Universitetet
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

BACHELOROPPGAVE

Study programme / specialisation: Computer Science	The Spring semester 2024 Open / Confidential
Author: Sigurd Heggemoen & Stian Vedelden Robberstad	
Supervisors: Alvaro Fernandez Quilez & Anna Kurbatskaya	
Thesis title: Deep learning-based EEG analysis for Parkinson's disease Credits (ECTS): 20	
Emneord: CNN, EEG, Deep Learning, Parkinson's Disease, Spectrogram	Pages: 70 Stavanger 15. mai 2024

Contents

Acronyms and Abbreviations

ANN Artificial Neural Network

AUC Area Under the Receiver Operating Characteristic Curve

Bradykinesia Slowed Movements

CNN Convolutional Neural Network

CT Computerized Tomography

CV Cross-Validation

CWT Continuous Wavelet Transform

DL Deep Learning

DNN Deep Neural Network

DWT Discrete Wavelet Transform

EEG Electroencephalogram (The recording of the gathered EEG data)

EEG Electroencephalography (The method of gathering the EEG data)

FFT Fast Fourier Transform

FN False Negatives

FP False Positives

FT Fourier Transform

MDS-UPDRS Movement Disorder Society Unified Parkinson's Disease Rating Scale

ML Machine Learning

MRI Magnetic Resonance Imaging

PD Parkinson's Disease

ResNet Residual Network

ROC Receiver Operating Characteristic

TN True Negatives

TP True Positives

Tremor Rhytmic Shaking

Abstract

Parkinson's Disease (PD) is the fastest-growing neurological condition, with over 10 million cases worldwide. The most common way of diagnosing PD has been based on the observation of symptoms of the subject. Electroencephalography (EEG) is a non-invasive technique that allows clinicians to analyse brain patterns and has the potential to detect PD. Manually analysing EEG signals requires a skilled and trained expert to interpret the results accurately. For this reason, Deep Learning (DL) has the potential to bypass this step and detect PD automatically based on the EEG signals. Such models have been developed, with most reaching very promising results. However, some of these models do not practice a strict division of subjects on the subject level, resulting in potential cross-contamination of data between the testing and training set, not representing a real test scenario.

In our work, we aim to compare the performance of DL models in detecting PD using EEG signals represented as time-frequency series and spectrograms as inputs. Additionally, we aim to assess the difference in model performance between 2D and 3D DL architectures. Two implementations were tested for the 3D models: subject and segment level. For that purpose, we used the datasets from Turku, California, and New Mexico, which contained in total 50 PD and 60 non-PD subjects.

Our results show that incorporating the splitting on the subject level reduces the model's performance compared to other approaches that do not perform that splitting, with the best Area Under the Receiver Operating Characteristic Curve (AUC) and subject level accuracy, respectively, being 0.69 ± 0.07 and $60.00\% \pm 4.97\%$ achieved by EEGNet run on 320x320 spec-

trograms. In addition, combining the 244x244 spectrograms on a subject level using the ResNet18 3D model showed a better result compared to the standard 2D ResNet18 model without ImageNet, with an AUC score of 0.68 ± 0.09 vs. 0.64 ± 0.09, respectively. These results indicate that EEG has the potential for detection and the performance benefits of representing EEG signals as spectrograms.

Acknowledgements

We want to thank our two supervisors, Álvaro Fernández Quílez and Anna Kurbatskaya, for their guidance and help with the thesis. We are especially grateful for quickly providing a good baseline, guiding us down the correct path, and for quick email responses.

Disclosure

Grammarly Premium, containing AI, was used in this project as a tool for proofreading and enhancing the readability of the thesis. ChatGPT has also been used to help with reformulations but not as a source of information for the project.

Chapter 1

Introduction

1.1 Motivation

Parkinson's Disease (PD) is a condition that gradually reduces the motor functions of the body, with 10 million cases worldwide and is the fastest-growing neurological condition in the world.[1] PD is a condition mainly affecting subjects above 60 years. The condition is mainly based on motor symptoms, and the most common classification is observing symptoms only after it take hold.[2] The average life expectancy after getting a diagnosis is 10-20 years[3], with no known cure.

The motivation for this project comes from the fact that there is no current definitive test for PD. Electroencephalography (EEG) is a non-invasive technique to analyse brain patterns, which could help differentiate PD and non-PD subjects. To analyse Electroencephalogram (EEG) series, a skilled and trained expert is required to analyse the recordings, making it prone to subjectivity and human errors. This is where Deep Learning (DL) has the potential to help in the automatic analysis to make EEG analysis more accessible, thus improving the accessibility of results.

1.2 Objectives

- To compare two different data input types for DL for both EEG as time-frequency series and EEG represented as spectrograms in terms of Area Under the Receiver Operating Characteristic Curve (AUC) scores.
- To compare different DL models and the benefit of using a pre-trained model on ImageNet concerning AUC scores for spectrograms and time-frequency series.
- To compare representations of the data in 2D and 3D regarding AUC model performance.

1.3 Related Work

There have been several articles covering DL approaches to classify PD and non-PD subjects from EEG recordings. The work of Shu Lih Oh and Co. [4] looks at using a 13-layer Convolutional Neural Network (CNN) model on a Malaysian dataset of PD and non-PD subjects. They achieved an accuracy of 88.26%, sensitivity of 84.71% and specificity of 91.77%.

The work of Majid Aljalal and Co. [5] based on the datasets from both California and New Mexico use Discrete Wavelet Transform (DWT) and get an accuracy of 99.89% on classifying non-PD and 94.21% for PD, with specificity and sensitivity being around the same numbers. They also achieve AUC scores of 0.98. Other studies using time-frequency series as input [6][7][8][9] have demonstrated high levels of accuracy, specificity, and sensitivity, ranging from the mid to high 90%, along with AUC scores in the high 0.9 range for those that reported them.

Marwa Obayya's work [10] includes spectrograms, the only related work included using this data representation. In this work, they use a Densely Linked Bidirectional Long Short-Term Memory (DLBLSTM), first representing each layer as the sum of its hidden state plus the hidden states of all the layers above for it to be represented to all layers above using recursion. It uses a six-fold Cross-Validation (CV) to classify PD and non-PD with an accuracy of 99.6% and an AUC score of 1.00.

These articles have in common that they divide train/test sets on the **epoch** level, while the work by Sugden and Co.[11] divides on the subject level. This method does not cross-contaminate the test and train sets with epochs of the same subjects. The results are lower than the other works, reaching an accuracy of 83% and an AUC score of 0.88.

Kurbatskaya's work [12] is based on a split on the subject level, using logistic regression (LR), with datasets from multiple centres, including Iowa, Medellin, California and Turku. They tested accuracy and AUC for each specific dataset and combined them. The combined dataset reached an accuracy of 72% and an AUC score of 0.72, while the best dataset, Turku, reached an accuracy of 81.3% and an AUC score of 0.82.

Reference	Classifier	Splitting level	Dataset	Subjects	Accuracy (%)	AUC
[4] 2018	CNN	Epoch	Malaysia	20 non-PD, 20 PD	88.25	-
[5] 2022	CNN	Epoch	California and New Mexico	43 non-PD, 42 PD	99.44	0.98
[6] 2021	CNN and LSTM	Epoch	TS and HOS	21 non-PD, 20 PD	99.2	0.99
[8] 2021	2D-CNN	Epoch	California	16 non-PD, 15 PD	99.90	1.00
[7] 2020	DGHNet	Epoch	New Mexico	28 PD	99.2	-
[9] 2021	PDCNNet	Epoch	California and Malaysia	36 non-PD, 35 PD	(100 and 99.7)	-
[10] 2023 Spectro.	DLBLSTM	Epoch	California	16 non-PDnon-PD, 15 PD	99.6	1.00
[11] 2023	CNN	Subject	Iowa, New Mexico	41 non-PD, 41 PD	83	0.88
[12] 2023	LR	Subject	Iowa, Medellin, California, Turku	85 non-PD and 84 PD	72.2 (60.6, 68.7, 77.7, 82.2)	0.72

Table 1.1: List of earlier works' datasets and achieved accuracy. (Unless in parenthesis, the result is combined for those with multiple datasets.)

This work differs from the previous works by testing different DL architectures and by comparing both EEG as time-frequency series and EEG represented as spectrograms. Methods like 3D-stacking are also used on both the segment and subject levels, and pre-trained models are used based on ImageNet while focusing on splitting the dataset into training and testing data at the subject level using k-fold CV. The dataset also included 50 PD and 60 non-PD subjects from three different centres, making it more robust and varied.

Chapter 2

Background

2.1 Clinical

2.1.1 Parkinson's Disease

PD is a condition where the subject's motor functionality is reduced. This happens as the brain's neurons start breaking down over time.[2] This decline is associated with a decrease in neuron production of dopamine, resulting in irregular brain activity, which reduces movement functionality. The symptoms may include rhythmic shaking (Tremor), loss of automated functions, writing and speech changes, impaired posture and balance, and slowed movements (Bradykinesia).

The direct causes of PD are currently unknown, but factors that appear to play a role are specific genes and environmental triggers.[2] However small, exposure to certain toxins might also increase the likelihood of developing PD. Some groups are more at risk than others, with risk factors including age, heredity, and sex. People over 60 years, males, and people with a family history of the disease are more likely to develop PD.

2.1.2 EEG

EEG involves placing electrodes on a subject's scalp to record the brain's activity. That enables the recording of neurons in the brain while communicating by generating microvoltages between them.[13] Analyzing EEG allows researchers to better understand epilepsy, sleep problems, and other brain-related states.[14] While skilled neurophysiologists can understand underlying medical states, other neurological disorders with more subtle shifts are harder to detect. This is where DL has the potential to assist neurophysiologists with assessing and classifying different medical states.[10]

EEG is a standard neuroimaging procedure that benefits from its non-invasiveness and affordability.[10] The placement and the number of electrodes in the neuroimaging procedure vary. The most common international layouts are 10-20 and 10-10, where the 10-20 layout is shown in Figure 2.1. The '10' and '20' signify the approximate distance between the electrode positions, and the systems are meant to standardize the electrode placements across different individuals and facilities.[15] These electrodes convert into channels, each representing a specific location on the scalp, allowing for the capture of electrical activity from different brain regions.

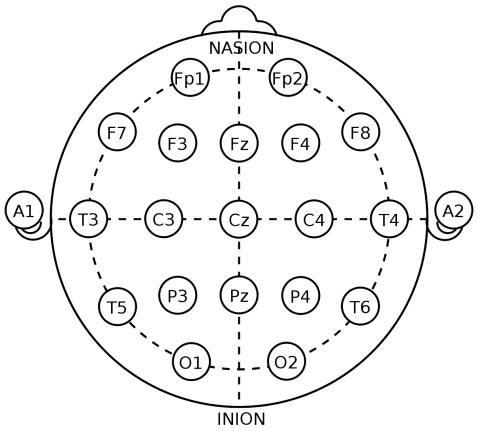

Figure 2.1: EEG layout depicted on a scalp using the international 10-20 layout. Image is obtained from [16] on the 9. April 2024.

2.2 EEG Pre-processing

Pre-processing is a time-frequency method that filters out noise, removes epochs containing artefacts, and enhances relevant features.[17] The time-frequency method establishes a link between the time and frequency domains. It analyses and processes the signals in both domains to extract the relevant features.

When measuring the voltage of the neurons in the brain, the signal is subject to noise interference. Line noises from the power grid must be filtered out to make the data usable. There is also a need to filter out noise from other movements, like winking, heart activity, eye movement, and muscle movements, which are known to be presented in EEG signals.[17][18] To address this, EEG data is divided into time intervals called epochs. These epochs allow for targeted noise reduction and analysis of specific parts of the EEG signal, allowing a more accurate interpretation of the EEG data. An epoch contains data for a single channel within a fixed or variable length time interval, making it better for analysis and allowing the possibility of dropping bad epochs.[19] Figure 2.2 illustrates the concept with the circle "A" representing a two-second epoch.

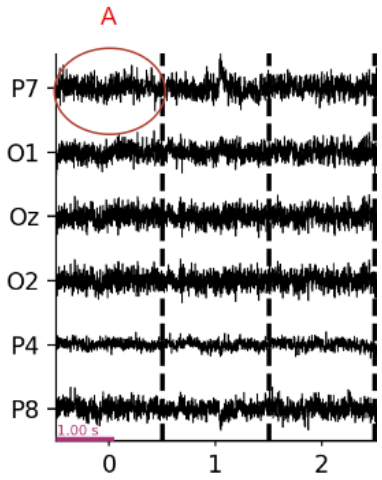

Figure 2.2: Pre-processed EGG signal of a PD subject divided into two-second length epochs represented by the dotted lines, where circle "A" represents an epoch.

2.2.1 Data Filtering

Using PREP by pyPREP[20] is a way of pre-processing data to standardize it into a useful medium while preserving as much of the original data as possible. One goal of PREP is to quickly determine if signal artefacts, noise, or inconsistencies are present in a given dataset, and it provides dataset summaries that allow pinpointing of given issues. High-pass filtering, commonly used to remove low-frequency drifts and artefacts, can also ruin the EEG signals.[21] By using PREP in the pre-processing, the EEG signals can be high-pass filtered while accounting for the line noise from the power voltage, a common source of interference. PREP achieves this by computing an estimated line noise signal to remove it.[22]

The goals of the PREP pipeline, as stated in their study[22], are as follows:

- Remove line noise without committing to a filtering strategy.
- Robustly reference the signal relative to an estimate of the "true" average reference.
- Detect and interpolate bad channels relative to this reference.
- Retain sufficient information to allow users to re-reference using another method or to undo interpolation of a particular channel.

MNE-ICALabel is a tool designed to automatically identify, label and remove independent components derived from EEG signals. It should be performed after removing bad segments to achieve the optimal result.[23][24] Each component is categorised as either brain-related or non-neural, where non-neural can be artefacts stemming from eye blinks or muscle movements. MNE-ICALabel enhances the reliability and reproducibility of the EEG signals by removing the need for manually labelling each component, improving signal analysis.

Autoreject is used for EEG data pre-processing to identify and handle bad epochs, complementing the functionality of PREP.[25][26] It targets the detection of bad epochs with potential abnormalities, including but not limited to artefacts. Autoreject utilises a peak-to-peak signal amplitude threshold to identify the bad epochs. It looks at the signal from multiple

sensors and marks the epoch as bad if most sensors show high-amplitude artefacts. This is similar to how an expert in the field would mark a bad trial based on observation.

2.2.2 Continuous Wavelet Transformation

Continuous Wavelet Transformation (CWT) is a signal processing technique similar to Fourier Transform (FT) in extracting frequency information from a time series.[27][28] However, a Fourier series represents signals using infinite-length cosine and sine functions, which lose temporal information. In contrast, the CWT preserves the time domain by decomposing the signal into smaller wavelets. Wavelets are dynamic components that focus on specific signal aspects, allowing them to hold more detailed information. This makes CWT good for analysing signals with varying properties over time. The computations in CWT are done using a Fast Fourier Transform (FFT).

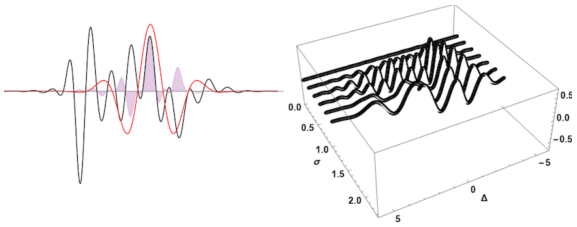

Figure 2.3: Illustration of a CWT, with the mother wavelet on the left, with the wavelets being constructed on the right. Image obtained from [29] where the original gif was converted to a picture, on the 9. April 2024.

2.3 Deep Learning

DL is a subset of Machine Learning (ML).[30] It involves using Deep Neural Networks (DNNs), also known as Artificial Neural Networks (ANNs), to learn complex patterns and relationships within data. ANNs are inspired by the human brain and contain one input and output layer with one or more hidden layers of nodes that transform and process data. DNNs can automatically improve and learn from experiences with the help of DL al-

gorithms. There are three different techniques that a DNN can use to learn: supervised, unsupervised, and reinforced learning.

2.3.1 Supervised Learning

Supervised learning is a technique that makes the model learn and predict based on labelled datasets. It can be categorised into regression and classification. Regression predicts continuous values, such as the stock market and housing prices.[31] Classification predicts categorical values like whether a subject has PD or not, and the predicted output will be a probability distribution between the two.[31] When the model trains, it tries to replicate the correct output from the labelled data using back-propagation. Back-propagation is an algorithm that calculates the gradient of the loss function and uses it to update the weights, aiming to minimize the loss function.[32]

Loss Function

The loss function is a mathematical visualization of the loss between the true labels and the model's predicted outputs.[33] If the loss function value is low, the model's prediction is good. During training, weights are updated to minimize the loss function and improve the accuracy of the predictions. There are many different loss functions available suitable for different tasks where binary and categorical cross-entropy are commonly used for classification tasks.

The Softmax Function

Softmax is a function often used in classification tasks for converting raw scores into a probability distribution.[34] It does this by exponentiating each raw score using Euler's number and dividing it by the sum of all exponentiated raw values. If one raw score is small compared to the others it will get a low probability and if it is large it will get a high probability, as shown in Figure 2.4 together with the softmax function.

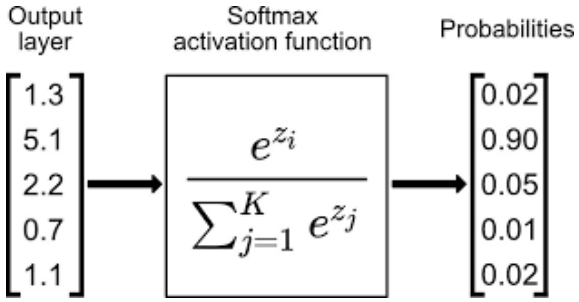

Figure 2.4: The softmax function is applied on the output array, producing a probability distribution of the output classes. Image obtained from [35] on the 9. April 2024.

2.3.2 Convolutional Neural Network

CNNs are a type of DNNs often used for image and video recognition and classification. In an DNN, each node in one layer is connected to every other node in the previous layer, resulting in many parameters. As the number of parameters increases, the model becomes computationally heavy and prone to overfitting. In a CNN, however, each node in one layer is only connected to a few in the previous layer, making it solve the limitations of DNN.[18] A CNN contains three types of layers: convolutional, pooling and fully connected layer, as shown in Figure 2.5.[36]

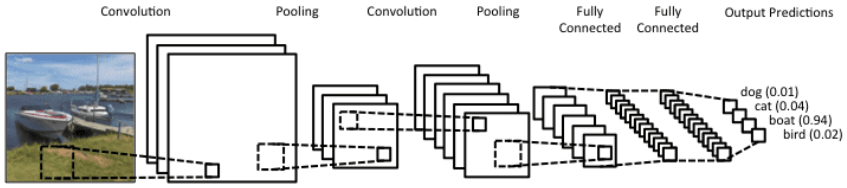

Figure 2.5: An illustration of what a CNN architecture looks like. Image obtained from [37] on the 15. March 2024.

2.3.2.1 Components

Convolutional Layer

In a CNN, the convolutional layer is a core component responsible for feature extraction, and a model usually contains many convolution layers.[36] It works by applying a filter to an area of the image and calculating the dot product between the filter and the area's pixels. The filter is called a kernel with trainable weights. The filter shifts by a stride value until it has gone through the whole image. The output is a series of dot products known as a feature map. Typically, a convolution layer contains multiple kernels, resulting in equally many feature maps. Applying the kernel to the raw input reduces its output dimension, as shown in Figure 2.6.

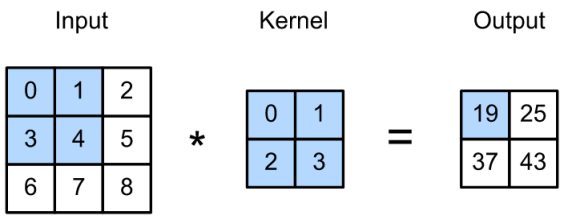

Figure 2.6: A kernel of size 2x2 is applied on a 3x3 matrix producing a 2x2 output. Image obtained from [38] on the 8. April 2024.

With a technique called zero-padding, it is possible to produce equal or larger output dimensions, as illustrated in Figure 2.7.[36]

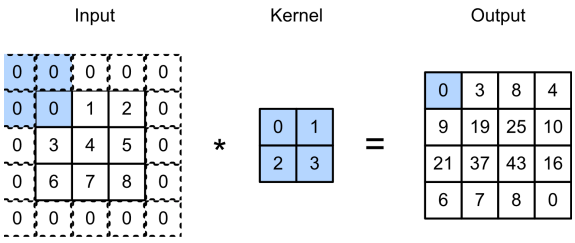

Figure 2.7: A kernel of size 2x2 is applied on a 3x3 matrix with one layer of zero padding added, producing a 4x4 output. Image obtained from [38] on the 8. April 2024.

Pooling Layer

The pooling layer, also known as the down-sampling layer, reduces the model's complexity.[36] It decreases the computation time by reducing the number of parameters in the input. Similar to the convolution layer, a filter sweeps across the image, but instead of calculating the dot product with the kernel parameters, the max or average value of the area is sent to the output array. Max pooling is shown in Figure 2.8 where a pooling window of 2x2 and a stride value of one is applied on a 3x3 matrix, reducing the input by 56%.

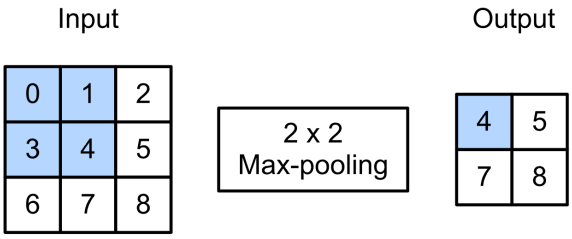

Figure 2.8: Max pooling is applied on a 3x3 matrix with a pooling window of 2x2 and a stride value of one, reducing the input by 56%. Image obtained from [38] on the 8. April 2024.

Fully Connected Layer

The fully connected layer connects every node in the previous layer to every node in the current layer.[39] It typically comes after the convolution and pooling layer, producing the model's final output prediction. Therefore, the final fully connected layer contains the same number of nodes as the output classes in a classification problem.

2.3.2.2 Convolutional Neural Network Architectures

ResNet

Residual Network (ResNet) is a CNN architecture supporting a dynamic range of convolution layers. Therefore, many different ResNet architectures exist, like ResNet18 and ResNet50, which have 18 and 50 layers, respectively. ResNet offers a solution to the "vanishing gradient" problem, which occurs when more layers are added, resulting in a loss in performance.[40] ResNet does this by adding the input to a set of convolution layers to the output of the same layers. This technique is called skip connections and is shown in Figure 2.9.[41]

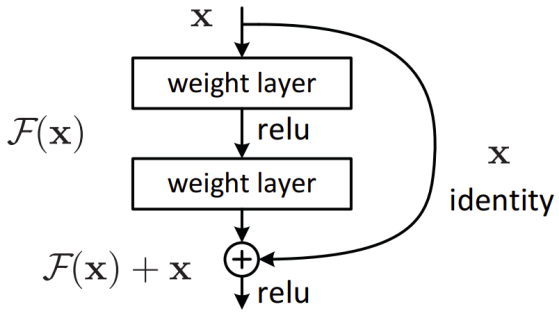

Figure 2.9: Visualization of skip connections. Image obtained from [42] on the 8. April 2024.

EEGNet

EEGNet is a compact 2D CNN designed for EEG signals.[43] It contains three convolutional layers, two pooling layers and one fully connected layer and is designed for an input format of [channels, signal length, 1]. A visual representation of the EEGNet architecture is shown in Figure 2.10.

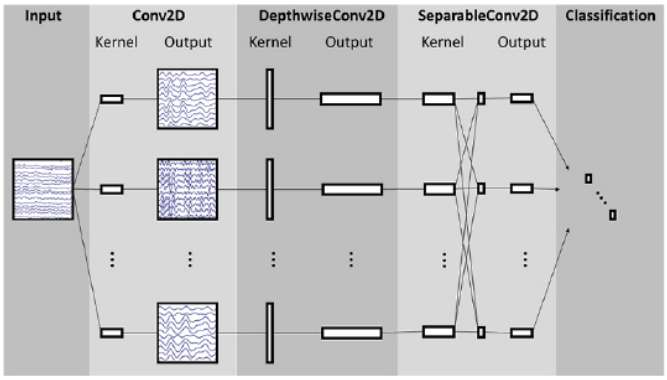

Figure 2.10: A visual representation of the EEGNet architecture. Image obtained from [42] on the 11. March 2024.

3D Models

A 3D CNN is designed to input a 3D volume or a sequence of 2D images, allowing it to learn spatial relationships within the data.[44][45] Therefore, it suits medical images such as Computerized Tomography (CT), Magnetic Resonance Imaging (MRI), and EEG signals.

2.3.3 Transfer Learning

Transfer learning is a DL technique that uses a pre-trained model trained on a large dataset, potentially increasing performance and efficiency, which makes it popular to use when limited data is available.[46] ImageNet is a dataset with over 14 million images and over 20,000 different categories, and many popular CNNs, such as VGG and ResNet, have been pre-trained using it.[47][48]

2.3.4 Checkpoints

In DL, checkpoints are used to save the model state throughout the training process.[49] This enables the model to be restored and resumed from where

it left off and called back to a previous state. This is important because overfitting can occur as the number of epochs increases during training. To prevent this and ensure that the best model is used for evaluation, checkpoints are used to save the model state, typically when the validation loss improves.

2.3.5 Evaluation Metrics

Calculating different metrics to evaluate a DL model's performance is important. Accuracy, sensitivity, specificity, F1 score and AUC are often calculated for supervised learning models that predict categorical values. These metrics are calculated from the components of the confusion matrix, which is a table that provides counts of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN).[50]

- **Accuracy**: Accuracy measures the proportion of correctly classified predictions done by the model.[51] It is calculated by dividing the sum of TP and TN by the total number of predictions.
- **Sensitivity** (True Positive Rate, Recall): Sensitivity measures the accuracy of actual positive cases that the model correctly identified.[52] For example, if the model tries to predict whether a person is sick, the sensitivity is the probability that the model can detect the disease if the person is sick. Sensitivity is calculated by dividing the TP by the sum of TP and FN.
- **Specificity** (True Negative Rate): Specificity measures the accuracy of actual negative cases that the model correctly identified.[52] For example, if the model tries to predict whether a person is sick, the specificity is the probability that the model can detect a healthy person if the person is healthy. Specificity is calculated by dividing the TN by the sum of TN and FP.
- **F1 Score**: The F1 score is the harmonic mean of precision and sensitivity.[53] Precision measures how accurately the model predicts positive instances and is computed by dividing the TP by the sum of TP and FP. The F1 score is a useful performance metric when the dataset is uneven or the values for precision and sensitivity are very different.

- **Area Under the ROC Curve (AUC)**: AUC measures the Area Under the Receiver Operating Characteristic (ROC) Curve.[54] The ROC curve plots the true positive rate against the false positive rate at different threshold settings. Therefore, the AUC provides more information on the model's performance regardless of the chosen threshold.

K-Fold Cross-Validation

To evaluate and train a DL model, it is important to split the dataset into training and validation.[55] K-fold CV is used to split the dataset into k equal-size folds. One fold is used for the validation set and the rest for the training set. The model gets evaluated and trained k times with a different validation fold each time, as shown in Figure 2.11. By calculating the average of the performance metrics, a more robust and unbiased estimate of the model's true performance can be obtained.

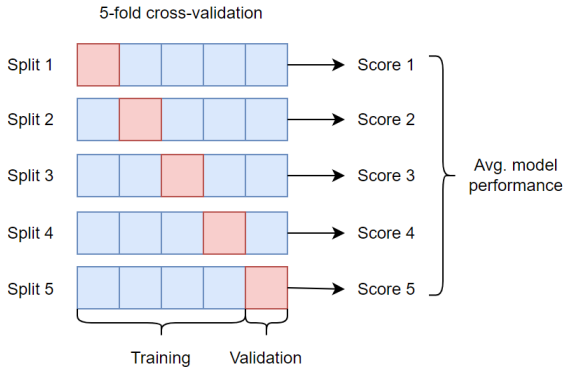

Figure 2.11: Visualization of 5-fold CV.

Stratified Splitting

Stratified splitting is often used in DL to ensure that the proportion of subgroups is equally represented in splitted training and validation data.[56] Typically, the dataset is stratified on the label but can also be stratified on other subgroups. For example, if a DL model tries to predict whether an image shows a dog or a cat, it is most important to have equal proportions

of dogs and cats in the training and validation data. However, dogs and cats can look very different, so it may be essential to stratify further based on colour, gender and race to ensure equal proportions of these subgroups.

Threshold Setting

A threshold value is used as the minimum probability required for a positive prediction for models that predict a probability distribution between the output categories.[57] For example, if a model predicts whether a person is sick, a threshold value of 0.5 will give a positive prediction for every instance where the model's output probability is 0.5 or higher. This can be problematic in real-life scenarios as it may lead to many false positives. Therefore, a threshold value should be chosen carefully based on the cost of false negatives and false positives.

Chapter 3

Data and Methods

3.1 Datasets

This thesis used three distinct EEG datasets from Turku, California, and New Mexico. Combining different datasets potentially enabled diversity in EEG patterns from different cultural and demographic environments, making the combined dataset more robust and varied. The datasets included information about each subject, including medication condition, whether eyes were open or closed during recording and Movement Disorder Society Unified PD Rating Scale (MDS-UPDRS) scores. With the use of medication, the subject's PD symptoms get reduced, resulting in a different EEG recording.[5] Therefore, this thesis only used EEG signals where subjects were off medication. Additionally, EEG data extracted from individuals whose eyes were open or closed yields different results.[58] Given that the combined number of subjects with eyes open was higher, this subset was selected.

The MDS-UPDRS can be subscaled into mild, moderate and severe, where a score of 32 and below is mild and 59 and above is severe.[59] Subjects with severe MDS-UPDRS stage were excluded, as there were only three subjects at this severity level. The flow of data extraction steps is illustrated in Figure 3.1. Each dataset contained a different amount of channels, from 41 to 67, with 32 channels in common, and the signal length varied for every

location.[60]

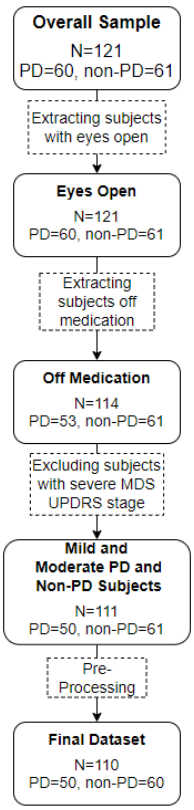

Figure 3.1: A visualization of the data extraction steps.

Turku Dataset

The dataset from Turku was collected in 2018 from the University of Turku and the Turku University Hospital.[60] It contained 20 PD and 20 non-PD subjects. Every EEG signal contained 66 channels and had an average length of 156 seconds. Extracting the subjects that were off medication, had eyes open, and had MDS-UPDRS stage mild and moderate resulted in 10 mild, one moderate and 19 non-PD subjects.

After pre-processing, two non-EEG channels were removed, and one of the non-PD subjects had to be dropped because the EEG signal had too many

noisy channels. As a result, 11 PD and 19 non-PD subjects with a signal containing 64 channels and an average signal length of 146 seconds was included from the Turku dataset.

California Dataset

The dataset from California was collected in 2013 and contained 15 PD and 16 non-PD subjects. The PD subjects were recruited from Scripps Clinic in La Jolla, and the non-PD subjects were recruited from the PD subjects spouses and the local community. Every EEG signal contained 41 channels with an average length of 195 seconds.[60] The MDS-UPDRS stage was not collected by a board-certified neurologist but by personnel who had completed online training.[61]

After extracting the relevant subjects, there were four mild, 10 moderate and 16 non-PD subjects. None of the subjects got dropped after pre-processing, but nine non-EEG channels were removed. This resulted in 14 PD and 16 non-PD subjects with a signal containing 32 channels and an average signal length of 176 second was included from the California dataset.

New Mexico Dataset

The dataset from New Mexico contained 25 PD and 25 non-PD subjects.[62] It was collected at the Cognitive Rhythms and Computation Lab at the University of New Mexico in 2015. The EEG signal contained 67 channels and had a fixed length of 60 seconds.

All the subjects correspond to the requirements, and none got dropped after pre-processing, but four non-EEG channels were removed. This resulted in 11 moderate, 14 mild and 25 non-PD subjects with an EEG signal containing 63 channels and an average signal length of 54 seconds were included from the New Mexico dataset.

Summary

Combining the subjects that fitted the requirements and passed the pre-processing from Turku, California and New Mexico, the final dataset contained 50 PD and 60 non-PD subjects. The 50 PD subjects can be further categorised into 28 mild and 22 moderate based on MDS-UPDRS stage. There were 32 common channels for the centre locations, the average signal length was 112 and the average amount of channels was 53. The subject level 3D model required a signal length of 54 seconds, excluding 10 subjects. Among these were two mild, three moderate and four non-PD subjects from New Mexico and one non-PD from California.

Table 3.1: Combined dataset information. The PD-Duration (Y) represents the duration of PD in years, and the symbol ♀ represents the number of females.

Location	PD	Non-PD	Mild	Moderate	PD-Duration (Y)	Age	♀(%)	Signal Length (sec)	Channels
Turku	11	19	10	1	5.1	67.2	18 (60.0)	146	64
California	14	16	4	10	4.7	63.4	17 (56.7)	176	32
New Mexico	25	25	14	11	5.4	69.5	18 (36.0)	54	63

3.2 Signal Pre-processing

3.2.1 Epoch and Segments

In this thesis, a segment is defined as a collection of epochs from all channels within a two-second time interval. In an EEG signal divided into fixed epoch lengths, where each epoch contains data from a single channel for a time period, a segment contains the data from all channels over that same time interval. This concept is illustrated in Figure 2.2, where circle 'B' represents a two-second segment and circle 'A' represents a two-second epoch.

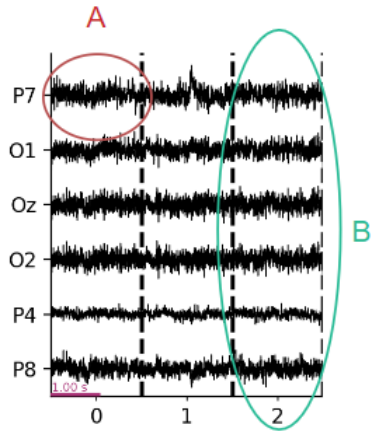

Figure 3.2: Pre-processed EEG signal of a PD subject divided into two-second length epochs represented by the dotted lines, with circle "A" representing an epoch and circle "B" representing a segment. A segment is defined as a collection of epochs from all channels within a two-second time interval.

3.2.2 Signal Preparation

The raw EEG signal contained noise and artefacts, as shown in Figure 3.4 and 3.6. Therefore, the data was subjected to PREP, mne-ICALabel, and Autoreject, which removed several channels and segments. PREP standardized the signal and filtered out the line noise specified by each centre's power-line voltage. After PREP was applied, the EEG data was split into segments of two seconds with size [channels, 1000], where 1000 is the number of data points. Where data points are the µV collected 500 times a second, resulting in 1000 samples for 2 seconds. Autoreject was then applied to identify and remove artefacts and bad segments as explained in paragraph 2.2.1, and mne-ICALabel was applied to remove non-neural artefacts. Lastly, autoreject was applied again to handle any remaining artefacts and bad segments. This resulted in a pre-processed EEG signal as shown in Figure 3.5 and 3.7. The segments were then split by every channel, resulting in separate epochs containing data from one channel with a size of [1000] data points.

Figure 3.3: The EEG signals presented were selected randomly from the California dataset. Figures 3.4 and 3.5 depict data from the same PD subject, while Figures 3.6 and 3.7 represent data from a non-PD subject.

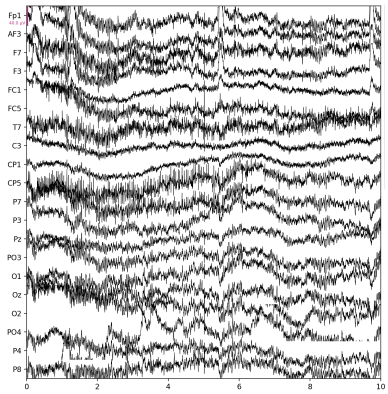

Figure 3.4: A 10-second EEG signal from a PD subject that contains 20 channels.

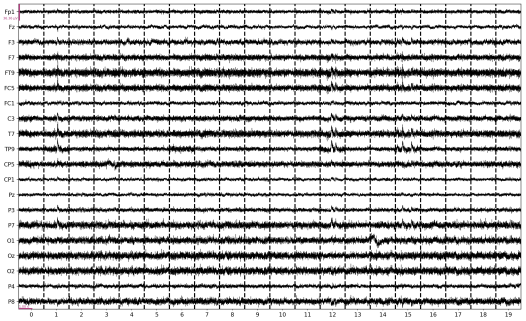

Figure 3.5: Pre-processed EGG signal from a PD subject that contains 20 segments and 20 channels.

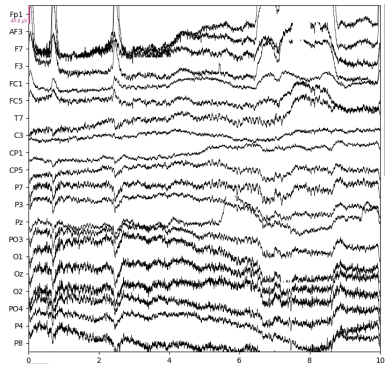

Figure 3.6: A 10-second EEG signal from a non-PD subject that contains 20 channels.

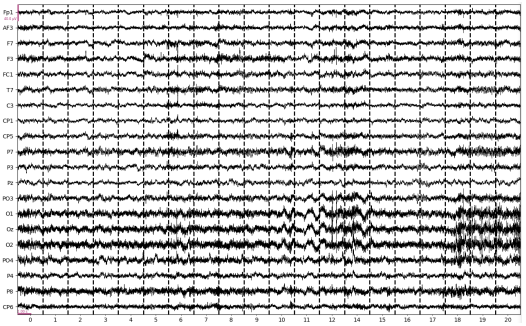

Figure 3.7: Pre-processed EGG signal from a non-PD subject that contains 20 segments and 20 channels.

3.2.3 Time-frequency Transformation

For a time-frequency transformation, the epochs are normalized to convert to a target range between 0 and 1 and saved separately as a .npy file. Figure 3.5 and Figure 3.7 show the transformed EEG signal of a PD and non-PD

subject, respectively.

3.2.4 Continuous Wavelet Transformation

Each epoch is transformed into a spectrogram using a CWT.[63] The transformation produces a spectrogram of size [1000, 238, 3] as seen in the top spectrogram in Figure 3.8. It is resized to an RGB format of [244, 244, 3] as shown in the left spectrogram in Figure 3.8 and to [320, 320, 3] as shown in the right spectrogram in Figure 3.8. The motivation behind converting it to RGB is that pre-trained CNNs are primarily trained on RGB images, and since we use ImageNet, it may improve performance. Resizing to 244x244 represents a standard image format commonly used in CNNs and typically performs better with square-shaped data, avoiding potential challenges with irregular sizes, with the added benefit of lowering the picture's overall size and avoiding problems with GPU capacity. Upsampling the image to 320x320 provides more data points that may improve the model's learning ability.

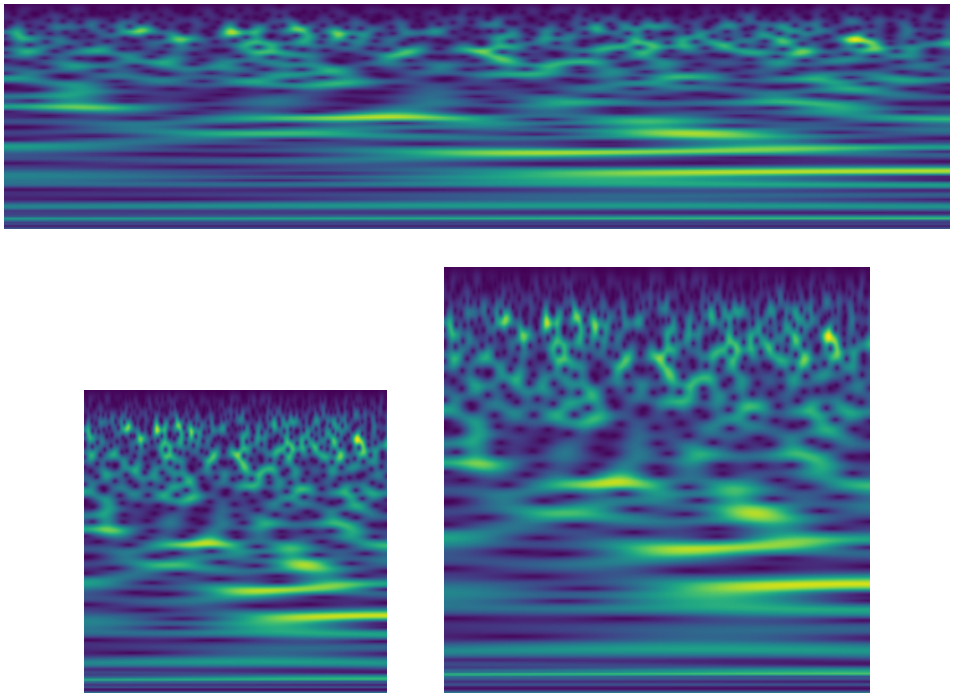

Figure 3.8: The top spectrogram illustrates the original spectrogram image with size [1000, 238, 3]. The left spectrogram illustrates the image with size [244, 244, 3], and the right spectrogram illustrates the image with size [320, 320, 3].

3.3 Methods

3.3.1 Data Splitting

The dataset was split at the subject level into training, testing, and validation sets using 5-fold CV while utilizing stratification. The stratification parameters included the centre of origin, gender, PD and non-PD label, with signal length further dividing subjects based on a threshold of 110 seconds. However, stratifying on the signal length was unnecessary for the subject-level 3D model because it used a fixed number of segments and channels per subject. Table 3.2 illustrates the subject statistics across training, testing,

and validation sets for each fold for the 2D models.

Table 3.2: Subject statistics across training, testing and validation sets for each fold for the 2D models. The symbol ♀ represents the number of females.

	PD	Non-PD	Turku	California	New Mexico	♀(%)	Epochs
Fold 1							
Training Set	31	39	19	20	31	35 (50.0)	197622
Validation Set	9	9	5	4	9	8 (44.4)	50060
Test Set	10	12	6	6	10	10 (45.5)	63338
Fold 2							
Training Set	31	39	19	20	31	34 (48.6)	196084
Validation Set	9	9	5	4	9	8 (44.4)	50123
Test Set	10	12	6	6	10	11 (50.0)	64813
Fold 3							
Training Set	32	38	19	20	31	33 (47.1)	199245
Validation Set	8	10	5	4	9	8 (44.4)	50002
Test Set	10	12	6	6	10	12 (54.5)	61773
Fold 4							
Training Set	32	38	18	21	31	34 (48.6)	199280
Validation Set	8	10	5	4	9	8 (44.4)	49285
Test Set	10	12	7	5	10	11 (50.0)	62455
Fold 5							
Training Set	32	38	19	20	31	33 (47.1)	196780
Validation Set	8	10	5	4	9	8 (44.4)	49398
Test Set	10	12	6	6	10	12 (54.5)	64842

The 5-fold CV technique split the dataset into five equal-sized boxes where 20% was used for the test set. The remaining 80% of the data were split into train and validation sets with an 80/20 split, respectively, while utilizing stratification. That way, the model was trained on the training set and was validated on the validation set. After completing all epochs, the model state was called back to the lowest validation loss. That model was evaluated against the test set, and performance metrics were calculated. For the next fold, the model was reset, and the test set was shifted by one box, ensuring a different test, training, and validation set for each fold. Figure 3.9 illustrates this with the number of subjects for the 2D and segment level 3D models.

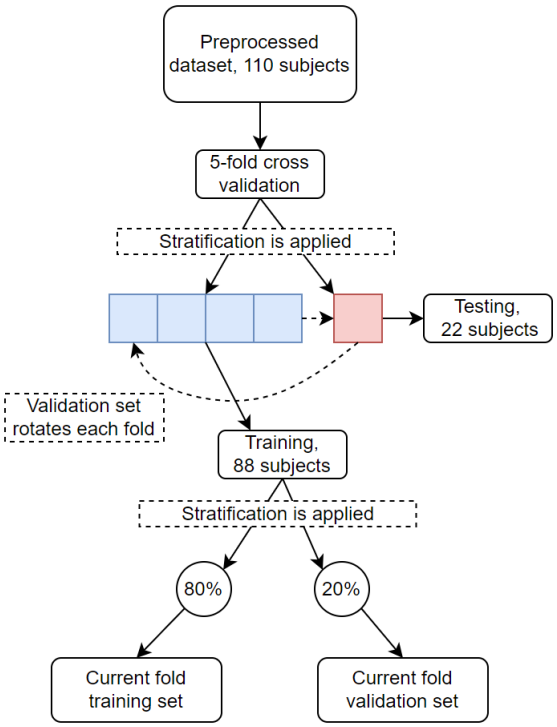

Figure 3.9: A visualization of how the dataset is split into train, test and validation sets.

3.3.2 Training of the Models

Python was the programming language selected for this project, and the open-source libraries Tensorflow [64] and Keras [65] were the building blocks for the models. TensorFlow is a framework by Google for building and training ML models and offers the baseline to build DL applications. Whereas Keras is a high-level neural network API that works on top of TensorFlow and provides the interface for designing, training and testing the DL models. The training was performed on the UiS Unix Server on NVIDIA Tesla V100-PCIE-32GB [66] GPUs.

3.3.3 Hyper-parameters and Checkpoints

For every model, the batch size was 32 and the learning rate was 0.0001[67]. The number of epochs used per fold varied based on the model and was selected by running each model by a large number of epochs on a single fold to determine the optimal amount. For ResNet18 2D with and without ImageNet, the epoch amount was 40. EEGNet was set to 100 epochs, and both of the 3D models used 50 epochs. A threshold value of 0.5 and categorical cross-entropy was used for all models. The softmax function was included in the output layer for every architecture for converting raw scores into a probability distribution between PD and non-PD. Checkpoints were used to save the model weights for the best validation loss.

3.3.4 2D Models

The 2D CNNs used in this thesis were ResNet18 with and without ImageNet and EEGNet. ResNet18 was used because it is a popular lightweight architecture that supports ImageNet. EEGNet was used due to its designed architecture and previous positive results for EEG.[43] Every epoch of the EEG signal were input separately, producing one prediction for each epoch. ResNet18 required input of [height, width, colour channels], making it compatible with the spectrograms of size [244, 244, 3] and [320, 320, 3] and was used for all 2D models. The time-frequency series needed to be converted from [1000] to [1000, 1, 1]. ResNet18 with ImageNet required RGB format, and the time-frequency series was expanded to [1000, 1, 3] by repeating the data for the three axes. EEGNet was designed for an input shape of [channels, signal length, 1], making it suitable to convert the time-frequency series to [1, 1000, 1].

3.3.5 3D Models

The 3D CNN used in this thesis was a 3D variant of ResNet18.[68] ResNet18 3D expects an input of the format [height, width, depth, colour channels], making it possible to input a sequence of spectrograms and time-frequency series for a subject. The motivation for 3D is that it enables the model to

potentially identify which channels hold greater value for detecting PD. The spectrograms of size [244, 244, 3] and [320, 320, 3] and the time-frequency series converted to [1000, 1, 1] were stacked along the third axis to fit the input format. This thesis combined the epochs for an EEG signal in two ways: subject and segment level.

Subject Level 3D

For the subject level 3D model, the epochs from the 32 common channels for 27 segments were stacked together in order of channels as illustrated in Figure 3.10. This produces a sequence of $32 \times 27 = 864$ 2D time-frequency series and spectrograms per subject. The reason for using a fixed segment length was to ensure a fixed input size for the model. Using a length of 27 segments offered a reasonable signal length without excluding too many subjects.

Segment Level 3D

For the segment level 3D model, the epochs from the 32 common channels for one segment were stacked in order as illustrated in Figure 3.10. This produces a sequence of 32 time-frequency series and spectrograms for one segment. This resulted in multiple sequences per subject equal to the number of segments in the EEG signal. The motivation behind stacking the epochs for one segment was to get more trainable data compared to the subject level 3D model while still offering the potential for the model to determine which channels hold greater relevance for detecting PD.

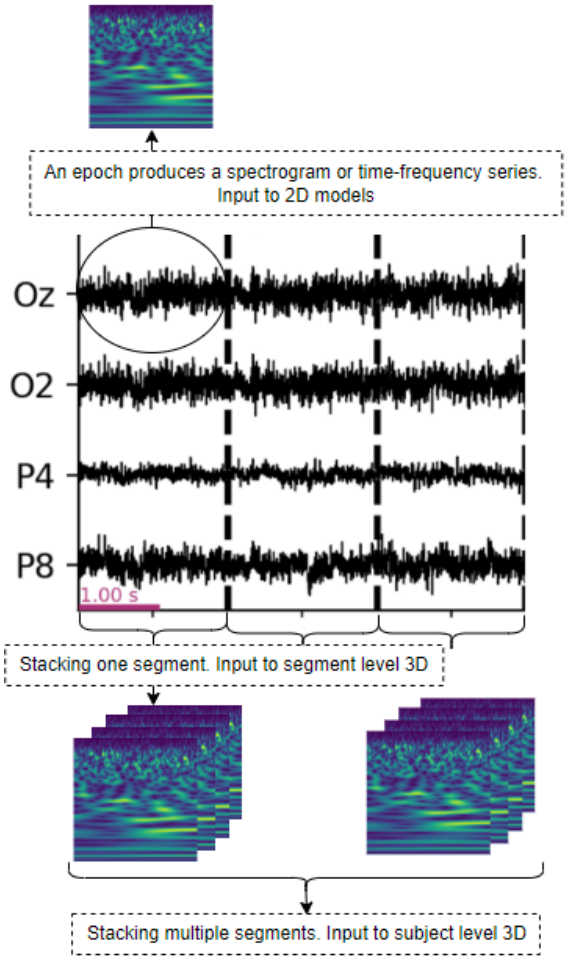

Figure 3.10: A visualization of the different ways to input and stack spectrograms and time-frequency series.

3.3.6 Evaluation Metrics

The metrics used to evaluate the model's performance were accuracy, sensitivity, specificity, F1 score and AUC, as well as a plot of the confusion matrix. For this thesis, given the nearly equal distribution of PD and non-PD subjects, we prioritized AUC as the most relevant metric. AUC was chosen because it measures the model's performance at different threshold

settings, providing a robust estimate of its effectiveness in distinguishing between PD and non-PD. This thesis used a 5-fold CV, resulting in a set of performance metrics for each fold. By calculating the average of these metrics, a more robust and unbiased estimate of the model's true performance can be obtained. Since every epoch of the EEG signal was input to the 2D models separately, there were equally many predictions. Similarly, for the segment level 3D model, every input segment gave a prediction. Both resulted in multiple predictions for a single subject. By applying a hard threshold of 0.5 to every prediction and taking the average of these to find the subject level prediction, as shown in equation (3.1).

$$\text{Subject Level Prediction} = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}1, & \text{if } P_i \geq 0.5\\ 0, & \text{otherwise}\end{cases} \tag{3.1}$$

As a result, it was possible to calculate metrics for the subject level. To gather information about the model's ability to predict for PD for different severity levels, separate metrics were calculated for mild PD subjects compared to non-PD subjects, as well as for moderate PD subjects compared to non-PD subjects.

Chapter 4

Results

4.1 ResNet18 2D ImageNet

The ResNet18 2D model, pre-trained on ImageNet, predicted on epoch level, and the subject level metrics were calculated with a hard threshold of 0.5.

4.1.1 Time-Frequency Series

Table 4.1 shows a subject level accuracy of $56.36\% \pm 13.09\%$ and a subject level AUC at 0.61 ± 0.17. Figure 4.1 shows that the model, on average, accurately predicts 9.6 out of the 12 non-PD subjects correctly and 2.8 out of the 10 PD subjects.

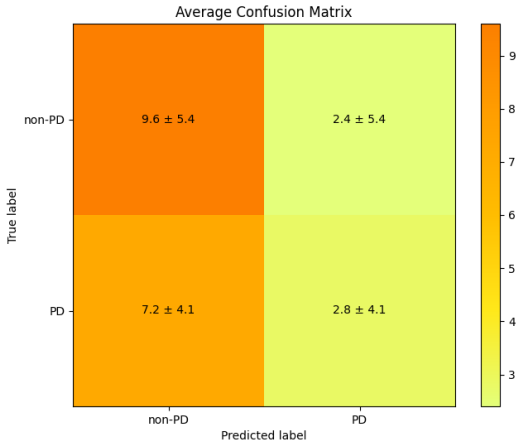

Figure 4.1: Average confusion metric across the five folds for the ResNet18 2D model with ImageNet using time-frequency series.

Metric	**Value and Standard Deviation**	
	Epoch Level	**Subjects Level**
Accuracy	$55.27\% \pm 5.04\%$	$56.36\% \pm 13.09\%$
F1 Score	0.33 ± 0.20	0.25 ± 0.34
Sensitivity	0.33 ± 0.29	0.28 ± 0.41
Specificity	0.71 ± 0.28	0.80 ± 0.45
AUC	0.54 ± 0.07	0.61 ± 0.17
	Subjects Level for:	
	PD Mild and Non-PD	**PD Moderate and Non-PD**
Accuracy	$60.54\% \pm 8.35\%$	$56.56\% \pm 12.69\%$
F1 Score	0.26 ± 0.35	0.25 ± 0.34
Sensitivity	0.27 ± 0.40	0.30 ± 0.45
Specificity	0.80 ± 0.45	0.80 ± 0.45
AUC	0.65 ± 0.19	0.62 ± 0.21

Table 4.1: Average values and standard deviations for the ResNet18 2D model with ImageNet using time-frequency series.

4.1.2 Spectrogram of Size 244x244

Table 4.2 shows a subject level accuracy of $51.82\% \pm 4.07\%$ and an AUC score of 0.64 ± 0.10. Figure 4.2 shows that the model, on average, accurately predicts 6.8 out of the 12 non-PD subjects correctly and 4.6 out of 10 PD subjects.

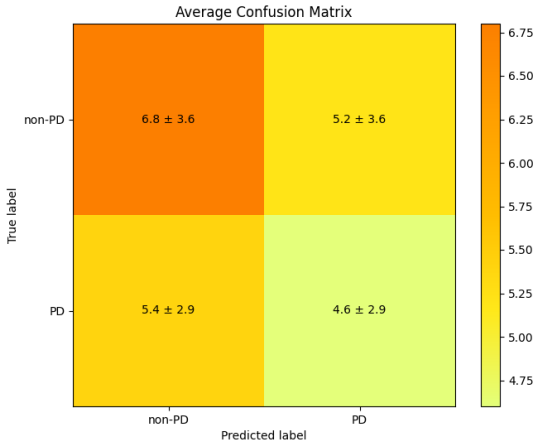

Figure 4.2: Average confusion metric across the five folds for the ResNet18 2D model with ImageNet using spectrograms of size 244x244.

Metric	Value and Standard Deviation	
	Epoch Level	**Subjects Level**
Accuracy	$56.10\% \pm 4.96\%$	$51.82\% \pm 4.07\%$
F1 Score	0.46 ± 0.05	0.42 ± 0.17
Sensitivity	0.46 ± 0.15	0.46 ± 0.29
Specificity	0.64 ± 0.19	0.57 ± 0.30
AUC	0.58 ± 0.05	0.64 ± 0.10
	Subjects Level for:	
	PD Mild and Non-PD	**PD Moderate and Non-PD**
Accuracy	$54.33\% \pm 11.10\%$	$52.75\% \pm 15.82\%$
F1 Score	0.41 ± 0.09	0.20 ± 0.20
Sensitivity	0.50 ± 0.22	0.40 ± 0.45
Specificity	0.57 ± 0.30	0.57 ± 0.30
AUC	0.67 ± 0.10	0.58 ± 0.17

Table 4.2: Average values and standard deviations for the ResNet18 2D model with ImageNet using spectrograms of size 244x244.

4.1.3 Spectrogram of Size 320x320

Table 4.3 shows a subject level accuracy of $60.91\% \pm 2.49\%$ and an AUC score of 0.61 ± 0.08. Figure 4.3 shows that the model, on average, accurately predicts 9.4 out of the 12 non-PD subjects correctly and 4 out of 10 PD subjects.

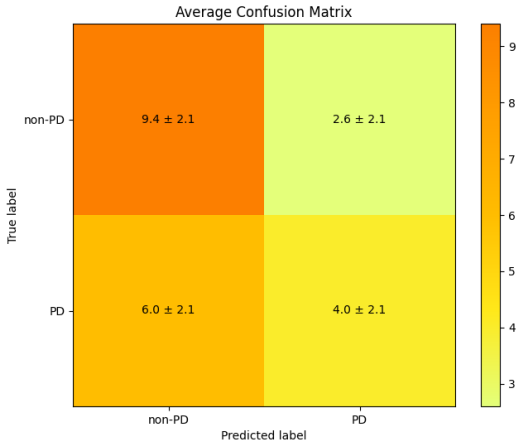

Figure 4.3: Average confusion metric across the five folds for the ResNet18 2D model with ImageNet using spectrograms of size 320x320.

Metric	**Value and Standard Deviation**	
	Epoch Level	**Subjects Level**
Accuracy	$56.51\% \pm 5.49\%$	$60.91\% \pm 2.49\%$
F1 Score	0.40 ± 0.13	0.45 ± 0.17
Sensitivity	0.38 ± 0.19	0.40 ± 0.21
Specificity	0.70 ± 0.18	0.78 ± 0.17
AUC	0.57 ± 0.08	0.61 ± 0.08
	Subjects Level for:	
	PD Mild and Non-PD	**PD Moderate and Non-PD**
Accuracy	$64.09\% \pm 3.32\%$	$70.83\% \pm 12.46\%$
F1 Score	0.37 ± 0.23	0.42 ± 0.05
Sensitivity	0.39 ± 0.28	0.43 ± 0.09
Specificity	0.78 ± 0.17	0.78 ± 0.17
AUC	0.63 ± 0.10	0.60 ± 0.14

Table 4.3: Average values and standard deviations for the ResNet18 2D model with ImageNet using spectrograms of size 320x320.

4.2 ResNet18 2D No-ImageNet

The ResNet18 model without ImageNet was predicted on the epoch level, and the subject level metrics were calculated with a hard threshold of 0.5.

4.2.1 Time-Frequency Series

Table 4.4 has a subject level accuracy reaching $60.91\% \pm 10.46\%$. With subject level AUC at 0.65 ± 0.06. Figure 4.4 shows that the model, on average, accurately predicts 9.2 out of the 12 non-PD subjects correctly and 4.2 out of 10 PD subjects.

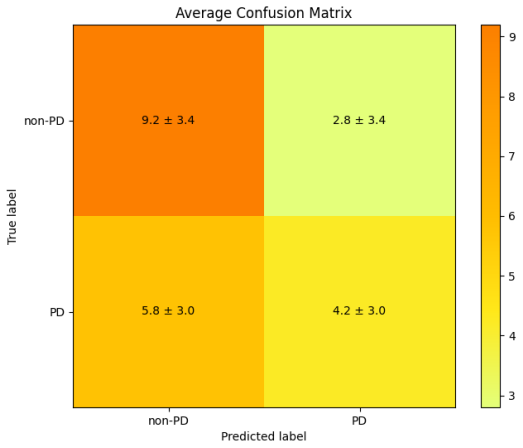

Figure 4.4: Average confusion metric across the five folds for the ResNet18 2D model without ImageNet using time-frequency series.

Metric	Value and Standard Deviation	
	Epoch Level	**Subjects Level**
Accuracy	$56.68\% \pm 3.61\%$	$60.91\% \pm 10.46\%$
F1 Score	0.41 ± 0.14	0.43 ± 0.28
Sensitivity	0.39 ± 0.20	0.42 ± 0.30
Specificity	0.70 ± 0.18	0.77 ± 0.29
AUC	0.58 ± 0.03	0.65 ± 0.06
	Subjects Level for:	
	PD Mild and Non-PD	**PD Moderate and Non-PD**
Accuracy	$62.92\% \pm 10.29\%$	$70.92\% \pm 20.28\%$
F1 Score	0.36 ± 0.22	0.41 ± 0.31
Sensitivity	0.39 ± 0.30	0.43 ± 0.09
Specificity	0.77 ± 0.29	0.77 ± 0.29
AUC	0.69 ± 0.04	0.57 ± 0.24

Table 4.4: Average values and standard deviations for the ResNet18 2D model without ImageNet using time-frequency series.

4.2.2 Spectrogram of Size 244x244

Table 4.5 shows a subject level accuracy of $60.00\% \pm 11.77\%$ and an AUC score of 0.64 ± 0.09. Figure 4.5 shows that the model, on average, accurately predicts 8.2 out of the 12 non-PD subjects correctly and 5 out of 10 PD subjects.

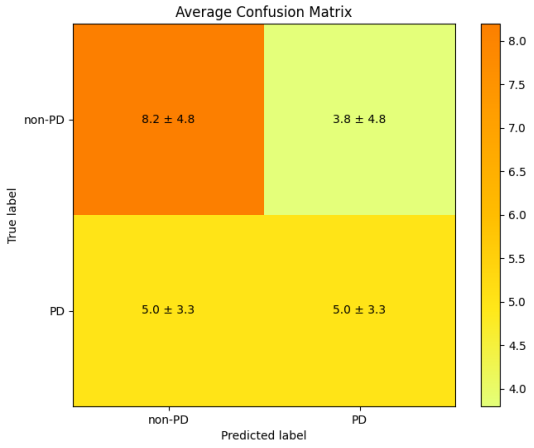

Figure 4.5: Average confusion metric across the five folds for the ResNet18 2D model without ImageNet using spectrograms of size 244x244.

Metric	**Value and Standard Deviation**	
	Epoch Level	**Subjects Level**
Accuracy	$57.15\% \pm 4.71\%$	$60.00\% \pm 11.77\%$
F1 Score	0.41 ± 0.13	0.48 ± 0.27
Sensitivity	0.38 ± 0.18	0.50 ± 0.33
Specificity	0.71 ± 0.18	0.68 ± 0.40
AUC	0.58 ± 0.04	0.64 ± 0.09
	Subjects Level for:	
	PD Mild and Non-PD	**PD Moderate and Non-PD**
Accuracy	$60.94\% \pm 16.63\%$	$65.42\% \pm 25.23\%$
F1 Score	0.40 ± 0.23	0.43 ± 0.29
Sensitivity	0.47 ± 0.33	0.58 ± 0.37
Specificity	0.68 ± 0.40	0.68 ± 0.40
AUC	0.65 ± 0.06	0.63 ± 0.20

Table 4.5: Average values and standard deviations for the ResNet18 2D model without ImageNet using spectrograms of size 244x244.

4.2.3 Spectrogram of Size 320x320

Table 4.6 shows a subject level accuracy of $59.09\% \pm 6.43\%$ and an AUC score of 0.63 ± 0.10. Figure 4.6 shows that the model, on average, accurately predicts 7.4 out of the 12 non-PD subjects correctly and 5.6 out of 10 PD subjects.

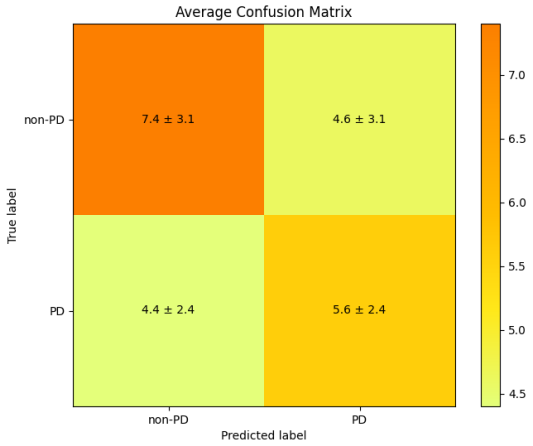

Figure 4.6: Average confusion metric across the five folds for the ResNet18 2D model without ImageNet using spectrograms of size 320x320.

Metric	Value and Standard Deviation	
	Epoch Level	**Subjects Level**
Accuracy	$56.36\% \pm 4.56\%$	$59.09\% \pm 6.43\%$
F1 Score	0.47 ± 0.06	0.54 ± 0.10
Sensitivity	0.47 ± 0.14	0.56 ± 0.24
Specificity	0.64 ± 0.16	0.62 ± 0.26
AUC	0.57 ± 0.05	0.63 ± 0.10
	Subjects Level for:	
	PD Mild and Non-PD	**PD Moderate and Non-PD**
Accuracy	$62.92\% \pm 11.23\%$	$58.24\% \pm 11.21\%$
F1 Score	0.59 ± 0.15	0.51 ± 0.15
Sensitivity	0.63 ± 0.25	0.55 ± 0.30
Specificity	0.61 ± 0.33	0.63 ± 0.30
AUC	0.68 ± 0.17	0.63 ± 0.13

Table 4.6: Average values and standard deviations for the ResNet18 2D model without ImageNet using spectrograms of size 320x320.

4.3 Subject Level ResNet18 3D

For the subject level ResNet18 3D model, the epochs from the 32 common channels for 27 segments were stacked together in order of channels, producing a sequence of $32 \times 27 = 864$ 2D time-frequency series and spectrograms per subject. Therefore, the prediction was on the subject level.

4.3.1 Time-Frequency Series

Table 4.7 shows a subject level accuracy of $55.00\% \pm 3.54\%$ and an AUC score of 0.41 ± 0.15. Figure 4.7 shows that the model, on average, accurately predicts 10.4 out of the 11 non-PD subjects correctly and 0.6 out of 9 PD subjects.

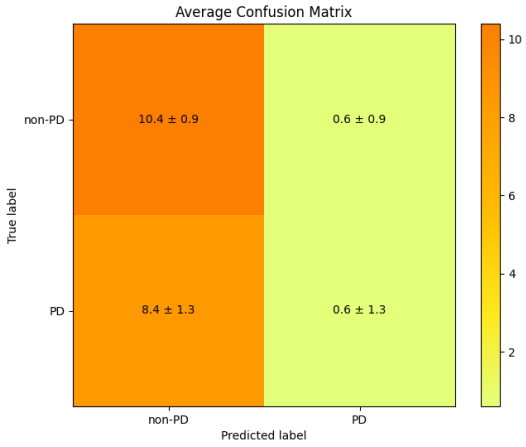

Figure 4.7: Average confusion metric across the five folds for the subject level ResNet18 3D model using time-frequency series.

Metric	Value and Standard Deviation
Subject Level	
Accuracy	$55.00\% \pm 3.54\%$
F1 Score	0.09 ± 0.19
Sensitivity	0.07 ± 0.15
Specificity	0.95 ± 0.08
AUC	0.41 ± 0.15
Subjects Level for PD Mild and Non-PD	
Accuracy	$65.15\% \pm 4.07\%$
F1 Score	0.08 ± 0.18
Sensitivity	0.07 ± 0.15
Specificity	0.95 ± 0.08
AUC	0.43 ± 0.16
Subjects Level for PD Moderate and Non-PD	
Accuracy	$73.62\% \pm 2.93\%$
F1 Score	0.07 ± 0.15
Sensitivity	0.07 ± 0.15
Specificity	0.95 ± 0.08
AUC	0.37 ± 0.15

Table 4.7: Average values and standard deviations for the subject level ResNet18 3D model using time-frequency series.

4.3.2 Spectrogram of Size 244x244

Table 4.8 shows a subject level accuracy of $67.00\% \pm 7.58\%$ and an AUC score of 0.68 ± 0.09. Figure 4.8 shows that the model, on average, accurately predicts 9.2 out of the 11 non-PD subjects and 4.2 out of 9 PD subjects.

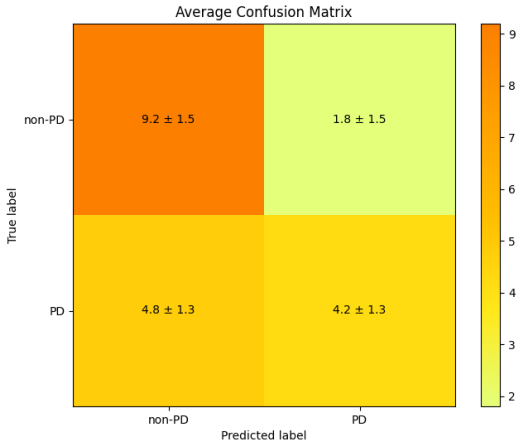

Figure 4.8: Average confusion metric across the five folds for the subject level ResNet18 3D model using spectrograms of size 244x244.

Metric	Value and Standard Deviation
Subjects Level	
Accuracy	$67.00\% \pm 7.58\%$
F1 Score	0.55 ± 0.11
Sensitivity	0.47 ± 0.14
Specificity	0.84 ± 0.13
AUC	0.68 ± 0.09
Subjects Level for PD Mild and Non-PD	
Accuracy	$72.43\% \pm 11.90\%$
F1 Score	0.56 ± 0.14
Sensitivity	0.51 ± 0.11
Specificity	0.84 ± 0.13
AUC	0.69 ± 0.12
Subjects Level for PD Moderate and Non-PD	
Accuracy	$73.71\% \pm 8.75\%$
F1 Score	0.34 ± 0.33
Sensitivity	0.43 ± 0.43
Specificity	0.84 ± 0.13
AUC	0.68 ± 0.19

Table 4.8: Average values and standard deviations for the subject level ResNet18 3D model using spectrograms of size 244x244.

4.3.3 Spectrogram of Size 320x320

Table 4.9 shows a subject level Accuracy of $57.00\% \pm 10.36\%$ and an AUC score of 0.57 ± 0.09. Figure 4.9 shows that the model, on average, accurately predicts 9 out of the 11 non-PD subjects correctly and 2.4 out of 9 PD subjects.

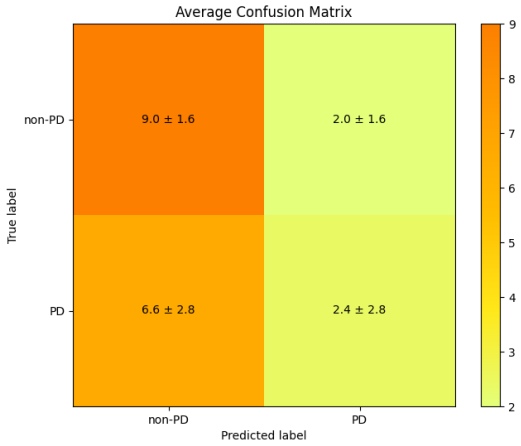

Figure 4.9: Average confusion metric across the five folds for the subject level ResNet18 3D model using spectrograms of size 320x320.

Metric	Value and Standard Deviation
Subjects Level	
Accuracy	$57.00\% \pm 10.36\%$
F1 Score	0.30 ± 0.28
Sensitivity	0.27 ± 0.31
Specificity	0.82 ± 0.14
AUC	0.57 ± 0.09
Subjects Level for PD Mild and Non-PD	
Accuracy	$64.93\% \pm 6.03\%$
F1 Score	0.32 ± 0.25
Sensitivity	0.31 ± 0.31
Specificity	0.82 ± 0.14
AUC	0.56 ± 0.13
Subjects Level for PD Moderate and Non-PD	
Accuracy	$66.95\% \pm 11.30\%$
F1 Score	0.17 ± 0.23
Sensitivity	0.20 ± 0.30
Specificity	0.82 ± 0.14
AUC	0.62 ± 0.17

Table 4.9: Average values and standard deviations for the subject level ResNet18 3D model using spectrograms of size 320x320.

4.4 Segment Level ResNet18 3D

The segment level ResNet18 3D model predicted on the segment level with 32 common channels, and therefore, the subject level metrics were calculated with a hard threshold of 0.5.

4.4.1 Time-Frequency Series

Table 4.10 shows a subject level accuracy of $54.55\% \pm 8.51\%$ and an AUC score of 0.60 ± 0.13. Figure 4.10 shows that the model, on average, accurately predicts 6.6 out of the 12 non-PD subjects correctly and 5.4 out of 10 PD subjects.

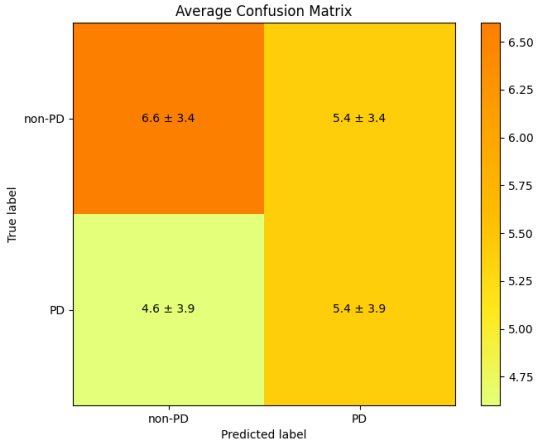

Figure 4.10: Average confusion metric across the five folds for the segment level ResNet18 3D model using time-frequency series.

Metric	Value and Standard Deviation	
	Segment Level	**Subjects Level**
Accuracy	$53.29\% \pm 4.21\%$	$54.55\% \pm 8.51\%$
F1 Score	0.48 ± 0.11	0.47 ± 0.23
Sensitivity	0.53 ± 0.24	0.54 ± 0.39
Specificity	0.54 ± 0.19	0.55 ± 0.28
AUC	0.55 ± 0.07	0.60 ± 0.13
	Subjects Level for:	
	PD Mild and Non-PD	**PD Moderate and Non-PD**
Accuracy	$54.14\% \pm 6.09\%$	$51.53\% \pm 11.41\%$
F1 Score	0.42 ± 0.28	0.45 ± 0.23
Sensitivity	0.50 ± 0.43	0.55 ± 0.42
Specificity	0.54 ± 0.33	0.50 ± 0.23
AUC	0.59 ± 0.10	0.55 ± 0.16

Table 4.10: Average values and standard deviations for the segment level ResNet18 3D model using time-frequency series.

4.4.2 Spectrogram of Size 244x244

Table 4.11 shows a subject level accuracy of $51.82\% \pm 10.46\%$ and an AUC score of 0.57 ± 0.10. Figure 4.11 shows that the model, on average, accurately predicts 5.6 out of the 12 non-PD subjects correctly and 5.8 out of 10 PD subjects.

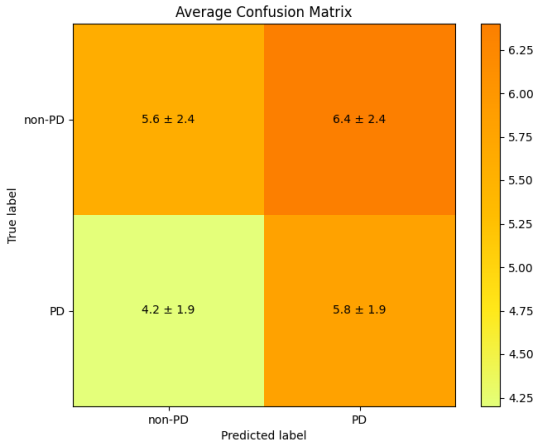

Figure 4.11: Average confusion metric across the five folds for the segment level ResNet18 3D model using spectrograms of size 244x244.

Metric	**Value and Standard Deviation**	
	Segment Level	**Subjects Level**
Accuracy	$53.20\% \pm 5.95\%$	$51.82\% \pm 10.46\%$
F1 Score	0.47 ± 0.11	0.52 ± 0.10
Sensitivity	0.50 ± 0.21	0.58 ± 0.19
Specificity	0.56 ± 0.18	0.47 ± 0.20
AUC	0.56 ± 0.09	0.57 ± 0.10
	Subjects Level for:	
	PD Mild and Non-PD	**PD Moderate and Non-PD**
Accuracy	$58.71\% \pm 10.73\%$	$45.67\% \pm 11.46\%$
F1 Score	0.58 ± 0.10	0.43 ± 0.18
Sensitivity	0.63 ± 0.17	0.50 ± 0.26
Specificity	0.55 ± 0.19	0.43 ± 0.18
AUC	0.62 ± 0.10	0.49 ± 0.12

Table 4.11: Average values and standard deviations for the segment level ResNet18 3D model using spectrograms of size 244x244.

4.4.3 Spectrogram of Size 320x320

Table 4.12 shows a subject level accuracy of $52.73\% \pm 6.89\%$ and an AUC score of 0.53 ± 0.04. Figure 4.12 shows that the model, on average, accurately predicts 8.2 out of the 12 non-PD subjects correctly and 3.4 out of 10 PD subjects.

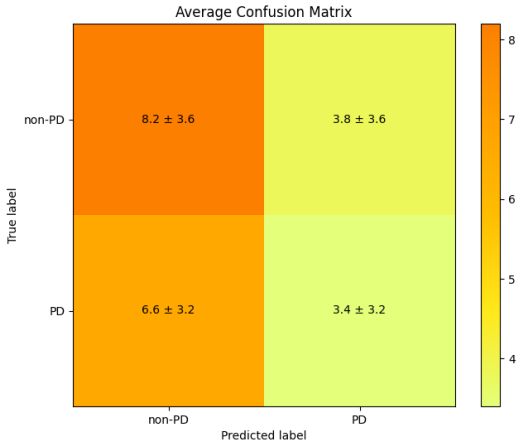

Figure 4.12: Average confusion metric across the five folds for the segment level ResNet18 3D model using spectrograms of size 320x320.

Metric	Value and Standard Deviation	
	Segment Level	**Subjects Level**
Accuracy	$51.49\% \pm 2.57\%$	$52.73\% \pm 6.89\%$
F1 Score	0.32 ± 0.18	0.31 ± 0.29
Sensitivity	0.31 ± 0.22	0.34 ± 0.32
Specificity	0.68 ± 0.20	0.68 ± 0.30
AUC	0.53 ± 0.03	0.53 ± 0.04
	Subjects Level for:	
	PD Mild and Non-PD	**PD Moderate and Non-PD**
Accuracy	$55.38\% \pm 8.25\%$	$53.02\% \pm 10.81\%$
F1 Score	0.33 ± 0.31	0.33 ± 0.31
Sensitivity	0.35 ± 0.34	0.39 ± 0.36
Specificity	0.69 ± 0.28	0.65 ± 0.31
AUC	0.56 ± 0.06	0.54 ± 0.11

Table 4.12: Average values and standard deviations for the segment level ResNet18 3D model using spectrograms of size 320x320.

4.5 EEGNet 2D

The EEGNet 2D model made predictions on the epoch level and the subject level metrics were calculated with a hard threshold of 0.5.

4.5.1 Time-Frequency Series

Table 4.13 shows subject level accuracy at $64.55\% \pm 8.74\%$ and an AUC score at 0.67 ± 0.08, meaning it is the best-performing time-frequency model tested. Figure 4.13 shows that the model, on average, accurately predicts 9.8 out of the 12 non-PD subjects correctly and 4.4 out of 10 PD subjects.

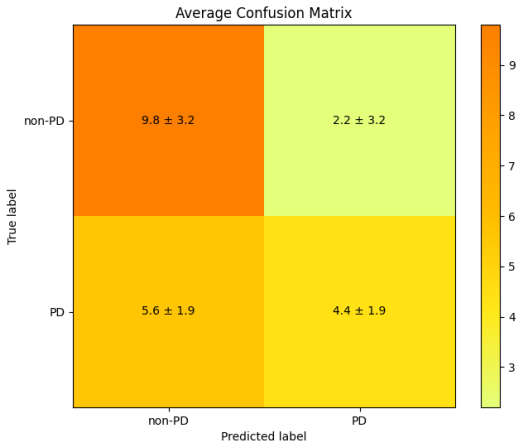

Figure 4.13: Average confusion metric across the five folds for the EEGNet 2D model using time-frequency series.

Metric	Value and Standard Deviation	
	Epoch Level	**Subjects Level**
Accuracy	$58.70\% \pm 4.66\%$	$64.55\% \pm 8.74\%$
F1 Score	0.46 ± 0.07	0.51 ± 0.13
Sensitivity	0.43 ± 0.13	0.44 ± 0.19
Specificity	0.71 ± 0.15	0.82 ± 0.27
AUC	0.59 ± 0.05	0.67 ± 0.08
	Subjects Level for:	
	PD Mild and Non-PD	**PD Moderate and Non-PD**
Accuracy	$65.96\% \pm 10.34\%$	$67.31\% \pm 11.68\%$
F1 Score	0.52 ± 0.20	0.56 ± 0.12
Sensitivity	0.47 ± 0.27	0.46 ± 0.13
Specificity	0.85 ± 0.24	0.83 ± 0.24
AUC	0.69 ± 0.12	0.69 ± 0.11

Table 4.13: Average values and standard deviations for the EEGNet 2D model using time-frequency series.

4.5.2 Spectrogram of Size 244x244

Table 4.14 shows a subject level accuracy of $60.00\% \pm 7.47\%$ and an AUC score of 0.67 ± 0.06. Figure 4.14 shows that the model, on average, accurately predicts 10.4 out of the 12 non-PD subjects and 2.8 out of 10 PD subjects.

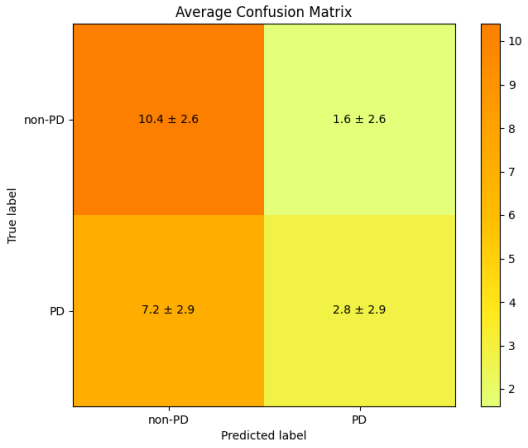

Figure 4.14: Average confusion metric across the five folds for the EEGNet 2D model using spectrograms of size 244x244.

Metric	**Value and Standard Deviation**	
	Epoch Level	**Subjects Level**
Accuracy	$57.52\% \pm 2.90\%$	$60.00\% \pm 7.47\%$
F1 Score	0.36 ± 0.10	0.32 ± 0.28
Sensitivity	0.30 ± 0.14	0.28 ± 0.29
Specificity	0.77 ± 0.14	0.87 ± 0.22
AUC	0.57 ± 0.03	0.67 ± 0.06
	Subjects Level for:	
	PD Mild and Non-PD	**PD Moderate and Non-PD**
Accuracy	$63.99\% \pm 11.21\%$	$59.13\% \pm 9.17\%$
F1 Score	0.34 ± 0.36	0.28 ± 0.30
Sensitivity	0.32 ± 0.37	0.25 ± 0.29
Specificity	0.88 ± 0.19	0.88 ± 0.22
AUC	0.73 ± 0.08	0.67 ± 0.07

Table 4.14: Average values and standard deviations for the EEGNet 2D model using spectrograms of size 244x244.

4.5.3 Spectrogram of Size 320x320

Table 4.15 shows a subject level accuracy of $60.00\% \pm 4.97\%$ and an AUC score of 0.69 ± 0.07, meaning it is the best spectrogram model tested. Figure 4.15 shows that the model, on average, accurately predicts 11 out of the 12 non-PD subjects and 2.2 out of 10 PD subjects.

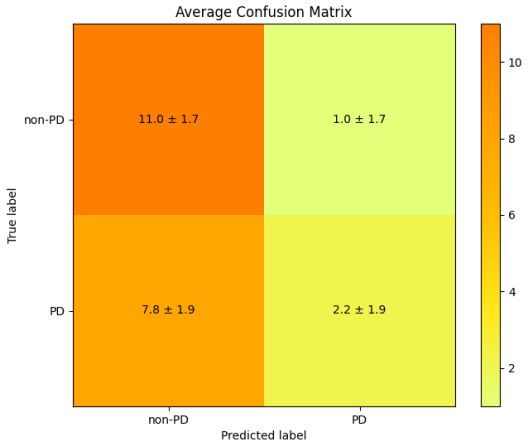

Figure 4.15: Average confusion metric across the five folds for the EEGNet 2D model using spectrograms of size 320x320.

Metric	Value and Standard Deviation	
	Epoch Level	**Subjects Level**
Accuracy	$58.27\% \pm 2.88\%$	$60.00\% \pm 4.97\%$
F1 Score	0.38 ± 0.08	0.30 ± 0.21
Sensitivity	0.32 ± 0.11	0.22 ± 0.19
Specificity	0.78 ± 0.10	0.92 ± 0.14
AUC	0.57 ± 0.04	0.69 ± 0.07
	Subjects Level for:	
	PD Mild and Non-PD	**PD Moderate and Non-PD**
Accuracy	$62.66\% \pm 9.61\%$	$61.57\% \pm 5.19\%$
F1 Score	0.34 ± 0.22	0.33 ± 0.21
Sensitivity	0.24 ± 0.18	0.24 ± 0.19
Specificity	0.93 ± 0.10	0.91 ± 0.14
AUC	0.76 ± 0.11	0.68 ± 0.06

Table 4.15: Average values and standard deviations for the EEGNet 2D model using spectrograms of size 320x320.

4.6 Best Results

Table 4.16 shows a comparison of the best performing time-frequency and spectrogram model. Both resulting from the EEGNet 2D model using time-frequency series and spectrograms of size 320x320. The accuracy's achieved was $64.55\% \pm 8.74\%$ and $60.00\% \pm 4.97\%$, respectively. The AUC was 0.67 ± 0.08 compared to 0.69 ± 0.07, respectively. Figures 4.16 and 4.17 show that the model, on average, accurately predicts 9.8 and 11 out of 12, respectively, for non-PD and 4.4 and 2.2 out of 10 for PD.

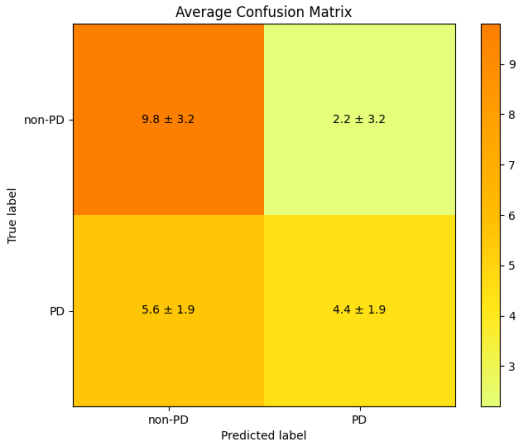

Figure 4.16: Average confusion metric for the EEGNet 2D model using time-frequency series

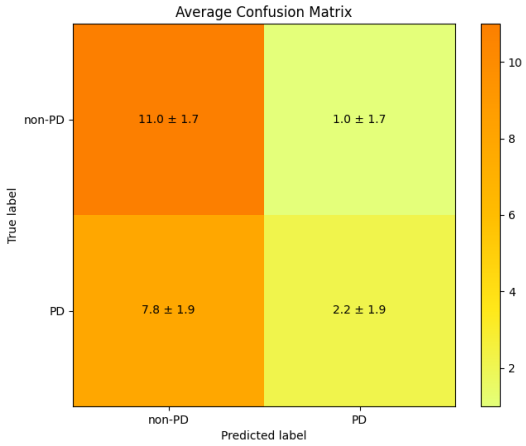

Figure 4.17: Average confusion metricfor for the EEG-Net 2D model using spectrograms of size 320x320.

Metric	Best Run For:	
Epoch Level	**Time-Frequency**	**Spectrograms**
Accuracy	$58.70\% \pm 4.66\%$	$58.27\% \pm 2.88\%$
F1 Score	0.46 ± 0.07	0.38 ± 0.08
Sensitivity	0.43 ± 0.13	0.32 ± 0.11
Specificity	0.71 ± 0.15	0.78 ± 0.10
AUC	0.59 ± 0.05	0.57 ± 0.04
Subjects Level		
Accuracy	$64.55\% \pm 8.74\%$	$60.00\% \pm 4.97\%$
F1 Score	0.51 ± 0.13	0.30 ± 0.21
Sensitivity	0.44 ± 0.19	0.22 ± 0.19
Specificity	0.82 ± 0.27	0.92 ± 0.14
AUC	0.67 ± 0.08	0.69 ± 0.07
PD Mild and Non-PD		
Accuracy	$65.96\% \pm 10.34\%$	$62.66\% \pm 9.61\%$
F1 Score	0.52 ± 0.20	0.34 ± 0.22
Sensitivity	0.47 ± 0.27	0.24 ± 0.18
Specificity	0.85 ± 0.24	0.93 ± 0.10
AUC	0.69 ± 0.12	0.76 ± 0.11
PD Moderate and Non-PD		
Accuracy	$67.31\% \pm 11.68\%$	$61.57\% \pm 5.19\%$
F1 Score	0.56 ± 0.12	0.33 ± 0.21
Sensitivity	0.46 ± 0.13	0.24 ± 0.19
Specificity	0.83 ± 0.24	0.91 ± 0.14
AUC	0.69 ± 0.11	0.68 ± 0.06

Table 4.16: Average values and standard deviations for the best performing models. Both resulting from the EEGNet 2D model using time-frequency series and spectrograms of size 320x320.

Chapter 5

Discussion

Our analysis revealed that EEGNet had the best 2D model performance. The AUC scores for time-frequency series and spectrograms of sizes 244x244 and 320x320 were 0.67 ± 0.08, 0.67 ± 0.06, and 0.69 ± 0.07, respectively, indicating that EEG has the potential for detecting PD. The AUC scores for mild PD subjects compared to non-PD subjects and for moderate PD subjects compared to non-PD subjects were close for both time-frequency series and spectrograms of size 244x244 and 320x320, with AUC scores of 0.69 ± 0.12, 0.73 ± 0.08 and 0.76 ± 0.11, respectively, for the mild PD comparison, and AUC scores of 0.69 ± 0.11, 0.67 ± 0.07 and 0.68 ± 0.06, for the moderate PD comparison. This suggests that the model's predictive performance across different PD severity levels remained consistent, indicating robustness in distinguishing between PD and non-PD subjects regardless of disease severity.

The ResNet18 2D model performed better for both time-frequency series and spectrograms of size 244x244 and 320x320 with AUC scores of 0.65 ± 0.06, 0.64 ± 0.09 and 0.63 ± 0.10, respectively, compared to ResNet18 2D model pre-trained on ImageNet with AUC score of 0.61 ± 0.17, 0.64 ± 0.10 and 0.61 ± 0.08. The reason could be that ImageNet does not include relevant images of EEG signals and spectrograms. However, it did not perform as well as EEGNet, possibly due to EEGNet's design for EEG signals. EEGNet is a more compact model with fewer layers, reducing overfitting, which is better for a smaller dataset.

Discussion

Our results show that time-frequency series as input to subject and segment level ResNet18 3D models got drastically different results with AUC scores of 0.41 ± 0.15 and 0.60 ± 0.13, respectively. This indicates that the smaller training set in the subject level ResNet18 3D negatively affects performance. However, the performance for the spectrograms as input to the 3D models contradicts this, with the best performing 3D models with the use of 244x244 spectrograms achieving an AUC score of 0.68 ± 0.09 and 0.57 ± 0.10 for subject level and segment level Resnet 18 3D models, respectively.

EEGNet was the best-performing model when comparing the 2D and 3D models for the time-frequency series and the spectrograms. The spectrograms of size 320x320 performed the best, with an AUC score of 0.69 ± 0.07. Notably, the subject level ResNet18 3D model utilizing spectrograms of size 244x244 performed nearly as well as EEGNet, with an AUC of 0.68 ± 0.09. While the results do not show an advantage in utilizing 3D models over 2D models, it is worth noting that the potential performance increase with 3D models could be bigger with a larger dataset. This is because combining epochs in 3D models reduces the amount of training data compared to the individual epoch input in the 2D models.

While EEGNet with spectrograms of size 320x320 achieved the highest AUC score of 0.69 ± 0.07 among all models. For the other models, including ResNet18 2D and 3D models, spectrograms of size 244x244 consistently yielded higher AUC scores compared to size 320x320. This indicates that despite the overall higher AUC score achieved by EEGNet with spectrograms of size 320x320, the spectrograms of size 244x244 were more effective for other model architectures in capturing relevant features for PD detection.

Comparing our best-performing model to Kurbatskaya's work[12], our AUC score was marginally lower, at 0.69 ± 0.07 compared to their 0.72. However, our score fell short compared to the work of Sugden [11], which achieved an AUC score of 0.88. Our model was trained on a larger dataset of 110 subjects across three centres, compared to Sugden's model, which utilized data from two centres with 82 subjects. However, compared to their work, our larger dataset size and increased number of centres potentially make our results more robust and generalizable.

Chapter 6

Conclusion

DL has demonstrated promising capabilities in classifying PD from EEG signals and the performance benefits of representing EEG signals as spectrograms for the input to DL models, underlain by the results from this thesis. Using EEGNet, both time-frequency series and spectrograms achieved respectable AUC scores of 0.67 ± 0.08 and 0.69 ± 0.07, with corresponding accuracies of $64.55\% \pm 8.74\%$ and $60.00\% \pm 4.97\%$. Using ResNet18, both with and without preset weights from ImageNet, led to AUC scores in the low to mid 0.6. Additionally, exploration of various 3D versions of ResNet18 yielded promising results at the subject level for spectrograms, but was less promising for the time-frequency inputs, with the best 3D model achieving an accuracy of $67.00\% \pm 7.58\%$ and an AUC score of 0.68 ± 0.09. While our comparison highlights strengths, it is important to note certain limitations. The dataset contains few subjects, especially for the subject level ResNet18 3D model. The subjects from the California dataset, the MDS-UPDRS score were not collected by a board-certified neurologist but by personnel who had completed online training.[61]

6.1 Future Work

This study has shown promising results for testing different input types against multiple different CNN models. However, there is a need for im-

provement, where some potential directions are as follows:

- Acquire a larger dataset.
- Experiment with different spectrogram transformation algorithms.
- Implement variable sampling of epochs to increase trainable data.
- Experiment with generating synthetic EEG data using different generative adversarial networks.
- Experiment with different augmentation methods

Bibliography

[1] Parkinsons Europe. *What is Parkinson's?* Parkinson's Europe. URL: https://parkinsonseurope.org/understanding-parkinsons/what-is-parkinsons/ (visited on 04/08/2024).

[2] Mayo Clinic. *Parkinson's disease - Diagnosis and treatment - Mayo Clinic*. URL: https://www.mayoclinic.org/diseases-conditions/parkinsons-disease/diagnosis-treatment/drc-20376062 (visited on 04/08/2024).

[3] Jeremy Rodriguez. *Parkinson's Disease Life Expectancy*. Griswold Home Care. URL: https://www.griswoldhomecare.com/blog/2022/december/parkinsons-disease-life-expectancy/ (visited on 03/08/2024).

[4] Shu Lih Oh et al. "A deep learning approach for Parkinson's disease diagnosis from EEG signals". In: *Neural Computing and Applications* 32.15 (Aug. 1, 2020), pp. 10927–10933. ISSN: 1433-3058. DOI: 10.1007/s00521-018-3689-5. URL: https://doi.org/10.1007/s00521-018-3689-5 (visited on 03/11/2024).

[5] Majid Aljalal et al. "Detection of Parkinson's disease from EEG signals using discrete wavelet transform, different entropy measures, and machine learning techniques". In: *Scientific Reports* 12.1 (Dec. 29, 2022). Publisher: Nature Publishing Group, p. 22547. ISSN: 2045-2322. DOI: 10.1038/s41598-022-26644-7. URL: https://www.nature.com/articles/s41598-022-26644-7 (visited on 03/11/2024).

[6] Soojin Lee et al. "A convolutional-recurrent neural network approach to resting-state EEG classification in Parkinson's disease". In: *Journal of Neuroscience Methods* 361 (Sept. 1, 2021), p. 109282. ISSN: 0165-0270. DOI: 10.1016/j.jneumeth.2021.109282. URL: https://www.

sciencedirect.com/science/article/pii/S016502702100217X (visited on 03/11/2024).

[7] Syed Aamir Ali Shah, Lei Zhang, and Abdul Bais. "Dynamical system based compact deep hybrid network for classification of Parkinson disease related EEG signals". In: *Neural Networks* 130 (Oct. 1, 2020), pp. 75–84. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2020.06.018. URL: https://www.sciencedirect.com/science/article/pii/S0893608020302331 (visited on 03/11/2024).

[8] Hui Wen Loh et al. "GaborPDNet: Gabor Transformation and Deep Neural Network for Parkinson's Disease Detection Using EEG Signals". In: *Electronics* 10.14 (Jan. 2021). Number: 14 Publisher: Multidisciplinary Digital Publishing Institute, p. 1740. ISSN: 2079-9292. DOI: 10.3390/electronics10141740. URL: https://www.mdpi.com/2079-9292/10/14/1740 (visited on 03/11/2024).

[9] Smith K. Khare, Varun Bajaj, and U. Rajendra Acharya. "PDCNNet: An Automatic Framework for the Detection of Parkinson's Disease Using EEG Signals". In: *IEEE Sensors Journal* 21.15 (Aug. 2021). Conference Name: IEEE Sensors Journal, pp. 17017–17024. ISSN: 1558-1748. DOI: 10.1109/JSEN.2021.3080135. URL: https://ieeexplore.ieee.org/document/9430513 (visited on 03/11/2024).

[10] Marwa Obayya et al. "A novel automated Parkinson's disease identification approach using deep learning and EEG". In: *PeerJ Computer Science* 9 (Nov. 22, 2023), e1663. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.1663. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10703017/ (visited on 03/15/2024).

[11] Richard James Sugden and Phedias Diamandis. "Generalizable electroencephalographic classification of Parkinson's disease using deep learning". In: *Informatics in Medicine Unlocked* 42 (Jan. 1, 2023), p. 101352. ISSN: 2352-9148. DOI: 10.1016/j.imu.2023.101352. URL: https://www.sciencedirect.com/science/article/pii/S2352914823001983 (visited on 03/21/2024).

[12] Anna Kurbatskaya et al. "Assessing Gender Fairness in EEG-Based Machine Learning Detection of Parkinson's Disease: A Multi-Center Study". In: *2023 31st European Signal Processing Conference (EUSIPCO)*. 2023 31st European Signal Processing Conference (EUSIPCO). ISSN: 2076-1465. Sept. 2023, pp. 1020–1024. DOI: 10.23919/EUSIPCO58844.2023.10289837. URL: https://ieeexplore.ieee.org/abstract/document/10289837 (visited on 04/22/2024).

[13] learningeeg. *Basic EEG Electrophysiology*. URL: https://www.learningeeg.com/basic-eeg-electrophysiology (visited on 04/22/2024).

[14] Anita Herigstad, Sigurbjørg Stefansdottir, and Harald Aurlien. "EEG – når og hvordan?" In: *Tidsskrift for Den norske legeforening* (Jan. 8, 2013). ISSN: 0029-2001. DOI: 10.4045/tidsskr.12.0087. URL: https://tidsskriftet.no/2013/01/tema-klinisk-nevrofysiologi/eeg-nar-og-hvordan (visited on 04/22/2024).

[15] tmsi. *The 10-20 System for EEG*. URL: https://info.tmsi.com/blog/the-10-20-system-for-eeg (visited on 04/22/2024).

[16] Wikipedia. *10–20 system (EEG)*. In: *Wikipedia*. Page Version ID: 1200194996. Jan. 29, 2024. URL: https://en.wikipedia.org/w/index.php?title=10%E2%80%9320_system_(EEG)&oldid=1200194996 (visited on 04/09/2024).

[17] Ahmad Chaddad et al. *Electroencephalography Signal Processing: A Comprehensive Review and Analysis of Methods and Techniques - PMC*. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10385593/ (visited on 03/04/2024).

[18] Audun Eltvik. "Deep Learning for the Classification of EEG Time-Frequency Representations". In: ().

[19] neuroimage. *Tutorials/Epoching - Brainstorm*. URL: https://neuroimage.usc.edu/brainstorm/Tutorials/Epoching (visited on 04/22/2024).

[20] Stefan Appelhoff et al. *PyPREP: A Python implementation of the preprocessing pipeline (PREP) for EEG data*. original-date: 2018-04-12T11:29:52Z. Mar. 4, 2024. URL: https://github.com/sappelhoff/pyprep (visited on 03/18/2024).

[21] Darren Tanner, Kara Morgan-Short, and Steven J. Luck. "How inappropriate high-pass filters can produce artifactual effects and incorrect conclusions in ERP studies of language and cognition". In: *Psychophysiology* 52.8 (Aug. 2015), pp. 997–1009. ISSN: 0048-5772. DOI: 10.1111/psyp.12437. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4506207/ (visited on 04/22/2024).

[22] Nima Bigdely-Shamlo et al. "The PREP pipeline: standardized preprocessing for large-scale EEG analysis". In: *Frontiers in Neuroinformatics* 9 (June 18, 2015). Publisher: Frontiers. ISSN: 1662-5196. DOI: 10.3389/fninf.2015.00016. URL: https://www.frontiersin.org/articles/10.3389/fninf.2015.00016 (visited on 04/11/2024).

[23] PyData Sphinx Theme. *MNE-ICALabel — MNE-ICALabel*. URL: https://mne.tools/mne-icalabel/dev/index.html (visited on 03/18/2024).

[24] Adam Li et al. "MNE-ICALabel: Automatically annotating ICA components with ICLabel in Python". In: *Journal of Open Source Software* 7.76 (Aug. 26, 2022), p. 4484. ISSN: 2475-9066. DOI: 10.21105/joss.04484. URL: https://joss.theoj.org/papers/10.21105/joss.04484 (visited on 04/11/2024).

[25] PyData Sphinx Theme. *autoreject.RejectLog — autoreject 0.5.0.dev15+gebb24fe documentation*. URL: https://autoreject.github.io/dev/generated/autoreject.RejectLog.html (visited on 03/18/2024).

[26] Mainak Jas et al. *Autoreject: Automated artifact rejection for MEG and EEG data - ScienceDirect*. URL: https://www.sciencedirect.com/science/article/abs/pii/S1053811917305013 (visited on 04/11/2024).

[27] weisang. *Continuous Wavelet Transform (CWT) - weisang.com*. URL: https://www.weisang.com/en/documentation/timefreqspectrumalgorithmscwt_en/ (visited on 04/11/2024).

[28] Mike Clayton. *The power of the Continuous Wavelet Transform (CWT) in machine learning | by Mike Clayton | Medium*. URL: https://medium.com/@maclayton/the-power-of-the-continuous-wavelet-transform-cwt-in-machine-learning-980e80ecfb8a (visited on 04/11/2024).

[29] *Continuous wavelet transform*. In: *Wikipedia*. Page Version ID: 1199163846. Jan. 26, 2024. URL: https://en.wikipedia.org/w/index.php?title=Continuous_wavelet_transform&oldid=1199163846 (visited on 04/08/2024).

[30] GeeksforGeeks. *Introduction to Deep Learning*. GeeksforGeeks. Section: AI-ML-DS. June 1, 2018. URL: https://www.geeksforgeeks.org/introduction-deep-learning/ (visited on 03/11/2024).

[31] GeeksforGeeks. *Supervised and Unsupervised learning*. GeeksforGeeks. Section: Computer Subject. Oct. 1, 2017. URL: https://www.geeksforgeeks.org/supervised-unsupervised-learning/ (visited on 04/03/2024).

[32] Tamanna. *Backpropagation in Neural Networks: A Comprehensive Guide*. Medium. May 4, 2023. URL: https://medium.com/@tamanna18/backpropagation-in-neural-networks-a-comprehensive-guide-3d36151b8fb4 (visited on 04/03/2024).

[33] Everton Gomede. *Understanding Loss Functions in Deep Learning*. The Modern Scientist. Mar. 9, 2024. URL: https://medium.com/the-modern-scientist/understanding-loss-functions-in-deep-learning-9f06e5090f20 (visited on 04/09/2024).

[34] Hunter Phillips. *A Simple Introduction to Softmax*. Medium. May 10, 2023. URL: https://medium.com/@hunter-j-phillips/a-simple-introduction-to-softmax-287712d69bac (visited on 04/09/2024).

[35] Dario Radečić. *Softmax Activation Function Explained*. Medium. Mar. 21, 2022. URL: https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60 (visited on 04/09/2024).

[36] IBM. *What are Convolutional Neural Networks? | IBM*. URL: https://www.ibm.com/topics/convolutional-neural-networks (visited on 03/15/2024).

[37] Pavan Vadapalli. *Using Convolutional Neural Network for Image Classification*. upGrad blog. Aug. 14, 2020. URL: https://www.upgrad.com/blog/using-convolutional-neural-network-for-image-classification/ (visited on 04/17/2024).

[38] washington.edu. *convolutional_networks*. URL: https://courses.cs.washington.edu/courses/cse446/21au/sections/08/convolutional_networks.html#Convolutions-for-Images (visited on 04/08/2024).

[39] Vaibhav Rastogi. *Fully Connected Layer*. Medium. Sept. 8, 2023. URL: https://medium.com/@vaibhav1403/fully-connected-layer-f13275337c7c (visited on 04/03/2024).

[40] Datagen. *ResNet: The Basics and 3 ResNet Extensions*. Datagen. URL: https://datagen.tech/guides/computer-vision/resnet/ (visited on 04/03/2024).

[41] Azeem- I. *Understanding ResNet Architecture: A Deep Dive into Residual Neural Network*. Medium. Nov. 14, 2023. URL: https://medium.com/@ibtedaazeem/understanding-resnet-architecture-a-deep-dive-into-residual-neural-network-2c792e6537a9 (visited on 04/03/2024).

[42] Kaiming He et al. *Deep Residual Learning for Image Recognition*. Dec. 10, 2015. arXiv: 1512.03385[cs]. URL: http://arxiv.org/abs/1512.03385 (visited on 04/08/2024).

[43] Vernon J. Lawhern et al. "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces". In: *Journal of Neural Engineering* 15.5 (July 2018). Publisher: IOP Publishing, p. 056013. ISSN: 1741-2552. DOI: 10.1088/1741-2552/aace8c. URL: https://dx.doi.org/10.1088/1741-2552/aace8c (visited on 03/11/2024).

[44] Saba Hesaraki. *3D CNN*. Medium. Nov. 11, 2023. URL: https://medium.com/@saba99/3d-cnn-4ccfab119cc2 (visited on 04/03/2024).

[45] Hasib Zunair. *Keras documentation: 3D image classification from CT scans*. Jan. 11, 2024. URL: https://keras.io/examples/vision/3D_image_classification/ (visited on 04/03/2024).

[46] Niklas Donges. *What Is Transfer Learning? A Guide for Deep Learning | Built In*. Dec. 9, 2022. URL: https://builtin.com/data-science/transfer-learning (visited on 03/19/2024).

[47] Matt Reynolds. *New computer vision challenge wants to teach robots to see in 3D*. New Scientist. Apr. 7, 2017. URL: https://www.newscientist.com/article/2127131-new-computer-vision-challenge-wants-to-teach-robots-to-see-in-3d/ (visited on 04/03/2024).

[48] John Markoff. "Seeking a Better Way to Find Web Images". In: *The New York Times* (Nov. 19, 2012). ISSN: 0362-4331. URL: https://www.nytimes.com/2012/11/20/science/for-web-images-creating-new-technology-to-seek-and-find.html (visited on 04/03/2024).

[49] Deepchecks. *Machine Learning Checkpointing*. Deepchecks. URL: https://deepchecks.com/glossary/machine-learning-checkpointing/ (visited on 04/17/2024).

[50] GeeksforGeeks. *Confusion Matrix in Machine Learning*. GeeksforGeeks. Section: Data Science. Oct. 15, 2017. URL: https://www.geeksforgeeks.org/confusion-matrix-machine-learning/ (visited on 04/03/2024).

[51] Aditya Mishra. *Metrics to Evaluate your Machine Learning Algorithm*. Medium. May 28, 2020. URL: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234 (visited on 04/09/2024).

[52] Vikram Singh. *Sensitivity vs. Specificity: What's the Difference? - Shiksha Online*. Oct. 13, 2023. URL: https://www.shiksha.com/online-courses/articles/sensitivity-vs-specificity/ (visited on 04/02/2024).

[53] Nikolaj Buhl. *F1 Score in Machine Learning*. July 18, 2023. URL: https://encord.com/blog/f1-score-in-machine-learning/ (visited on 04/02/2024).

[54] Shailey Dash. *Understanding the ROC and AUC Intuitively*. Medium. Feb. 7, 2023. URL: https://medium.com/@shaileydash/understanding-the-roc-and-auc-intuitively-31ca96445c02 (visited on 04/03/2024).

[55] Shanthababu Pandian. *K-Fold Cross Validation Technique and its Essentials*. Analytics Vidhya. Feb. 17, 2022. URL: https://www.analyticsvidhya.com/blog/2022/02/k-fold-cross-validation-technique-and-its-essentials/ (visited on 03/21/2024).

[56] Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. "On the Stratification of Multi-label Data". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Dimitrios Gunopulos et al. Berlin, Heidelberg: Springer, 2011, pp. 145–158. ISBN: 978-3-642-23808-6. DOI: 10.1007/978-3-642-23808-6_10.

[57] evidentlyai. *How to use classification threshold to balance precision and recall*. URL: https://www.evidentlyai.com/classification-metrics/classification-threshold (visited on 04/09/2024).

[58] Robert J. Barry et al. "EEG differences between eyes-closed and eyes-open resting conditions". In: *Clinical Neurophysiology* 118.12 (Dec. 1, 2007), pp. 2765–2773. ISSN: 1388-2457. DOI: 10.1016/j.clinph.2007.07.028. URL: https://www.sciencedirect.com/science/article/pii/S1388245707004002 (visited on 03/01/2024).

[59] Pablo Martínez-Martín et al. "Parkinson's disease severity levels and MDS-Unified Parkinson's Disease Rating Scale". In: *Parkinsonism & Related Disorders* 21.1 (Jan. 1, 2015). Publisher: Elsevier, pp. 50–54. ISSN: 1353-8020, 1873-5126. DOI: 10.1016/j.parkreldis.2014.10.026. URL: https://www.prd-journal.com/article/S1353-8020(14)00411-8/abstract (visited on 05/05/2024).

[60] Alberto Jaramillo-Jimenez et al. "Spectral features of resting-state EEG in Parkinson's Disease: A multicenter study using functional data analysis". In: *Clinical Neurophysiology* 151 (July 1, 2023), pp. 28–40. ISSN: 1388-2457. DOI: 10.1016/j.clinph.2023.03.363. URL:

https://www.sciencedirect.com/science/article/pii/S1388245723005989 (visited on 03/04/2024).

[61] Alexander P. Rockhill et al. *UC San Diego Resting State EEG Data from Patients with Parkinson's Disease - OpenNeuro*. OpenNeuro. 2021. URL: https://openneuro.org/datasets/ds002778/versions/1.0.5 (visited on 03/21/2024).

[62] James F Cavanagh Jcavanagh@Unm.Edu. *EEG: 3-Stim Auditory Oddball and Rest in Parkinson's*. 2021. DOI: 10.18112/OPENNEURO.DS003490.V1.1.0. URL: https://openneuro.org/datasets/ds003490/versions/1.1.0 (visited on 05/09/2024).

[63] John Muradeli. *OverLordGoldDragon/ssqueezepy: QoL, cleanups, fixes*. Version 0.6.3. Language: en. Jan. 23, 2022. DOI: 10.5281/ZENODO.5080508. URL: https://zenodo.org/record/5080508 (visited on 04/11/2024).

[64] Google. *TensorFlow*. TensorFlow. URL: https://www.tensorflow.org/ (visited on 04/05/2024).

[65] Chollet. *Keras: Deep Learning for humans*. URL: https://keras.io/ (visited on 04/05/2024).

[66] NVIDIA. *NVIDIA Tesla V100*. NVIDIA. URL: https://www.nvidia.com/en-gb/data-center/tesla-v100/ (visited on 04/05/2024).

[67] Martin Ytredal Tord and Nikolaas Lindeijer Tim. *Multi-planar segmentation of prostate MRI images*. Stavanger: University of Stavanger, May 2023.

[68] Solovyev et al. *3D convolutional neural networks for stalled brain capillary detection*. In: *Computers in Biology and Medicine*. Vol. 141. Elsevier, 2022, p. 105089.