosebrock, 2017).

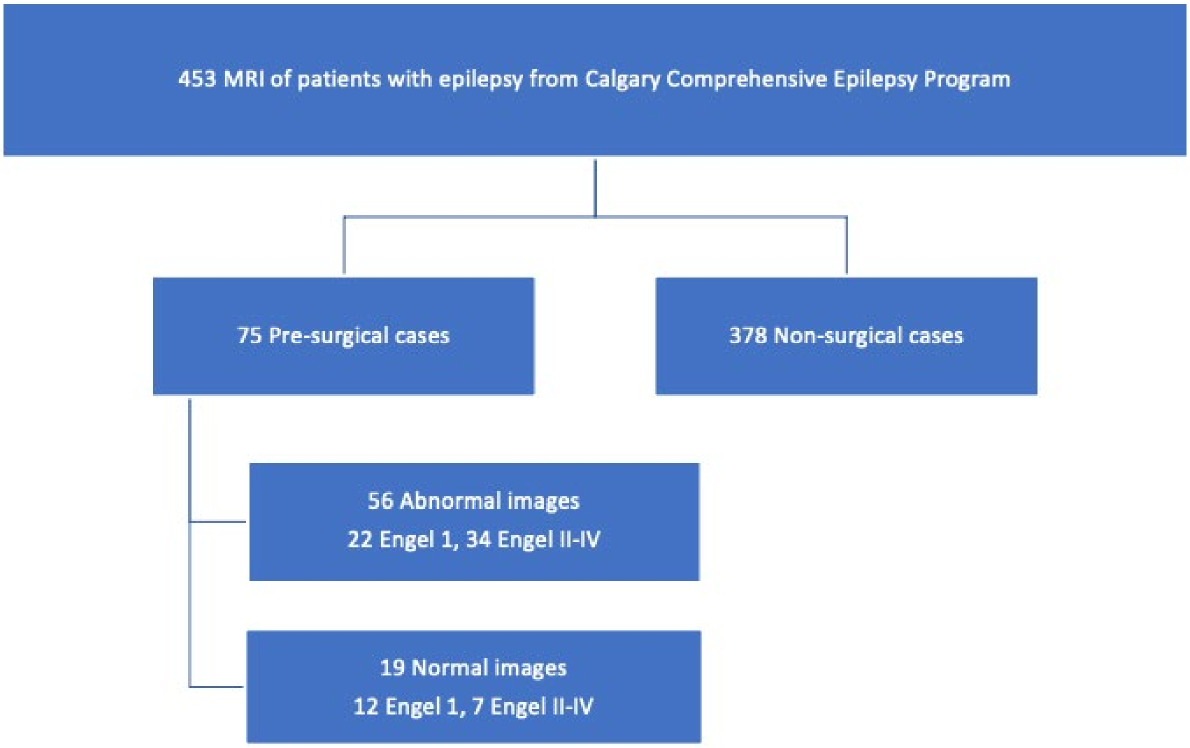

453 MRI of patients with epilepsy from Calgary Comprehensive Epilepsy Program
75 Pre-surgical cases
378 Non-surgical cases
56 Abnormal images
22 Engel 1, 34 Engel II-IV
19 Normal images
12 Engel 1, 7 Engel II-IV

Figure 5.1 Information about each group of patients and the result of surgery in each group.

Variable	**Patients (n=75)**	**Abnormal MRI Cases (n=56)**	**Normal MRI Cases (n=19)**
PSex (female) n	49% 37	50% 28	47% 9
Age at surgery Mean (SD)	35.8 (13.5)	37.6 (13.4)	29.3 (12.3)
Age at onset Mean (SD)	16.3 (12)	17.1 (12)	13.6 (12)
Hemisphere of seizure focus Number (percentage)	Left:33 (44%) Right: 36 (48 %) Bilateral: 6 (8%)	Left:29 (51.7%) Right:22 (39.2%) Bilateral:5 (8.9%)	Left:4 (21%) Right:14(74%) Bilateral:1 (5%)
Seizure focus Number (percentage)	Temporal: 44 (58.7%) Frontal: 9 (12%) Parietal: 2 (2.7%) Insular: 1 (1.3%) Multi-focal/Hemispheric/Regional: 19 (25.3%)	Temporal: 38 (67.9%) Frontal: 5 (8.9%) Parietal:2 (3.6%) Insular:1 (1.8%) Multi-focal: 10 (17.9%)	Temporal: 6 (31.6%) Frontal:4 (21.1%) Parietal:0 Insular:0 Multi-focal: 9 (47.4%)

Table 5.1: Demographics and Clinical Characteristics of 75 patients to whom we had access to their preoperative MRI.

First, we trained a CNN network to determine if preoperative MRI data could predict surgical outcome (dichotomized Engel classification) using a Densenet-121 architecture (Huang et al., 2017). As expected, given our relatively small sample size, it was not possible to train a CNN. We, therefore, decided to use the concept of transfer learning while changing the second section of the CNN to another classifier. Transfer learning uses an already trained network and fine-tunes it for another task (Chollet, 2021), and this may result in requiring a smaller number of images for the target task (Koul et al., 2019). In this method, first, a network is trained for an initial task, and in the second step, the learning parameters of the trained network will be used for the target task (Figure 5.2). During the training of a CNN, the lower convolutional layers (layers closer to input) learn simpler patterns, such as edges, while the higher layers learn more sophisticated patterns (Zeiler & Fergus, 2014). It has been shown that in transfer learning, the lower layers of the network have little change during the second phase of training (training for target task). This means that features learned in lower layers during the first phase of training are being reused in the second phase (Raghu et al., 2019).

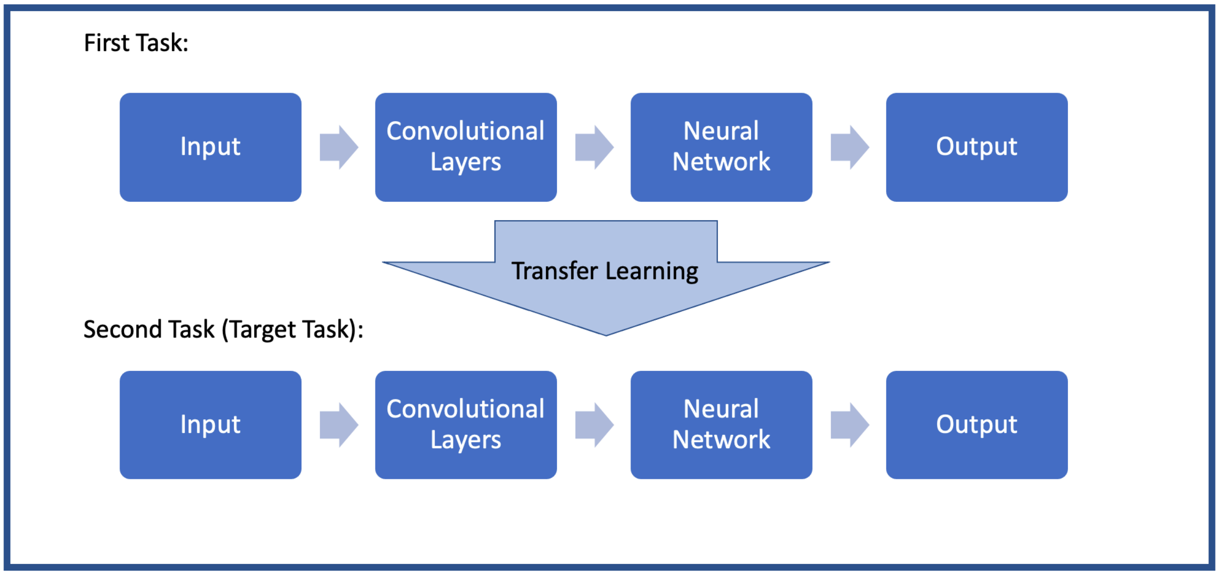


Figure 5.2 - Transfer Learning: using the learning parameters of the network of the first task for training the network of the target task.

Considering the mentioned concepts, we used a CNN network to extract features from MRIs and passed them to a classifier. We used the 387 images of non-operated patients to train a Densenet-121 to classify MRI of male and female patients. The network was trained on batches of images. The size of the images and hardware limitations allowed us to put only two images in each batch, and each batch contained an image of each class. Before passing the images to the network, the images were normalized for image dimension (256*256*152) and scaled the voxel intensity to be between 0 and 1. This data was divided into training and validation sets, and we used the MRI of 75 patients who had surgery as a test set. We used the cross-entropy loss function and Adam optimization to train the network, and the learning rate was set to $1*10^{-5}$.

The preoperative MRI of patients who had surgery were divided into two sets of normal and abnormal MRIs. After training the male versus female network, both datasets were fed to the network, and the output of the section which extracts features were passed to a Max Pooling Layer and were stored to be used by other classifiers. The Max Pooling Layer had a kernel size of 4 and a stride of 16. Each MRI has around 10 million datapoints, and the output of the stored features was a 1024-dimension vector. The 1024 datapoints were passed to six different classification algorithms, including Naïve Bayes, Logistic Regression, SVM (radial basis function kernel), Decision Tree, Random Forest, and a Gradient Boosting algorithm (Figure 5.3). The algorithms were chosen from three major groups of algorithms in machine learning. The first group is Probabilistic Models, the second group is Kernel Methods, and the third group is Decision Trees, Random Forest, and Gradient Boosting Machines. For Naïve Bayes, Logistic Regression, SVM, Decision Tree, and Random Forest, we used Scikit Learn (Pedregosa et al., 2011). For the Gradient Boosting algorithm, we used XGBoost (T. Chen & Guestrin, 2016). We applied the leave-one-out method and calculated accuracy, F1-Score, and Matthew's correlation coefficient (MCC) to evaluate the results. The 95% confidence interval was calculated as exact Clopper-Pearson confidence interval.

Sex differences in adult human brains are observed in overall and subcortical brain volumes, along with surface area and cortical thickness of subregional brain sections (Ritchie et al., 2018). Furthermore, the volume and surface area of different brain regions change in patients with epilepsy (Dreifuss et al., 2001; Pardoe et al., 2009; Pulsipher et al., 2007). We hypothesized that, as the network uses MRI features related to brain volume and surface to classify male *versus* female, these features could further help predict the result of epilepsy surgery.

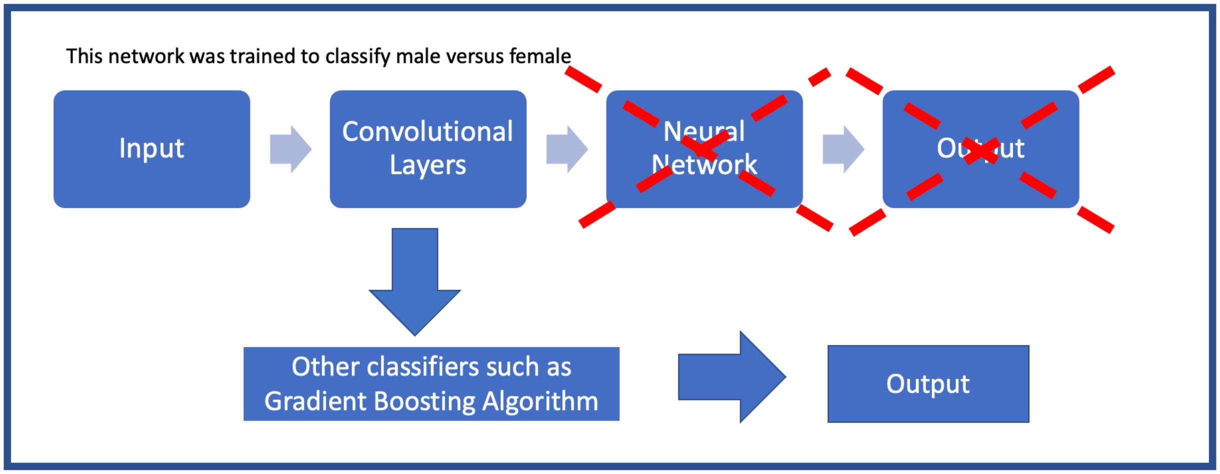

This network was trained to classify male versus female
Input
Convolutional Layers
Neural Network
Output
Other classifiers such as Gradient Boosting Algorithm
Output

Figure 5.3 - Extracted features are passed to a classifier to predict the result of surgery.

4.4 Results

Densenet-121, which was trained to classify male MRI and female MRI, was tested on 75 MRIs of patients who had surgery. The test set consisted of 37 female subjects and 38 male subjects. It yielded 0.96 (95% CI, 0.89–0.99) with F1-Score of 0.96 (95% CI, 0.89–0.99) and MCC of 0.92.

This accuracy was achieved after 150 epochs, which means that the network has seen all the images 150 times. The classification report and more details about the performance of the network are presented in table 5.2. The results are in accordance with other analyses for this task (DLTK, 2018).

Performance of six different machine learning algorithms (from three major types of machine learning algorithms) to predict the result of surgery were compared. The algorithms were applied on extracted features from both normal and abnormal preoperative MRIs, and the results are reported in Table 5.3. The features were extracted using the mentioned Densenet-121 network. The results indicated that XGBoost outperforms other algorithms in all three metrics in both normal and abnormal images dataset. After XGBoost, Decision Trees had the best performance. Considering the results, it is worth mentioning that XGBoost is an implementation of gradient boosted Decision Trees. Our results are in accordance with the general notion that gradient boosting applied on Decision Trees usually outperforms Random Forests (Chollet, 2021). The performance of XGBoost in our results agrees with the general notion that Gradient Boosting algorithms outperform other types of algorithms when dealing with tabular and non-perceptual datasets.

Table 5.4 contains the classification report for XGBoost performance for abnormal MRIs. The algorithm yielded 77% (95% CI, 0.64–0.87) accuracy with F1-Score of 0.75 (95% CI, 0.62-0.86) and MCC of 0.5**.** For the normal MRI cases, the accuracy was 89% (95% CI, 0.67–0.99) with the F1-Score of 0.88 (95% CI, 0.65–0.98) and MCC of 0.78. More detail information about the performance of XGBoost on normal MRI dataset can be found in Table 5.5. In our experiments, XGBoost was set to be a tree booster, with a learning rate of $1*10^{-2}$, and the number of times for boosting was set to 20. We also tested 2D FLAIR and 2D T2-weighted preoperative MRIs of the same patients with the same pipeline. The plan was to ensemble the result from all three sequences to improve the overall results. The pipeline did not work for 2D FLAIR and 2D T2-weighted MRI, which may relate to acquisition mode.

	Precision	Recall	F1-score	Number of cases
Female	0.95	0.97	0.96	37
Male	0.97	0.95	0.96	38
Average	0.96	0.96	0.96	

Accuracy = 0.96 (95% CI, 0.89–0.99), F1-Score = 0.96 (95% CI, 0.89–0.99), MCC = 0.92

Table 5.2: Result of DenseNet-121 network for classifying male vs female using only MRI.

Group of Algorithms	Algorithm	Library	Predicting Result of Surgery for Abnormal MRI cases (n=56)			Predicting Result of Surgery for Normal MRI cases (n=19)		
			Accuracy	F1-Score	MCC	Accuracy	F1-Score	MCC
Probabilistic Modeling	Naïve Bayes	Scikit Learn	0.61	0.57	0.15	0.53	0.42	-0.13
	Logistic Regression	Scikit Learn	0.70	0.68	0.35	0.68	0.63	0.28
Kernel Methods	SVM	Scikit Learn	0.64	0.54	0.19	0.58	0.37	-0.18
Decision Trees, Random Forest, and Gradient Boosting Machines	Decision Trees	Scikit Learn	0.71	0.69	0.38	0.74	0.72	0.45
	Random Forest	Scikit Learn	0.59	0.55	0.1	0.53	0.42	-0.13
	Gradient Boosting	XGBoost	0.77	0.75	0.5	0.89	0.88	0.78

Table 5.3: The features of the MRI of patients who had surgery were extracted using the convolutional layers section of male vs female network and were fed to the different classification algorithms.

	Precision	Recall	F1-score	Number of cases
Engel II-IV	0.71	0.68	0.70	22
Engel I	0.80	0.82	0.81	34
Average	0.76	0.75	0.75	

Accuracy = 0.77 (95% CI, 0.64–0.87), F1-Score = 0.75 (95% CI, 0.62-0.86), MCC = 0.5

Table 5.4: XGBoost performance for predicting the outcome of surgery using extracted features from preoperative lesional MRI cases. The features were extracted using a CNN model. The model is evaluated using leave-one-out cross-validation.

	Precision	Recall	F1-score	Number of cases
Engel II-IV	1.00	0.71	0.83	7
Engel I	0.86	1.00	0.92	12
Average	0.93	0.86	0.88	

Accuracy = 0.89 (95% CI, 0.67–0.99), F1-Score = 0.88 (95% CI, 0.65-0.98), MCC = 0.78

Table 5.5: XGBoost performance for predicting the outcome of surgery using extracted features from preoperative MRI of 19 patients with non-lesional MRI. The features were extracted using a CNN model. The model is evaluated using leave-one-out cross-validation.

4.5 Attempts which did not work

It was mentioned before that the first attempt was using the 75 images to train a network which predicts the result of surgery. The images were divided into training, validation, and test sets. The validation set was set to 12 images (6 images of each class), and the test set was also set to be ten images (5 images of each class). The training set was balanced by duplicating some of the instances of the class with a lower number of images. We made sure that each batch had an equal number of images of each class when presented to the network. As the training loss and validation accuracy graphs shown in Figure 5.4, this attempt did not work. In the graph, the training loss decreases gradually overtime, while the validation accuracy fluctuates around 50%. This indicates that the network is overfitting on the training data. Given the small number of cases, this was expected.

In the second attempt, we trained a network on male vs female data as described previously and then let the network train on the target task. Again, as Figure 5.5 shows, the network is overfitted on the training data, and validation accuracy fluctuated around 50%.

In the third attempt, we first trained the network on male vs female data and then used the same network to train it for the target task, which is predicting the result of surgery, but this time with a different strategy. We froze the weights and biases of the convolution layers and only let the classification layer to be trained. As Figure 5.6 shows, this attempt was also futile. We tried other different combinations of transfer learning, such as freezing weights and biases of the first few layers of the convolution layers and then training the network. We also changed the learning rate and tested different methods for initiating weights before the training starts. None of the mentioned experiments resulted in meaningful results.

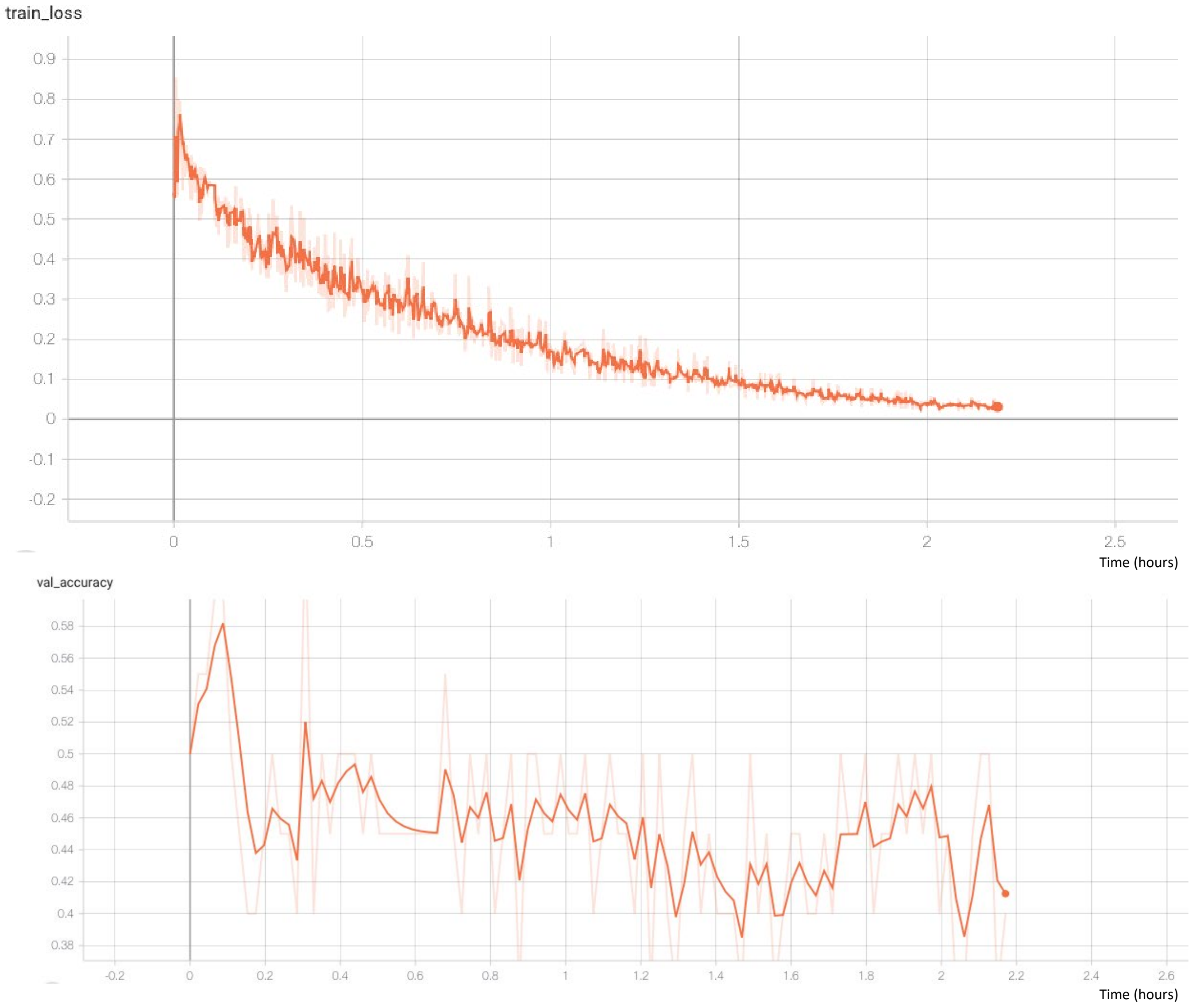


Figure 5.4: Training loss and validation accuracy graphs for a network which was trained to predict the result of surgery. Given the small number of samples, as it was expected, the training loss shows that the network is overfitting as the network is trying to minimize the loss and iterates over the training set.

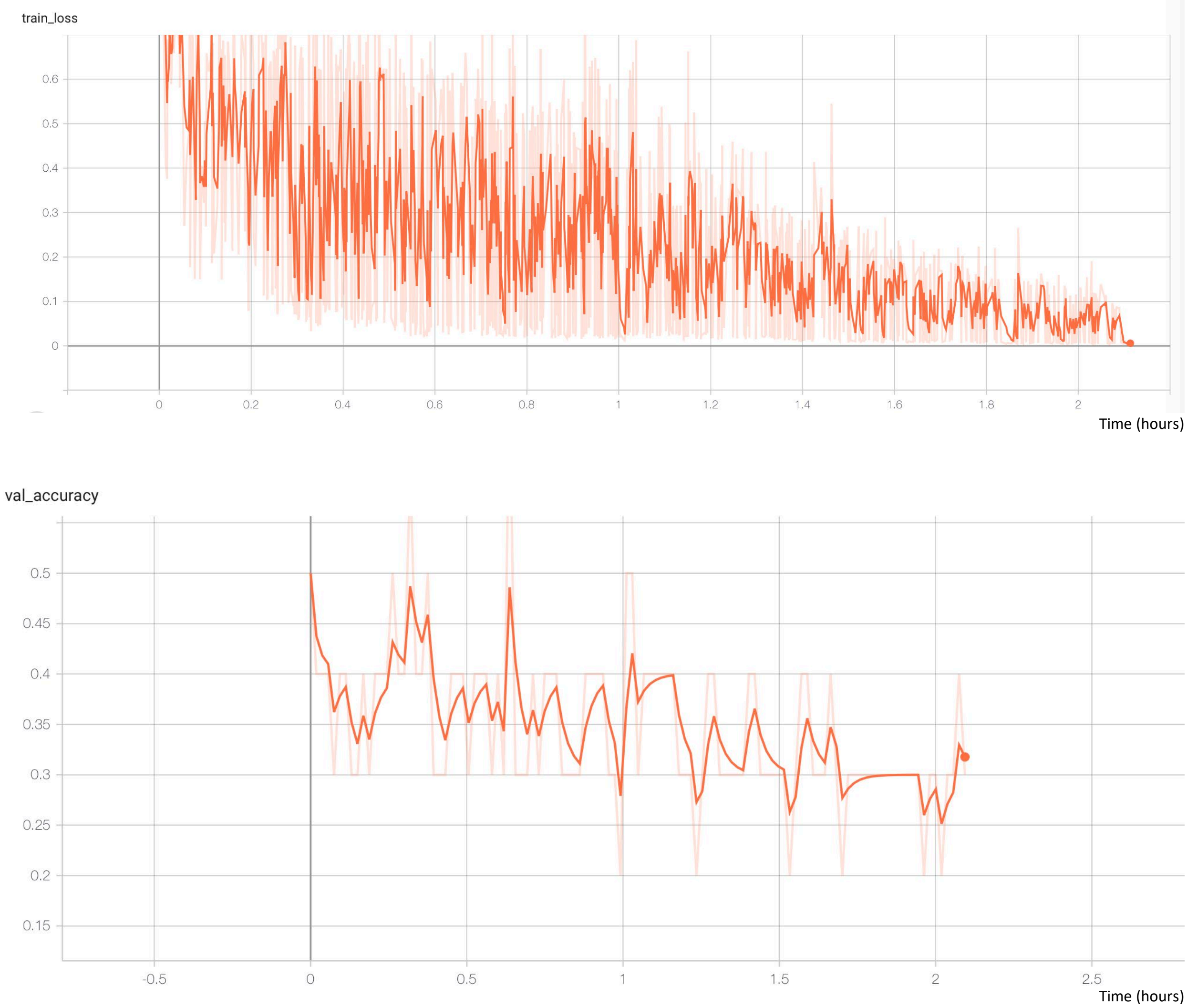


Figure 5.5: The training loss and validation accuracy of a network which first was trained to predict male vs female and then was trained to predict the result of surgery. As the graphs show, the network overfits on training data.

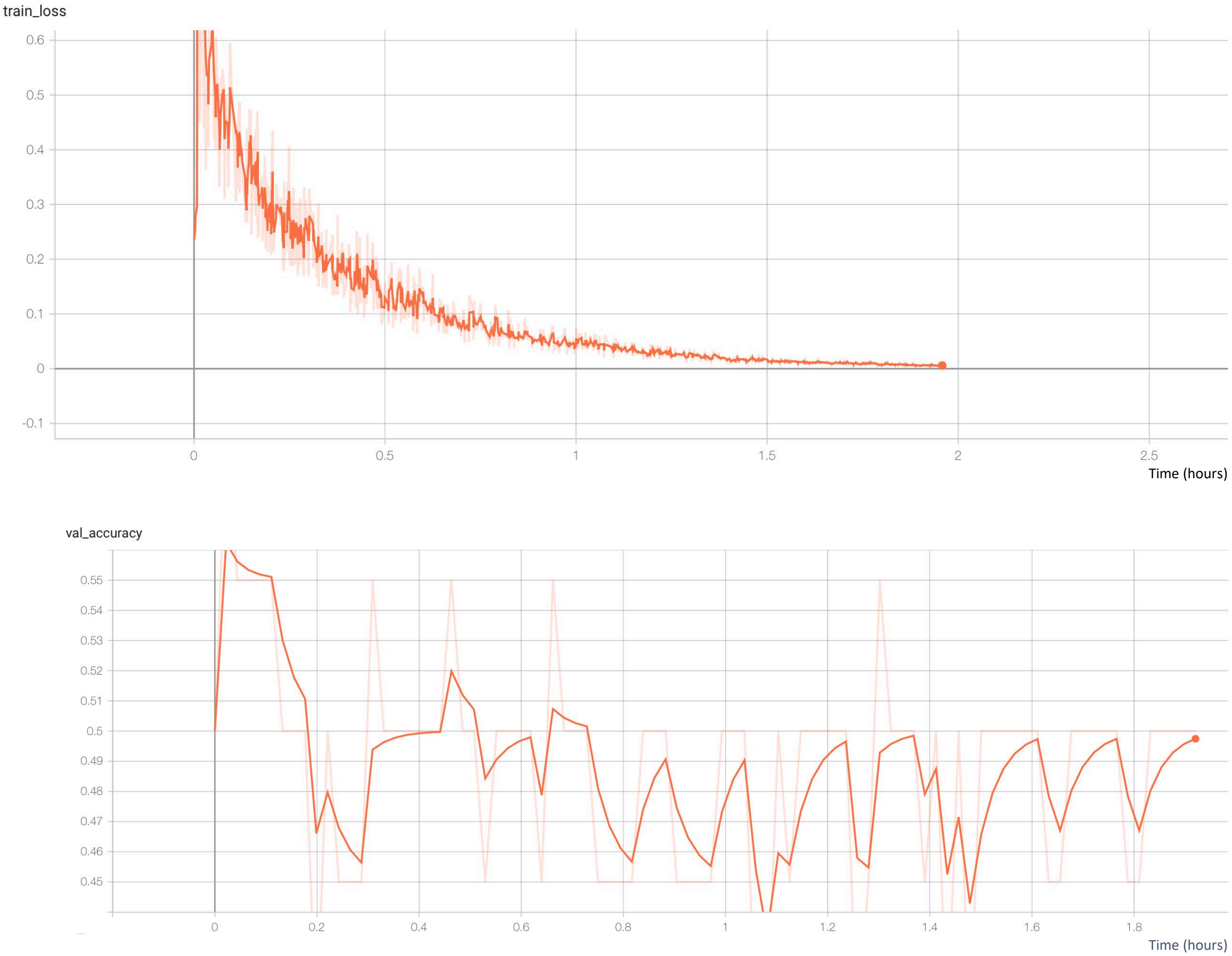


Figure 5.6 Training loss and validation accuracy for predicting the result of surgery using transfer learning. First, the network was trained on a male vs female task and then it was trained on the target task. The convolution layer parameters were frozen during the second phase of training. The graphs indicate that the network was overfitting on the training data.

After noticing that using traditional methods of transfer learning is not solving our problem, we decided to use the extracted features from the convolution layers and pass them to other classifiers. Given that the number of features was very high compared to the number of samples, we decided to use Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) to reduce the number of features before classifying. Testing different combinations for these methods, we noticed they were not useful for us. As was mentioned before, we used Max Pooling to reduce the number of features which was retrieved from the convolution layers. It is worth noting that we also tested Average Pooling and Adaptive Pooling as well, but Max Pooling produced the best results.

Alongside testing different algorithms by passing features as input to them, we tried to see if there is a way to use distant measurements such as cosine similarity to classify the features, and it did not work. Another major algorithm which we experimented with was Hopfield networks. In the following, we have discussed how Hopfield networks work and how we tried to use them for our research.

Hopfield Networks

Hopfield networks are a type of neural network that, unlike feedforward networks, are not learning patterns through an iterative process. They are presented with patterns once and can retrieve the pattern when similar data is presented to them afterwards. The retrieving part is an iterative process. Let us assume that we have a network which has stored P patterns. When it is presented with a new pattern, it goes through iterations of updating states of the neurons (nodes) and produces a pattern which most resembles the new pattern. They are basically associative networks which associate the input to its most similar pattern. For example, as is shown in Figure 5.8, if the network is trained to learn the four images on the first column, and then the noisy version of any of the images is presented to the network, the network retrieves the learned image without the noise.

Hopfield networks are point attractor networks. Assuming that our network has N neurons, then the state of the neurons can be imagined in an N-dimensional space. In this space, there are points to which the network is attracted after going through iterations of the update rule for each neuron. These points can be the learned patterns or could be somewhere else in the N-dimensional space. After training, if the presented state to the network is close to one of the stored patterns, then the update rule carries the starting points into those stored points. If the stored patterns in our network are not orthogonal to each other, it can cause the creation of attractor points.

Structure, Update Rule, Learning Rule, and Capacity

In Hopfield networks, all the neurons are connected to each other, and the connections between the nodes are weighted. In this architecture, there is no self-connection. The state of the node *n* is determined by a method which takes the state of all the other nodes, multiplies them by the weight of the connection between them and node *n* and then adds them up. During the retrieving phase, this method is used iteratively until the network is in an attractor point of the mentioned space.

One of the main learning rules for such networks is the Hebbian learning rule. This rule is set based on the idea that the weight of each connection is relative to the activity of both ends of the connection. If we only have one pattern, implementing the rule and identifying the weight between the two nodes is easy. But how does these types of network use this learning rule for more than one pattern? To address this question, all the weights which are created by different patterns for one connection would be added up. As it is expected, these weights would be used to update the state of the nodes during retrieving (update rules). The capacity of a network in terms of storing patterns and retrieving them depends on the number of nodes that the network has.

Our idea was to use the extracted features from the convolution layers produced by the network, which was trained to classify male vs female MRIs and identify if we can use classical Hopfield networks to predict the result of surgery. The first attempt was training one network with samples of features of subjects who had successful surgeries and features of subjects who did not have a successful surgery and identifying if we presented features of any of the subjects, the network would retrieve any of the samples. This idea did not work, and we trained one network with features of subjects who had successful surgery and another network with features of subjects who did not have successful surgery and identified if any of the networks could retrieve the samples. Our experiments based on these ideas did not work.

Another idea that we tested was the following. We tried to see if it is possible to create one set of features for each class which would be the main form of that class, and see if the network trained for that set can retrieve the right class when presented with the features of subjects from that class. Different combinations for this idea were tested, and none of them worked.

Recently an improved version of Hopfield networks was introduced (Ramsauer et al., 2020). The new network has a larger capacity for the number of patterns that it can learn. As is shown in Figure 5.9, using the traditional Hopfield network, the network can learn six patterns. When half of the image of one of the patterns (Homer Simpson) is presented to the network, it can retrieve the whole image. However, in the new version, a network with the same number of nodes (neurons) can store 24 patterns and still retrieve one of the inputs when half the image is provided to the network as an input. Using this modern version of Hopfield was also tested, and we did not achieve meaningful results.

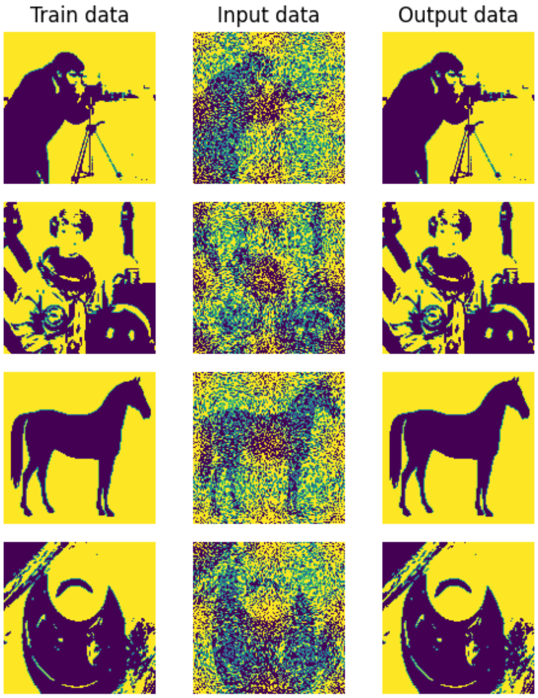
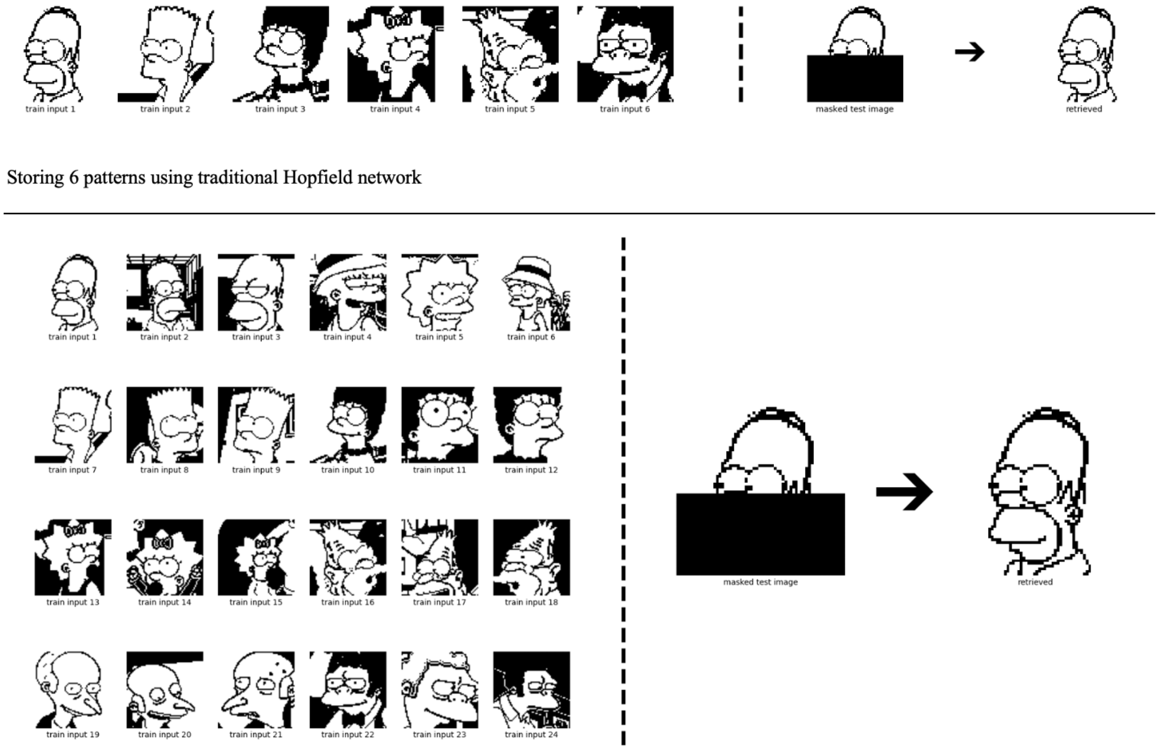


Figure 5.8: If the images in the input column (second column) were presented to a Hopfield network which was trained with the four images in the train data presented in the first column, the network would retrieve the corresponding images in the output column. Source of image: (*GitHub - Takyamamoto/Hopfield-Network: Hopfield Network Implemented with Python*, n.d.)



Storing 6 patterns using traditional Hopfield network

Storing 24 patterns in the new version of Hopfield network with the same number of nodes (neurons).

Figure 5.9: The different between the traditional Hopfield network and the new version introduced in 2020. The new version has a higher capacity. Source of image (*Hopfield Networks Is All You Need | Hopfield-Layers*, n.d.)

Gradient Boosting Algorithms and Extracted Features

It has been constantly shown that Deep Learning algorithms are the best candidates to be used when working with perceptual data such as images and videos. On the other hand, gradient boosting algorithms are known to be one of the better algorithms when dealing with tabular data. As was discussed previously, our experiments also confirm that. We also ran another experiment which extracts features and passes them to different classifiers and received the same results. We trained a CNN model to classify images of handwritten digits of zero and one from the MNIST dataset. MNIST is a set of images of handwritten digits from zero to nine (Figure 5.10). The main idea was to choose two sets of images which would cover different general shapes when training the model for the first task. The images are 28*28 pixels which makes each image to have 784 dimensions. When using UMAP to reduce the dimension to visualize the data, it is evident that one and zero are not very close to each other, which makes them good candidates for our plan (Figure 5.11).

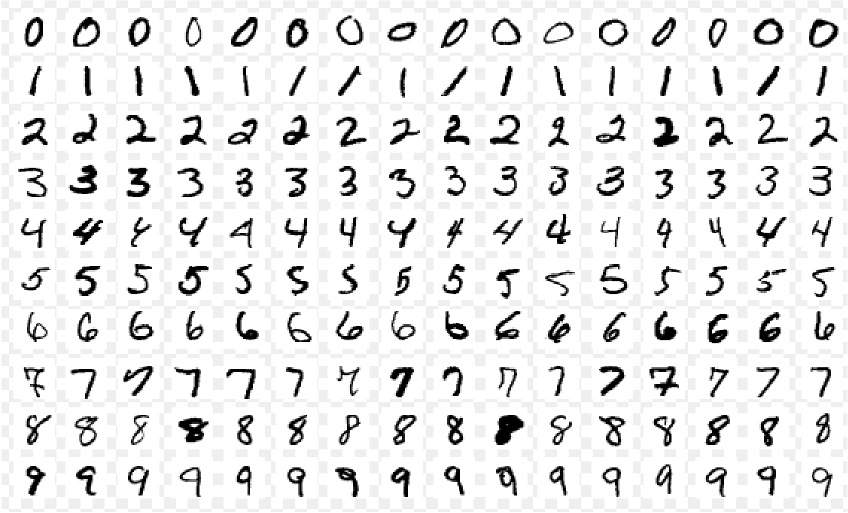


Figure 5.10: Examples of handwritten digits of MNIST.

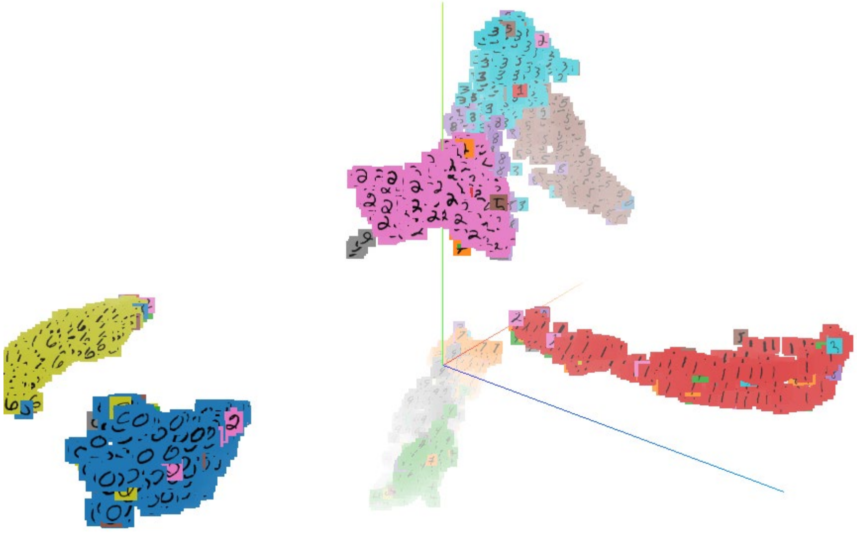
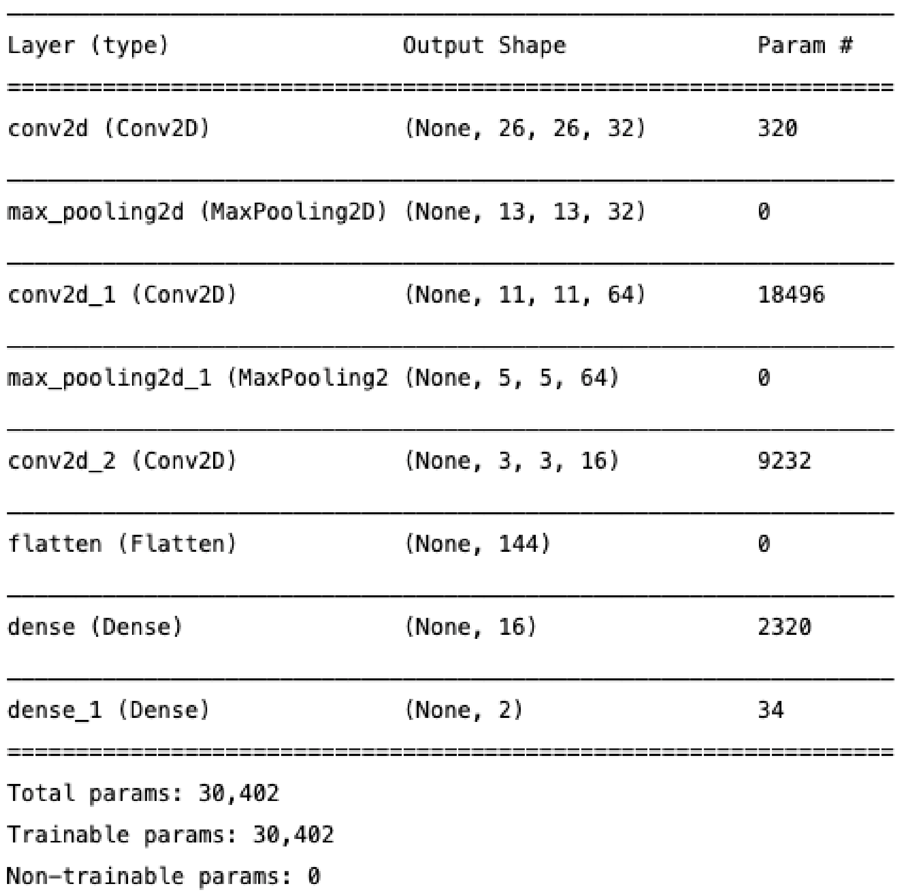


Figure 5.11: Reducing the dimensions of MNIST data to visualize it. The blue bobble on the left button of the image represents images of zero, and the red bobble presents images of one.

The architecture of the network that we used is shown in Figure 5.12. The target task was to classify images of four and five. As has been explained in the previous sections, we used the convolution layers of the network, to extract the feature and passed them to different algorithms. In this experiment also, XGBoost performed better than the other algorithms (Table 5.6). The two experiments suggest that anytime this type of structure is being tested, it would be a good idea to start with testing XGBoost.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 3, 3, 16)          9232
_________________________________________________________________
flatten (Flatten)            (None, 144)               0
_________________________________________________________________
dense (Dense)                (None, 16)                2320
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 34
=================================================================
Total params: 30,402
Trainable params: 30,402
Non-trainable params: 0
```

Figure 5.12: Structure of the network being used to classify images of zero and one.

Group of Algorithms	Algorithm	Library	MNIST (4 and 5) - Accuracy, F1
Probablistic Modeling	Naïve Bayes	Scikit Learn	0.52, 0.34
	Logistic Regression	Scikit Learn	0.52, 0.34
Kernel Methods	SVM	Scikit Learn	0.53, 0.36
Decision Trees, Random Forest, and Gradient Boosting Machines	Decision Trees	Scikit Learn	0.71, 0.69
	Random Forest	Scikit Learn	0.67, 0.61
	Gradient Boosting	XGBoost	0.75, 0.73

Table 5.6: Result of performance of each algorithm applied on the features extracted from images of four and five. The extracted features were gathered by passing the images through the convolution layers of a network trained to classify zero and one images.

4.6 Conclusion

Around 40% of patients with epilepsy are drug-resistant, and it has been shown that surgery can be beneficial to these people. Depending on the anatomical region, approximately 60% of surgeries result in seizure freedom, but predicting this on an individual level has been elusive. Hence, decision support tools which can predict the result of surgery for individual patients would be an extremely valuable asset in clinical care. Using only MRI data, we have designed a pipeline which can predict the result of surgery for cases with abnormal MRI with 77% (95% CI, 0.64–0.87) accuracy. We applied the mentioned pipeline to non-lesional MRIs and achieved 89% (95% CI, 0.67–0.99) accuracy. Given that only 35% to 50% of surgeries for normal MRI cases are successful, this is a notable result. Considering both experiments, our work suggests that there is a signal in MRI which can help predict the result of surgery.

Other studies have used MRI data to predict the result of surgery in two ways, either the interpretation of the MRI by a physician is fed to the prediction algorithm or features from MRI are extracted and provided as input. The examples of extracted features are volumetric features or connectome map of the brain. Deep Learning has proven to be a very powerful tool for automatically finding and extracting the most important features of data sets, such as images and videos where the computer is handling a perceptual task. In this study, we have used features extracted by Convolutional Neural Networks (CNN), which are a type of Deep Leaning (DL), to predict the result of surgery. Our results confirm the power of CNNs in automatically extracting features and reaffirming the capability of gradient boosting algorithms when dealing with tabular and non-perpetual datasets.

One of the main constraints of our study was the lack of a large number of preoperative MRIs of patients who had surgery to train the machine learning algorithms. Small samples affect the training, validation, and generalization of the models. We have used the Clopper-Pearson method to calculate the confidence interval, and it is one of the most conservative methods for calculating the confidence interval. A small number of samples results in large intervals when calculating the 95% confidence interval, and in our case, this interval covers around 35 points. It is important to note that even the lower bound of the reported intervals in our studies shows that there is a signal in MRI to predict the result of sugary.

More images and further investigation can solidify our findings and perhaps improve the result of the prediction. The MRI that we used for training and testing was from one medical center. To test the scalability of the solution, images from different medical centres must be gathered and used in the training and testing process. Furthermore, methods for adopting trained models in different medical centres should be tested and investigated. A model which is proven to work properly in one center can be deployed to other centers by using finetuning and transfer learning, with the assumption that there would be enough data in other centers for finetuning.

Results from models which drive prediction from MRI data, such as the work in this chapter, can be ensembled with other models which use data from other diagnostic tools to have better and more accurate predictions. Testing these systems with large datasets, which are gathered from different epilepsy centers, can pave the way for developing decision support tools which are ready to be deployed in clinical settings to assist clinicians when identifying the best surgical candidates.

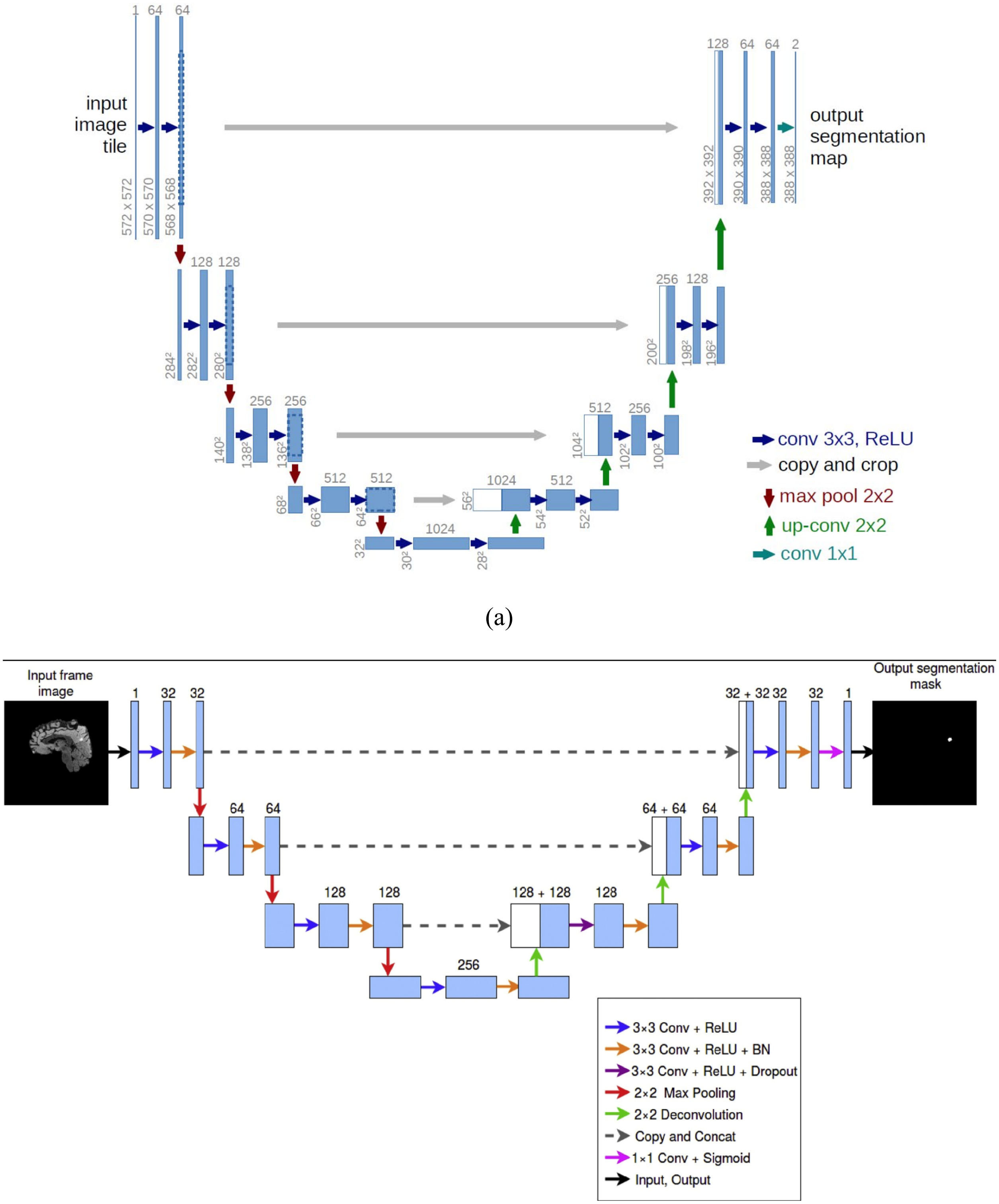
5 Finding the Abnormality (Project 3)

5.1 Introduction

As it was mentioned before, around 40% of patients with epilepsy are drug-resistant and are considered to be a candidate for having surgery to either eliminate the seizures or decrease the frequency and severity of seizures. One of the main diagnostic tools which help clinicians to evaluate candidates and plan for surgery is MRI. Finding epilepsy-related abnormality in MRI helps physicians to locate the area where seizures start and therefore increases the chances of having an effective operation. On the other hand, a lack of abnormality reduces the chance of successful surgery. For example, between 60% to 70% of patients with temporal lobe epilepsy become seizure free after surgery, but this number is around 40% for patients who don't have any abnormalities in their MRI. Clinicians cannot find abnormality in around 44% of patients, but the interesting point is that this number decreased when they were asked to re-evaluate the MRI after they are provided by the histopathology report. In a study, only 16% of patients did not have abnormality reported either in MRI or in the histopathology report (Bien et al., 2009). This suggests that there may be ways to improve the evaluation and examination of MRI, so clinicians can find the abnormality in more cases, which consequently would improve the result of surgery cases.

Voxel-wise comparison between MRIs with abnormality and healthy control groups has identified that such analysis can become useful in identifying the area of the abnormality (Focke et al., 2009; Martin et al., 2017). Considering that the reported metrics in these studies had room to improve, machine learning algorithms were utilized to improve the analysis. For example, in one study, SVM was used to lateralize the temporal lobe epilepsy of lesional MRIs (Focke et al., 2012), or in another study, SVM was used to identify the area of abnormality in MRI of lesional cases (Perera-Ortega et al., 2021). Random forest was also used for the purpose of lateralization in non-lesional MRIs (Bennett et al., 2019).

In recent years Deep Learning algorithms have also been used to find abnormalities in MRI. For example, in one research study, U-Net architecture was used for the segmentation of focal cortical dysplasia (FCD) in MRIs. U-Net architecture has two main sections. The first section extracts features (encoding), and the second section constructs an image using the extracted features (decoding) (Figure 6.1) (Ronneberger et al., 2015). The network was trained and evaluated on marked MRIs and reached a Dice Score of 52.47 (Bijay Dev et al., 2019). Dice score is used to measure the performance of algorithms which are used for segmentation. In essence, it evaluates the overlap between the area which was marked by the algorithm and the actual marked area ($(2\cdot|A\cap B|)/(|A|+|B|)$). In another study, the researchers used a CNN for classifying the MRIs with FCD and without FCD. They proposed a method to mark the FCD area and achieved Dice score of 78 (Wang et al., 2020). Another method was proposed by Feng et al. to mark the FCD in an MRI, and they have reached Dice score of 52.68 for non-lesional MRIs. Although in the paper, the Dice score was reported for FCD of non-lesional MRI, they used resected area during the surgery and not the area acquired based on the histopathology report (Feng et al., 2020).



(a)

(b)

Figure 6.1: (a) The original U-Net architecture (source: (Ronneberger et al., 2015)*. (b) The modified U-Net architecture to mark the FCD in lesional MRI cases. (Source:* (Bijay Dev et al., 2019)*)*

We planned to design a system that can help clinicians find abnormalities in non-lesional MRIs. We hypothesized that a system which can differentiate between MRIs of healthy brains and MRIs of epilepsy patients, in which no abnormality is reported on them, uses the lesional area to distinguish between the two classes. To test our idea, we decided to train a Deep Learning model which differentiates between the two classes and then identify which area of the MRI causes the network to classify an MRI as belonging to the epilepsy group. We planned to compare the identified area of MRIs of patients who had surgery with the following areas and determine if we can reach any conclusion:

- Area of the brain where the seizure starts based on intracranial EEG report
- The resected area during the surgery
- Histopathology report

We also planned to test and see if the features resulting which are triggering the discrimination between the two classes would lead to finding any sign of abnormality in the MRI by re-examining the MRI. There are different methods to find the areas of an image which cause the classification in CNNs. Three of the methods are reviewed in the following.

5.1.1 Visualizing The Reason Behind

As the potential of Deep Learning algorithms for computer vision tasks such as classification was revealed, understanding the reason behind the results produced by the algorithms has become an important research area. A few main approaches are being deployed by the Deep Learning community to open the black box and provide more explanation behind the decision made by the networks, and in the following three of them are explored.

Occlusion Sensitivity

In this technique, portions of the image are occluded systematically, and the effect of the covered portion on the output of the network is measured. The areas which affected the outcome the most are the regions of the image which have contributed the most in the process of classification. Figure 6.2 shows an example of using Occlusion sensitivity for identifying a cat in the image (Selvaraju et al., 2017).

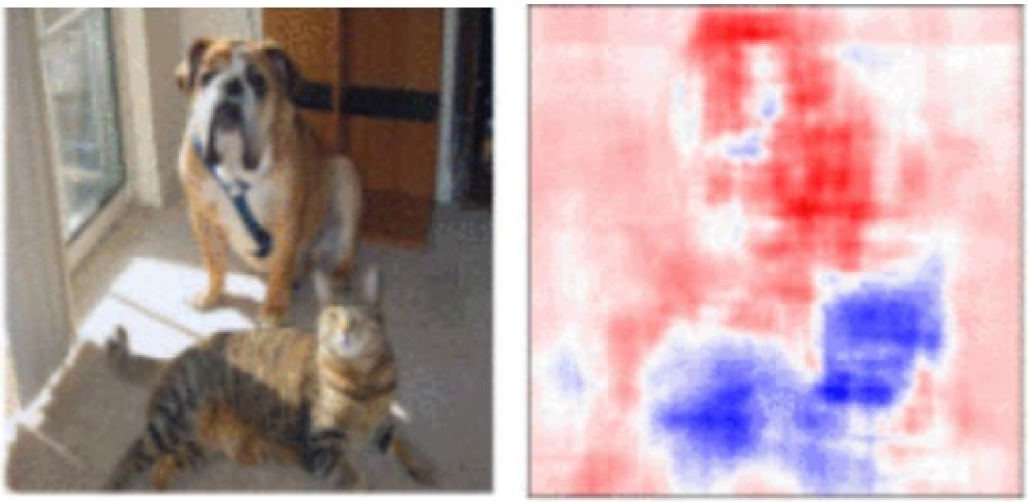


Figure 6.2: Using Occlusion sensitivity to locate the cat in the image. The source of the image is (Selvaraju et al., 2017)

Class Activation Mapping

Class Activation Mapping (CAM) creates heatmaps based on the activation of neurons of convolutional filters in the last layer of the CNN for the predicted category. In the process of calculating the heatmap of CNNs, the network structure would be changed, and this will reduce the accuracy (Zhou et al., 2015).

Gradient-weighted Class Activation Mapping

Gradient-weighted Class Activation (Grad CAM) generalizes CAM and does not change the architecture of the CNN when calculating the heatmap. It uses the gradient of the last convolutional layer of the CNN and is applicable to different families of CNN (Selvaraju et al., 2016). Newer and more advanced visualization methods, such as Grad-CAM++ or Grad-CAM++ with Mask Regional Convolutional Neural Network (GC-MRCNN), have been developed recently (Inbaraj et al., 2021). Figure 6.3 shows examples of CAM, Grad-CAM, Grad-CAM++ and GC-MRCNN.

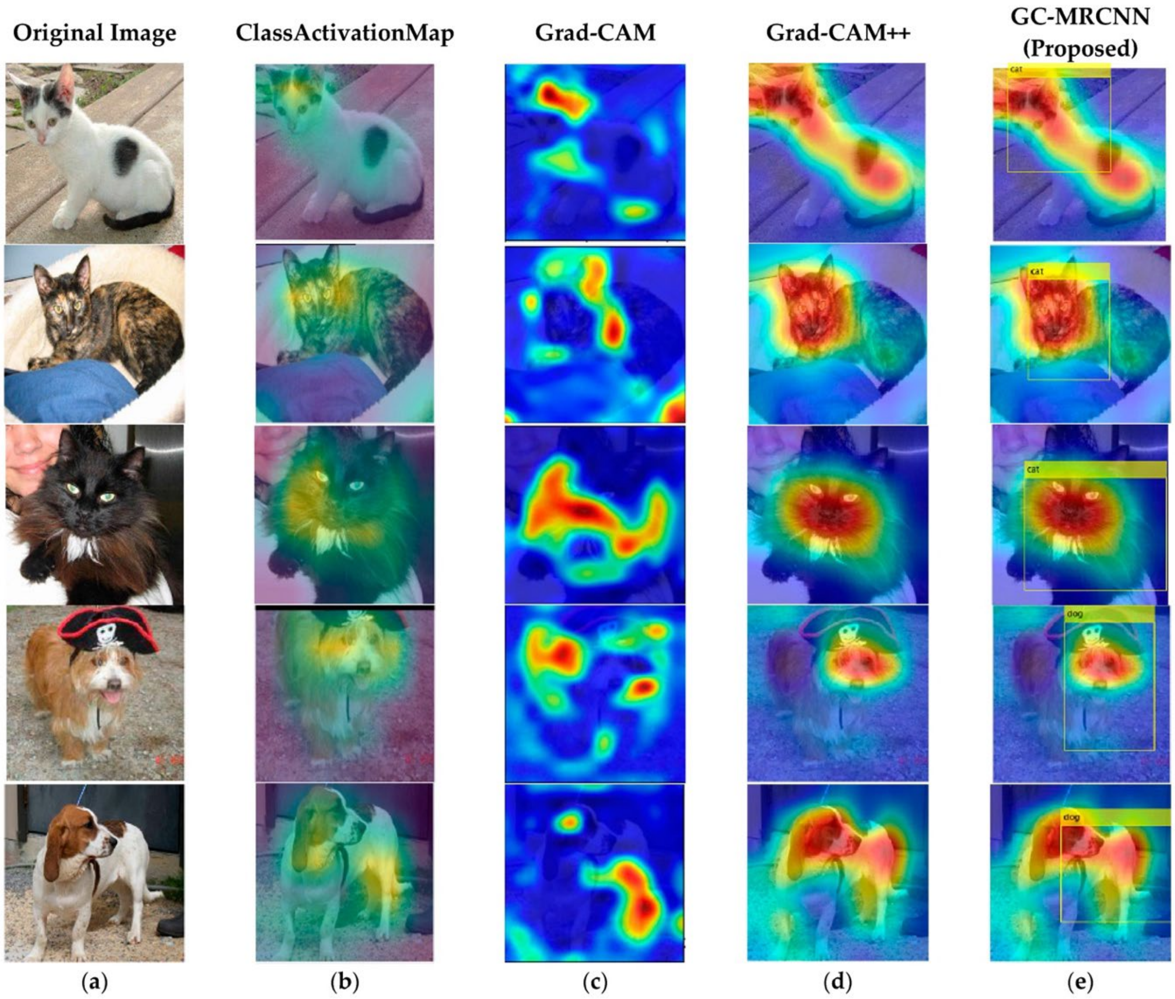


Figure 6.3: Original image and different methods of visualization. The image is borrowed from (Inbaraj et al., 2021)

Identifying the relevance and importance of each area of the image to the classification outcome has been used for medical images. For example, visualization methods have been used to bring insight to CNNs, which were trained for Alzheimer's disease classification based on MRI (C. Yang et al., 2018). In another instant, the methods were used to identify the areas of the lung affected by Covid-19. Figure 6.4 shows the heatmap produced by the occlusion sensitivity method on a network trained to segregate Covid-19 CT scans from CT scans of normal lungs and CT scans of lungs with pneumonia (Aminu et al., 2021). In another example, Grad-CAM was used to investigate a network which differentiates dementia with Lewy bodies from Alzheimer's disease using brain perfusion single photon emission tomography images (Figure 6.5) (Iizuka et al., 2019). It has to be mentioned that there are also points of view which argue that the mentioned visualization tools should only be used for "model troubleshooting and systems audit" and not for the purpose of "human-comprehensible explanations for complex, black-box machine learning algorithms" (Ghassemi et al., 2021).

5.2 Methodology and Results

We had access to 353 T1-weighted MRIs of healthy brains and 213 T1-weighted non-lesional MRIs of patients with epilepsy. The healthy brain MRI scans were provided by Dr. Richard Frayne, and the location of the scans was Seaman Family MR Centre in Foothills Medical Centre in Calgary, Alberta (*Home | Vascular Imaging | Cumming School of Medicine | University of Calgary*, n.d.). The MRI scans for the patients with epilepsy were obtained from "Federico Lab" (*Federico Epilepsy Neuroimaging | Home | Cumming School of Medicine | University of Calgary*, n.d.). The average age of individuals, who were recruited for healthy brain MRI scans, was 49.7, and the average age of patients with epilepsy was 40.5. In this document, non-lesional is referred to MRIs where no epilepsy-related abnormality is reported for them (Table 6.1). The dimension of healthy MRIs was mostly 166, 256, 256 (for x, y and z dimensions, respectively), and the dimension of the non-lesional MRIs were mostly 256, 256, 148 or 256, 256 152 (for x, y, and z dimensions respectively). The dataset was divided into three sections of training, validation, and test sets. Three hundred thirteen of the healthy MRIs were allocated to the training set, and 173 of the non-lesional MRIs were put in the training set. Twenty images of each class were allocated to the validation set, which resulted in a total of 40 images for the validating set. Forty images were assigned to the test set, and it included 20 images of each class.

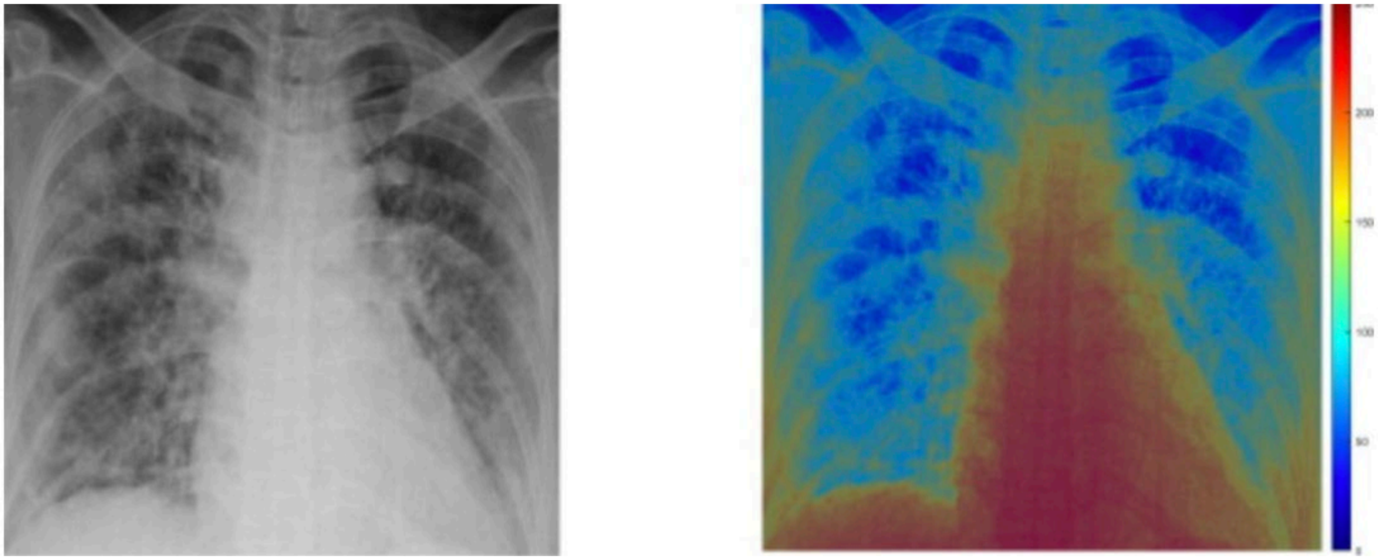


Figure 6.4: Heatmap of a CT scan of a lung produced based on the output of a CNN, which was trained to classify normal class, pneumonia class and Covid-19 class images of lungs. The lung in this CT scan was from the Covid-19 class. The source of the image is: (Aminu et al., 2021).

We used DensNet-121 network architecture as the other projects showed that it could be a good candidate. The review of the structure of the network is not provided here, as the structure of DensNet-121 has been discussed in previous chapters. The CNNs are trained in batches of images, and given the size of MRIs and the hardware limitations for training a CNN with the mentioned MRIs, each batch could only have two MRIs. Given the mechanics of the backpropagation algorithm and the fact that the gradient descent algorithm is applied to each batch, it made sense to make sure that each batch has both classes, so each time that the algorithm is trying to minimize the error between the predicted result and expected result it would be based on a sample of each class. To achieve this, we duplicated some of the non-lesional MRIs to increase them to a total of 313. The duplicated images were chosen randomly. It is important to mention that before training, we normalized the MRIs for scale intensity and resized all of them to 256, 256, and 152.

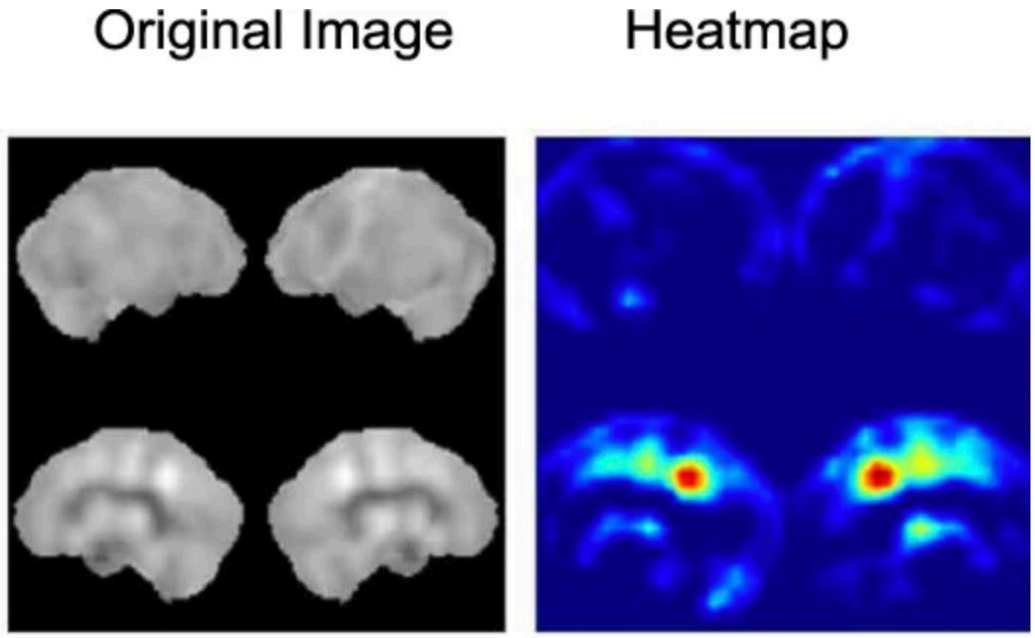


Figure 6.5: Heatmap for a network which distinguishes between dementia with Lewy bodies from Alzheimer's using brain perfusion single photon emission tomography images. The source of the image is (Iizuka et al., 2019).

	Age	Sex

Non-lesional (n = 242)	Avg = 40.5
	SD = 14.23
		57% female
		138

Healthy (n = 382)	Avg = 49.7
	SD = 17.50
		55% female
		211

Table 6.1: Demographics of the two classes of MRI. Non-lesional MRIs are MRI of patients with epilepsy which did not have any abnormality reported on them, and healthy is MRI of normal brains.

After training the network, it was identified that the validation accuracy was 100%. Further investigations indicated that the network had achieved this status in only two epochs. Each epoch is defined as one round of presenting all the images to the network for training. Figure 6.6 shows training loss and validation accuracy graphs during the training of the network. Running the network against the test set also presents a 100% accuracy (Table 6.2). This indicates that there must be a very strong signal which is not related to subtle differences but probably to a major difference in the MRI of each class. A closer examination of both classes revealed that most of the healthy MRIs include the neck of the subjects (Figure 6.6). This led to the second attempt to train a network after cropping the images and removing the neck area of the MRIs. We used dcm2nixx tool to remove the neck area from the MRIs. Figure 6.7 shows a sample of an MRI of the healthy brain after removing the neck area. Unfortunately, that did not solve the problem either, and the network still converged very quickly (Figure 6.8).

We decided to train a network which distinguishes between healthy control MRIs (not cropped) and non-lesional MRIs with fewer images and to observe the accuracy of the validation set over time. To do this, we randomly chose 14 images (7 of each class) and 60 validation images. The main purpose of the practice was to make sure that we had not made any mistakes in our code and to try to gain a better understanding of the results. We adjusted the dimensions of the healthy control MRI to the non-lesional MRI's dimensions (256, 256, 152). Figure 6.9 illustrates that the network achieves 100% accuracy on the validation set after 16 epochs. We repeated the same experiment and normalized all the images to (211, 256, 204) dimensions which are the average values of dimensions in both classes. In the previous experiment, we picked the dimensions of the non-lesional MRIs and normalized the healthy control images based on non-lesional MRIs dimensions (which would result in losing information in the healthy control group). Choosing the middle ground also did not solve the issue, and the network archived 100% accuracy after ten epochs (Figure 6.10). We ran similar experiments on a cropped version of healthy control images. The dimension of cropped healthy control images was (166,256, 170). Although removing the neck area resulted in achieving 100% accuracy in epoch 200, which indicates that removing was useful, but still the difference between the MRI of the two classes are very pronounced (Figure 6.11 and 6.12).

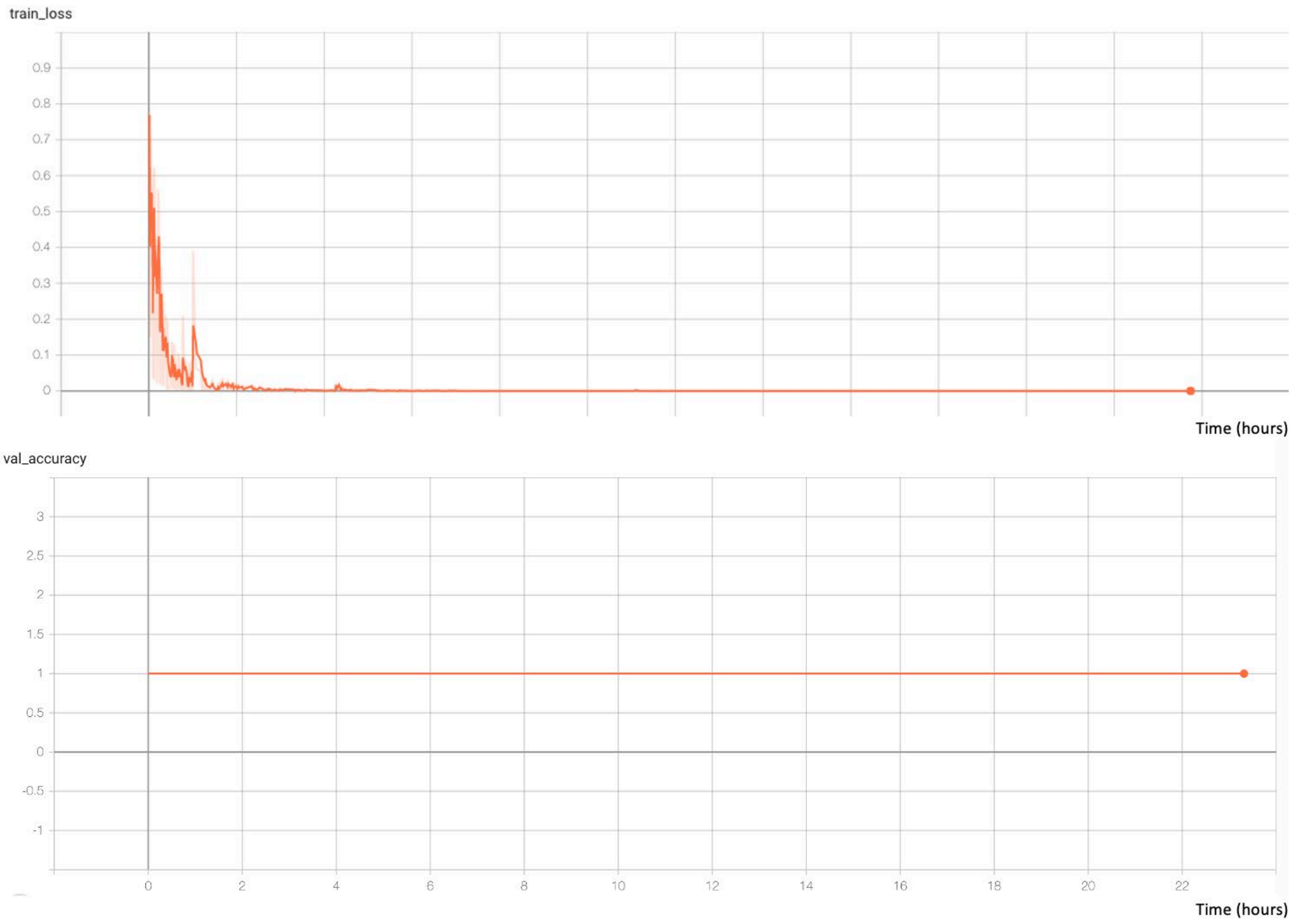


Figure 6.5: The validation accuracy and training loss graphs during the training of the network which is aimed to distinguish between healthy brain MRIs and non-lesional MRIs of patients with epilepsy. The graphs are smoothed by 0.6.

	Precision	**Recall**	**F1-score**	**Number of cases**
Non-lesional	1	1	1	20
Healthy	1	1	1	20
Average	1	1	1	

Table 6.2: Result of running the trained network against the test set of healthy brain MRIs and non-lesional MRIs of patients with epilepsy.

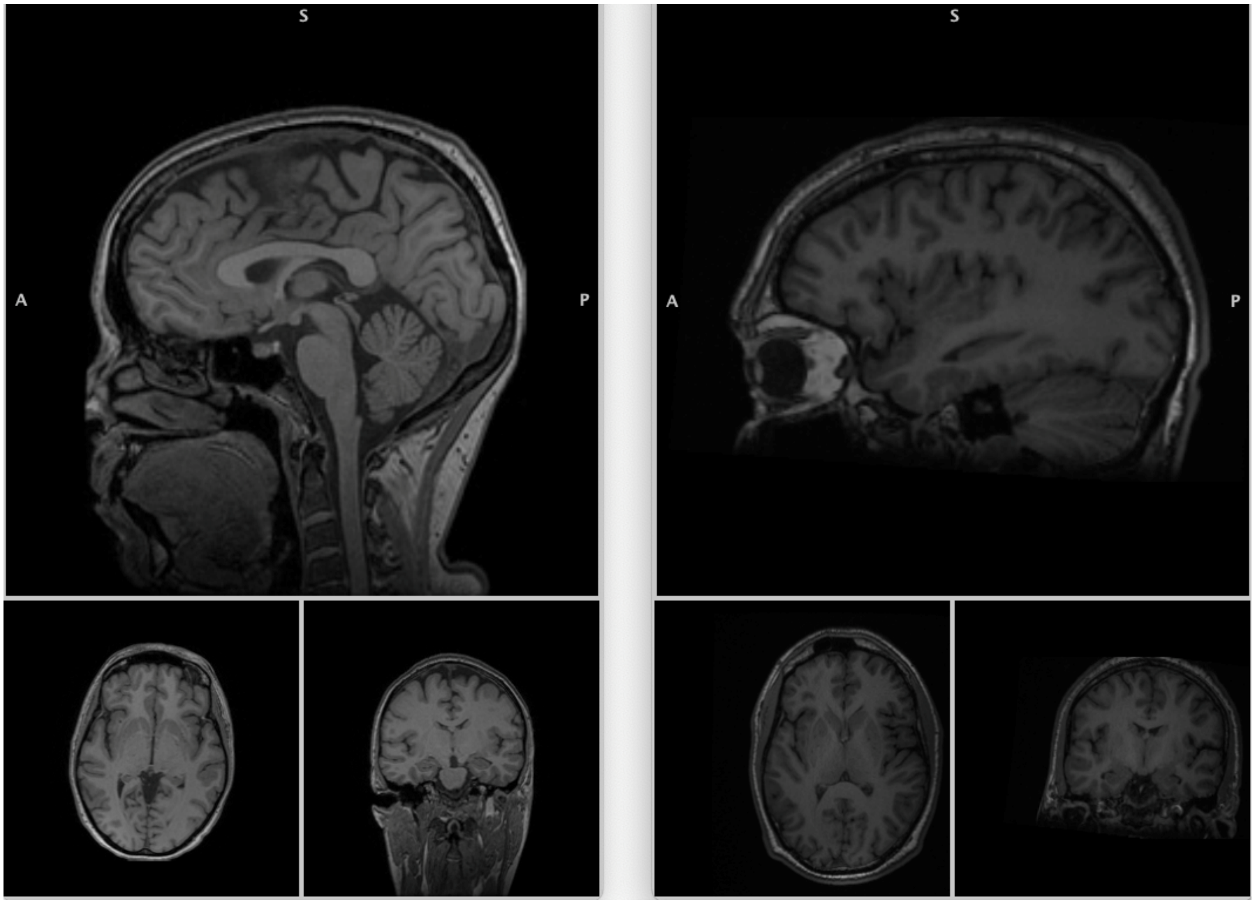


Figure 6.6: The left-hand side of the image shows the MRI of a healthy brain, and the right-hand side shows the non-lesional MRI of a patient with epilepsy. The MRIs of healthy brains contain the neck area.

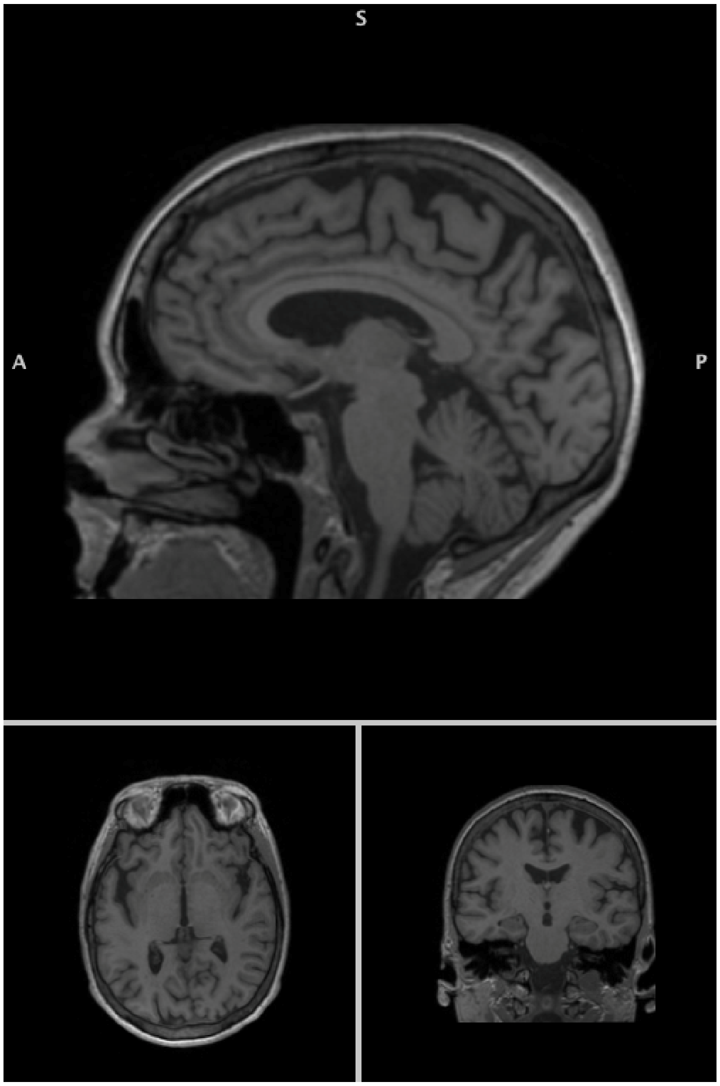


Figure 6.7: A sample of a healthy brain MRI in which its neck area is removed.

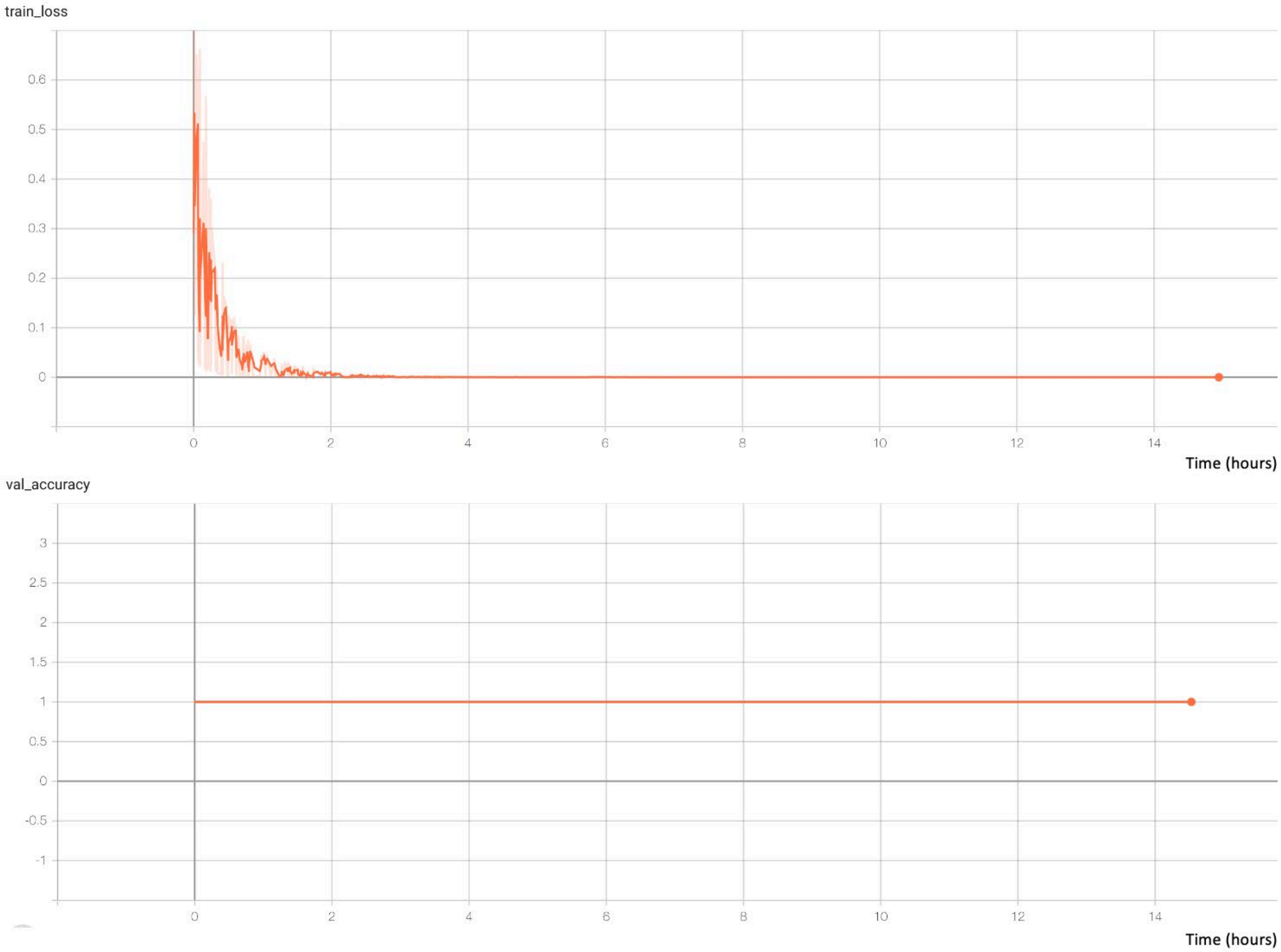


Figure 6.8: Validation accuracy and training loss graphs during training of a network which is aimed to distinguish between the cropped version of healthy control MRI and non-lesional MRI. The graphs are smoothed by 0.6.

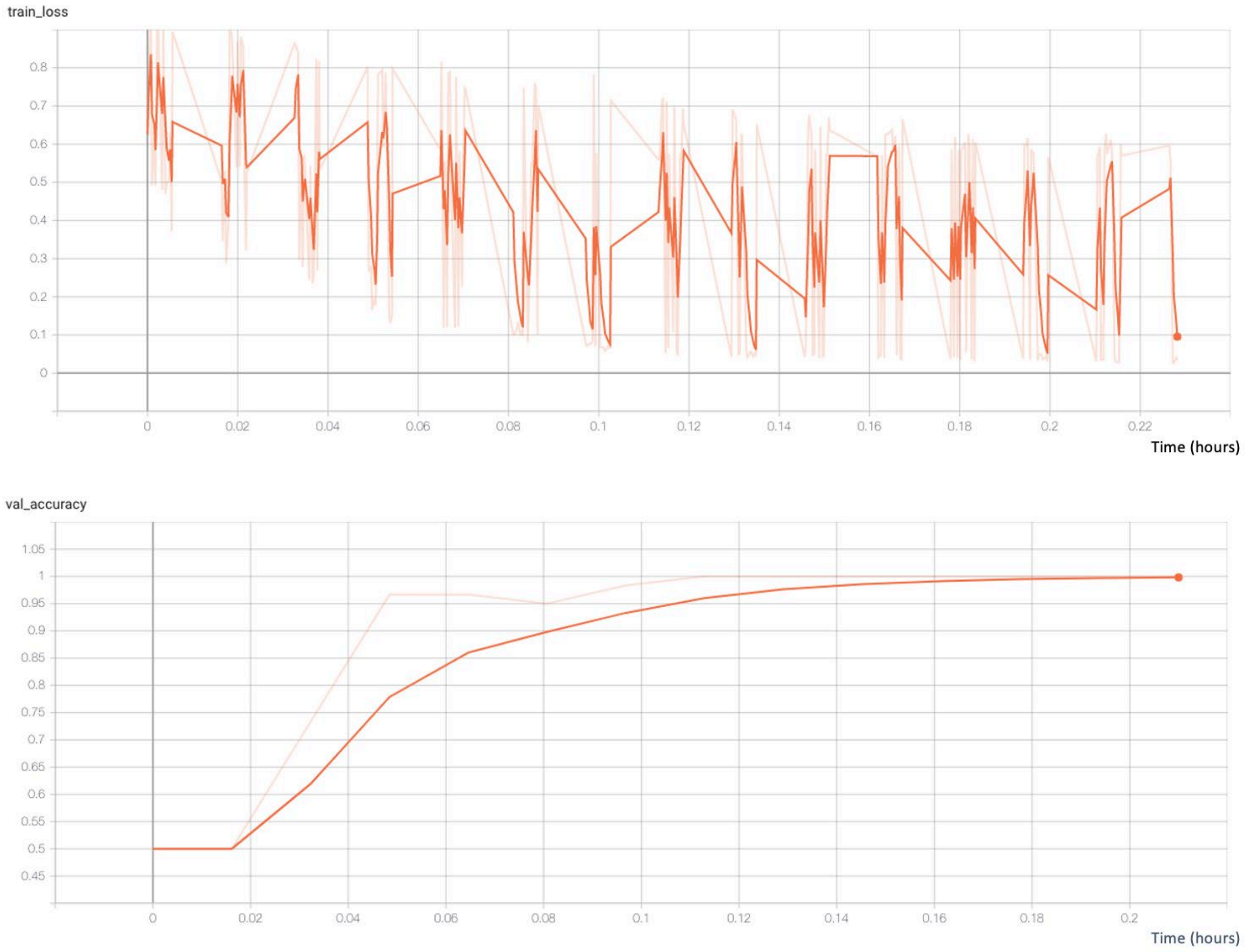


Figure 6.9: Validation accuracy and training loss graphs during training a network which is aimed to distinguish between healthy brain MRI and non-lesional MRI. The training set had 14 images. The graphs are smoothed by 0.6.

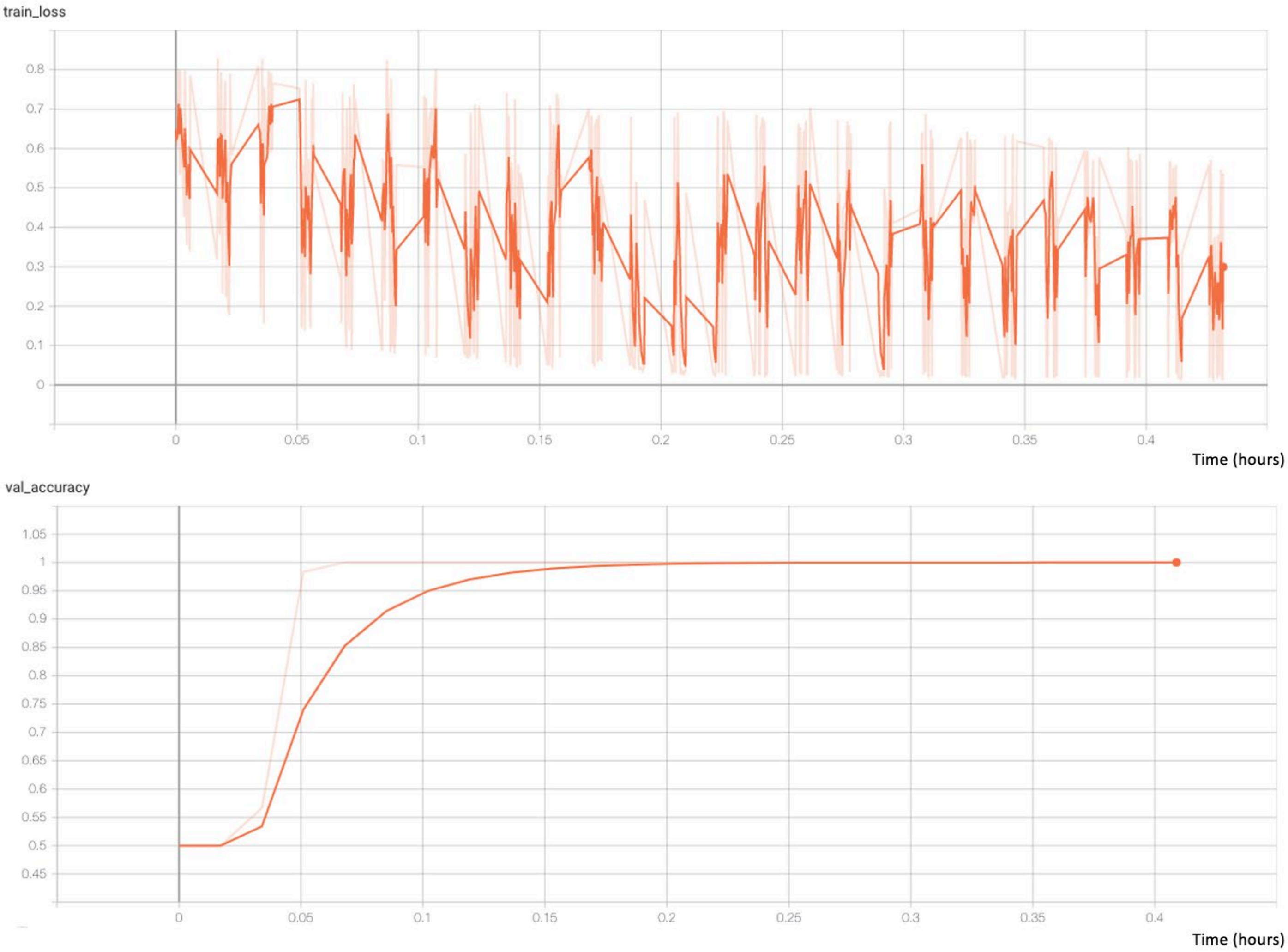


Figure 6.10: Validation accuracy and training loss graphs during training a network which is aimed to distinguish between healthy brain MRI and non-lesional MRI. The training set had 14 images. The dimensions of the MRIs were set to (211, 256, 204). The graphs are smoothed by 0.6.

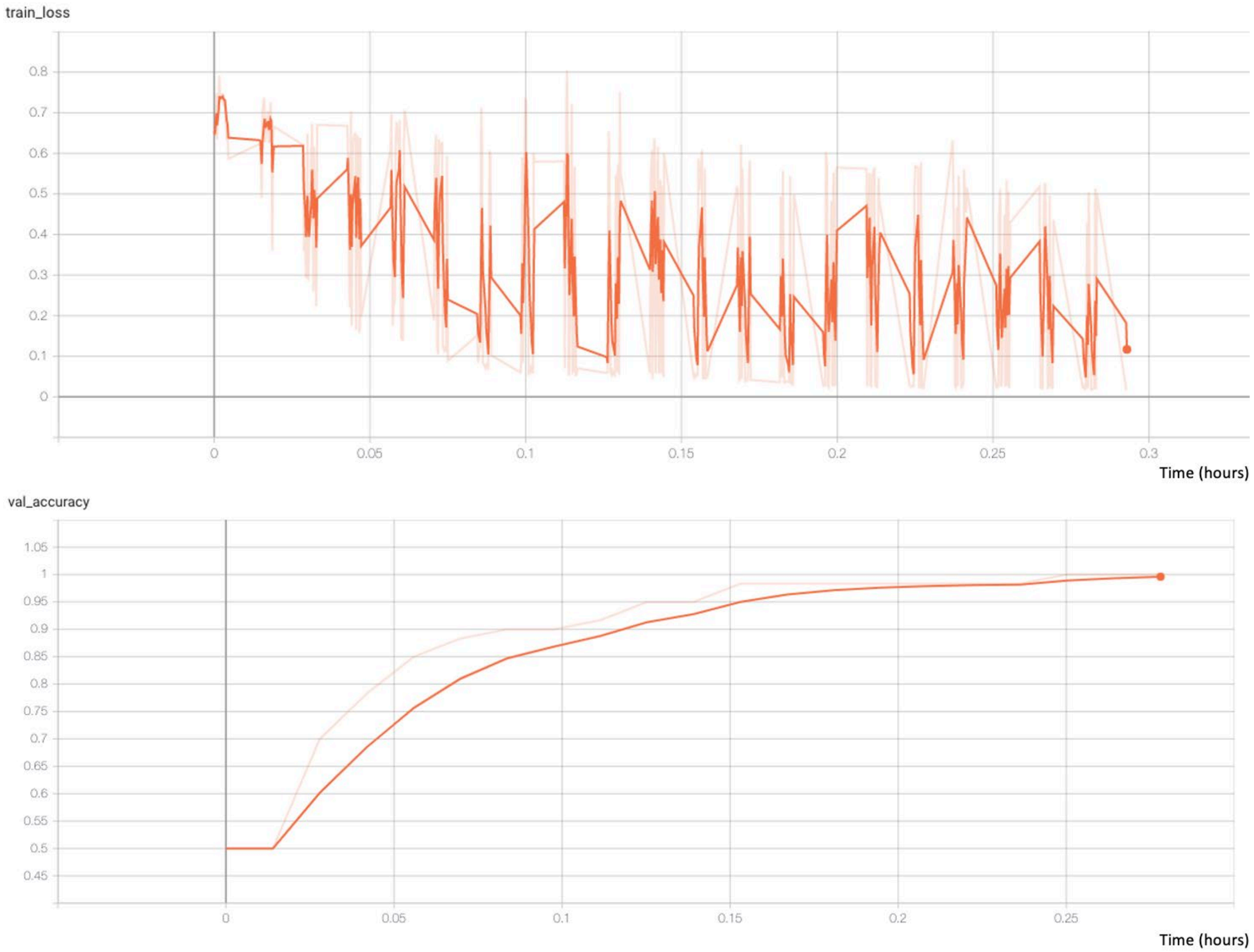


Figure 6.11: Validation accuracy and training loss graphs during the training of a network which is aimed to differentiate between cropped healthy brain MRI and non-lesional MRI. The training set had 14 images. The dimensions of the MRIs were set to the dimension of non-lesional MRIs. The graphs are smoothed by 0.6.

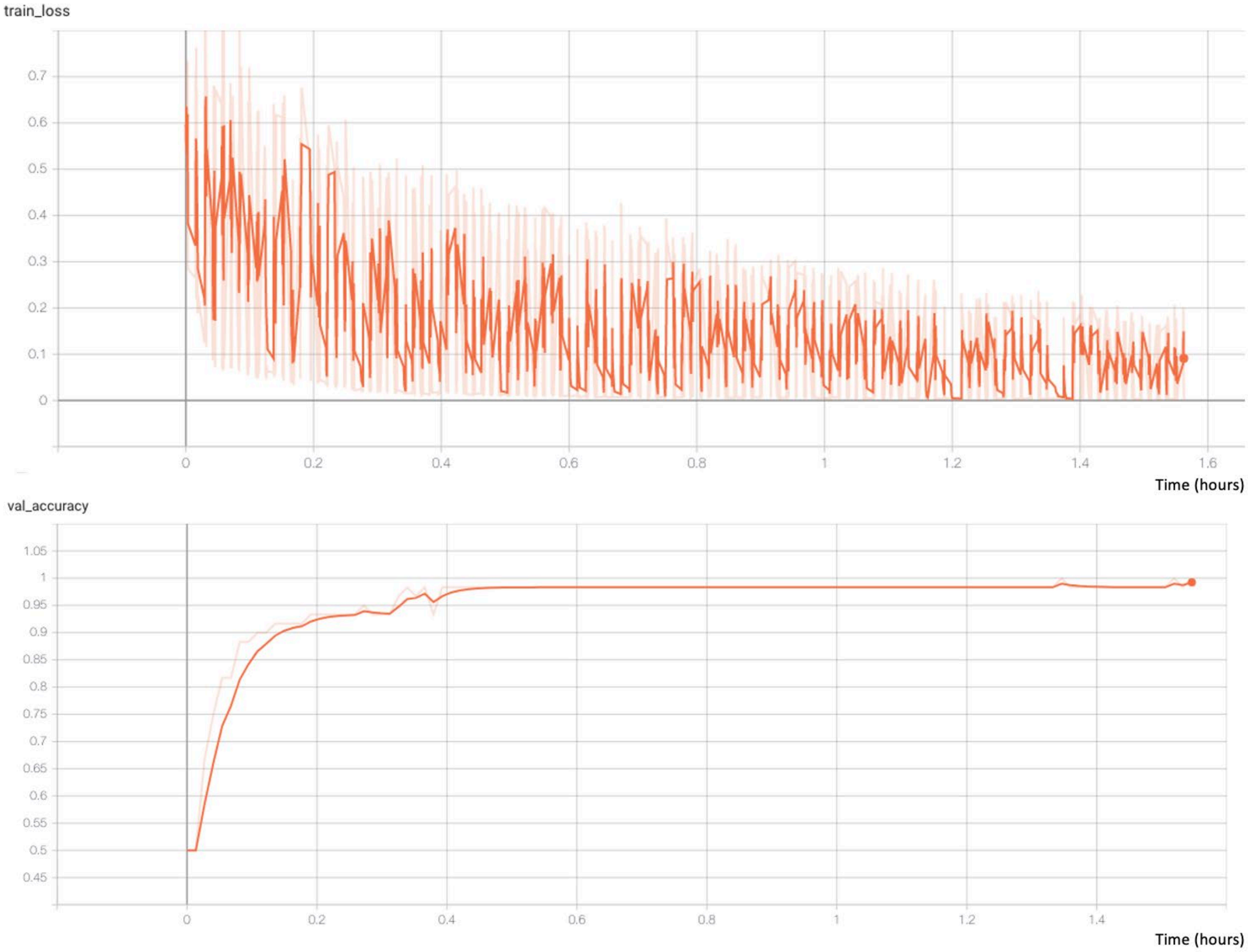


Figure 6.12: Validation accuracy and training loss during training of a network which is aimed to differentiate between cropped healthy brain MRI and non-lesional MRI. The training set had 14 images. The dimensions of the MRIs were set to (211, 256,161 . The graphs are smoothed by 0.6.

All the above experiments suggested that there are significant differences between the two sets of MRIs. Given that we received the images from two different settings, the differences can be related to the characteristics of the images, such as dimensions. Using the trained networks and deploying the Occlusion Sensitivity method, we created the heatmaps for MRIs. The heatmap confirmed the theory as it showed that in many layers of MRI, all the sections of the brain area are triggering the classification (Figure 13).

The following is a summary of the steps taken during this work. Compiling the steps would help to clarify why we concluded that using the images that we have access to would not help answer our questions.

- As the first step, we trained a network using a set of non-lesional MRIs and healthy control MRIs. The goal was to have a network which can differentiate between the two sets. During the training, we noticed that the network converges very quickly and achieves 100% accuracy on both the validation and test sets. This shows that the network is discriminating based on very pronounced differences, while we were hoping that the network would pick on subtle differences which would lead to finding the lesion.
- In the second step, we thought maybe improving the database and homogenizing both sets would help. We noticed that most MRIs from the healthy brain set contain the neck area. As a result, we removed the necks to see if we can overcome the problem. It did not solve the problem.
- In the last step, we visualized the reason behind the classifications. This visualization confirmed that the network is picking on very major differences.

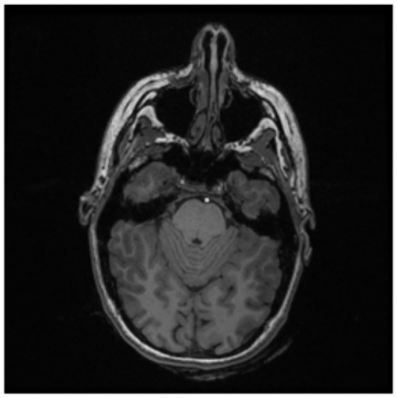
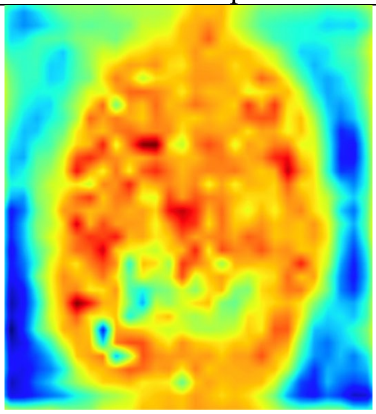
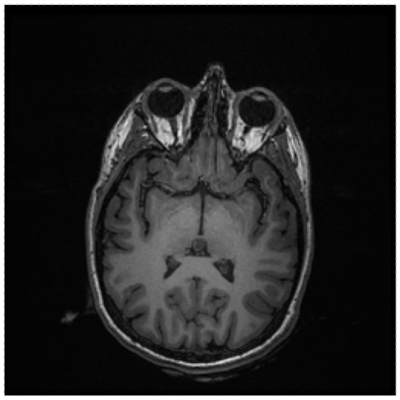
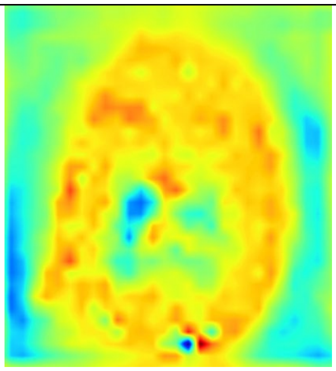
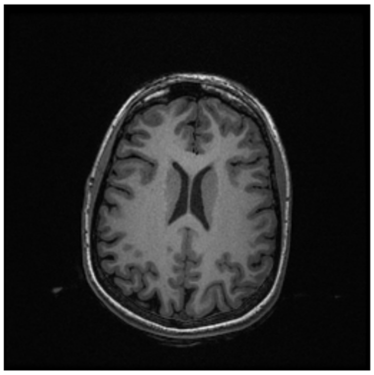
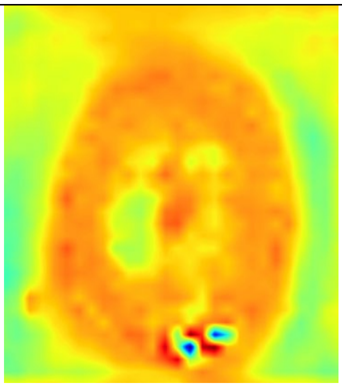
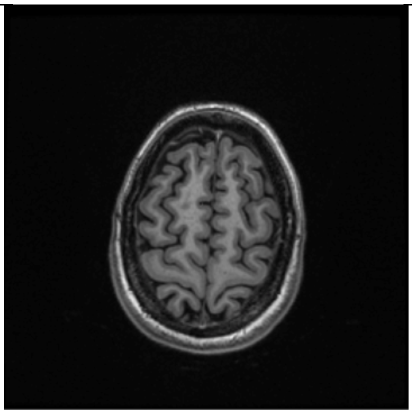
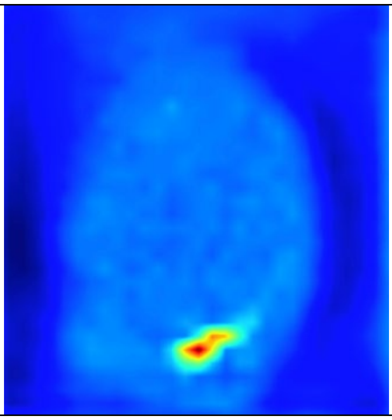
Original MRI	Heatmap
	
	
	
	

Figure13: Heatmap of MRI for the network which is trained to distinguish between healthy contorl MRI and non-lesional MRI

We did not pursue other visualization methods for a few reasons. Two of the reasons are listed below:

- We know the network is not using the features that we are interested in
- Although the Occlusion method seems to be the most simplistic method, there are medical image related studies which have identified that it worked better than the other methods for them (Aminu et al., 2021).

The current issues could be resolved if the healthy control images were from more similar settings compared to the non-lesional group. On the other hand, there is a low probability that healthy brains and normal looking brains, which have epilepsy, have major differences.

5.3 Conclusion

A subgroup of patients with epilepsy who are not responsive to available drugs may be candidates for surgery. In the clinical workflow, when deciding which patients would be the best candidate, different diagnostic tools are used. These tools include EEG, Video EEG, MRI, fMRI, PET, and SPECT. Using the mentioned tools, clinicians determine the location of surgery. Finding abnormality in the MRI would have a paramount value in indicating the location of the surgery and is indicative of the chances of success of the surgery. As a result, designing tools which can help clinicians find the abnormality in MRI would have a huge impact on the surgery outcome, specifically for cases where clinicians cannot find any abnormality.

The main purpose of the mentioned attempts in this chapter was to use Deep Learning to find an abnormality or location of seizure in MRI of patients that there was no abnormality reported for them. To design such a tool, we decided to train a network to distinguish between the mentioned

MRIs and MRIs of healthy brains and then use the different methods to identify which area of the MRI has caused the classification. We speculated that the abnormal area probably would play a role as a distinguishing factor. Unfortunately, when training the networks, a very strong signal not related to the subtle difference was playing the role of the distinguishing factor, and consequently, we could not further analyze the work.

It seems to us that the idea of using Deep Learning to find abnormalities in MRI still has the potential to be worked on the different set of MRIs. MRIs produced with more consistent variables from the same machine should be investigated and compared using Deep Learning. For example, as the first step, the MRIs of healthy brains and the MRIs of patients with epilepsy should be gathered in the same settings with the exact same MRI variables. In the second step, it would be a good idea to focus on the specific type of epilepsy, such as temporal epilepsy, as opposed to using MRI of different types of epilepsy. Furthermore, if necessary, MRIs should be cropped, and only specific areas of MRIs should be fed to the network.

6 Conclusion and Future Work

Epilepsy is the most reported neurological disorder after migraine, stroke, and Alzheimer's. Around 40% of patients with epilepsy are drug-resistant and can be candidates for surgery. In epilepsy centers, usually, a team of clinicians provides consultations on the benefits and risks of having surgery for cases which are considered to be candidates for surgery. Diagnostic tools such as MRI, EEG, intracranial EEG, and PET help clinicians provide recommendations. Between 60% to 70% of patients who undergo surgery would become seizure-free after surgery. This number is substantially lower for patients with no abnormality reported in their MRI. As a result, providing decision support tools which can help physicians in the process of evaluating patients with epilepsy for surgery would be very useful.

In general, clinical decision support systems are designed to improve healthcare outcome. For example, they may be designed to prevent mistakes or to bring consistency in providing care. Depending on the nature of the tools, they also can help healthcare systems in areas of the world which are challenged by lack of proper resources. New advancements in AI algorithms, along with using them in clinical decision support tools, have been very promising, especially in the imaging area. With the rise of Deep Learning and reinforcement learning algorithms, complex clinical tasks are being automated by machines, and advanced diagnostic tools are being designed and developed. Consequently, using machine learning to provide tools to deal with epilepsy has been the focus of many studies in the last decade. In this work, we have worked on using machine learning algorithms to provide tools for physicians when providing care for patients with epilepsy.

Using Deep Learning, we have worked on models which can be the core of decision support tools that can help when dealing with drug-resistant epilepsy cases. More specifically, we have worked on three main topics.

- Research Topic One: To investigate if it is possible to differentiate between normal and abnormal MRIs using Deep Learning.
    - Findings:

      We developed a model which would distinguish between MRI of patients with epilepsy with abnormality and MRI of patients without any abnormality. The purpose of this type of model is to be a decision support tool for physicians when they are seeking lesions on MRI of focal epilepsy cases. Our work indicates that there is a signal in MRI which can be harnessed for this purpose.
    - Limitations and future work:
        - Usually, training Deep Learning models requires a large number of data and more data results in better precision. Considering this fact, it would make sense to gather more data and train models to investigate if it is possible to achieve better results.
        - Training a model using MRIs from different centers would help to conclude if our findings can be generalized. Future work can investigate this by using data from different centers.
        - Probably training models using MRI of similar types of epilepsy would result in better accuracy. More data from different centers would give a chance to train CNNs for a specific type of epilepsy.

        - Having access to better hardware can help improve the performance of the model. In our case, we could only have 2 MRIs for each batch. This can be increased and probably would help with the performance of the model.
        - Training models which can differentiate between MRIs of brains with histopathological lesions and MRIs of brains with no histopathological lesion. When a clinician does not find any abnormality on MRI, but the model suggests that there may be a lesion, the clinician would be encouraged to reevaluate the MRI. It has been shown that, in some cases, reevaluation of non-lesional MRIs after knowing that there are histopathological lesions would result in finding an abnormality on the MRI (Bien et al., 2009).

- Research Topic Two: Is it possible to predict the result of surgery using MRI?
    - Findings:
        - Our finding indicates that there is a signal in both lesional and non-lesional MRIs to predict the result of surgery.
    - Limitations and future work:
        - More MRIs would help to improve the results and enables the researchers to use better evaluation methods. As stated before, our finding indicates that there is a signal in MRIs which can be harnessed for our task. Having access to enough data probably eliminated the need to use transfer learning, and we can train a CNN from scratch to predict the result of surgery. In our project, we have used the leave-one-out methodology. With enough data, better evaluation methods can be deployed. Also, using

        data from multiple centers would help to evaluate if our findings can be generalized. Furthermore, more data would also enable the researchers to train the models for specific types of epilepsy.

- Research Topic Three: Investigating the possibility of helping physicians in finding abnormalities in MRIs which are reported to be non-lesional by comparing MRIs of healthy brains and MRIs of non-lesional brains using Deep Learning.
    - Our finding: Our data set of two classes had major differences, which resulted in not being able to achieve what we were looking for.
    - Future work: Trying to see if there are other ways to deploy such ideas to find abnormalities.

As explained in the above sections, our experiments show that in the first two projects, there is a signal in MRI to be harnessed for the desired classifications, and we speculate that more data would improve the performance of the models. To solidify the findings of models, such as the ones explored in our work, the models need to be tested against data from different centers. There are a few methods to use data from different centers. One way would be to combine and normalize data from different centres and then use them for training. The second method would be to train a model in one center, use transfer learning and then test on the data of other centres. Federated learning is another option which can be considered for learning from different centers. Federated learning is a machine learning method which trains a model across multiple data centers while not storing data from different centers in one place. In this method, different models are trained in local centers, and learning parameters are shared between the models to generate a global model.

Further research can enhance the experiments that we have conducted. For example, data from different sources, such as EEG or clinical notes, can be fed to the models as input. At least two different architectures can be used to combine the data. One method would be training different models for each type of data and then ensemble the results from the different models (Figure 7.1a). Another possible option would be, using data from different resources and training one model (Figure 7.1b). For example, features extracted from image data using the convolution layers can be combined with the other data types and then be provided to a classifier.

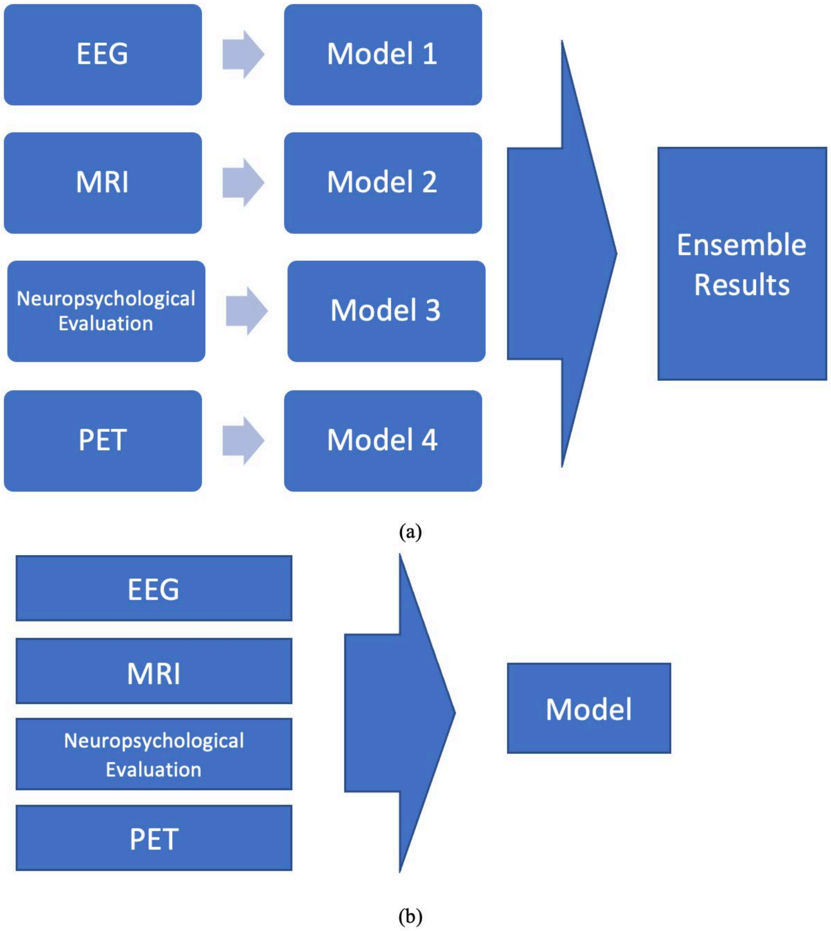


Figure 7.1: Methods to use data from different sources. Training different models each for a specific data type and then ensemble the results(a). Feeding different data types to one model and train it (b).

Data, the Bottleneck

The biggest challenge that we encountered throughout the projects was the lack of enough data, which is needed when working with Deep Learning algorithms. It seems that lack of access to a large number of data is the main issue for the medical image community as well. With the rise of Deep Learning, it was expected that decision support systems would bring huge changes to practices related to medical images and radiology. Although most probably, these changes will be materialized, more data is needed to build and deploy such systems, and gathering such data requires more time. It can be argued that the data probably exists, but proper processes should be established to provide access to the data.

Outside of the medical world, machine learning based models are seamlessly integrated into our lives. Examples of such models include Alexa and Siri. There are also models with astonishing output which have attracted media attention. An example of such models is LaMDA, a language model developed by Google. LaMDA is not the only example, there are other interesting models which have been trained by a very large number of data points. GPT-3 (a language model developed by OpenAI), Imagen (a text-to-image model developed by Google), or DALL-E (a text-to-image model developed by OpenAI) are other examples of such models.

Comparing the mentioned models with models which have been trained and deployed to medical settings indicates a huge difference between the two. One of the main explanations could be related to the availability of data for training models in the medical world. The comparison also shows the untapped potential for training models in medical fields once enough data is available. This does not mean that currently, there are no impactful, deployed models in clinical settings.

Examples of such models include an early warning system developed and deployed in St. Michael's Hospital in Toronto (Antoniou & Mamdani, 2021), skin lesion analysis applications, or pathology-related applications available on the market. More of such applications and devices will be used in medical settings soon. In basic science fields, which can affect medical settings in the future, the impact of sophisticated ML models has been materialized. An example of that type of model is AlphaFold, which recently has been used to publish the predicted structure of more than 200 million proteins (*AlphaFold Reveals the Structure of the Protein Universe*, n.d.). As the health record is moving toward digitalization, Electronic Health Record (HER) systems which are used for a large number of patients, or projects such as Observational Health Data Science and Informatics (OHDSI) (*OHDSI – Observational Health Data Sciences and Informatics*, n.d.) are going to be the basis for many future ML projects. Considering the current efforts for data gathering in healthcare systems and data gathered through wearable devices, the future of healthcare data would present a different landscape.

Other Challenges

Improving the results of the projects worked on in this thesis can be enabled by focusing on the following:

1. Using more and better data
2. Using better methods and algorithms
3. Using better hardware

There are also other challenges which must be considered when working on machine learning models that are aimed to be deployed in medical settings. The following is a short list of such challenges.

Patient Privacy and Model Inversion Methods

Protecting the privacy of patient data is an essential part of research projects, and any models which are trained with the aim of deployment in healthcare settings are bound by ethical constraints to protect the patient data. In the last decade, a new research area has emerged which investigates the possibility of recovering the training data from a trained model (Fredrikson et al., 2015). There are methods to prevent such actions. One example is using federated learning which was explained before. Another option would be adding noise to the data to minimize what can be retrieved. Preventing recovering data from a trained model is an ongoing and interesting field of research.

Representation of Different Demographic Groups

ML models targeted for medical settings must be trained on datasets which represent all ethnic and marginalized groups. For example, the dataset of MRIs, which is used to train a model, must include all groups within the target population and not just patients who can access the healthcare system due to socioeconomic factors. In general, when training models using regional data, the fact that training data is biased toward regional populations must be considered. The effects of demographic composition of datasets and how to address the problem of imbalanced datasets must be carefully considered.

Operationalizing Medical Models

Operationalizing medical models which have passed all the tests and are ready to be deployed has proven to be a challenging task. For example, a report from MIT Technology Review indicated that none of the predictive models developed during the COVID pandemic made a real difference (*Hundreds of AI Tools Have Been Built to Catch Covid. None of Them Helped. | MIT Technology Review*, n.d.). Along with acquiring approvals from regulatory bodies, proper multidisciplinary teams should be assembled to deploy such models in clinical settings. After deploying the model, it must be constantly monitored and improved based on new data and information. Monitoring, maintaining, and improving the model is an essential part of installing models in medical settings. This process is part of a well-known ML workflow (Figure 7.2). In some cases, after installing the models into the healthcare settings, proper change management approaches should be deployed to improve the efficiency of the clinical workflow. The impact of the installed systems on healthcare should also be evaluated and studied for improvement.

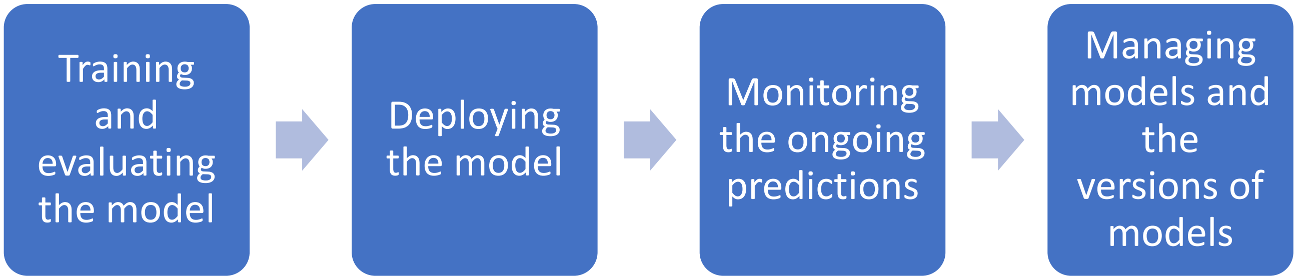

Training and evaluating the model
Deploying the model
Monitoring the ongoing predictions
Managing models and the versions of models

Figure 7.2: The ML workflow based on Google AI Guidelines - source: (*Machine Learning Workflow | AI Platform | Google Cloud*, n.d.)

Some of the above considerations are applicable to non-medical fields as well. Given the easier access to data in other fields compared to medicine, groups which work on them have valuable experience that can be used by groups that work on healthcare-related models. Advancements in ML and AI have opened new horizons for many fields, and it is expected that the effects of machine learning on medicine to be experienced in the upcoming decade. Assuming that is the case, patient care outcomes for neurological diseases such as epilepsy might also be substantially improved in the near future.

7 Bibliography

AlphaFold reveals the structure of the protein universe. (n.d.). Retrieved August 16, 2022, from https://www.deepmind.com/blog/alphafold-reveals-the-structure-of-the-protein-universe?utm_campaign=AF+200+Million&utm_medium=bitly&utm_source=LinkedIn+Organic

Aminu, M., Ahmad, N. A., & Mohd Noor, M. H. (2021). Covid-19 detection via deep neural network and occlusion sensitivity maps. *Alexandria Engineering Journal*, *60*(5), 4829–4855. https://doi.org/10.1016/J.AEJ.2021.03.052

Amit, G., Ben-Ari, R., Hadad, O., Monovich, E., Granot, N., & Hashoul, S. (2017). Classification of breast MRI lesions using small-size training sets: comparison of Deep Learning approaches. *Medical Imaging 2017: Computer-Aided Diagnosis*, *10134*, 101341H. https://doi.org/10.1117/12.2249981

Antoniou, T., & Mamdani, M. (2021). Evaluation of machine learning solutions in medicine. *CMAJ*, *193*(36), E1425–E1429. https://doi.org/10.1503/CMAJ.210036/TAB-RELATED-CONTENT

Arle, J. E., Kenneth, P., Devinsky, O., & Doyle, W. K. (1999). Neural network analysis of preoperative variables and outcome in epilepsy surgery in: Journal of Neurosurgery Volume 90 Issue 6 (1999). *Journal of Neurosurgery*, *90*(6), 998–1004. https://thejns.org/view/journals/j-neurosurg/90/6/article-p998.xml

Armañanzas, R., Alonso-Nanclares, L., DeFelipe-Oroquieta, J., Kastanauskaite, A., de Sola, R. G., DeFelipe, J., Bielza, C., & Larrañaga, P. (2013). Machine Learning Approach for the Outcome Prediction of Temporal Lobe Epilepsy Surgery. *PLoS ONE*, *8*(4). https://doi.org/10.1371/journal.pone.0062819

Arya, R., Leach, J. L., Horn, P. S., Greiner, H. M., Gelfand, M., Byars, A. W., Arthur, T. M., Tenney, J. R., Jain, S. V., Rozhkov, L., Fujiwara, H., Rose, D. F., Mangano, F. T., & Holland, K. D. (2016). Clinical factors predict surgical outcomes in pediatric MRI-negative drug-resistant epilepsy. *Seizure*, *41*, 56–61. https://doi.org/10.1016/j.seizure.2016.07.004

Atif, M., Sarwar, M. R., & Scahill, S. (2016). The relationship between epilepsy and sexual dysfunction: a review of the literature. *SpringerPlus*, *5*(1), 1–10. https://doi.org/10.1186/S40064-016-3753-5/METRICS

Bennett, O. F., Kanber, B., Hoskote, C., Cardoso, M. J., Ourselin, S., Duncan, J. S., & Winston, G. P. (2019). Learning to see the invisible: A data-driven approach to finding the underlying patterns of abnormality in visually normal brain magnetic resonance images in patients with temporal lobe epilepsy. *Epilepsia*, *60*(12), 2499–2507. https://doi.org/10.1111/EPI.16380

Bernasconi, A., Bernasconi, N., Bernhardt, B. C., & Schrader, D. (2011). Advances in MRI for "cryptogenic" epilepsies. In *Nature Reviews Neurology* (Vol. 7, Issue 2, pp. 99–108). https://doi.org/10.1038/nrneurol.2010.199

Bernasconi, A., Cendes, F., Theodore, W. H., Gill, R. S., Koepp, M. J., Hogan, R. E., Jackson, G. D., Federico, P., Labate, A., Vaudano, A. E., Blümcke, I., Ryvlin, P., & Bernasconi, N. (2019). Recommendations for the use of structural magnetic resonance imaging in the care of patients with epilepsy: A consensus report from the International League Against Epilepsy Neuroimaging Task Force. *Epilepsia*, *60*(6), 1054–1068. https://doi.org/10.1111/EPI.15612

Bernhardt, B. C., Hong, S. J., Bernasconi, A., & Bernasconi, N. (2015). Magnetic resonance imaging pattern learning in temporal lobe epilepsy: Classification and prognostics. *Annals of Neurology*, *77*(3), 436–446. https://doi.org/10.1002/ana.24341

Bien, C. G., Szinay, M., Wagner, J., Clusmann, H., Becker, A. J., & Urbach, H. (2009). Characteristics and surgical outcomes of patients with refractory magnetic resonance imaging-negative epilepsies. *Archives of Neurology*, *66*(12), 1491–1499. https://doi.org/10.1001/archneurol.2009.283

Bijay Dev, K. M., Jogi, P. S., Niyas, S., Vinayagamani, S., Kesavadas, C., & Rajan, J. (2019). Automatic detection and localization of Focal Cortical Dysplasia lesions in MRI using fully convolutional neural network. *Biomedical Signal Processing and Control*, *52*, 218–225. https://doi.org/10.1016/j.bspc.2019.04.024

Bonilha, L., Helpern, J. A., Sainju, R., Nesland, T., Edwards, J. C., Glazier, S. S., & Tabesh, A. (2013). Presurgical connectome and postsurgical seizure control in temporal lobe epilepsy. *Neurology*, *81*(19), 1704–1710. https://doi.org/10.1212/01.wnl.0000435306.95271.5f

Bonilha, L., Jensen, J. H., Baker, N., Breedlove, J., Nesland, T., Lin, J. J., Drane, D. L., Saindane, A. M., Binder, J. R., & Kuzniecky, R. I. (2015). The brain connectome as a personalized biomarker of seizure outcomes after temporal lobectomy. *Neurology*, *84*(18), 1846–1853. https://doi.org/10.1212/WNL.0000000000001548

Boylan, L. S., Flint, L. A., Labovitz, D. L., Jackson, S. C., Starner, K., & Devinsky, O. (2004). Depression but not seizure frequency predicts quality of life in treatment-resistant epilepsy. *Neurology*, *62*(2), 258–261. https://doi.org/10.1212/01.WNL.0000103282.62353.85

Brain lobes - Mayo Clinic. (n.d.). Retrieved May 2, 2022, from https://www.mayoclinic.org/brain-lobes/img-20008887

Cervenka, M. C., Henry, B. J., Felton, E. A., Patton, K., & Kossoff, E. H. (2016). Establishing an Adult Epilepsy Diet Center: Experience, efficacy and challenges. *Epilepsy & Behavior : E&B*, *58*, 61–68. https://doi.org/10.1016/J.YEBEH.2016.02.038

Chapman, K., Wyllie, E., Najm, I., Ruggieri, P., Bingaman, W., Lüders, J., Kotagal, P., Lachhwani, D., Dinner, D., & Lüders, H. O. (2005). Seizure outcome after epilepsy surgery in patients with normal preoperative MRI. *Journal of Neurology, Neurosurgery and Psychiatry*, *76*(5), 710–713. https://doi.org/10.1136/jnnp.2003.026757

Chen, C., Qin, C., Qiu, H., Tarroni, G., Duan, J., Bai, W., & Rueckert, D. (2019). *Deep Learning for cardiac image segmentation: A review*. http://arxiv.org/abs/1911.03723

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-August-2016*, 785–794. https://doi.org/10.1145/2939672.2939785

Chollet, F. (2021). *Deep Learning with Python* (Second). Manning Publications Co.3 Lewis Street Greenwich, CTUnited States.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1023/A:1022627411411

Cossu, M., Lo Russo, G., Francione, S., Mai, R., Nobili, L., Sartori, I., Tassi, L., Citterio, A., Colombo, N., Bramerio, M., Galli, C., Castana, L., & Cardinale, F. (2008). Epilepsy surgery in children: Results and predictors of outcome on seizures. *Epilepsia*, *49*(1), 65–72. https://doi.org/10.1111/j.1528-1167.2007.01207.x

De Tisi, J., Bell, G. S., Peacock, J. L., McEvoy, A. W., Harkness, W. F., Sander, J. W., & Duncan, J. S. (2011). The long-term outcome of adult epilepsy surgery, patterns of seizure remission, and relapse: A cohort study. *The Lancet*, *378*(9800), 1388–1395. https://doi.org/10.1016/S0140-6736(11)60890-8

Definition of Epilepsy 2014 // International League Against Epilepsy. (n.d.). Retrieved April 30, 2022, from https://www.ilae.org/guidelines/definition-and-classification/definition-of-epilepsy-2014

DeSalvo, M. N., Douw, L., Tanaka, N., Reinsberger, C., & Stufflebeam, S. M. (2014). Altered structural connectome in temporal lobe epilepsy. *Radiology*, *270*(3), 842–848. https://doi.org/10.1148/radiol.13131044

Devinsky, O., Vezzani, A., O'Brien, T. J., Jette, N., Scheffer, I. E., De Curtis, M., & Perucca, P. (2018). Epilepsy. *Nature Reviews Disease Primers 2018 4:1*, *4*(1), 1–24. https://doi.org/10.1038/nrdp.2018.24

DLTK. (2018). *DLTK*. https://github.com/DLTK/DLTK/tree/master/examples/applications/IXI_HH_sex_classification_resnet

Dolz, J., Desrosiers, C., & Ayed, I. Ben. (2016). 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study. *NeuroImage*, *170*, 456–470. https://doi.org/10.1016/j.neuroimage.2017.04.039

Dong, H., Yang, G., Liu, F., Mo, Y., & Guo, Y. (2017). Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks. *Communications in Computer and Information Science*, *723*, 506–517. http://arxiv.org/abs/1705.03820

Dou, Q., Chen, H., Yu, L., Zhao, L., Qin, J., Wang, D., Mok, V. C. T., Shi, L., & Heng, P. A. (2016). Automatic Detection of Cerebral Microbleeds from MR Images via 3D Convolutional Neural Networks. *IEEE Transactions on Medical Imaging*, *35*(5), 1182–1195. https://doi.org/10.1109/TMI.2016.2528129

Dreifuss, S., Vingerhoets, F. J. G., Lazeyras, F., Andino, S. G., Spinelli, L., Delavelle, J., & Seeck, M. (2001). Volumetric measurements of subcortical nuclei in patients with temporal lobe epilepsy. *Neurology*, *57*(9), 1636–1641. https://doi.org/10.1212/WNL.57.9.1636

Dwivedi, R., Ramanujam, B., Chandra, P. S., Sapra, S., Gulati, S., Kalaivani, M., Garg, A., Bal, C. S., Tripathi, M., Dwivedi, S. N., Sagar, R., Sarkar, C., & Tripathi, M. (2017). Surgery for Drug-Resistant Epilepsy in Children. *The New England Journal of Medicine*, *377*(17), 1639–1647. https://doi.org/10.1056/NEJMOA1615335

Ebrahimighahnavieh, M. A., Luo, S., & Chiong, R. (2020). Deep Learning to detect Alzheimer's disease from neuroimaging: A systematic literature review. *Computer Methods and Programs in Biomedicine*, *187*, 105242. https://doi.org/10.1016/j.cmpb.2019.105242

Engel, J. (Ed.). (1993). *Surgical Treatment of the Epilepsies* (2nd Editio). Raven Press.

Engel, J., McDermott, M. P., Wiebe, S., Langfitt, J. T., Stern, J. M., Dewar, S., Sperling, M. R., Gardiner, I., Erba, G., Fried, I., Jacobs, M., Vinters, H. V., Mintzer, S., & Kieburtz, K. (2012). Early surgical therapy for drug-resistant temporal lobe epilepsy: a randomized trial. *JAMA*, *307*(9), 922–930. https://doi.org/10.1001/JAMA.2012.220

Engel Surgical Outcome Scale – MGH Epilepsy Service. (n.d.). Retrieved July 1, 2020, from http://seizure.mgh.harvard.edu/engel-surgical-outcome-scale/

Epilepsy surgery - Mayo Clinic. (n.d.). Retrieved May 15, 2022, from https://www.mayoclinic.org/tests-procedures/epilepsy-surgery/about/pac-20393981

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, *542*(7639), 115–118. https://doi.org/10.1038/nature21056

Farabet, C., Couprie, C., Najman, L., & Lecun, Y. (2013). Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1915–1929. https://doi.org/10.1109/TPAMI.2012.231

Farooq, A., Anwar, S., Awais, M., & Rehman, S. (2017). A deep CNN based multi-class classification of Alzheimer's disease using MRI. *IST 2017 - IEEE International Conference on Imaging Systems and Techniques, Proceedings*, *2018-January*, 1–6. https://doi.org/10.1109/IST.2017.8261460

Fazel, S., Wolf, A., Långström, N., Newton, C. R., & Lichtenstein, P. (2013). Premature mortality in epilepsy and the role of psychiatric comorbidity: A total population study. *The Lancet*, *382*(9905), 1646–1654. https://doi.org/10.1016/S0140-6736(13)60899-5/ATTACHMENT/1669EB7A-FC21-45EC-8F64-35C651E6860D/MMC1.PDF

Federico Epilepsy Neuroimaging | Home | Cumming School of Medicine | University of Calgary. (n.d.). Retrieved January 12, 2023, from https://cumming.ucalgary.ca/labs/federico-epilepsy-neuroimaging/federico-lab

Feis, D. L., Schoene-Bake, J. C., Elger, C., Wagner, J., Tittgemeyer, M., & Weber, B. (2013). Prediction of post-surgical seizure outcome in left mesial temporal lobe epilepsy. *NeuroImage: Clinical*, *2*(1), 903–911. https://doi.org/10.1016/j.nicl.2013.06.010

Feng, C., Zhao, H., Li, Y., & Wen, J. (2020). Automatic localization and segmentation of focal cortical dysplasia in FLAIR-negative patients using a convolutional neural network. *Journal of Applied Clinical Medical Physics*, *21*(9), 215–226. https://doi.org/10.1002/ACM2.12985

Fiest, K. M., Dykeman, J., Patten, S. B., Wiebe, S., Kaplan, G. G., Maxwell, C. J., Bulloch, A. G. M., & Jette, N. (2013). Depression in epilepsy: a systematic review and meta-analysis. *Neurology*, *80*(6), 590–599. https://doi.org/10.1212/WNL.0B013E31827B1AE0

Fiest, K. M., Sauro, K. M., Wiebe, S., Patten, S. B., Kwon, C. S., Dykeman, J., Pringsheim, T., Lorenzetti, D. L., & Jetté, N. (2017). Prevalence and incidence of epilepsy. In *Neurology* (Vol. 88, Issue 3, pp. 296–303). Lippincott Williams and Wilkins. https://doi.org/10.1212/WNL.0000000000003509

Filters in Convolutional Neural Networks. (n.d.). Retrieved August 28, 2022, from https://kharshit.github.io/blog/2018/12/14/filters-in-convolutional-neural-networks

Finding Non-Linear Decision Boundary in SVM | by Sourodip Kundu | Medium. (n.d.). Retrieved June 12, 2022, from https://medium.com/@KunduSourodip/finding-non-linear-decision-boundary-in-svm-a89a97a006d2

Fisher, R. S., Cross, J. H., French, J. A., Higurashi, N., Hirsch, E., Jansen, F. E., Lagae, L., Moshé, S. L., Peltola, J., Roulet Perez, E., Scheffer, I. E., & Zuberi, S. M. (2017). Operational classification of seizure types by the International League Against Epilepsy: Position Paper of the ILAE Commission for Classification and Terminology. *Epilepsia*, *58*(4), 522–530. https://doi.org/10.1111/EPI.13670

Fisher, R., Salanova, V., Witt, T., Worth, R., Henry, T., Gross, R., Oommen, K., Osorio, I., Nazzaro, J., Labar, D., Kaplitt, M., Sperling, M., Sandok, E., Neal, J., Handforth, A., Stern, J., DeSalles, A., Chung, S., Shetter, A., … Young, C. (2010). Electrical stimulation of the anterior nucleus of thalamus for treatment of refractory epilepsy. *Epilepsia*, *51*(5), 899–908. https://doi.org/10.1111/J.1528-1167.2010.02536.X

Focke, N. K., Bonelli, S. B., Yogarajah, M., Scott, C., Symms, M. R., & Duncan, J. S. (2009). Automated normalized FLAIR imaging in MRI-negative patients with refractory focal epilepsy. *Epilepsia*, *50*(6), 1484–1490. https://doi.org/10.1111/J.1528-1167.2009.02022.X

Focke, N. K., Yogarajah, M., Symms, M. R., Gruber, O., Paulus, W., & Duncan, J. S. (2012). Automated MR image classification in temporal lobe epilepsy. *NeuroImage*, *59*(1), 356–362. https://doi.org/10.1016/J.NEUROIMAGE.2011.07.068

Fredrikson, M., Jha, S., & Tech, C. (2015). Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures Thomas Ristenpart. *Proceedings of the 22nd ACM*

SIGSAC Conference on Computer and Communications Security. https://doi.org/10.1145/2810103

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4), 193–202. https://doi.org/10.1007/BF00344251

Fulton, L. V., Dolezel, D., Harrop, J., Yan, Y., & Fulton, C. P. (2019). Classification of alzheimer's disease with and without imagery using gradient boosted machines and resnet-50. *Brain Sciences*, *9*(9). https://doi.org/10.3390/brainsci9090212

Ghafoorian, M., Karssemeijer, N., Heskes, T., Van Uder, I. W. M., De Leeuw, F. E., Marchiori, E., Van Ginneken, B., & Platel, B. (2016). Non-uniform patch sampling with deep convolutional neural networks for white matter hyperintensity segmentation. *Proceedings - International Symposium on Biomedical Imaging*, *2016-June*, 1414–1417. https://doi.org/10.1109/ISBI.2016.7493532

Ghassemi, M., Oakden-Rayner, L., & Beam, A. L. (2021). The false hope of current approaches to explainable artificial intelligence in health care. *The Lancet Digital Health*, *3*(11), e745–e750. https://doi.org/10.1016/S2589-7500(21)00208-9/ATTACHMENT/F419A161-CE0C-4D86-8197-39B3190CA57B/MMC1.PDF

GitHub - martisak/dotnets: Create simple drawings of neural networks using graphviz. (n.d.). Retrieved June 12, 2022, from https://github.com/martisak/dotnets

GitHub - takyamamoto/Hopfield-Network: Hopfield network implemented with Python. (n.d.). Retrieved July 22, 2022, from https://github.com/takyamamoto/Hopfield-Network

Gleichgerrcht, E., Munsell, B., Bhatia, S., Vandergrift, W. A., Rorden, C., McDonald, C., Edwards, J., Kuzniecky, R., & Bonilha, L. (2018). Deep Learning applied to whole-brain connectome to determine seizure control after epilepsy surgery. *Epilepsia*, *59*(9), 1643–1654. https://doi.org/10.1111/epi.14528

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, *3*(January), 2672–2680. https://doi.org/10.3156/jsoft.29.5_177_2

Graves, A. (2013). *Generating Sequences With Recurrent Neural Networks*. http://arxiv.org/abs/1308.0850

Grewal, S. S., Alvi, M. A., Perkins, W. J., Cascino, G. D., Britton, J. W., Burkholder, D. B., So, E., Shin, C., Marsh, R. W., Meyer, F. B., Worrell, G. A., & Van Gompel, J. J. (2019). Reassessing the impact of intraoperative electrocorticography on postoperative outcome of patients undergoing standard temporal lobectomy for MRI-negative temporal lobe epilepsy. *Journal of Neurosurgery*, 1–10. https://doi.org/10.3171/2018.11.jns182124

Grigsby, J., Kramer, R. E., Schneiders, J. L., Gates, J. R., & Smith, W. B. (1998). Predicting outcome of anterior temporal lobectomy using simulated neural networks. *Epilepsia*, *39*(1), 61–66. https://doi.org/10.1111/j.1528-1157.1998.tb01275.x

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., Kim, R., Raman, R., Nelson, P. C., Mega, J. L., & Webster, D. R. (2016). Development and validation of a Deep Learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA - Journal of the American Medical Association*, *316*(22), 2402–2410. https://doi.org/10.1001/jama.2016.17216

Hauser, W. A., & Beghi, E. (2008). First seizure definitions and worldwide incidence and mortality. *Epilepsia*, *49 Suppl 1*(SUPPL. 1), 8–12. https://doi.org/10.1111/J.1528-1167.2008.01443.X

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision*, *2015 Inter*, 1026–1034. https://doi.org/10.1109/ICCV.2015.123

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Home | Vascular Imaging | Cumming School of Medicine | University of Calgary. (n.d.). Retrieved January 12, 2023, from https://cumming.ucalgary.ca/labs/vascular-imaging/vascular-imaging

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, *79*(8), 2554. https://doi.org/10.1073/PNAS.79.8.2554

Hopfield Network (HN) - PRIMO.ai. (n.d.). Retrieved June 12, 2022, from http://primo.ai/index.php?title=Hopfield_Network_%28HN%29

Hopfield Networks is All You Need | hopfield-layers. (n.d.). Retrieved July 21, 2022, from https://ml-jku.github.io/hopfield-layers/

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-Excitation Networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 7132–7141. https://doi.org/10.1109/CVPR.2018.00745

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). *Densely Connected*

Convolutional Networks (pp. 4700–4708). https://github.com/liuzhuang13/DenseNet.

Hundreds of AI tools have been built to catch covid. None of them helped. | MIT Technology Review. (n.d.). Retrieved July 27, 2022, from https://www.technologyreview.com/2021/07/30/1030329/machine-learning-ai-failed-covid-hospital-diagnosis-pandemic/

Iizuka, T., Fukasawa, M., & Kameyama, M. (2019). Deep-learning-based imaging-classification identified cingulate island sign in dementia with Lewy bodies. *Scientific Reports 2019 9:1*, *9*(1), 1–9. https://doi.org/10.1038/s41598-019-45415-5

Inbaraj, X. A., Villavicencio, C., Macrohon, J. J., Jeng, J. H., & Hsieh, J. G. (2021). Object Identification and Localization Using Grad-CAM++ with Mask Regional Convolution Neural Network. *Electronics 2021, Vol. 10, Page 1541*, *10*(13), 1541. https://doi.org/10.3390/ELECTRONICS10131541

Jayakar, P., Dunoyer, C., Dean, P., Ragheb, J., Resnick, T., Morrison, G., Bhatia, S., & Duchowny, M. (2008). Epilepsy surgery in patients with normal or nonfocal MRI scans: Integrative strategies offer long-term seizure relief. *Epilepsia*, *49*(5), 758–764. https://doi.org/10.1111/j.1528-1167.2007.01428.x

Jeha, L. E., Najm, I., Bingaman, W., Dinner, D., Widdess-Walsh, P., & Lüders, H. (2007). Surgical outcome and prognostic factors of frontal lobe epilepsy surgery. *Brain*, *130*(2), 574–584. https://doi.org/10.1093/brain/awl364

Kang, H. C., Chung, D. E., Kim, D. W., & Kim, H. D. (2004). Early- and late-onset complications of the ketogenic diet for intractable epilepsy. *Epilepsia*, *45*(9), 1116–1123. https://doi.org/10.1111/J.0013-9580.2004.10004.X

Kearney, H., Byrne, S., Cavalleri, G. L., & Delanty, N. (2019). Tackling Epilepsy With High-definition Precision Medicine: A Review. *JAMA Neurology*, *76*(9), 1109–1116. https://doi.org/10.1001/JAMANEUROL.2019.2384

Korolev, S., Safiullin, A., Belyaev, M., & Dodonova, Y. (2017). Residual and plain convolutional neural networks for 3D brain MRI classification. *Proceedings - International Symposium on Biomedical Imaging*, 835–838. https://doi.org/10.1109/ISBI.2017.7950647

Koul, A., Ganju, S., & Kasam, M. (2019). *Practical Deep Learning for Cloud, Mobile & Edge* (First). O'Reilly.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 1097–1105. http://code.google.com/p/cuda-convnet/

Kwan, P., & Brodie, M. J. (2000). Early identification of refractory epilepsy. *New England Journal of Medicine*, *342*(5), 314–319. https://doi.org/10.1056/NEJM200002033420503

Lascano, A. M., Perneger, T., Vulliemoz, S., Spinelli, L., Garibotto, V., Korff, C. M., Vargas, M. I., Michel, C. M., & Seeck, M. (2016). Yield of MRI, high-density electric source imaging (HD-ESI), SPECT and PET in epilepsy surgery candidates. *Clinical Neurophysiology*, *127*(1), 150–155. https://doi.org/10.1016/j.clinph.2015.03.025

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2323. https://doi.org/10.1109/5.726791

Lefevre, F., & Aronson, N. (2000). Ketogenic diet for the treatment of refractory epilepsy in children: A systematic review of efficacy. *Pediatrics*, *105*(4). https://doi.org/10.1542/PEDS.105.4.E46

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak,

J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on Deep Learning in medical image analysis. *Medical Image Analysis*, *42*(1995), 60–88. https://doi.org/10.1016/j.media.2017.07.005

Liu, M., Chen, Z., Beaulieu, C., & Gross, D. W. (2014). Disrupted anatomic white matter network in left mesial temporal lobe epilepsy. *Epilepsia*, *55*(5), 674–682. https://doi.org/10.1111/epi.12581

Lo, S. C. B., Lou, S. L. A., Chien, M. V., & Mun, S. K. (1995). Artificial Convolution Neural Network Techniques and Applications for Lung Nodule Detection. *IEEE Transactions on Medical Imaging*, *14*(4), 711–718. https://doi.org/10.1109/42.476112

Lopinto-Khoury, C., Sperling, M. R., Skidmore, C., Nei, M., Evans, J., Sharan, A., & Mintzer, S. (2012). Surgical outcome in PET-positive, MRI-negative patients with temporal lobe epilepsy. *Epilepsia*, *53*(2), 342–348. https://doi.org/10.1111/j.1528-1167.2011.03359.x

Machine learning workflow | AI Platform | Google Cloud. (n.d.). Retrieved August 16, 2022, from https://cloud.google.com/ai-platform/docs/ml-solutions-overview

Majkowska, A., Mittal, S., Steiner, D. F., Reicher, J. J., McKinney, S. M., Duggan, G. E., Eswaran, K., Cameron Chen, P.-H., Liu, Y., Kalidindi, S. R., Ding, A., Corrado, G. S., Tse, D., & Shetty, S. (2020). Chest Radiograph Interpretation with Deep Learning Models: Assessment with Radiologist-adjudicated Reference Standards and Population-adjusted Evaluation. *Radiology*, *294*(2), 421–431. https://doi.org/10.1148/radiol.2019191293

Marson, A., Jacoby, A., Johnson, A., Kim, L., Gamble, C., & Chadwick, D. (2005). Immediate versus deferred antiepileptic drug treatment for early epilepsy and single seizures: a randomised controlled trial. *Lancet (London, England)*, *365*(9476), 2007–2013. https://doi.org/10.1016/S0140-6736(05)66694-9

Martin, P., Winston, G. P., Bartlett, P., de Tisi, J., Duncan, J. S., & Focke, N. K. (2017). Voxel-based magnetic resonance image postprocessing in epilepsy. *Epilepsia*, *58*(9), 1653–1664. https://doi.org/10.1111/epi.13851

McKee, H. R., & Privitera, M. D. (2017). Stress as a seizure precipitant: Identification, associated factors, and treatment options. *Seizure*, *44*, 21–26. https://doi.org/10.1016/J.SEIZURE.2016.12.009

Memarian, N., Kim, S., Dewar, S., Engel, J., & Staba, R. J. (2015). Multimodal data and machine learning for surgery outcome prediction in complicated cases of mesial temporal lobe epilepsy. *Computers in Biology and Medicine*, *64*, 67–78. https://doi.org/10.1016/j.compbiomed.2015.06.008

Milby, A. H., Halpern, C. H., & Baltuch, G. H. (2009). Vagus nerve stimulation in the treatment of refractory epilepsy. *Neurotherapeutics*, *6*(2), 228. https://doi.org/10.1016/J.NURT.2009.01.010

Minsky, M., & Papert, S. (1969). *Perceptrons; an introduction to computational geometry*. 258.

MRI Basics. (n.d.). Retrieved May 18, 2022, from https://case.edu/med/neurology/NR/MRI Basics.htm#:~:text=The most common MRI sequences,longer TE and TR times.

Munsell, B. C., Wee, C. Y., Keller, S. S., Weber, B., Elger, C., da Silva, L. A. T., Nesland, T., Styner, M., Shen, D., & Bonilha, L. (2015). Evaluation of machine learning algorithms for treatment outcome prediction in patients with epilepsy based on structural connectome data. *NeuroImage*, *118*, 219–230. https://doi.org/10.1016/j.neuroimage.2015.06.008

Ngugi, A. K., Bottomley, C., Kleinschmidt, I., Sander, J. W., & Newton, C. R. (2010). Estimation of the burden of active and life-time epilepsy: a meta-analytic approach. *Epilepsia*, *51*(5), 883–890. https://doi.org/10.1111/J.1528-1167.2009.02481.X

Nie, D., Zhang, H., Adeli, E., Liu, L., & Shen, D. (2016). 3D Deep Learning for multi-modal imaging-guided survival time prediction of brain tumor patients. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9901 LNCS*, 212–220. https://doi.org/10.1007/978-3-319-46723-8_25

Noe, K., Sulc, V., Wong-Kisiel, L., Wirrell, E., Van Gompel, J. J., Wetjen, N., Britton, J., So, E., Cascino, G. D., Marsh, R., Meyer, F., Horinek, D., Giannini, C., Watson, R., Brinkmann, B. H., Stead, M., & Worrell, G. A. (2013). Long-term outcomes after nonlesional extratemporal lobe epilepsy surgery. *JAMA Neurology*, *70*(8), 1003–1008. https://doi.org/10.1001/jamaneurol.2013.209

OHDSI – Observational Health Data Sciences and Informatics. (n.d.). Retrieved August 16, 2022, from https://www.ohdsi.org/

Original Guidelines - Ontario Epilepsy Guidelines. (n.d.). Retrieved May 9, 2022, from https://ontarioepilepsyguidelines.ca/original-guidelines/

Pardoe, H. R., Pell, G. S., Abbott, D. F., & Jackson, G. D. (2009). Hippocampal volume assessment in temporal lobe epilepsy: How good is automated segmentation? *Epilepsia*, *50*(12), 2586–2592. https://doi.org/10.1111/J.1528-1167.2009.02243.X

Park, S. A., Lim, S. R., Kim, G. S., Heo, K., Park, S. C., Chang, J. W., Chung, S. S., Choi, J. U., Kim, T. S., & Lee, B. I. (2002). Ictal electrocorticographic findings related with surgical outcomes in nonlesional neocortical epilepsy. *Epilepsy Research*. https://doi.org/10.1016/S0920-1211(02)00006-2

Pedregosa, F., Michel, V., Grisel OLIVIERGRISEL, O., Blondel, M., Prettenhofer, P., Weiss, R., Vanderplas, J., Cournapeau, D., Pedregosa, F., Varoquaux, G., Gramfort, A., Thirion, B., Grisel, O., Dubourg, V., Passos, A., Brucher, M., Perrot andÉdouardand, M., Duchesnay, A., & Duchesnay EDOUARDDUCHESNAY, Fré. (2011). Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot. *Journal of Machine Learning Research*, *12*, 2825–2830. http://scikit-learn.sourceforge.net.

Perera-Ortega, A., Sedghi, A., Isen, J., Vos, S. B., Mousavi, P., & Winston, G. P. (2021). Machine learning to detect brain lesions in focal epilepsy. *Https://Doi.Org/10.1117/12.2581075*, *11598*, 297–302. https://doi.org/10.1117/12.2581075

Pinaya, W. H. L., Tudosiu, P.-D., Dafflon, J., da Costa, P. F., Fernandez, V., Nachev, P., Ourselin, S., & Cardoso, M. J. (2022). *Brain Imaging Generation with Latent Diffusion Models*. https://doi.org/10.48550/arxiv.2209.07162

Pitkänen, A., Löscher, W., Vezzani, A., Becker, A. J., Simonato, M., Lukasiuk, K., Gröhn, O., Bankstahl, J. P., Friedman, A., Aronica, E., Gorter, J. A., Ravizza, T., Sisodiya, S. M., Kokaia, M., & Beck, H. (2016). Advances in the development of biomarkers for epilepsy. *The Lancet. Neurology*, *15*(8), 843–856. https://doi.org/10.1016/S1474-4422(16)00112-5

Pulsipher, D. T., Seidenberg, M., Morton, J. J., Geary, E., Parrish, J., & Hermann, B. (2007). MRI volume loss of subcortical structures in unilateral temporal lobe epilepsy. *Epilepsy & Behavior*, *11*(3), 442–449. https://doi.org/10.1016/J.YEBEH.2007.08.007

Raghu, M., Zhang, C., Kleinberg, J., & Bengio, S. (2019). *Transfusion: Understanding Transfer Learning for Medical Imaging*. http://arxiv.org/abs/1902.07208

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M. P., & Ng, A. Y. (2017). *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*. http://arxiv.org/abs/1711.05225

Ramachandrannair, R., Otsubo, H., Shroff, M. M., Ochi, A., Weiss, S. K., Rutka, J. T., & Snead, O. C. (2007). MEG predicts outcome following surgery for intractable epilepsy in children with normal or nonfocal MRI findings. *Epilepsia*, *48*(1), 149–157. https://doi.org/10.1111/j.1528-1167.2006.00901.x

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). *Hierarchical Text-Conditional Image Generation with CLIP Latents*. https://doi.org/10.48550/arxiv.2204.06125

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., Greiff, V., Kreil, D., Kopp, M., Klambauer, G., Brandstetter, J., & Hochreiter, S. (2020). *Hopfield Networks is All You Need*. https://doi.org/10.48550/arxiv.2008.02217

Ritchie, S. J., Cox, S. R., Shen, X., Lombardo, M. V, Reus, L. M., Alloza, C., Harris, M. A., Alderson, H. L., Hunter, S., Neilson, E., Liewald, D. C. M., Auyeung, B., Whalley, H. C., Lawrie, S. M., Gale, C. R., Bastin, M. E., McIntosh, A. M., & Deary, I. J. (2018). Sex Differences in the Adult Human Brain: Evidence from 5216 UK Biobank Participants. *Cerebral Cortex (New York, NY)*, *28*(8), 2959. https://doi.org/10.1093/CERCOR/BHY109

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9351*, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Rosebrock, A. (2017). *Deep Learning for Computer Vision with Python*. PYIMAGESEARCH.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408.

https://doi.org/10.1037/h0042519

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Neurocomputing: Foundations of Research. In J. A. Anderson & E. Rosenfeld (Eds.), *Neurocomputing: Foundations of Research* (pp. 696–699). MIT Press. http://dl.acm.org/citation.cfm?id=65669.104451

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Ryvlin, P., So, E. L., Gordon, C. M., Hesdorffer, D. C., Sperling, M. R., Devinsky, O., Bunker, M. T., Olin, B., & Friedman, D. (2018). Long-term surveillance of SUDEP in drug-resistant epilepsy patients treated with VNS therapy. *Epilepsia*, *59*(3), 562–572. https://doi.org/10.1111/EPI.14002

Samala, R. K., Chan, H. P., Hadjiiski, L., Helvie, M. A., Wei, J., & Cha, K. (2016). Mass detection in digital breast tomosynthesis: Deep convolutional neural network with transfer learning from mammography. *Medical Physics*, *43*(12), 6654–6666. https://doi.org/10.1118/1.4967345

See, S. J., Jehi, L. E., Vadera, S., Bulacio, J., Najm, I., & Bingaman, W. (2013). Surgical outcomes in patients with extratemporal epilepsy and subtle or normal magnetic resonance imaging findings. *Neurosurgery*, *73*(1), 68–76. https://doi.org/10.1227/01.neu.0000429839.76460.b7

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2016). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*, *128*(2), 336–359.

https://doi.org/10.1007/s11263-019-01228-7

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-October*, 618–626. https://doi.org/10.1109/ICCV.2017.74

Senders, J. T., Staples, P. C., Karhade, A. V., Zaki, M. M., Gormley, W. B., Broekman, M. L. D., Smith, T. R., & Arnaout, O. (2018). Machine Learning and Neurosurgical Outcome Prediction: A Systematic Review. *World Neurosurgery*, *109*, 476-486.e1. https://doi.org/10.1016/j.wneu.2017.09.149

Seo, J. H., Noh, B. H., Lee, J. S., Kim, D. S., Lee, S. K., Kim, T. S., Kim, S. H., Kang, H. C., & Kim, H. D. (2009). Outcome of surgical treatment in non-lesional intractable childhood epilepsy. *Seizure*, *18*(9), 625–629. https://doi.org/10.1016/j.seizure.2009.07.007

Shi, J., Lacuey, N., & Lhatoo, S. (2017). Surgical outcome of MRI-negative refractory extratemporal lobe epilepsy. *Epilepsy Research*. https://doi.org/10.1016/j.eplepsyres.2017.04.010

Simonovsky, M., Gutiérrez-Becker, B., Mateus, D., Navab, N., & Komodakis, N. (2016). A Deep Metric for Multimodal Registration. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9902 LNCS*, 10–18. http://arxiv.org/abs/1609.05396

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Singh, A., & Trevick, S. (2016). The Epidemiology of Global Epilepsy. *Neurologic Clinics*, *34*(4), 837–847. https://doi.org/10.1016/J.NCL.2016.06.015

Sixt, L., Wild, B., & Landgraf, T. (2016). RenderGAN: Generating Realistic Labeled Data. *5th International Conference on Learning Representations, ICLR 2017 - Workshop Track Proceedings*. http://arxiv.org/abs/1611.01331

Skandarani, Y., Jodoin, P.-M., & Lalande, A. (2021). *GANs for Medical Image Synthesis: An Empirical Study*. https://doi.org/10.48550/arxiv.2105.05318

Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958. https://doi.org/10.5555/2627435

Stafstrom, C. E., & Carmant, L. (2015). Seizures and Epilepsy: An Overview for Neuroscientists. *Cold Spring Harbor Perspectives in Medicine*, *5*(6), 1–19. https://doi.org/10.1101/CSHPERSPECT.A022426

Strine, T. W., Kobau, R., Chapman, D. P., Thurman, D. J., Price, P., & Balluz, L. S. (2005). Psychological distress, comorbidities, and health behaviors among U.S. adults with seizures: results from the 2002 National Health Interview Survey. *Epilepsia*, *46*(7), 1133–1139. https://doi.org/10.1111/J.1528-1167.2005.01605.X

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07-12-June-2015*, 1–9. https://doi.org/10.1109/CVPR.2015.7298594

Tavakol, S., Royer, J., Lowe, A. J., Bonilha, L., Tracy, J. I., Jackson, G. D., Duncan, J. S., Bernasconi, A., Bernasconi, N., & Bernhardt, B. C. (2019). Neuroimaging and connectomics of drug-resistant epilepsy at multiple scales: From focal lesions to macroscale networks. *Epilepsia*, *60*(4), 593–604. https://doi.org/10.1111/epi.14688

Taylor, P. N., Sinha, N., Wang, Y., Vos, S. B., de Tisi, J., Miserocchi, A., McEvoy, A. W., Winston, G. P., & Duncan, J. S. (2018). The impact of epilepsy surgery on the structural connectome and its relation to outcome. *NeuroImage. Clinical*, *18*, 202–214. https://doi.org/10.1016/j.nicl.2018.01.028

Téllez-Zenteno, J. F., Dhar, R., & Wiebe, S. (2005). Long-term seizure outcomes following epilepsy surgery: a systematic review and meta-analysis. *Brain*, *128*(5), 1188–1198. https://doi.org/10.1093/brain/awh449

Téllez-Zenteno, J. F., Ronquillo, L. H., Moien-Afshari, F., & Wiebe, S. (2010). Surgical outcomes in lesional and non-lesional epilepsy: A systematic review and meta-analysis. *Epilepsy Research*, *89*(2–3), 310–318. https://doi.org/10.1016/j.eplepsyres.2010.02.007

Theodore, W. H., Spencer, S. S., Wiebe, S., Langfitt, J. T., Ali, A., Shafer, P. O., Berg, A. T., & Vickrey, B. G. (2006). Epilepsy in North America: A Report Prepared under the Auspices of the Global Campaign against Epilepsy, the International Bureau for Epilepsy, the International League Against Epilepsy, and the World Health Organization. *Epilepsia*, *47*(10), 1700–1722. https://doi.org/10.1111/j.1528-1167.2006.00633.x

Tonini, C., Beghi, E., Berg, A. T., Bogliun, G., Giordano, L., Newton, R. W., Tetto, A., Vitelli, E., Vitezic, D., & Wiebe, S. (2004). Predictors of epilepsy surgery outcome: a meta-analysis. *Epilepsy Research*, *62*(1), 75–87. https://doi.org/10.1016/J.EPLEPSYRES.2004.08.006

Vagus Nerve Stimulation (VNS) Therapy - Epilepsy Queensland. (n.d.). Retrieved May 18, 2022, from https://epilepsyqueensland.com.au/about-epilepsy-epilepsy-queensland/epilepsy-treatment/vagus-nerve-stimulation-vns-therapy/

Wang, H., Ahmed, S. N., & Mandal, M. (2020). Automated detection of focal cortical dysplasia using a deep convolutional neural network. *Computerized Medical Imaging and Graphics : The Official Journal of the Computerized Medical Imaging Society*, *79*. https://doi.org/10.1016/J.COMPMEDIMAG.2019.101662

Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard University.

Wiebe, S., Bellhouse, D. R., Fallahay, C., & Eliasziw, M. (1999). Burden of epilepsy: The Ontario Health Survey. *Canadian Journal of Neurological Sciences*, *26*(4), 263–270. https://doi.org/10.1017/S0317167100000354

Wiebe, S., Blume, W. T., Girvin, J. P., & Eliasziw, M. (2001). A Randomized, Controlled Trial of Surgery for Temporal-Lobe Epilepsy. *New England Journal of Medicine*, *345*(5), 311–318. https://doi.org/10.1056/NEJM200108023450501

Wieser, H. G., Blume, W. T., Fish, D., Goldensohn, E., Hufnagel, A., King, D., Sperling, M. R., Lüders, H., & Pedley, T. A. (2001). Proposal for a new classification of outcome with respect to epileptic seizures following epilepsy surgery. *Epilepsia*, *42*(2), 282–286. https://doi.org/10.1046/j.1528-1157.2001.35100.x

Willard, A., Antonic-Baker, A., Chen, Z., O'Brien, T. J., Kwan, P., & Perucca, P. (2022). Seizure Outcome After Surgery for MRI-Diagnosed Focal Cortical Dysplasia: A Systematic Review and Meta-analysis. *Neurology*, *98*(3), e236–e248. https://doi.org/10.1212/WNL.0000000000013066

Wolterink, J. M., Leiner, T., de Vos, B. D., van Hamersvelt, R. W., Viergever, M. A., & Išgum, I. (2016). Automatic coronary artery calcium scoring in cardiac CT angiography using paired convolutional neural networks. *Medical Image Analysis*, *34*, 123–136.

https://doi.org/10.1016/j.media.2016.04.004

Yang, C., Rangarajan, A., & Ranka, S. (2018). Visual Explanations From Deep 3D Convolutional Neural Networks for Alzheimer's Disease Classification. *AMIA ... Annual Symposium Proceedings. AMIA Symposium*, *2018*, 1571–1580.

Yang, W., Chen, Y., Liu, Y., Zhong, L., Qin, G., Lu, Z., Feng, Q., & Chen, W. (2017). Cascade of multi-scale convolutional neural networks for bone suppression of chest radiographs in gradient domain. *Medical Image Analysis*, *35*, 421–433. https://doi.org/10.1016/j.media.2016.08.004

Yang, Y., & Wang, X. (2016). Sexual dysfunction related to antiepileptic drugs in patients with epilepsy. *Expert Opinion on Drug Safety*, *15*(1), 31–42. https://doi.org/10.1517/14740338.2016.1112376

Yankam Njiwa, J., Gray, K. R., Costes, N., Mauguiere, F., Ryvlin, P., & Hammers, A. (2015). Advanced [18F]FDG and [11C]flumazenil PET analysis for individual outcome prediction after temporal lobe epilepsy surgery for hippocampal sclerosis. *NeuroImage: Clinical*, *7*, 122–131. https://doi.org/10.1016/j.nicl.2014.11.013

Yi, X., Walia, E., & Babyn, P. (2018). Generative Adversarial Network in Medical Imaging: A Review. *Medical Image Analysis*, *58*. https://doi.org/10.1016/j.media.2019.101552

Zakaria, T., Noe, K., So, E., Cascino, G. D., Wetjen, N., Van Gompel, J. J., Marsh, W. R., Meyer, F. B., Giannini, C., Watson, R. E., & Worrell, G. A. (2012). Scalp and Intracranial EEG in Medically Intractable Extratemporal Epilepsy with Normal MRI. *ISRN Neurology*, *2012*, 1–9. https://doi.org/10.5402/2012/942849

Zeiler, M. D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. *Computer Vision–ECCV 2014*, *8689*(PART 1), 818–833. https://doi.org/10.1007/978-3-

319-10590-1_53

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2015). Learning Deep Features for Discriminative Localization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 2921–2929. http://arxiv.org/abs/1512.04150

Zilly, J., Buhmann, J. M., & Mahapatra, D. (2017). Glaucoma detection using entropy sampling and ensemble learning for automatic optic cup and disc segmentation. *Computerized Medical Imaging and Graphics*, *55*, 28–41. https://doi.org/10.1016/j.compmedimag.2016.07.012

8 Appendix A

```
DenseNet(
  (features): Sequential(
    (conv0): Conv3d(1, 64, kernel_size=(7, 7, 7), stride=(2, 2, 2), paddin
g=(3, 3, 3), bias=False)
    (norm0): BatchNorm3d(64, eps=1e-05, momentum=0.1, affine=True, track_r
unning_stats=True)
    (relu0): ReLU(inplace=True)
    (pool0): MaxPool3d(kernel_size=3, stride=2, padding=1, dilation=1, cei
l_mode=False)
    (denseblock1): _DenseBlock(
      (denselayer1): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(64, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(64, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1)
, bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer2): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(96, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(96, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1)
, bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer3): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(128, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
```

```
        )
      )
      (denselayer4): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(160, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer5): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(192, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer6): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(224, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(224, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
    )
    (transition1): _Transition(
      (norm): BatchNorm3d(256, eps=1e-05, momentum=0.1, affine=True, track
_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv): Conv3d(256, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1), bi
as=False)
      (pool): AvgPool3d(kernel_size=2, stride=2, padding=0)
    )
    (denseblock2): _DenseBlock(
      (denselayer1): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(128, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer2): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(160, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer3): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(192, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer4): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(224, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(224, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
```

```
        )
      )
      (denselayer5): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(256, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer6): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(288, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(288, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer7): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(320, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(320, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer8): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(352, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(352, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer9): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(384, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer10): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(416, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(416, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer11): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(448, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(448, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer12): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(480, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
```

```
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
    )
    (transition2): _Transition(
      (norm): BatchNorm3d(512, eps=1e-05, momentum=0.1, affine=True, track
_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv): Conv3d(512, 256, kernel_size=(1, 1, 1), stride=(1, 1, 1), bi
as=False)
      (pool): AvgPool3d(kernel_size=2, stride=2, padding=0)
    )
    (denseblock3): _DenseBlock(
      (denselayer1): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(256, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer2): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(288, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(288, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer3): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(320, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(320, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer4): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(352, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(352, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer5): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(384, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer6): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(416, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(416, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer7): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(448, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(448, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
```

```
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer8): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(480, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer9): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(512, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer10): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(544, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(544, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer11): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(576, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer12): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(608, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(608, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer13): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(640, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(640, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer14): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(672, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer15): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(704, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(704, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer16): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(736, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(736, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer17): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(768, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(768, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer18): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(800, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(800, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer19): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(832, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(832, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer20): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(864, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(864, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer21): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(896, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(896, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer22): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(928, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(928, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
```

```
        )
      )
      (denselayer23): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(960, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer24): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(992, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(992, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
    )
    (transition3): _Transition(
      (norm): BatchNorm3d(1024, eps=1e-05, momentum=0.1, affine=True, trac
k_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv): Conv3d(1024, 512, kernel_size=(1, 1, 1), stride=(1, 1, 1), b
ias=False)
      (pool): AvgPool3d(kernel_size=2, stride=2, padding=0)
    )
    (denseblock4): _DenseBlock(
      (denselayer1): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(512, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer2): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(544, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(544, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer3): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(576, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer4): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(608, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(608, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer5): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(640, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(640, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
```

```
        )
      )
      (denselayer6): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(672, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer7): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(704, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(704, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer8): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(736, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(736, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer9): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(768, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(768, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer10): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(800, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(800, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer11): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(832, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(832, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer12): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(864, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(864, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer13): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(896, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(896, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
```

```
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer14): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(928, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(928, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer15): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(960, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
      (denselayer16): _DenseLayer(
        (layers): Sequential(
          (norm1): BatchNorm3d(992, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu1): ReLU(inplace=True)
          (conv1): Conv3d(992, 128, kernel_size=(1, 1, 1), stride=(1, 1, 1
), bias=False)
          (norm2): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (relu2): ReLU(inplace=True)
          (conv2): Conv3d(128, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1)
, padding=(1, 1, 1), bias=False)
        )
      )
    )
    (norm5): BatchNorm3d(1024, eps=1e-05, momentum=0.1, affine=True, track
_running_stats=True)
  )
)
```