



People's Democratic Republic of Algeria

Larbi Tebessi University

Faculty of exact sciences and natural and life sciences

Department of Math and computer science

2021/2022



# Master's thesis

In computer science

**Specialty:** Systems and Multimedia (SYM)

---

## Detection of neurodegenerative diseases by the automatic analysis of handwriting

---

**Realized by:** Habes Salah Eddine

**Infront of jury members:**

Dr. Mekhaznia. T

Mr. Abbas. F

Dr. Bennour. A

Mr. Zemmar. A

**President (MCA)**

**Examiner (MCB)**

**Thesis supervisor**

**Co-supervisor**

# Acknowledgment

---

I would like to express my gratitude and appreciation to my thesis supervisor Dr. Bennour Akram for his dedicated support and guidance, his continuous encouragement and unwavering patience.

I would also like to thank my friends who, despite their busy schedules, still managed to find the time to provide me with different tips and ideas.

But most importantly, I would like to express my utmost appreciation to my dear parents and sisters for all the support they provided me with through-out the years.

# Abstract

---

Neurodegenerative diseases (ND) are a serious issue which encompasses a myriad of complex and incurable disorders. In this thesis, we focus on Parkinson's disease (PD), specifically; the detection of PD through the automatic analysis of offline handwriting. To accomplish this task; we propose Park-Net, our own convolutional neural network (CNN) architecture. We proceed to test this CNN on three PD handwriting datasets before comparing the results to state-of-the-art works, and with a **98%** accuracy, and to the best of our knowledge; Park-Net outperforms studies as recent as (2022).

**Keywords:** Neurodegenerative diseases, Parkinson's disease, Artificial intelligence, Image processing, Pattern recognition, Handwriting analysis, Deep learning, Convolutional neural networks, HandPD, NewHandPD, Parkinson's drawings.

# Résumé

---

Les maladies neurodégénératives (ND) sont un problème grave qui englobe une myriade de troubles complexes et incurables. Dans cette thèse, nous nous concentrons sur la maladie de Parkinson (MP), plus précisément ; la détection de la MP par l'analyse automatique de l'écriture manuscrite hors ligne. Afin d'accomplir cette tâche, nous proposons Park-Net ; notre propre architecture de réseau neuronal convolutionnel (RNC). Nous procédons au test de ce RNC sur trois ensembles de données d'écriture manuscrite parkinsonienne avant de comparer les résultats à l'état de l'art, et avec une précision de 98 %, et au meilleur de nos connaissances ; Park-Net surpasse des études aussi récentes que (2022).

**Mots clés :** Maladies neurodégénératives, Maladie de Parkinson, Intelligence artificielle, Traitement d'images, Reconnaissance de formes, Analyse de l'écriture manuscrite, Apprentissage en profondeur, Réseaux de neurones convolutifs, HandPD, NewHandPD, Parkinson's drawings.

# نبذة مختصرة

تعد أمراض التنكس العصبي مشكلة خطيرة تشمل عددًا لا يحصى من الاضطرابات المعقدة وغير القابلة للشفاء. في هذه الأطروحة ، نركز على مرض باركنسون. على وجه التحديد ، سنحاول الكشف عن هذا المرض من خلال التحليل التلقائي للكتابة اليدوية. لإنجاز هذه المهمة. نقترح Park-Net ، وهي بنية شبكتنا العصبية الملتفة (CNN) شرعنا في اختبار هذا CNN على ثلاث مجموعات بيانات بخط يد مرضى الباركنسون قبل مقارنة النتائج بأحدث الأعمال ، وبدقة 98% ، وعلى حد علمنا ؛ يتفوق أداء Park-Net على دراسات حديثة مثل دراسات 2022.

**الكلمات الرئيسية:** الأمراض العصبية التنكسية ، مرض باركنسون ، الذكاء الاصطناعي ، معالجة الصور ، التعرف على الأنماط ، تحليل خط اليد ، التعلم العميق ، الشبكات العصبية التلافيفية ، HandPD ، NewHandPD ، Parkinson's drawings.

# Table of Contents

---

<b>General Introduction</b> .....	<b>10</b>
<b>Chapter 01: Theoretical Concepts</b> .....	<b>12</b>
1.1. Introduction.....	13
1.2. Neurodegenerative diseases.....	13
1.2.1. Alzheimer’s disease.....	13
1.2.2. Parkinson’s disease.....	14
1.2.3. Amyotrophic lateral sclerosis.....	14
1.3. Handwriting recognition.....	15
1.3.1. Dataset types.....	15
1.3.2. Online.....	15
1.3.3. Off-line.....	16
1.4. Convolutional neural networks.....	16
1.4.1. CNN Layers.....	17
1.4.1.1. Convolutional.....	17
1.4.1.2. Pooling Layer.....	17
1.4.1.3. Flatten layer.....	18
1.4.1.4. Fully-connected layers.....	18
1.4.2. Epoch.....	18
1.4.3. Batch size.....	19
1.4.4. Iteration.....	19
1.4.5. Examples of existing architectures.....	19
1.4.5.1. Vgg.....	19
1.4.5.2. ResNet.....	20
1.4.5.3. AlexNet.....	21
1.4.6. CNN training methods.....	22

1.4.6.1.	<i>Training from scratch</i> .....	22
1.4.6.2.	<i>Transfer learning</i> .....	22
1.5.	<b>Over-fitting</b> .....	23
1.5.1.	Over-fitting prevention .....	24
1.5.1.1.	<i>Network-reduction</i> .....	24
1.5.1.2.	<i>Data-augmentation</i> .....	24
1.5.1.3.	<i>Callbacks</i> .....	25
1.5.1.3.1.	Checkpoints .....	25
1.5.1.3.2.	Early-stopping.....	25
1.5.1.4.	<i>Regularization</i> .....	25
1.6.	<b>Conclusion</b> .....	26
<b>Chapter 02: State of the Art</b> .....		<b>27</b>
2.1.	<b>Introduction</b> .....	28
2.2.	<b>Related works</b> .....	28
2.2.1.	A new modality for quantitative evaluation of Parkinson's disease: In-air movement (2013).....	28
2.2.2.	Prediction potential of different handwriting tasks for diagnosis of Parkinson's (2013).	29
2.2.3.	Analysis of in-air movement in handwriting: A novel marker for Parkinson's disease (2014) .....	29
2.2.4.	Decision Support Framework for Parkinson's Disease Based on Novel Handwriting Markers (2014).....	30
2.2.5.	Contribution of different handwriting modalities to differential diagnosis of Parkinson's disease (2015) .....	30
2.2.6.	Evaluation of handwriting kinematics and pressure for differential diagnosis of Parkinson's disease (2016) .....	31
2.2.7.	A new computer vision-based approach to aid the diagnosis of Parkinson's disease (2016).....	31
2.2.8.	Deep Learning-Aided Parkinson's Disease Diagnosis from Handwritten Dynamics (2016).....	32

2.2.9. Feature selection for an improved Parkinson's disease identification based on handwriting (2017).....	32
2.2.10. Reliable Parkinson’s Disease Detection by Analyzing Handwritten Drawings: Construction of an Unbiased Cascaded Learning System based on Feature Selection and Adaptive Boosting Model (2019) .....	33
2.2.11. Dynamically enhanced static handwriting representation for Parkinson’s disease detection (2019) .....	33
2.2.12. Detection of Parkinson’s disease from handwriting using deep learning: a comparative study (2020) .....	34
2.2.13. Refining Parkinson’s Neurological Disorder Identification Through Deep Transfer Learning (2020) .....	35
2.2.14. Parkinson’s disease diagnosis using convolutional neural networks and figure-copying tasks (2022).....	35
2.2.15. Multiple-Fine-Tuned Convolutional Neural Networks for Parkinson’s Disease Diagnosis from Offline Handwriting (2022) .....	36
2.3. State of the art Summary .....	37
2.4. Conclusion .....	38
<b>Chapter 03: Our approach.....</b>	<b>39</b>
3.1. Introduction.....	40
3.2. Training method.....	40
3.3. Data sets used.....	41
3.3.1. HandPD.....	41
3.3.2. NewHandPD .....	42
3.3.3. Parkinson’s Drawings .....	42
3.4. Data treatment.....	44
3.4.1. Image pre-processing .....	44
3.4.2. Data augmentations.....	45
3.5. VGG16.....	47
3.6. Proposed CNN Architecture (Park-Net).....	48



3.7.	Park-Net with SVM .....	49
3.8.	Confusion matrix .....	50
3.9.	Conclusion .....	52
<b>Chapter 04: Experimental Study.....</b>		<b>53</b>
1.1	Introduction.....	54
4.1.	Execution environment .....	54
4.1.1.	Hardware.....	54
4.1.2.	Software .....	54
4.2.	Data splits.....	55
4.3.	Experimental results .....	56
4.3.1.	VGG16.....	56
4.3.2.	Park-Net .....	57
4.3.3.	Park-Net + SVM .....	59
4.3.4.	Additional tests .....	60
4.4.	Comparisons .....	65
4.5.	Conclusion .....	67
<b>General Conclusion.....</b>		<b>68</b>
<b>Bibliography .....</b>		<b>70</b>
<b>Table of figures.....</b>		<b>75</b>

# General Introduction

---

Neurodegenerative Disorders encompass a wide range of conditions that result from progressive damage to cells and nervous system connections that are essential for mobility, coordination, strength, sensation, and cognition. And although millions are affected, we are yet to fully grasp the complexity of most of these neurological diseases. Some of these illnesses include: Alzheimer's disease, Amyotrophic lateral sclerosis, and Parkinson's disease [1]. Only the latter of which, Parkinson's disease, will be the main focus of this master's thesis.

With modern-day rough estimates ranging between 10 and 800 per 100000 [2], Parkinson's disease (PD) is the second most common neurodegenerative disease after Alzheimer's disease. Nonetheless, the precise pathogenetic mechanisms underlying the selective cell loss in PD are yet to be fully understood. Thus, dependable and easily applicable diagnostic checks or markers for PD aren't yet available [3]. With that said, research shows that PD particularly, and in fact greatly, affects motor skills, more precisely, handwriting [4]. This gave researchers the inspiration to establish noninvasive systems of quantifying motor functions derived from the analysis of the handwriting of PD patients, such as the Spiral analysis [5].

This is where Machine Learning (ML) comes in. Modern health care has, over the years, greatly benefited from the progress in theory and practice of health information systems. These health informatics (alias Health IT, or HIT) have been shown to have the potential for positive impact on the quality and efficiency of patient care [6].

The aim of this work is to propose a Convolutional Neural Network (CNN) capable of taking in a PD patients' handwritings and/or drawings and form a correct prediction (positive or negative) with high accuracy and minimal loss. The main contributions of this work are as follows:

- Using a proposed CNN architecture (Park-Net) trained from scratch on PD handwriting datasets.
- Using the linear function in combination with the hinge loss function in order to emulate Support Vector Machine (SVM) classifier to better classify PD handwriting features.
- A study on the state-of-the-art of the detection of PD through handwriting analysis.

With that said, chapters in this thesis will be organized as such:

In Chapter01: in this first chapter, we present the theoretical concepts and abstract ideas relevant to this work as well as provide general definitions.

In Chapter02: in this second chapter, we detail previous works done on specifically the detection of PD through the analysis of handwriting, what datasets they used, what methods they've employed, and what results they've achieved.

In Chapter03: in this third chapter, we will provide details about the different datasets used in this work, and the pre-treatments that were applied on said data. We will also present the trained CNNs architectures; VGG16 with ImageNet weights, our proposed CNN (Park-net), and Park-net with SVM.

In Chapter04: in this final chapter, we showcase the obtained results from the employed training methods and classification techniques along with insightful observation and justification. We end this chapter with a comparison to state-of-the-art.

We conclude with a reminder of the importance of this study and many others like it in detecting PD and why this subject is not receiving the attention that it deserves. We will also present a short recapitulation of what we carried out in this work, as well as ideas that could improve on it which, due to time constraints, we were not able to test.

# **Chapter 01: Theoretical Concepts**

---

## 1.1. Introduction

In this chapter, we provide a brief introduction to the ideas and methods relevant to this work. In order to build a neural network capable of providing correct predictions and achieving a high accuracy, it is natural that we first get a proper grasp on Parkinson's disease, its symptoms, and what is to be expected from someone afflicted with this illness. Then we move on to define handwriting analysis and its relevance to the diagnosis of PD and the already existing datasets related to this subject, as well as the different techniques employed in processing said data. Next step is to understand convolutional networks, certain concepts relating to them, in addition to the pre-existing models and how they can be put to use through the idea of transfer learning. Either for making predictions or to extract features that are then fed to a classifier. Finally, we move on to introducing the issue of over-fitting and the methods that can be employed to mitigate the effects of this problem.

## 1.2. Neurodegenerative diseases

Neurodegenerative disease is an umbrella term that encompasses a wide array of incurable and debilitating conditions which primarily affect the human brain, causing an irreversible progressive deterioration of the neurons. These illnesses are classified into two categories; Ataxias which affect movement, and Dementias which affect cognitive functioning [7]. Examples of neurodegenerative disorders include:

### 1.2.1. Alzheimer's disease

Alzheimer's disease (AD) is the number one most common neurodegenerative disease and the leading cause of dementia worldwide, it makes up 60% –80% of all dementia cases, affecting an estimated 24 million people globally with varying ages although it is statistically more prevalent with older women. This pattern of memory loss reflects a dysfunction of mesial temporal structures and manifests in numerous ways: misplacing objects, repeating conversations or questions, or difficulty keeping track of dates and appointments, etc. In the most advanced cases, AD can lead to the loss of bodily functions and ultimately the death of the individual and while the speed of this progression can vary; life expectancy following a proper diagnosis is typically three to nine years [8]. During handwriting, AD patients were

observed to often repeat separate syllables many times and completely omits others. Different studies highlighted a pattern of lexical agraphia marked by an increased error rate on irregular words or orthographically ambiguous [9].

### **1.2.2. Parkinson's disease**

Parkinson's disease is a progressive neurological disorder that causes unintended or uncontrollable movements, such as shaking, stiffness, and difficulty with balance and coordination. Symptoms usually begin gradually and worsen over time. As the disease progresses, people may have difficulty walking and talking. They may also have mental and behavioral changes, sleep problems, depression, memory difficulties, and fatigue. Therefore, it would be expected that this degree of motor discoordination would have devastating effects on handwriting, which is a skill that requires highly refined movements, both sequential (strokes) and simultaneous (fingers, wrist and arm) components. This impairment is known as Micrographia. This motor loss is characterized by a progressive decrease in letter size, fluctuating changes in writing baseline, and slowness [4].

The progression of PD is customarily divided into three stages, referred to as SL for Severity Level; SL-1, SL-2, and SL-3. These are assessed using two rating scales: The Unified Parkinson's Disease Rating Scale (UPDRS) and the Hoehn and Yahr (H-Y) scale [10]. In the initial stage, PD symptoms typically affect only one side of the body, and as it progresses to both sides of the body, in the second stage, individuals afflicted with this condition start losing their ability to perform basic tasks without assistance. In the third stage of PD, movement is affected. In the last two stages, PD patients are unable to perform daily activities without assistance [4].

### **1.2.3. Amyotrophic lateral sclerosis**

This neurodegenerative disease is characterized by progressive muscular paralysis reflecting a deterioration of motor neurons in the primary motor cortex, corticospinal tracts, brainstem and spinal cord. The term "Amyotrophy" refers to the atrophy of the denervated muscle fibers as their corresponding motor neurons degenerate, thus rendering them weak and often unfit for daily tasks. "Lateral sclerosis" on the other hand, refers to hardening of the anterior and lateral corticospinal tracts, this happens when the deteriorated motor neurons

found in those are replaced by gliosis. ALS has a reported average Incidence of 1.89 per 100,000/year and an average prevalence of 5.2 per 100,000 in western countries in 1990's [11].

Out of the three mentioned neural disorders, Parkinson's disease will be the main focus of this research. This is due to two reasons; the first reason is the time constraint as we couldn't cover all of them, the second reason is that PD is known to have a substantial effect on the handwriting skill and, over the years, multiple drawing tasks and handwriting tests were conceptualized and tested and, for us, this translates to dataset availability.

## **1.3. Handwriting recognition**

Handwriting recognition is the task of transforming a language represented in its spatial form of graphical marks into its symbolic representation. For languages that use the Latin alphabet, this symbolic representation is typically the 8-bit ASCII code. This field is generally divided into two sub-fields in relation to the input method used; off-line or on-line, and while in recent years, with the ever-growing popularity of hand-held devices, real-time on-line handwriting recognition has become significantly more popular, both methods still have their respective use-cases [12].

### **1.3.1. Dataset types**

Several research studies have made use of the quick emergence of digital technologies to analyze the writing disorder in PD patients. Two fundamental technological approaches are commonly used in literature [12], these two being:

### **1.3.2. Online**

Online datasets involve a special type of electronic pen (stylus) and/or tablets during their acquisition, these tools will record not just the candidate's handwriting but also kinematic and spatiotemporal parameters related to the interaction between the stylus and the

surface of said tablet. The advantage of online datasets is that a lot more information is recorded on a patient therefore the recognition rates reported are much higher for the online case in comparison with the offline case. The disadvantage is that special equipment is required [12]. Examples of PD online datasets are: PaHaW, and NewHandPD.

### 1.3.3. Off-line

Offline handwriting recognition, often referred to as optical character recognition (OCR), is performed after the writing is completed, the handwritten document is then captured and processed as an image. The advantage of OCR and offline datasets is that it can be performed on any written document while not requiring any form of special equipment during the capture process, the disadvantage of Offline handwriting recognition systems is that they are often less accurate than online systems; this is due to the fact that OCR are restricted to image data and nothing else [13]. Examples of PD offline datasets would be: HandPD, NewHandPD, Parkinson's drawings.

Note that certain online datasets can be utilized as offline datasets by using the handwritings available in image format and disregarding any additional metadata.

## 1.4. Convolutional neural networks

Convolutional neural networks (CNNs) are comparable to traditional ANNs (Artificial Neural Networks) in that they are comprised of neurons that self-optimize through learning. From the input raw image vectors to the final output of the class score, the entire network will express a single perceptive score function; the weight. The last layer will contain a loss function relative to the number of classes, and all of the regular tips and tricks developed for traditional ANNs still apply. The over-all architecture of a CNN is comprised of convolutional layers, pooling layers, a flatten layer, and fully-connected layers [14]. as shown in the following Figure:



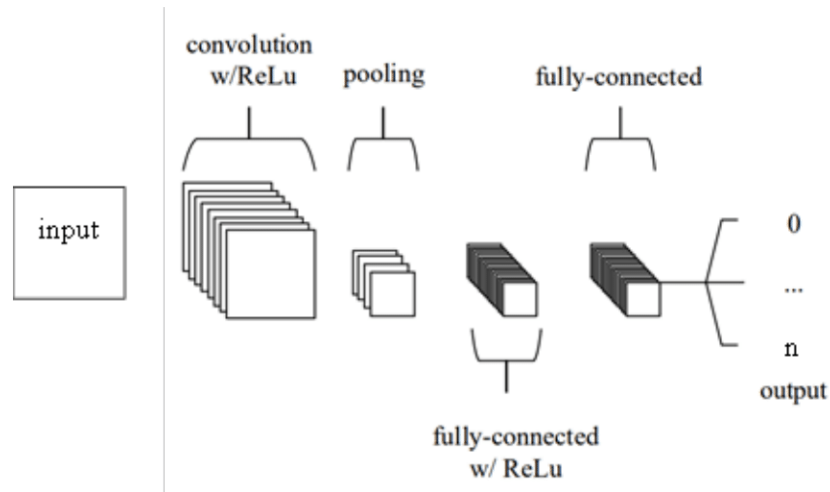


Figure 1: Basic 5 player CNN architecture outline [50]

### 1.4.1. CNN Layers

There are multiple types of CNN layers. However, for the sake of brevity we will only consider the ones that are relevant to this thesis:

#### 1.4.1.1. Convolutional

Determines the output of neurons of which are connected to local regions of the input through the calculation of the scalar product between their weights and the region connected to the input volume. ReLu (rectified linear unit) is commonly used as an activation function. The layers parameters focus around the use of learnable kernels. These kernels are usually small in spatial dimensionality, but spreads along the entirety of the depth of the input. The layer convolves each filter across the spatial dimensionality of the input to produce a 2D activation map [14].

#### 1.4.1.2. Pooling Layer

This layer serves to down sample the results of the previous layer along the spatial dimensionality of the given input, further reducing the number of parameters within that activation. This step can be set as “max pooling” or “average pooling” [14], the difference between these is illustrated in the following figures:



Figure 2: "max pooling" example



Figure 3: "average pooling" example

#### 1.4.1.3. **Flatten layer**

converting the data into a 1-dimensional array for inputting it to the next layer. We flatten the output of the convolutional layers to create a single long feature vector. And it is connected to the final classification model, which is called a fully-connected layer.

#### 1.4.1.4. **Fully-connected layers**

The fully-connected layer, also known as a dense layer, contains neurons which are directly connected to the neurons in the two adjacent layers but not to neurons within the same layer. The objective of this layer is to produce class scores from the activations, to be used for the classification.

Further concepts relating to the CNN training process [15].

### 1.4.2. **Epoch**

One Epoch is accounted for when the entirety of the training dataset is passed forward and backward through the neural network only once and an opportunity to update the internal model parameters. An epoch is comprised of one or more batches.

### 1.4.3. Batch size

Batch size is the total number of training examples present in a single batch. This parameter can be manually set before training, and according to the batch size, the denomination of the name of the learning algorithm may vary:

- **Batch Gradient Descent:** Batch Size = Size of Training Set
- **Stochastic Gradient Descent:** Batch Size = 1
- **Mini-Batch Gradient Descent:**  $1 < \text{Batch Size} < \text{Size of Training Set}$

With the most common batch size for the Mini-batch Gradient Descent being: 32, 64, and 128.

### 1.4.4. Iteration

Iterations, simply put, is the number of batches needed to complete one epoch.

## 1.4.5. Examples of existing architectures

For the sake of this brevity, we will only be looking at the three instances that we considered for the context of this work:

#### 1.4.5.1. *Vgg*

VGG Net is the name of a pre-trained convolutional neural network (CNN) invented by Simonyan and Zisserman from Visual Geometry Group (VGG) at University of Oxford in 2014. VGG 16 and VGG 19 have 16 and 19 weight layers respectively. VGG Net takes input of  $224 \times 224$  RGB images and passes them through a stack of convolutional layers with the fixed filter size of  $3 \times 3$  and the stride of 1. There are five max pooling filters embedded between convolutional layers in order to down-sample the input representation (image, hidden-layer output matrix, etc.). followed by 3 fully connected layers, having 4096, 4096 and 1000 channels, respectively [16].

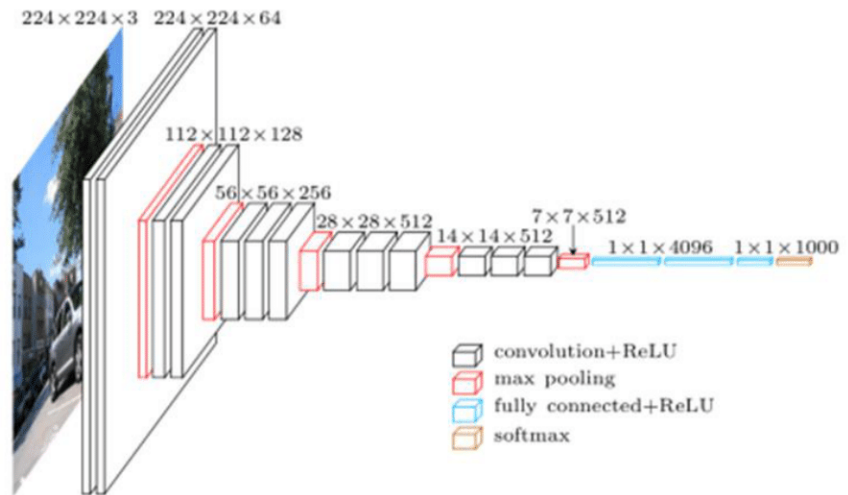


Figure 4: VGG16 architecture [17]

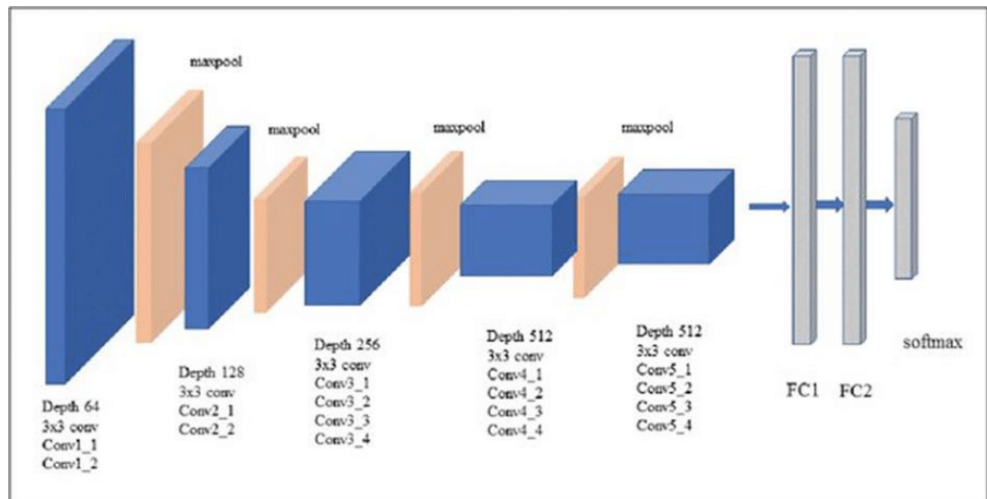


Figure 5: VGG19 architecture [18]

### 1.4.5.2. ResNet

ResNets or Residual Networks, learn residual functions with reference to the layer inputs, instead of learning unreferenced functions. Instead of hoping each few stacked layers directly fit a desired underlying mapping, residual nets let these layers fit a residual mapping. They stack residual blocks on top of each other to form network: e.g. a ResNet-50 has fifty layers using these blocks [19], its architecture is illustrated in the following figure:

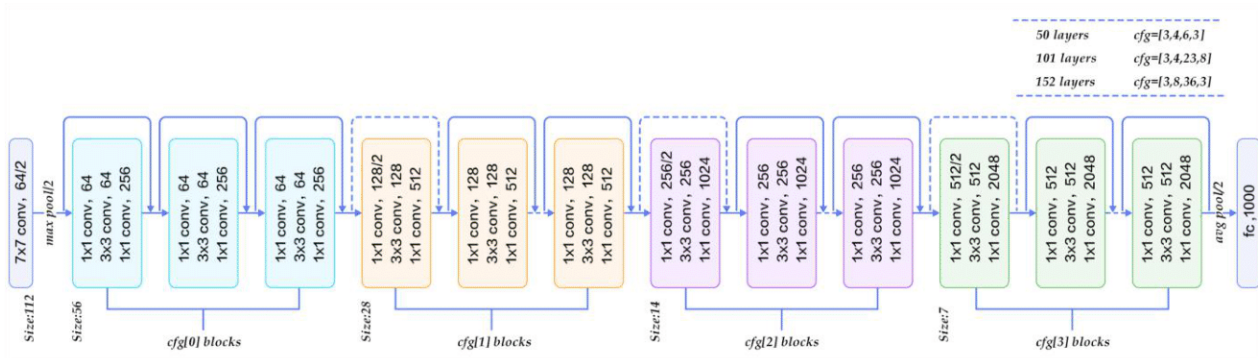
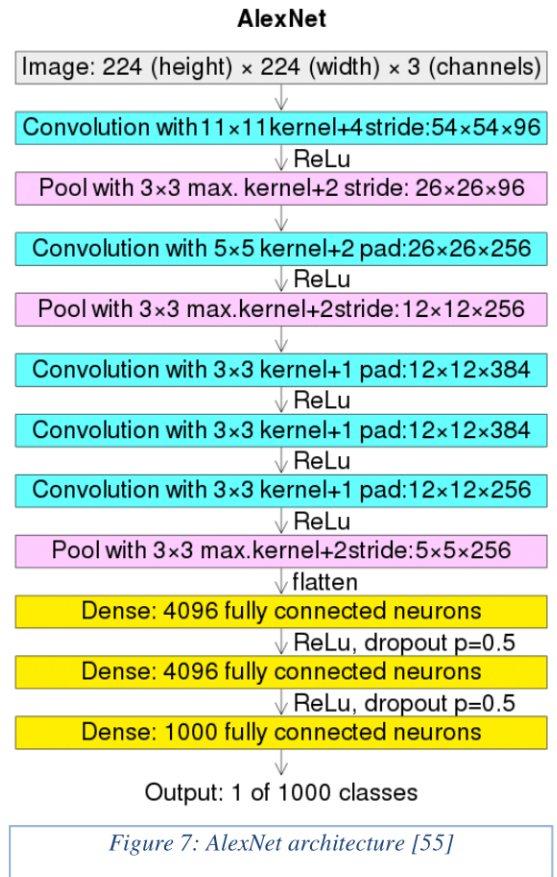


Figure 6: ResNet50 architecture [51]

### 1.4.5.3. AlexNet

This CNN was designed by Alex Krizhevsky in collaboration with Ilya Sutskever and Geoffrey Hinton [20]. This network managed to achieve a top-5 error of 15.3% in the 2012 Large Scale Visual Recognition Challenge. The network achieved a top-5 error of 15.3% beating the competition by over 10.8% [21]. AlexNet is characterized by its 650,000 neurons and 60 million parameters, spread over three blocks with one, one, and three convolutional layers respectively, which is five in total, each block is followed by a max-pooling layer. The network concludes with three fully connected layers with a final 1000-way softmax.



### **1.4.6. CNN training methods**

Within this research, we considered two training approaches; transfer learning, and learning from scratch. Both methods have their advantages and disadvantages.

#### ***1.4.6.1. Training from scratch***

Training from scratch is akin to conventional machine learning training, where the model has only access to the relevant datasets. Weights are initialized with random values then gradually adjusted in correlation with the training data provided.

#### ***1.4.6.2. Transfer learning***

Transfer learning is the act of reusing knowledge from past related tasks in an attempt to leverage the previously gained learning and experience to more efficiently learn the new task. The benefit of this approach, more often than not, is the reduction of the number of training samples needed to achieve a desired performance on a series of related learning tasks. This concept essentially mirrors the way humans learn, seeing that usually just a few training examples of a new idea are sufficient for a someone with prior knowledge on the related concepts to very quickly grasp the new concept. it is common practice to freeze most layers and selectively retrain only certain layers in a process called fine-tuning [22].

Fine-tuning consists of manually selecting individual layers of a pre-trained CNN and further tuning (training) them to better fit a given task and while it is common practice to focus the finetuning on the last layers of the model (dense layers and activation layer), any layers of the network can be finetuned. However, there is no general rule on which layers should be selected in order to reach optimal performance so fine tuning is a process of trial and error [23].

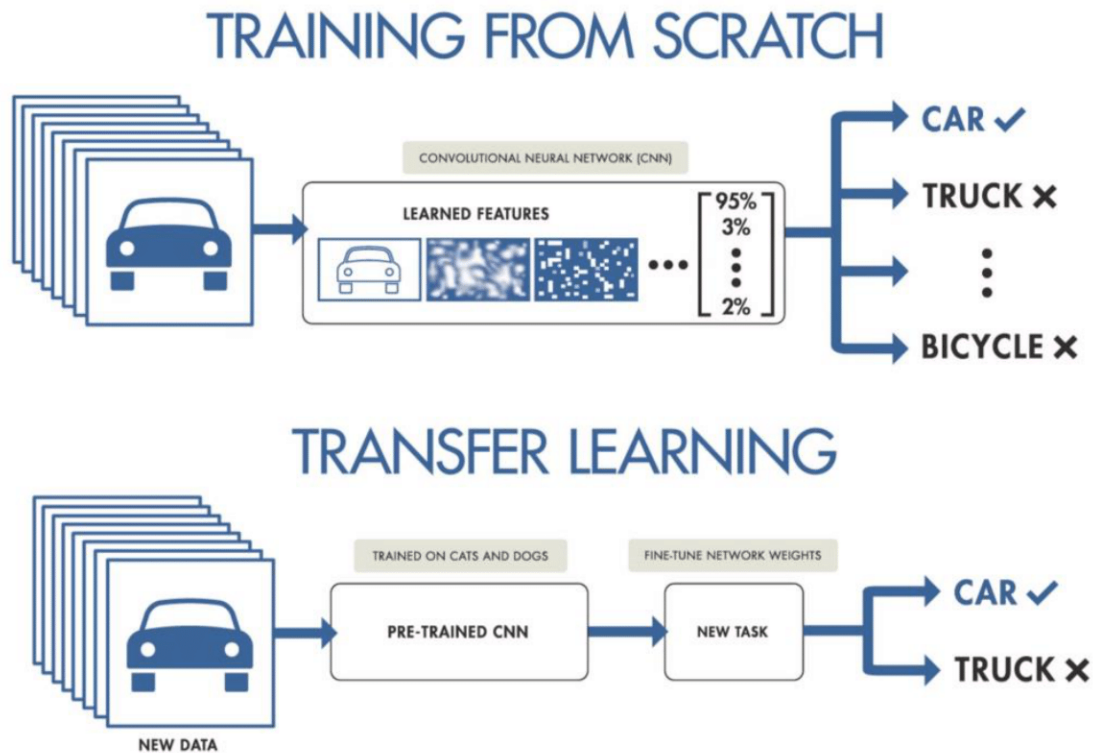


Figure 8: difference between training from scratch and transfer learning [54]

While training a network from scratch is time-consuming and labor-intensive, transfer-learning is not always advantageous, especially when the data set used for the pre-training is drastically different from the new task. Furthermore, training from scratch allows for a better understanding of the network [24].

## 1.5. Over-fitting

Overfitting is a central problem in supervised machine learning (learning from labeled training data). It is observed when a model ends up over-learning, and memorizing undesired aspects of the data thus preventing itself from generalizing the models to well fit not just the training data but also unseen data contained in the validation/testing set. Not to be confused with Under-fitting, a similar problem that occurs when a model is under-trained. Overfitting usually occurs due to; poorly pre-processed data containing too much noise, a small training set, and/or the use of an overly complex model architecture [25].

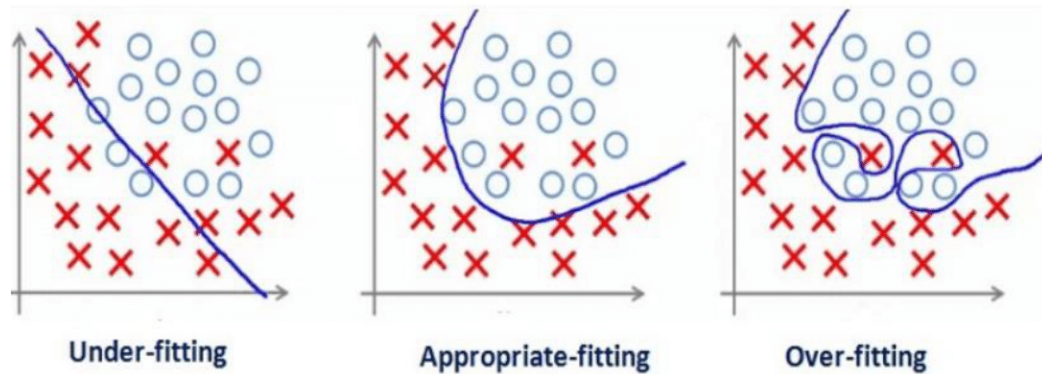


Figure 9: illustrative graphs displaying the different between Under-fitting, Optimal-fitting, and Over-fitting [52]

### 1.5.1. Over-fitting prevention

Over the years, researchers have discovered various strategies to reduce the effects of overfitting:

#### 1.5.1.1. Network-reduction

Network-reduction entails scaling down the classification complexity of the model. And while general machine learning proposes multiple ways of implementing this concept, in the case of CNNs it can be achieved through; varying the complexity of the model by changing the number of adaptive parameters in a process called structural stabilization, or choosing a smaller model architecture whether it's a pre-existing architecture of manually removing layers to reduce the number of neurons [26].

#### 1.5.1.2. Data-augmentation

Data augmentation is a set of techniques used to artificially increase the amount of data by generating new data points from existing data. This includes applying small changes to data or using deep learning models to generate new data points. These methods are often employed as a solution to over-fitting [27]. Further details on the data augmentation methods used in this work are available in “3.4.2 Data augmentations”.



### 1.5.1.3. Callbacks

Callbacks are methods that can perform certain actions at various stages of training, e.g; at the start or end of an epoch. In this dissertation, we consider two types of Callbacks:

#### 1.5.1.3.1. Checkpoints

Model Checkpoints in keras are callbacks used to save a model or weights (in a checkpoint file) at some interval, usual so that the model or weights can be loaded later to continue the training from the state saved. For details about callbacks used in this work, see “3.2 Training method”.

#### 1.5.1.3.2. Early-stopping

In short, this method will automatically stop the training as soon as the training metrics stop improving. Thus, preventing noise-learning. This technique is also very effective at avoiding under-fitting since it halts the training at the most opportune time, when metrics have reached their apparent peak (best value) [28].

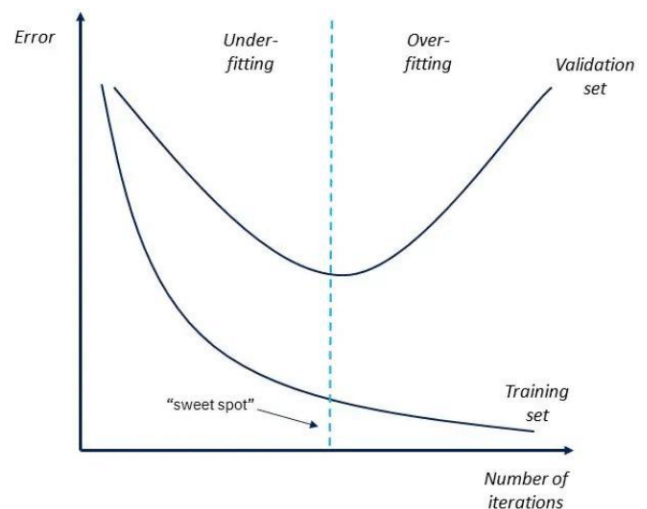


Figure 10: example graph of an Early-stopping in the case of Training-loss vs Validation-loss [53]

### 1.5.1.4. Regularization

The model output is affected by multiple features. When the number of features increases, so does the model complexity. An overfitting model tends to take all the features into consideration, even though some of them have very limited effect on the final output. Some may also be considered noise which are not only meaningless to the output but can even harm the performance. So, in order to limit these cases, there are two solutions; the first one is to select only the useful features and remove the useless features from our model through manual or automatic data pre-processing, the second one is to minimize the weights of the features which have little influence on the final classification [25].

## 1.6. Conclusion

In this chapter we provided a short introduction to Parkinson's disease and the different ideas and methods essential for the comprehension of this work. This included handwriting analysis and its relevance to the diagnosis of PD and the already existing datasets containing PD patients' handwritings, as well as the different techniques employed in processing said data. Following that we had a brief look at convolutional networks concepts in addition to a few examples of pre-existing models and how they can be put to use through the idea of transfer learning. Finally, we talked about the issue of Over-fitting and provided four ways of mitigating this problem; Network-reduction, Data-augmentation, Callbacks, and Regularization. In the next chapter we will discuss the state of the art.

## **Chapter 02: State of the Art**

---

## 2.1. Introduction

In recent times, use of machine learning and deep neural networks in various clinical applications has gained increasing popularity. There has been significant research interest in the creation of automated systems for early-stage detection of PD based on voice, gait, and handwriting data. In this chapter we detail previous works done specifically on the detection of PD through the analysis of handwriting, what datasets they used, what methods they've employed, and what results they've achieved. This section will also serve as an evolutionary history of such as works will be sorted from oldest to most recent.

## 2.2. Related works

### 2.2.1. A new modality for quantitative evaluation of Parkinson's disease: In-air movement (2013)

In this 2013 study [29], In-air trajectory during handwriting was proposed as an efficient PD diagnose. They showed in their experimental results that analysis of in-air trajectories is capable of determining subtle motor abnormalities connected to PD. Furthermore, by analyzing both in-air trajectories and on-surface handwriting they built a predictive model with PD classification accuracy of **80%**. The

methodology consisted of computing 600 handwriting features then selecting a smaller subset of these features using two feature selection algorithms: Mann-Whitney U-test filter and relief algorithm, and map these feature subsets to binary classification response using support vector machines (SVM).

Their data set consisted of 37 Parkinsonian patients (19 men/18 women) and 38 (20 men/18 women) healthy control with matching ages. All participants were right-handed.

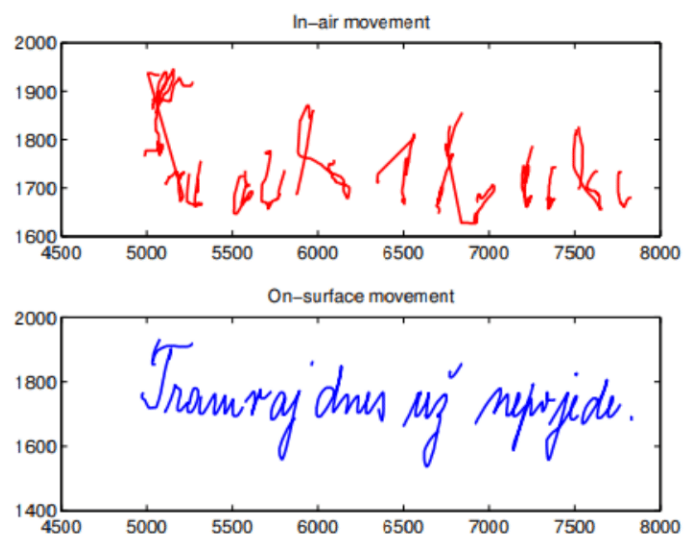


Figure 11: handwriting sample from in-air movements dataset [29]

All subjects were asked to write the same sentence in their native language (Czech); “Tramvaj dnes unepojede” (The tram won’t go today). Handwritten signals were acquired using the Intuos 4M digital tablet, from Wacom technology, in the x-y plane and in the pressure axis.

### 2.2.2. Prediction potential of different handwriting tasks for diagnosis of Parkinson’s (2013)

In this conference proceeding [30], a template for acquiring PD patients’ handwriting during different tasks was proposed. This template consists 8 tasks. first one being an Archimedean spiral. In the next 3 tasks (2, 3, 4), participants wrote cursive letters or bi/tri-grams of letters. The Next 3 tasks are words, that can be written as one long stroke, in this

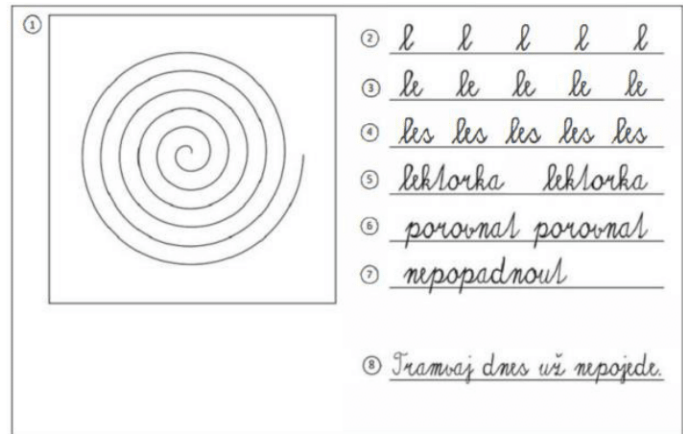


Figure 12: template proposed by "Prediction potential of different handwriting tasks for diagnosis of Parkinson's (2013)" [30]

specific study, they were: “lektorka”, “porovnat”, “nepopadnout”. Three Czech words (native language of participants), which respectively translate to: “lector(female)”, “to compare”, and “do not catch”. Finally, the last task is long sentence, which in this specific study was: “Tramvaj dnes uz nepojede” (The tram won't go today).

Their dataset consisted of 8 different handwriting samples from 75 subjects. The archived results show **80%** overall classification accuracy using the SVM classifier on manually extracted features.

### 2.2.3. Analysis of in-air movement in handwriting: A novel marker for Parkinson’s disease (2014)

This paper [31] is an improved version of the 2013 study from the same authors [29], they made use of the same dataset with 37 PD patients and 38 age- and gender-matched

healthy controls, where they were asked to write the same sentence in Czech. Here also, both on-surface and in-air trajectories data were exploited.

In this particular research, the authors showcased that through the use of; sequential forward feature selection (SFFS), minimum-redundancy-maximum-relevance (mRMR) as feature selection algorithms, and SVM as a classifier, they managed to achieve a classification accuracy of 84% while only considering in-air movements and a 78% accuracy while only considering on-surface data. Furthermore, combining the two led to a prediction accuracy of 85.61%.

#### **2.2.4. Decision Support Framework for Parkinson's Disease Based on Novel Handwriting Markers (2014)**

This is yet another study [32] from the authors of the three previously mentioned works [29], [30], and [31]. In this research, they had two goals in mind. The first one, was to find a subset of handwriting features suitable for identifying subjects with PD and the second one was to build a predictive model for the diagnose of PD. The dataset used here is the same one used in [29] and [31]. Containing handwriting samples from 37 medicated PD patients and 38 age- and sex- matched controls. They extracted handwriting measures, conventional kinematic, and spatio-temporal handwriting measures from each sample in the dataset. they also computed novel handwriting measures based on entropy, signal energy, and empirical mode decomposition of the handwriting signals. These features were then fed to an SVM with radial Gaussian kernel for automated diagnosis.

Their best results showed an accuracy of 88.13%, using SVM with the highest values of sensitivity and specificity equal to 89.47% and 91.89%, respectively.

#### **2.2.5. Contribution of different handwriting modalities to differential diagnosis of Parkinson's disease (2015)**

Still with the authors of the last four studies mentioned in this chapter, Peter et al. and in this particular research [33], they have yet again utilized the same dataset formulated from

the template in Figure 12, first introduced in [30]. To quickly reiterate, this dataset contains 37 PD patients (19 men/18 women) and 38 healthy controls (20 men/18 women).

The goal was to evaluate the contribution of different handwriting modalities for PD diagnosis. In their previous studies, they showed the relevancy of pressure and in-air movement in PD detection. And in this one, they experimented with, at the time, novel characteristics based on entropy and empirical mode decomposition of the handwriting signal.

Their best results showed a performance of 89% below the Receiver operating characteristic curve (AUC) for on-surface data, 74% for in-air data, and 84% for pressure data.

### **2.2.6. Evaluation of handwriting kinematics and pressure for differential diagnosis of Parkinson's disease (2016)**

Within this 2016 article from Artificial Intelligence in Medicine [3], Peter Drotár et al, presented their PaHaW (Parkinson's disease handwriting) dataset. which consists of handwriting samples from 37 PD patients and 38 healthy controls each asked to fill out the template [30] in Figure 12. The tasks included drawing an Archimedean spiral, repetitively writing orthographically simple syllables and words, and writing of a sentence. Both kinematic features and pressure features were extracted. three different classifiers were compared: K-nearest neighbors (K-NN), ensemble AdaBoost classifier, and support vector machines (SVM). SVM was the the best performing model with a classification accuracy of 81.3% (sensitivity = 87.4%, specificity = 80.9%). When evaluated separately, pressure features proved to be relevant for PD diagnosis, yielding an accuracy of 82.5% compared to 75.4% using kinematic features.

### **2.2.7. A new computer vision-based approach to aid the diagnosis of Parkinson's disease (2016)**

This 2016 Computer Methods and Programs in Biomedicine article [34] is associated with the HandPD dataset. the goal was to propose a method of automatically identifying and separating both the template and the drawings using image processing techniques without

user intervention. The focus of this study was mainly image processing. Nonetheless, according to experimental results from the recognition tests done, the best one was achieved by the SVM classifier with a **66.72%** on the Meander task.

### **2.2.8. Deep Learning-Aided Parkinson's Disease Diagnosis from Handwritten Dynamics (2016)**

This IEEE 2016 29th SIBGRAPI conference proceeding [35] is associated with the NewHandPD dataset. Their methodology for PD detection consisted of using drawings and signals converted into images, these signals were acquired by means of a smart pen composed of a series of sensors used during the creation of the NewHandPD dataset. They've achieved an accuracy of **84%** using the graph-based pattern recognition technique; OPF (Optimum-Path Forest).

### **2.2.9. Feature selection for an improved Parkinson's disease identification based on handwriting (2017)**

This Study [36] was performed for the purpose of finding a subset of handwriting features suitable for consistently identifying PD subjects. They made use of the PDMultiMC database collected in Lebanon. This dataset is composed of 16 PD patients (12 male/ 4 female) and 16 age-matched healthy controls (5 male/ 11 female), so a total of 32 subjects (17 male/ 15 female). Each individual was tasked with copying specific handwritten patterns, copying words in Arabic, and writing full names. The Wacom Intuos 5 tablet was used to collect spatial displacement (x, y positions), pen pressure, time stamp, pen status, and pen-tip angle (altitude, azimuth) measurements. These tasks were then studied and analyzed, and the following data was extracted: kinematic, spatio-temporal, pressure, energy, entropy, and intrinsic features.

The selection of features was done in two steps; the first stage selected a subset using statistical analysis, while the second step selected the most relevant features of said subset by a suboptimal approach. The determined characteristics were then fed to an SVM classifier



with RBF kernel, as an attempt to identify the PD subjects. The reported accuracy was 96.875% (sensitivity = 93.75 %, specificity = 100%).

### **2.2.10. Reliable Parkinson’s Disease Detection by Analyzing Handwritten Drawings: Construction of an Unbiased Cascaded Learning System based on Feature Selection and Adaptive Boosting Model (2019)**

In order to improve the PD detection accuracy, this 2019 study [37] proposes a cascaded learning system that cascades a Chi2 model with adaptive boosting (Adaboost) model. This model ranks and selects a subset of relevant features from the feature space, an Adaboost model is then used to classify the given sample in accordance with the subset of selected characteristics.

Their results showed a classification accuracy of  $ACC_{bal}=76.44\%$ , sensitivity of 70.94% and specificity of 81.94%, using an under-sampling of the meander task on the HandPD dataset as a proposed way to solve the biasedness caused by imbalances in both the training and validation data. These results were achieved with the proposed cascaded system (Chi2, Adaboost).

### **2.2.11. Dynamically enhanced static handwriting representation for Parkinson’s disease detection (2019)**

In this paper [38], the authors study the effectiveness of “dynamically enhanced” static images of handwriting in the detection and diagnosis of PD. They proposed a static representation that embeds dynamic information based on drawing the points of the samples, instead of linking them, so as to retain temporal/velocity information; in addition to adding pen-ups for the same purpose.

In order to evaluate the effectiveness of this approach, they conducted tests on the PaHaW dataset. The classification workflow involved transfer learning to extract meaningful

features from multiple representations of the sample data. Chained to an ensemble of different classifiers to achieve the final predictions.

Their best results included 86.67% accuracy with a sensitivity of 89.17% and a specificity of 80.83% using SVM with the ensemble of the best five with both static and dynamic characteristics.

### 2.2.12. Detection of Parkinson's disease from handwriting using deep learning: a comparative study (2020)

This study [39] served to introduce the HandPDMultiMC dataset, a sub-set of the PDMultiMC used in [36], which includes handwriting samples from 42 subjects (21 PD and 21 controls). Among the experiments carried within the context of this research are various Deep learning architectures, namely the CNN and the CNN-BLSTM, for the detection of PD through time series classification. Spectrograms was applied to encode pen-based signals into images that were then fed into the CNN model, while the raw time series are directly used in the CNN-BLSTM. In conjunction to this, multiple data augmentation approaches for pen-based signals were proposed.

Their best results (97.62% accuracy) were achieved through a combination of CNN-BLSTM models trained with Jittering and Synthetic data augmentation approaches.

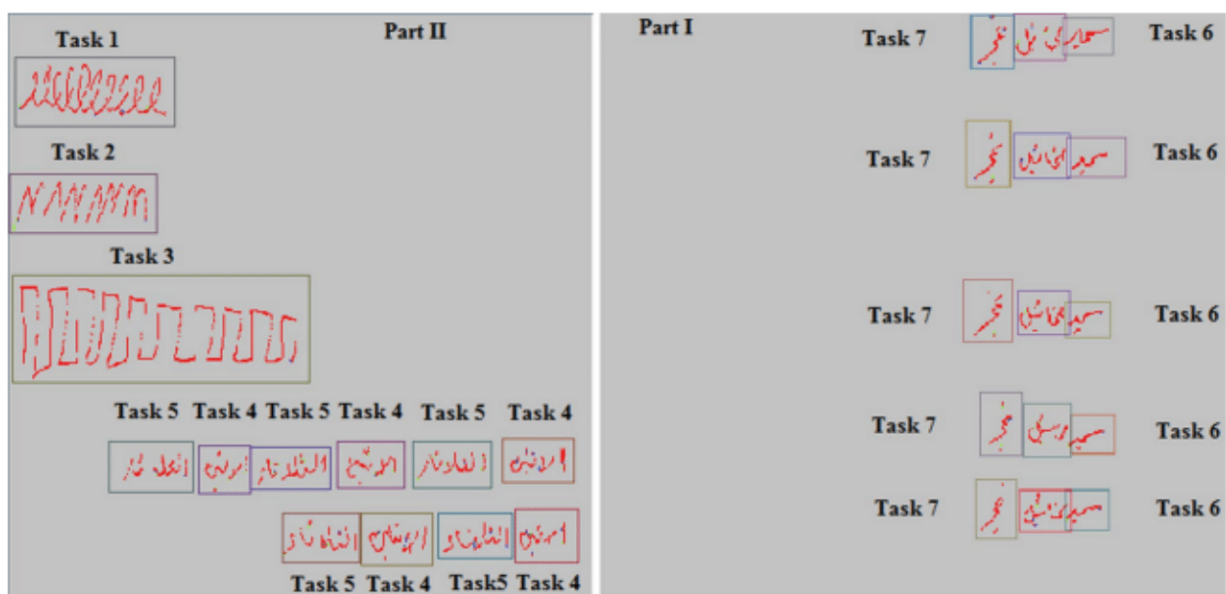


Figure 13: HandPDMultiMC dataset sample, the seven tasks were segmented from a sheet filled by a PD subject [39]

### 2.2.13. Refining Parkinson’s Neurological Disorder Identification Through Deep Transfer Learning (2020)

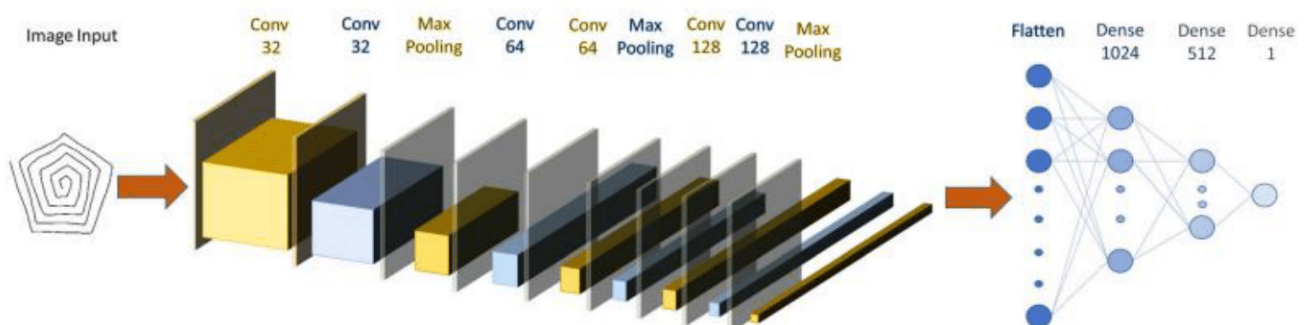
In this paper [40], a deep convolutional neural network classifier with transfer learning and data augmentation techniques was proposed to improve the identification of PD through the use of handwriting images. Both freeze and fine-tuning of transfer learning were investigated during the performed trainings, using ImageNet and MNIST dataset as sources for the pretrained weights, and the PD dataset used for feature extraction is the PaHaW dataset [3].

Their results showed an impressive 98.28% accuracy using fine-tuned AlexNet based approach with ImageNet weight. However, it is worth noting that the researchers behind this work, used augmented versions of the same images for training and also for testing. This is considered bad practice; as it creates a high positive bias and therefore these results should not be generalized.

### 2.2.14. Parkinson’s disease diagnosis using convolutional neural networks and figure-copying tasks (2022)

The aim of this work [41] was to demonstrate the effectiveness of each of the wire cube and spiral pentagon hand drawing tasks’ when it comes to PD discrimination. This was approached through both pre-existing CNNs and their proposed CNN model.

This proposed model was composed of three blocks, each with two convolutional layers with equal filter size (32, 64, 128) and a max pooling layer. These blocks are then followed by 2 convolutional layers with a size of 1024 and 512 respectively, and finally a prediction layer with a size of 1.



In terms of image size, they experimented with three different resolutions; 32x32, 64x64, and 128x128.

The dataset was collected using a digital Wacom tablet at the Leeds Teaching Hospitals NHS and it contains 87 subjects (58 patients and 29 aged-matched healthy controls). All individuals were tasked with copying the wire cube from a sample image and drawing the Archimedean spiral pentagon on top of a template image.

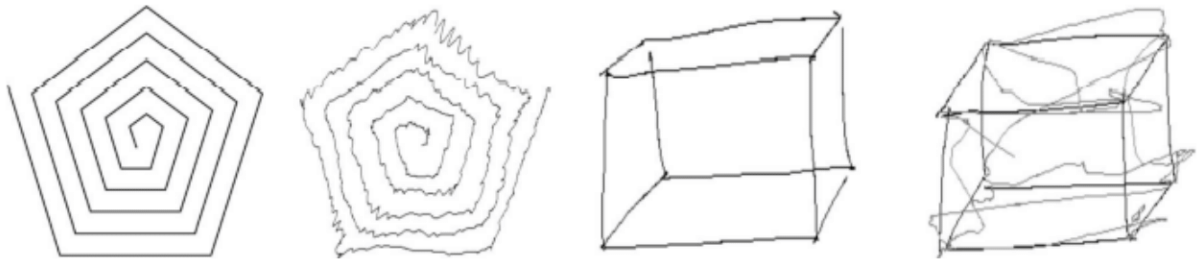


Figure 15: Mohamad et al, pentagon/cube data set samples [41]

Their performance result being 93.53% using the proposed CNN architecture on the pentagon task with augmented data at a resolution of 32x32.

### 2.2.15. Multiple-Fine-Tuned Convolutional Neural Networks for Parkinson's Disease Diagnosis from Offline Handwriting (2022)

In this study [42], M. Gazda, M. Hireš, and P. Drotár present an approach in which end-to-end processing by a convolutional neural network (CNN) is utilized to diagnose PD exclusively from offline handwriting images which eliminates any need for specialized devices or feature engineering. Their proposed CNN architecture (E-CNN), which is based on a multiple-fine-tuned CNNs approach, achieved **94.7%** on NewHandPD meander task with an 80% train to 20% validation split.

## 2.3. State of the art Summary

For illustration purposes, the following figure briefly summarizes all the works mentioned in this chapter:

	<b>Study</b>	<b>Year</b>	<b>Approach</b>	<b>Dataset</b>	<b>Data type</b>	<b>Accuracy%</b>
1	Peter et al. [29]	2013	SVM	PaHaW	Online	80%
2	Peter et al. [30]	2013	SVM	PaHaW	Online	80%
3	Peter et al. [31]	2014	SVM	PaHaW	Online	85.61%
4	Peter et al. [32]	2014	SVM	PaHaW	Online	88.13%
5	Peter et al. [33]	2015	AUC	PaHaW	Online	89%
6	Peter et al. [3]	2016	SVM	PaHaW	Online	81.30%
7	Pereira et al. [34]	2016	SVM	HandPD	Offline	66.72%
8	Pereira et al. [35]	2016	OPF	NewHandPD	Online + Offline	84.00%
9	Taleb et al. [36]	2017	SVM	PDMultiMC	Online + Offline	96.87%
10	Ali et al. [37]	2019	Chi2+Adaboost	HandPD	Offline	76.44%
11	Diaz et al. [38]	2019	SVM	PaHaW	Online	86.67%
12	Taleb et al. [39]	2020	CNN-BLSTM	HandPDMultiMC	Online + Offline	97.62
13	Naseer et al. [40]	2020	AlexNet	PaHaW	Online	98.28%
14	Mohamad et al. [41]	2020	proposed CNN	NHS dataset	Offline	93.53%
15	Matej et al. [42]	2022	E-CNN	NewHandPD	Offline	94.70%

Figure 16: State of the art summary table

## 2.4. Conclusion

In this chapter, we looked at different state of the art studies ranging from 2013 up to the year of writing this dissertation (2022), what datasets they used, what methods they've employed, and what results they've achieved. This section also served to show the evolutionary history of PD detection systems through the automatic analysis of handwriting, we concluded by compiling every paper cited in this chapter into a single summary table for illustration purposes. In the next chapter, we will be explaining our own approach, from the training method, to the datasets used, and the way we preprocessed and augmented said data, finally we will present the models used.

## **Chapter 03: Our approach**

---

## 3.1. Introduction

In this chapter, we will detail the different approaches used in this work, from data processing to the training steps and our proposed CNN architectures; a mildly finetuned VGG16 model with pre-loaded ImageNet weights, Park-net, and Park-net + SVM. Furthermore, we will provide information's about the datasets used as well as the data augmentation techniques that we've attempted. Finally, we will explain what a confusion matrix is, and how we will use it in our results acquisition.

## 3.2. Training method

An 80% to 20% split for training set to validation set respectively was used for all trainings performed, furthermore all models were given a maximum of 100 epochs with a fixed batch size of 32 as well as 4 callbacks were applied (3 checkpoints and 1 earlystop). For an explanation of these concepts, see Chapter01.

All three checkpoint had the `save_best_only` attribute set to true, and `save_weights_only` set to false, and these checkpoints were:

- **Accuracy:** Accuracy refers to the accuracy score achieved by the model at a given epoch on the training dataset. Our goal is to get the Maximum accuracy.
- **Validation accuracy:** Validation accuracy refers to the accuracy score achieved by the model at a given epoch on the validation data set. Just like with the Accuracy checkpoint, we also want the Maximum possible Validation accuracy.
- **Validation loss:** Validation loss refers to the error rate attained by the model at a given epoch on the validation data set. Unlike the two previous checkpoints, in this case we are looking for the Minimum Validation loss.

The one early-stop that we used was for monitoring Accuracy with a patience of 5; meaning that if the accuracy of the model during the training doesn't improve for 5 epochs, the training is automatically stopped.



### 3.3. Data sets used

In this work we made use of 3 widely available PD handwriting datasets: HandPD [34], NewHandPD [43], and Parkinson's drawing [10]. While we did not use them in our trainings, NewHandPD does include signal recordings making it valid as both an online and an offline data set.

#### 3.3.1. HandPD

The HandPD dataset [34] is comprised of handwritten exams from both healthy individuals (18) and PD patients (74). The task that was given to these 92 individuals was to fill out a form composed by four spirals and four meanders, which were then carefully cropped out from the form and stored in "jpg" format.

- **Healthy Group:** 6 male and 12 female individuals with ages ranging from 19 to 79 years old (average age of  $44.22 \pm 16.53$  years). Among those individuals, 2 are left-handed and 16 are right-handed.

The image shows a handwritten form for a test. At the top, the date '22/11/2010' is written. Below it, there are fields for 'No:' and 'RG:' with handwritten entries. The text 'Distonia e Hiperinesia Medicamentosa' is written in the 'RG:' field. To the right, there is a logo for 'Field study: Unesp 2010' and text for 'University of Applied Sciences Regensburg Biometric Smart Pen Project' and 'Universidade Estadual Paulista Faculdade de Medicina (FMB), Botucatu'. Below this, the age 'Idade: 56' and dominant hand 'Mão dominante: (X) direita ( ) esquerda' are noted. The form contains several drawing tasks:
 

- Task 'b': 'Desenhar círculo 12 vezes no mesmo lugar sem parar.' and 'Desenhar círculo no ar 12 vezes no mesmo lugar sem parar.' with a drawing of a circle and a small circle.
- Task 'c': 'Desenhar espiral após sinal sonoro, de dentro para fora.' with four spiral drawings.
- Task 'd': 'Desenhar meander após sinal sonoro, de dentro para fora.' with four meander drawings.
- Task 'e': 'Diadococinesia: Mão direita 20 segundos.'
- Task 'f': 'Diadococinesia: Mão esquerda 20 segundos.'

 There are speaker icons next to tasks b, c, d, e, and f, indicating audio signals.

Figure 17: HandPD test template [34]

- **Patient Group:** 59 male and 15 female individuals with ages ranging from 38 to 78 years old (average age of  $58.75 \pm 7.51$  years). Among those individuals, 5 are left-handed and 69 are right-handed.

All tests were performed at the Botucatu Medical School, São Paulo State University - Brazil.

### 3.3.2. NewHandPD

The NewHandPD dataset [43] is an improved version of the HandPD dataset. This time composed of 66 individuals, 31 of which are PD patients while the rest (36) are healthy. Each one was asked to complete 12 exams, 4 of them being related to spirals, 4 related to meanders, and 2 circular movements (one circle in the air and another on the paper), and left and right-handed diadochokinesis. During the exam, handwritten dynamics were recorded by means of a smart pen (BiSP), making NewHandPD a valuable option even for Online datasets studies. It is fair to note that NewHandPD is more balanced than the original HandPD dataset:

- **Healthy Group:** 18 male and 17 female individuals with ages ranging from 14 to 79 years old (average age of  $44.05 \pm 14.88$  years). Among those individuals, 5 are left-handed and 30 are right-handed.
- **Patient Group:** 21 male and 10 female individuals with ages ranging from 38 to 78 years old (average age of  $57.83 \pm 7.85$  years). Among those individuals, 2 are left-handed and 29 are right-handed.

### 3.3.3. Parkinson's Drawings

The Parkinson's drawings dataset [10] is the result of a survey that was approved and conducted RMIT University Human Research Ethics Committee and in accordance with Declaration of Helsinki (revised 2004). A total of Fifty-five (55) age-matched, 28 control group (CG) and 27 PD patients. All PD patients were recruited from PD outpatient clinic at Dandenong Neurology, Melbourne, Australia, while the Control Group (CG) subjects were from multiple aged-care facilities. All subjects were right hand dominant. The CG subjects were recruited to approximately match the age distribution and gender of the PD patients.

the following figure briefly details the contents of the three data sets used with image previews included:

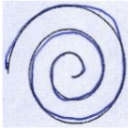

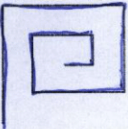
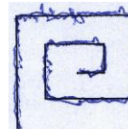

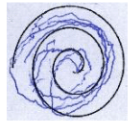
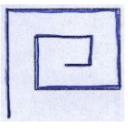

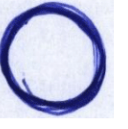



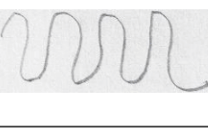
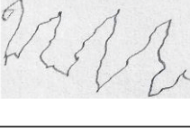
Data Set	Data Types	Healthy (Negative)	Parkinson (Positive)	Total Sample Size
HandPD [34]	Spirals			368
		72	296	
	Meanders			368
		72	296	
NewHandPD [43]	Spirals			263
		139	124	
	Meanders			263
		139	124	
	Circles			73
		35	38	
Parkinson's Drawings [10]	Spirals			102
		51	51	
	Waves			102
		51	51	

Table 1: Summary table of Data sets used

## 3.4. Data treatment

### 3.4.1. Image pre-processing

Image pre-processing refers to the steps taken to clean raw image data from any false, missing, or incomplete values or from noise, prior to its use in models and feature extractors. While smart application of these techniques may greatly benefit recognition systems, it is worth pointing out that this phase is optional depending on the input data at hand [44].

Due to the way kernels operate, CNNs use relatively little to no pre-processing compared to other image classification algorithms [45]. Nonetheless, one way of implementing image pre-processing in CNNs is by having an initial layer of predefined filters that are kept fixed during training. This way, additional information besides the raw input image can be provided to the network, e.g., edges and gradients [46].

With that said, the data samples contained in the datasets used in our experiments did not require any form of data segmentation as they were already separated into samples, however we did rescale all samples to a standardized size of 244x244 pixels as is it the recommended input shape for the VGG16 architecture. All rescaling was done using the resize method from the OpenCV python library (version 4.1.2).

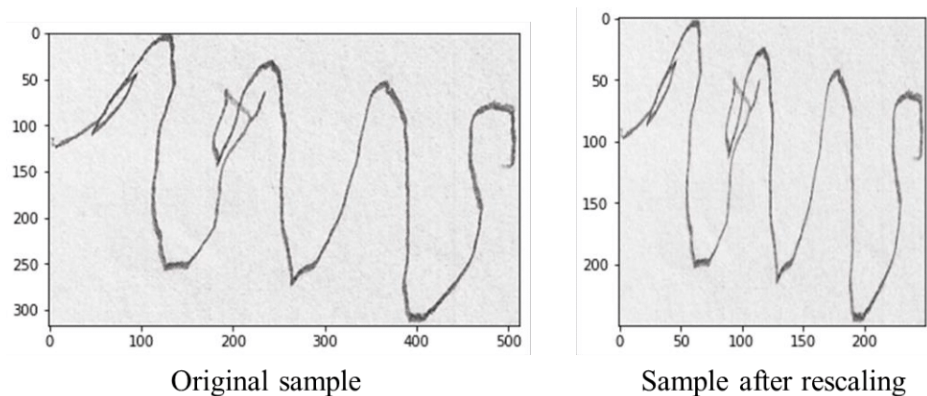


Figure 18: image resizing example (sample taken from Parkinson's drawings dataset)

In addition to that, we opted to Serialize all our data in advance by using the pickle library.

### 3.4.2. Data augmentations

Data augmentation was implemented through the **ImageDataGenerator** class and the **Flow** method from the Keras deep learning library. This method allows for a dynamic augmentation of the data sample during the fitting of the model.

The **Flow** iterator will return one batch of augmented images for each iteration. The size of this augmented batch is equal to the number of samples in the original data set divided by the batch size. Thus, the total size of the augmented dataset is equal to the number of epochs times the number of steps per epoch.

$$\text{Steps\_per\_epoch} = (\text{dataset\_size}) / (\text{batch\_size})$$

$$\text{Augmented\_set} = (\text{Steps\_per\_epoch}) * (\text{Epochs})$$

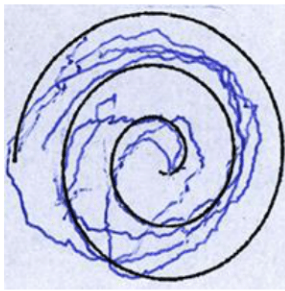
The images in the dataset are not used directly. Instead, only augmented images are provided to the model. Because the augmentations are performed randomly, this allows both modified images and close facsimiles of the original images to be generated and used during training.

Furthermore, data augmentation techniques were only applied on the training data set and not on the validation, as it is common practice. We chose to use the five following transformation techniques:

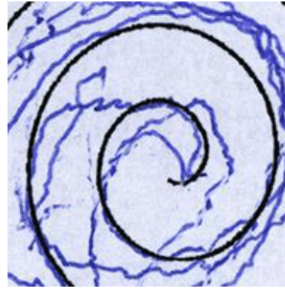
- **Image Zoom:** Range for random zoom. This argument was set to `zoom_range=0.5`.
- **Image Rotation:** Range for random rotation. This parameter was set `rotation_range=180`.
- **Image Shift:** both height and width shifts were applied during the data generation with both of these arguments being set to 0.5, this means that the max range of an image shift is equal to 50% of the total height/width.
- **Image Flip:** both vertical and horizontal flips were set to “true”.
- **Image Shear:** defines the shear Intensity and was set to `shear_range=0.5`.

Fill mode was set to “nearest”. Any parameter we did not mentioned was left in its default state as of Keras 2.8.0 and Tensorflow 2.8.0.

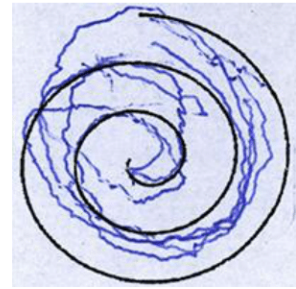
The following figures showcases the different image transformation techniques used:



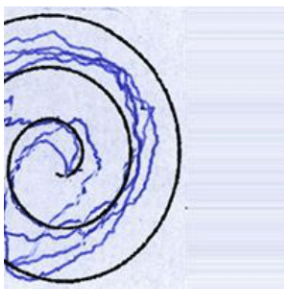
Original



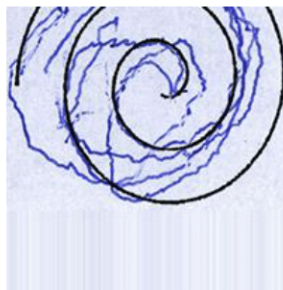
Zoom



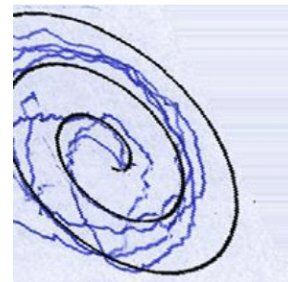
Rotation



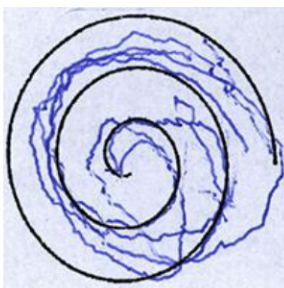
Width Shift



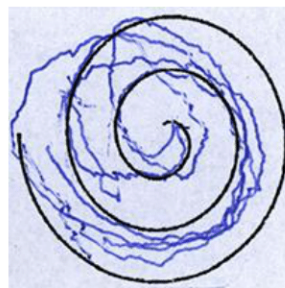
Height Shift



Shear



Horizontal Flip



Vertical Flip

Figure 19: example figures for image transformation techniques (sample taken from HandPD dataset)

### 3.5. VGG16

When it came to existing architectures, we opted to use VGG16. The architecture of this ILSVR(Imagenet) 2014 competition runner-up is available in Keras api along with the possibility of loading pretrained Imagenet weights. The fine tuning was done exclusively on the last layer that is the prediction layer with a sigmoid activation function [16].

Layer name	Layer type	Size	Kernel size	Stride	Parameters
Input_1	Input layer	244x244x3	-	-	-
Block1_conv1	Convolution 2D	244x244x64	3x3	1	1792
Block1_conv2	Convolution 2D	244x244x64	3x3	1	36928
Block1_pool	Max pooling 2D	112x112x64	2x2	2	-
Block2_conv1	Convolution 2D	112x112x128	3x3	1	73856
Block2_conv2	Convolution 2D	112x112x128	3x3	1	147584
Block2_pool	Max pooling 2D	56x56x128	2x2	2	-
Block3_conv1	Convolution 2D	56x56x256	3x3	1	295168
Block3_conv2	Convolution 2D	56x56x256	3x3	1	590080
Block3_conv2	Convolution 2D	56x56x256	3x3	1	590080
Block3_pool	Max pooling 2D	28x28x256	2x2	2	-
Block4_conv1	Convolution 2D	28x28x512	3x3	1	1180160
Block4_conv2	Convolution 2D	28x28x512	3x3	1	2359808
Block4_conv3	Convolution 2D	28x28x512	3x3	1	2359808
Block4_pool	Max pooling 2D	14x14x512	2x2	2	-
Block5_conv1	Convolution 2D	14x14x512	3x3	1	2359808
Block5_conv2	Convolution 2D	14x14x512	3x3	1	2359808
Block5_conv3	Convolution 2D	14x14x512	3x3	1	2359808
Block5_pool	Max pooling 2D	7x7x512	2x2	2	-
flatten	Flatten layer	25088	-	-	-
Dense1	Dense layer	4096	-	-	102764544
Dense2	Dense layer	4096	-	-	16781312
Prediction	Output layer	1x1	-	-	4097

Table 3: VGG16 architecture used in this work

VGG is a very popular CNN model, that proved to be efficient in many computers vision tasks. This architecture was notably used in [38], [41], and [42].

### 3.6. Proposed CNN Architecture (Park-Net)

In view of the small size of available data on this subject that is the detection of Parkinson's disease through the analysis of handwriting, we propose a relatively small CNN with an architecture comprised of 3 blocks each having one convolution ( $k = 3 \times 3$ , activation = ReLu) with a fixed stride of 1 and a kernel size of 64, 128, and 256 respectively. Each one of convolution is followed by a max pooling layer ( $k = 2 \times 2$ , stride =  $2 \times 2$ ), we've also set the input dimensions to be 224 by 244.

This model was trained from scratch using only the three datasets that were taken into consideration for this thesis (HandPD, NewHandPD, Parkinson's drawings)

Layer name	Layer type	Size	Kernel size	Stride	Parameters
Input	Input layer	244x244x3	-	-	-
Conv1	Convolution 2D	244x244x64	3x3	1	1792
Pool1	Max pooling 2D	244x244x64	2x2	2	-
Conv2	Convolution 2D	112x112x128	3x3	1	73856
Pool2	Max pooling 2D	112x112x128	2x2	2	-
Conv3	Convolution 2D	56x56x256	3x3	1	295168
Pool3	Max pooling 2D	28x28x256	2x2	2	-
Flatten	Flatten layer	200704	-	-	-
Dense1	Dense layer	512	-	-	102760960
Dense2	Dense layer	256	-	-	131328
Dense3	Dense layer	128	-	-	32896
Prediction	Output layer	1x1	-	-	129

Table 4: proposed Park-net architecture



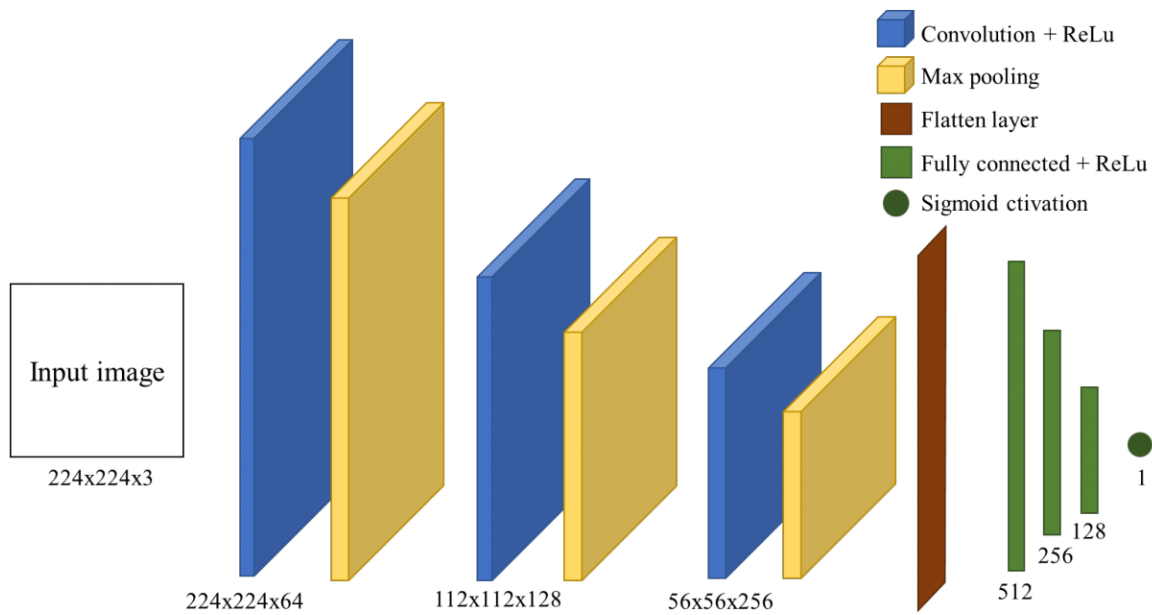


Figure 20: proposed Park-net architecture

### 3.7. Park-Net with SVM

Our next attempt was to implement an SVM classifier as the last layer of the CNN, this was done in Keras by setting the activation layer as linear and adding a 0.01 L2 kernel regularizer. During the compiling of the model, we've set the loss function to be "hinge" instead of binary cross-entropy [47]. And besides from these changes, the rest of the architecture was kept the same.

The idea of chaining a classifier to CNN feature extractor to increase performance, isn't unheard of. In fact we see it in [35], in fact, we see SVM getting used as a complement to a feature extracting CNN in Dynamically enhanced static handwriting representation for Parkinson's disease detection (2019) [38].

## 3.8. Confusion matrix

The confusion matrix, or error matrix is table layout that allows for a convenient visualization of the performance of a model, this consent is more commonly utilized with supervised learning. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa [48].

- **Positives (P):** total number of Positives, Parkinson afflicted individuals. Alternatively:  $P = TP + FN$
- **Negatives (N):** total number of Negatives, in the context on this work; Healthy individuals. Alternatively:  $N = TN + FP$
- **True Positives (TP):** the number of positive examples that the model correctly classified as positive.
- **True Negatives (TN):** the number of negative examples that the model correctly classified as negative.
- **False Positives (FP):** the number of negative examples that the model incorrectly classified as positive.
- **False Negatives (FN):** the number of positive examples that the model incorrectly classified as negative.

### Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

While there are multiple data points that can be extracted from a confusion matrix, in this work, we chose to focus on the following five:

- **Accuracy:** Known as conventional accuracy, often abbreviated as ACC, it is determined by dividing the number of correct predictions by the total number of the samples in the dataset. It can also be acquired by subtracting the error rate from 1:

$$ACC = \frac{TP + TN}{P + N} \dots \dots \dots (1)$$

- **Error rate:** Also referred to with ERR, is the division of the number of all incorrect predictions by the total number of samples in the dataset used. In correlation with ACC, the best error rate is 0.0, whereas the worst is 1.0.

$$ERR = \frac{FP + FN}{P + N} \dots \dots \dots (2)$$

- **Sensitivity:** Also known as Recall or **TPR** (short for True Positive Rate), denotes all Real Positive cases. The best TPR is 1.0, whereas the worst is 0.0. It is defined as:

$$TPR = \frac{TP}{P} \dots \dots \dots (3)$$

- **Specificity:** Also known as Selectivity or **TNR** (short for True Negative Rate), its aim is to identify the proportion of Predicted Positive cases that are correctly Real Positives. Again, the best-case scenario is TNR = 1.0, It is defined as:

$$TNR = \frac{TN}{N} \dots \dots \dots (4)$$

- **Balanced Accuracy** [37]: Often abbreviated as  $ACC_{bal}$ , it aims to identify the performance of a model regardless of data imbalance between the number of negatives and positives. And it is defined as:

$$ACC_{bal} = \frac{TNR + TPR}{2} \dots \dots \dots (5)$$

## 3.9. Conclusion

In this chapter we discussed the different data-preprocessing and training methods used in this work, we also provided vital information about the data sets used during said training and the augmentation that were applied. We've also introduced Park-net, our CNN model that we are proposing in the context of this thesis as well as the concept of combining a CNN for feature extraction and SVM as a classifier, we then moved on to aboard the concept of finetuning a pre-existing and pre-trained VGG16 model. Finally, we explained the concept of a confusion matrix and how it will be used in our results acquisition. All of this in preparation for the next chapter which will contain the experimental results acquired through our training methods as well as details on the execution environment in which the model trainings were performed.

## **Chapter 04: Experimental Study**

---

# 1.1 Introduction

In this chapter, we will provide information about the execution environment that we used, both the hardware and the software, as well as the exact data splits samples numbers, secondly, we will showcase the different acquired training results which we will judge on 4 metrics; Accuracy, Sensitivity, Specificity, and Balanced accuracy. We conclude with a state-of-the-art comparison.

## 4.1. Execution environment

In this section we detail the hardware and software used for all training done in the context of this thesis:

### 4.1.1. Hardware

All trainings were done in google colab's execution environment, and as of the time of this thesis, the free version of Google's colab provides a Tesla T4 gpu or equivalent with 16gb of vram. 2 threads of an Intel Xeon cpu with 12 to 13gb of system ram. And approximately 41gb of cloud disk storage available for 12 hours a day. The gpu however is only available for 6 hours.



*Figure 21: google colab logo*

### 4.1.2. Software

The IDE that we mostly used is google colab's notebook manager. With Python 3.7.13 and the 2.8.0 version of the Tensorflow and Keras libraries. However, for the brief instances where we had to use a local execution environment, we opted to use Anaconda Navigator (anaconda3) version 3.9.7 with Jupiter notebook version 6.4.8. the Tensorflow and Keras version in this case were 2.6.0.

## 4.2. Data splits

In this work, we made use of three datasets; HandPD, NewHandPD, and Parkinson's drawing. In addition to different combinations between these combine the three. And as previously mentioned in the Training method section, all experiments were conducted with a dataset split of 80% to 20% for the training and validation sets respectively. The following table servers to illustrate the exact sample numbers for each case of each scenario:

	datasets	Cases	datatypes	training set			validation set			contents		
				N	P	T	N	P	T	N	P	T
(a)	HandPD	Case1	Spirals	58	236	294	14	60	74	72	296	368
		Case2	Meanders	44	250	294	28	46	74	72	296	368
		Case3	Spirals + Meanders	102	487	589	42	105	147	144	592	736
(b)	NewHandPD	Case1	Spirals	111	99	210	28	25	53	139	124	263
		Case2	Meanders	111	99	210	28	25	53	139	124	263
		Case3	Circles	28	30	58	7	8	15	35	38	73
		Case4	Spirals + Meanders	223	198	421	55	50	105	278	248	526
		Case5	Spirals + Meanders + Circles	250	229	479	63	57	120	313	286	599
(c)	Parkinson's drawings	Case1	Spirals	40	42	82	11	9	20	51	51	102
		Case2	Waves	41	41	82	10	10	20	51	51	102
		Case3	Spirals + Waves	81	82	163	21	20	41	102	102	204
(d)	HendPD + NewHandPD	Case1	Spirals	169	336	505	42	84	126	211	420	631
		Case2	Meanders	156	349	505	55	71	126	211	420	631
		Case3	Spirals + Meanders	327	683	1010	95	157	252	422	840	1262
		Case4	Spirals + Meanders + Circles	356	712	1068	101	166	267	457	878	1335
(e)	HandPD + NewHandPD + Parkinson's drawings	Case1	Spirals	209	377	586	53	94	147	262	471	733
		Case2	Spirals + Meanders + Circles + Waves	437	794	1231	122	186	308	559	980	1539

Table 5: data splits for all data used for training and validating

- **N** = Negatives, Healthy Control samples.
- **P** = Positive, PD afflicted patients' samples.
- **T** = **N** + **P**, Total samples in a given section.

## 4.3. Experimental results

In this section we will go, step by step, through all the different results that we have reached through our methodology, providing clarifications, and justifications. For results acquisition formulas, see “3.8 Confusion matrix”.

Our initial tests were run on the HandPD and NewHandPD dataset and the combination of the two. Note that we chose to exclude the NewHandPD’s Cirles task at first. However, we did end up including it along with the Parkinson’s drawings as both separate and unique scenarios. See *Table 15* and *Table 16*.

### 4.3.1. VGG16

Our first experiments involved the use of the pre-existing **VGG16** architecture with pre-trained (**ImageNet**) weights. And using only the HandPD and NewHandPD dataset, in addition to a few combinations between the two. We managed to get the following results:

If we first take a look at the results acquired from the HandPD dataset alone, we notice that both the accuracy and sensitivity are quite high, even reaching ACC=90.54% on the spiral task, and a TNR=100%

on both of the latter cases; meanders, and spirals + meanders. However, these results are dismissible due to the low specificity. this is in part due to the HandPD dataset being very poorly balanced in terms of the number of positive vs negative samples it contains.

Moving to the second scenario, this time we used the NewHandPD dataset. this set has a better balanced compared to its HandPD counterpart. We notice VGG16 performed best on the

	HandPD		
	spirals	meanders	spirals + meanders
accuracy	90.54%	75.68%	75.51%
sensitivity	98.33%	100.00%	100.00%
specificity	57.14%	35.71%	14.29%

*Table 6: training results for VGG16 on the HandPD dataset*

	NewHandPD		
	spirals	meanders	spirals + meanders
accuracy	96.23%	88.68%	73.33%
sensitivity	100.00%	100.00%	100.00%
specificity	92.86%	78.57%	49.09%

*Table 7: training results for VGG16 on the NewHandPD dataset*



spirals task, achieving ACC=96.23% this time with an acceptable sensitivity to specificity ratio of TNR=100% and TPR=92.86%. the meanders task didn't fare as well, especially when combined with the spiral task. VGG16's affinity towards the spiral task may indicate the efficiency of the spiral drawings as a reliable test for PD diagnosis, it may also indicate the presence of spiral shapes in ImageNet (the dataset this model was pre-trained on).

We then attempted training on both datasets combined; HandPD and NewHandPD. And the only result worth looking at is the spirals task with an accuracy of ACC=89.68%, a sensitivity of TNR=100%, and a decent specificity of TPR=69.05%. the two other cases, in our opinion are not worth looking at due to the specificity being too low.

	HandPD + NewHandPD		
	spirals	meanders	spirals + meanders
accuracy	89.68%	59.52%	68.65%
sensitivity	100.00%	100.00%	100.00%
specificity	69.05%	5.45%	16.84%

Table 8: training results for VGG16 on both the HandPD and NewHandPD datasets combined

From here, there were two ways to improve these results; further finetuning of VGG16 or reducing the complexity of the model architecture thus creating a new model, we chose the latter.

### 4.3.2. Park-Net

This next section will showcase the results obtained using our proposed CNN Park-Net which architecture can be found in Figure 20: proposed Park-net architecture in the Proposed CNN Architecture (Park-Net) section of "Chapter 03: Our approach "

In this first scenario, we notice that Park-Net performed better than out VGG16 instance in two out of these three cases, with its best performance being spirals + meanders; netting us an accuracy of

	HandPD		
	spirals	meanders	spirals + meanders
accuracy	89.18%	86.48%	89.80%
sensitivity	88.33%	95.65%	92.38%
specificity	92.85%	71.43%	83.33%

Table 9: training results for Park-Net on the HandPD dataset

ACC=89.80% with a very well-balanced TNR and TPR of 92.38% and 83.33% respectively. And while it seems, at first glance, that VGG16 performed better in the spiral task with its 90.54% accuracy, Park-Net did achieve a superior sensitivity to specificity balance; with TNR=88.33% and TPR=92.85%.

When using the NewHandPD dataset, we noticed a significant performance boost for both that meander task and the combined meander + spiral case with a well-balanced TNR and TPR. However, in the case of the spiral task; VGG16 does out-perform Park-Net.

	NewHandPD		
	spirals	meanders	spirals + meanders
accuracy	88.68%	86.79%	91.43%
sensitivity	80.00%	72.00%	84.00%
specificity	96.43%	100.00%	98.18%

Table 10: training results for Park-Net on the NewHandPD dataset

When using a combination of the two databases, we see a significant performance increase in two out of the three scenarios. The exception being; the spiral task, although we must note that Park-Net does achieve a better sensitivity to specificity balance, it being TNR=82.14% and TPR=90.48% relative to VGG16 which achieved TNR=100% and TPR=69.05%.

	HandPD + NewHandPD		
	spirals	meanders	spirals + meanders
accuracy	84.92%	93.65%	95.24%
sensitivity	82.14%	91.55%	94.90%
specificity	90.48%	96.36%	95.79%

Table 11: training results for Park-Net on both the HandPD and NewHandPD datasets combined

While Park-Net did not surpass VGG16 in all cases, such as; HandPD:Spiral, and NewHandPD:Spiral. we did notice a significant performance increase in both the meander task and meander+spiral case, and this time with a much more balanced sensitivity and specificity.

### 4.3.3. Park-Net + SVM

Training results from the initial Park-Net model seemed promising. So, in an attempt to improve on them, we implemented a semi-SVM on the last layer of this CNN (see “3.7 Park-Net with SVM”). We did in fact observe improvements and, in this section, we will showcase the results:

Immediately, we notice an increase in performance on all three cases compared to both VGG16 and normal Park-Net. With the best performance in this case being the combination of the spiral and meander tasks, yielding an accuracy of ACC=93.20%, the best sensitivity was observed on the spiral task, and the best sensitivity to specificity balance in the meanders task.

	HandPD		
	spirals	meanders	spirals + meanders
accuracy	93.24%	89.19%	93.20%
sensitivity	96.67%	95.65%	95.24%
specificity	78.57%	78.57%	88.10%

Table 12: training results for Park-Net + SVM on the HandPD

When using the NewHandPD dataset, we again notice an increase in performance on all three experiments. With the best results being the spirals task with an accuracy of ACC=98.11% and well-balanced sensitivity to specificity ratio of 96.00% and 100% respectively.

	NewHandPD		
	spirals	meanders	spirals + meanders
accuracy	98.11%	96.23%	94.29%
sensitivity	96.00%	100.00%	92.00%
specificity	100.00%	92.86%	96.36%

Table 13: training results for Park-Net + SVM on the HandPD

Finally, when attempting the same model with both the HandPD and the NewHandPD datasets combined we achieved decently high and stable results; both spiral and meander tasks achieving an accuracy of 95.24%, with the combination of the two taking the lead at 95.63% with a sensitivity of 94.90% and a specificity of 96.84%.

	HandPD + NewHandPD		
	spirals	meanders	spirals + meanders
accuracy	95.24%	95.24%	95.63%
sensitivity	95.24%	97.18%	94.90%
specificity	95.24%	92.73%	96.84%

Table 14: training results for Park-Net + SVM on the HandPD

#### 4.3.4. Additional tests

For testing and analysis purposes, we proceeded to performed other trainings which included the use of the previously described “Parkinson’s drawings” Dataset, as well as several combinations of all three datasets; HandPD, NewHand, and Parkinson’s drawings.

The scenario where we fused all three datasets (HandPD, NewHandPD, and Parkinson’s drawing) included two cases; Case1 takes into account all the spiral tasks from all the three datasets, while Case2 accounts for all samples from all the different tasks in the three datasets.

Another experiment that we’ve attempted involved using an SVM classification layer as the last layer of the VGG16 model like we did with Park-Net. Unfortunately, this particular method of implementing the SVM classifier did not return any results worth showcasing. Further research is needed.

We also teste decided to test the predecessor of VGG16; the VGG19 model, and just like its counterpart; we used the ImageNet database pre-trained weights, and finetuned it on the last layer level.

And finally, we took the scenario containing the combination of all three datasets and applied the data augmentation techniques described in “3.4.2 Data augmentations”.

For a better visualization, we present *Table 15* which illustrates all the trainings mentioned up to this point:

- Results cells are organized as follow:

ACC%

TPR%

TNR%

				Park-Net	Park-Net + SVM	VGG16 (ImageNet)	VGG19 (ImageNet)
(a)	HandPD	Case1	Spirals	89.19%	93.24%	90.54%	<b>97.30%</b>
				88.33%	98.33%	98.33%	100%
				92.56%	64.28%	57.14%	85.71%
	Case2	Meanders	86.48%	<b>89.19%</b>	75.68%	71.62%	
			95.65%	95.65%	100%	100%	
			71.43%	78.57%	35.71%	25.00%	
	Case3	Spirals + Meanders	89.80%	<b>93.20%</b>	75.51%	75.51%	
			92.38%	95.24%	100%	100%	
			83.33%	88.10%	14.29%	14.29%	
(b)	NewHandPD	Case1	Spirals	88.68%	<b>98.11%</b>	96.23%	88.69%
				80.00%	96.00%	100%	80.00%
				96.43%	100%	92.86%	96.43%
		Case2	Meanders	86.79%	<b>96.23%</b>	88.68%	50.94%
				72.00%	100%	100%	100%
			100%	92.86%	78.57%	7.14	
	Case3	Circles	86.67%	86.67%	<b>100.00%</b>	93.33%	
			100%	100%	100%	100%	
			71.43%	71.43%	100%	85.71%	
	Case4	Spirals + Meanders	91.43%	<b>94.29%</b>	73.33%	76.19%	
			84.00%	92.00%	100%	98.00%	
			98.18%	96.36%	45.45%	52.73%	
	Case5	Spirals + Meanders + Circles	80.00%	<b>93.33%</b>	93.33%	79.17%	
			61.40%	89.47%	96.49%	100%	
			96.83%	96.83%	90.48%	60.31%	
(c)	Parkinson's drawings	Case1	Spirals	55.00%	90.00%	<b>100.00%</b>	85.00%
				0.00%	100%	100%	66.67%
				100%	81.82%	100%	100%
	Case2	Waves	50.00%	65.00%	95.00%	<b>100.00%</b>	
			0.00%	50.00%	100%	100%	
			100%	80.00%	90.00%	100%	
	Case3	Spirals + Waves	51.23%	58.54%	92.68%	95.12%	
			0.00%	50.00%	100%	95.00%	
			100%	66.67%	66.67%	95.23%	

(d)	(a) + (b)	Case1	Spirals	84.92%	<b>95.24%</b>	89.68%	88.10%
				82.14%	95.24%	100%	96.43%
		Case2	Meanders	93.65%	<b>95.24%</b>	59.52%	56.35%
				91.55%	97.18%	100%	100%
		Case3	Spirals + Meanders	95.24%	<b>95.63%</b>	68.65%	63.49%
				94.90%	94.90%	100%	100%
		Case4	Spirals + Meanders + Circles	91.76%	<b>93.63%</b>	77.15%	68.54%
				90.96%	93.37%	100%	100%
Case4	Spirals + Meanders + Circles	93.07%	94.06%	32.67%	16.83%		
(e)	(a)+(b)+(c)	Case1	Spirals	93.20%	<b>94.56%</b>	91.16%	81.63%
				94.68%	97.87%	100%	100%
		Case2	Spirals + Meanders + Circles + Waves	87.01%	<b>90.90%</b>	84.41%	74.68%
				90.32%	91.94%	96.77%	100%
Case2	Spirals + Meanders + Circles + Waves	81.97%	89.34%	65.57%	14.75%		

Table 15: gobar results table (ACC%, TNR%, TPR%)

Immediately we see VGG19 taking the lead in scenario(a) case1, on the HandPD spiral task, with an accuracy of ACC=97.30% with TPR=100% and TNR=85.71%. Unfortunately, that is one of only two cases where VGG19 performed better than other models. With the second case being scenario(c) case2 on the parkinson's drawings wave task.

From the other results, we notice a significant improvement achieved by adding an SVM classier to the base Park-Net model, it was also the most stable and performed better than or equal to our other models in twelve out of the seventeen presented cases, with its best performance reaching 98.11% on the first case of scenario(b), the NewHandPD's spiral task.

In scenario(b): case3 and all three cases of scenario(c), we observe that VGG16 and VGG19 achieved much higher accuracy than Park-Net. All four of these cases have very low sample counts, with scenario(b): case3 (the circles task) only having 73 samples in total and Parkinson's drawings dataset only reaching 204 sample even with both spiral and wave tasks are combined (see Data splits section). Therefore, the size of these scenarios makes them unfit for training a CNN from scratch such is the case for both Park-Net and Park-Net+SVM.

This also explains the decrease that we see in scenario(b): case5, adding the circles task to the spiral and meander tasks introduces a variety that without the proper number of samples, does nothing but harm the model’s learning curve and lowers the overall accuracy.

The fact that the performance recorded on the spiral tasks was, in average, better than the accuracy achieved on the other takes may indicate the Archimedean spiral shape’s superiority when it comes to the detection of PD through the analysis of hand-drawings.

Another way of extracting accuracy is through balanced accuracy ( $ACC_{bal}$ ; see 0“3.8 Confusion matrix” formula nb5). The following figure showcases results from the same tests, but this time using  $ACC_{bal}$  instead of the conventional accuracy:

			Park-net	Park-net + SVM	VGG16 (ImageNet)	VGG19 (ImageNet)	
(a)	HandPD	Case1	Spirals	90.45%	81.31%	77.74%	<b>92.86%</b>
		Case2	Meanders	83.54%	<b>87.11%</b>	67.86%	62.50%
		Case3	Spirals + Meanders	87.86%	<b>91.67%</b>	57.15%	57.15%
(b)	NewHandPD	Case1	Spirals	88.22%	<b>98.00%</b>	96.43%	88.22%
		Case2	Meanders	86.00%	<b>96.43%</b>	89.29%	53.57%
		Case3	Circles	85.72%	85.72%	<b>100.00%</b>	92.86%
		Case4	Spirals + Meanders	91.09%	<b>94.18%</b>	72.73%	75.37%
		Case5	Spirals + Meanders + Circles	79.12%	93.15%	<b>93.49%</b>	80.16%
(c)	Parkinson's drawings	Case1	Spirals	50.00%	90.91%	<b>100.00%</b>	83.34%
		Case2	Waves	50.00%	65.00%	95.00%	<b>100.00%</b>
		Case3	Spirals + Waves	50.00%	58.34%	83.34%	<b>95.12%</b>
(d)	(a) + (b)	Case1	Spirals	86.31%	<b>95.24%</b>	84.53%	83.93%
		Case2	Meanders	93.96%	<b>94.96%</b>	53.64%	50.00%
		Case3	Spirals + Meanders	95.35%	<b>95.87%</b>	58.42%	51.58%
		Case4	Spirals + Meanders + Circles	92.02%	<b>93.72%</b>	66.34%	58.42%
(e)	(a) + (b) + (c)	Case1	Spirals	92.63%	<b>93.28%</b>	87.74%	74.53%
		Case2	Spirals + Meanders + Circles + Waves	86.15%	<b>90.64%</b>	81.17%	57.38%

Table 16: gobal results table with Balanced Accuracy ( $ACC_{bal}$ %)

Switching to  $ACC_{bal}$ , we see that there is not much of a difference; in Scenario(b) Case4 VGG16 give 93.49%, surpassing Park-Net+SVM which gave 93.15% however, we do note that even when using the conventional accuracy metric, they were both equals, scoring 93.33 each.

In this following section, we took into consideration Scenario(e) to serve as a base for data augmentation. The data generation techniques used consisted of: vertical flip, horizontal flip, image shear, 50% zoom range, 180° rotation range, 50% shear range (see “3.4.2 Data augmentations”). The following figure illustrate the acquired results:

				Park-Net	Park-Net + SVM	VGG16 (ImageNet)	VGG19 (ImageNet)
(e)	HandPD + NewHandPD + Parkinson’s drawings	Case1	Spirals	76.87%	82.99%	<b>87.76%</b>	82.31%
				86.17%	85.11%	98.94%	78.72%
		Case2	Spirals + Meanders + Circles + Waves	60.38%	79.25%	67.92%	78.72%
				79.55%	83.77%	<b>90.58%</b>	75.00%
			76.34%	80.65%	95.70%	97.85%	
			84.43%	88.52%	81.97%	40.16%	

Table 17: Results Table of training done on augmented datasets (ACC, TNR, TPR)

Form the results attained in “Table 17: Results Table of training done on augmented datasets”, we notice that the best results were achieved by VGG16, 87.76% (TPR=98.94%, TNR=67.92%) on Case1 and 90.58% (TPR=95.70%, TNR=81.97%) on Case2, showing that Park-Net is not fit to be used on large data-sets.

And if we were to consider balanced accuracy instead, we still get the same results with VGG16 staying in the lead. Refer to the following figure:

				Park-net	Park-net + SVM	VGG16 (ImageNet)	VGG19 (ImageNet)
(e)	HandPD + NewHandPD + Parkinson’s drawings	Case1	Spirals	73.27%	82.18%	<b>83.43%</b>	78.72%
		Case2	Spirals + Meanders + Circles + Waves	80.39%	84.58%	<b>88.83%</b>	69.01%

Table 18: Results Table of training done on augmented datasets (ACC, TNR, TPR)



## 4.4. Comparisons

In this section, we provide a state-of-the-art comparison from works that have utilized the same data set as we did i.e., HandPD, and NewHandPD as they rank amongst the most used datasets in PD offline handwriting recognition domain.

First, HandPD. Presented in *Table 19* are the balanced accuracy results from our work as well as a few other papers that used the HandPD dataset in their tests. All mentioned tests used the original dataset with no augmentation.

		Approach	Spirals	Meanders	Spirals+Meanders
0	Our work	Park-Net	90.45%	83.54%	87.86%
		Park-Net+SVM	81.31%	<b>87.11%</b>	<b>91.67%</b>
		VGG16	77.74%	67.86%	57.15%
		VGG19	<b>92.86%</b>	62.50%	57.15%
1	Pereira et al. (2016) [34]	Naïve Bayes(spirals) SVM(meanders)	66.37%	66.72%	58.61%
2	Ali et al. (2019) [37]	Chi2+Adaboost	72.46%	78.04%	/

*Table 19: HandPD state-of-the-art comparison, using the balanced accuracy metric ( $ACC_{bal}$ )*

Secondly, NewHandPD. Compared to HandPD, NewHand is much more balanced in terms of PD to healthy samples ratio, thus eliminating the need to use balanced accuracy instead of the conventional accuracy. However, for the sake of consistency, we nonetheless chose to use  $ACC_{bal}$ .

		Data type	Approach	Spirals	Meanders
0	Our work	Offline	Park-Net	88.22%	86.00%
			Park-Net+SVM	<b>98.00%</b>	<b>96.43%</b>
			VGG16	96.43%	89.29%
			VGG19	88.22%	53.57%
1	Papa et al. (2016) [43]	Offline + Online	OPF	83.77%	84.42%
2	Gazda et al. (2022) [42]	Offline	E-CNN	92.70%	94.70%

*Table 20: NewHandPD state-of-the-art comparison, using the balanced accuracy metric ( $ACC_{bal}$ )*

We see that our approach gave better results without the need for data augmentation. Furthermore, entry (2) is as recent as 2022 and in this paper, Gazda et al. stated; “To the best of our knowledge, this is the highest accuracy achieved with the NewHandPD dataset.” which as far as we know, was still factual at the time of putting together this thesis. Meaning that our approach achieved the highest accuracy rate on the NewHandPD dataset with both the meander and spiral tasks.

## 4.5. Conclusion

In this chapter, we started by describing the execution environment used for trainings, we then proceeded to detail the datasets splits before showcasing the training results. We looked at four different evaluation metrics; ACC, TPR, TNR, and  $ACC_{bal}$ . with the latter being especially useful for evaluating models trained on imbalanced datasets (number of PD samples greater than Healthy samples of vice versa). Our best performing model was Park-Net, with balanced accuracy of  $ACC_{bal}=98\%$  on the NewHandPD spiral task, and  $ACC_{bal}=96.43\%$  on the meander task from the same dataset. we then compared these results, along with other, to state-of-the-art works and observed that Park-Net outperformed works as recent as 2022 which is the year of writing this thesis.

# General Conclusion

---

Progressive neurological disorders in general but more specifically Parkinson's disease is a serious problem which affects millions, and yet we are still to fully understand it's complexity and intricacies. The main goal of this thesis was working towards a way to automatically detect this illness through the analysis of non-intrusive static handwriting tests.

we presented a general domain study of Parkinson's disease identification through handwriting, as well as, various state of the art works on this particular subject. Some used static offline drawings while others utilized meta online data, such as; pen pressure, and in air movement amongst many. In our case, we focused on the treatment of static offline drawings data only, and for that we used three relatively popular datasets; HandPD, NewHandPD, and Parkinson's drawings.

Our method consisted of testing fine-tuned VGG16 and VGG19 models with pretrained ImageNet weights, along with our proposed Park-Net model that we then paired with a semi-SVM classifier as the last layer. We went in-depth over our methods; from data pre-treatment and augmentation, to training, and eventually results acquisition. Details of the execution environment used were also provided.

We managed to achieve satisfactory results using our training method on the non-augmented data, with our proposed Park-Net CNN paired to a semi-SVM classifier reaching; a balanced accuracy of  $ACC_{bal}=91.67\%$  on the Spiral-Meander combination of the HandPD dataset, and  $ACC_{bal}=98.00\%$  on the Spiral task of the NewHandPD dataset. In addition to considering each individual dataset separately, we have also carried out trainings on different combinations of the available tasks, which from what we have seen is not in many, all while examining four different evaluation metrics; ACC,  $ACC_{bal}$ , TPR, and TNR.

Due to time limitation, there are a few ideas which we were unable to test that we think may improve upon this work of ours:

- Further finetuning the VGG16 model by activating layers other than the dense and activation ones.
- Using VGG16 in combination with SVM; while we did reach satisfactory results with Park-Net + SVM, it would be interesting to see how far can the VGG16 be pushed.

- Using Park-Net in combination with other classifiers is something that we initially planned on doing but the time constraints and the unavailability of keras compatible approaches made it this idea un-reachable.
- Testing Park-Net on different disorders depending on dataset availability.
- Applying detailed data-augmentation approach, where every type of augmentation is tested separately for experimental purposes.
- Attempting data optimization though selectively removing bad samples from the train dataset before re-doing the training, any issues caused by the shrunken dataset can be solved with data augmentation, the subtracted samples can then be re-added at a later stage.

# Bibliography

---

- [1] M. G. Erkinen, M.-O. Kim and M. D. Geschwind, "Clinical Neurology and Epidemiology of the Major Neurodegenerative Diseases.," 2017.
- [2] L. M. d. Lau and M. M. Breteler., Epidemiology of Parkinson's disease., 2006.
- [3] P. Drotár, J. Mekyska, I. Rektorová, L. Masarová, Z. Smékal and M. Faundez-Zanuy, "Evaluation of handwriting kinematics and pressure for differential diagnosis of Parkinson's disease," *Artificial Intelligence in Medicine*, 2016.
- [4] J. Contreras-Vidal and G. E. Stelmach, "Effects of parkinsonism on motor control.," 1995.
- [5] R. Saunders-Pullman, C. Derby, K. Stanley, A. Floyd, S. Bressman, R. B. Lipton, A. Deligtisch, L. Severt, Q. Yu, M. Kurtis and S. L. Pullman, "Validity of spiral analysis in early Parkinson's disease.," 2008.
- [6] E. Ammenwerth, P. Nykänen, M. Rigby and N. de Keizer, "Clinical decision support systems: Need for evidence, need for evaluation. Artificial Intelligence in Medicine," 2013.
- [7] "WHAT IS NEURODEGENERATIVE DISEASE?," JPND Research, [Online]. Available: <https://www.neurodegenerationresearch.eu/what/>. [Accessed 31 05 2022].
- [8] M. G. K. M.-O. & G. M. D. Erkinen, "Clinical Neurology and Epidemiology of the Major Neurodegenerative Diseases," *Cold Spring Harbor Perspectives in Biology*, 2017.
- [9] J. Lambert, B. Giffard, F. Nore, V. d. l. Sayette, F. Pasquier and F. Eustache, "Central and Peripheral Agraphia in Alzheimer's Disease: From the Case of Auguste D. to a Cognitive Neuropsychology Approach.," *Cortex*, Vols. 43,7, p. 0–951, 2007.
- [10] K. D. D. P. P. A. S. a. R. S. Zham P, "Distinguishing Different Stages of Parkinson's Disease Using Composite Index of Speed and Pen-Pressure of Sketching a Spiral," *Front. Neurol*, no. 8:435. doi: 10.3389/fneur.2017.00435, 2017.
- [11] L. C. & L. P. N. Wijesekera, "Amyotrophic lateral sclerosis," *Orphanet Journal of Rare Diseases*, 2009.
- [12] R. Plamondon and S. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," no.

. , 22(1), 0–84. doi:10.1109/34.824821, 2000.

- [13] C. C. Tappert, Text Entry Systems || English Language Handwriting Recognition Interfaces, vol. 6, 2007, p. 123–137.
- [14] R. N. Keiron O’Shea, "An Introduction to Convolutional Neural Networks," *ArXiv e-prints*, no. arXiv:1511.08458v2, 2015.
- [15] B. Jason, "What is the Difference Between a Batch and an Epoch," 2018.
- [16] K. S. a. A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, San Diego, 2015.
- [17] "vgg16," programmersought, [Online]. Available: <https://programmersought.com/article/12141264861/>. [Accessed 31 05 2022].
- [18] I. a. C. J. a. B. A. Constantin, "On the Use of Deep Active Semi-Supervised Learning for Fast Rendering in Global Illumination," *Journal of Imaging*, vol. 6, 2020.
- [19] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016.
- [20] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60 (6), p. 84–90, (2017-05-24).
- [21] "ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012)," image-net.org, [Online]. Available: <https://image-net.org/challenges/LSVRC/2012/results.html>. [Accessed 04 06 2022].
- [22] L. Yang, S. Hanneke and J. Carbonell, "A theory of transfer learning with applications to active learning," *Machine Learning*, p. 161–189, 2013.
- [23] G. Vrbancic and V. Podgorelec, "Transfer Learning With Adaptive Fine-Tuning," *IEEE Access*, vol. 8, p. 196197–196211, (2020).
- [24] K. Zhao, Y. Zhou and X. Chen, "Object Detection: Training From Scratch," *IEEE Access*, vol. 8, p. 157520–157529, 2020.
- [25] X. Ying, "An Overview of Overfitting and its Solutions," *Journal of Physics: Conference Series*, no. 1168, 2019.

- [26] C. M. Bishop, *Neural Networks for Pattern Recognition*, 1995, p. 332.
- [27] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, no. , 6(1), 60–. doi:10.1186/s40537-019-0197-0, 2019.
- [28] "Callbacks," Keras, [Online]. Available: <https://keras.io/api/callbacks/>. [Accessed 14 05 2022].
- [29] P. Drotar, J. Mekyska, I. Rektorova, L. Masarova, Z. Smekal and M. Faundez-Zanuy, "A new modality for quantitative evaluation of Parkinson's disease: In-air movement," in *IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE)*, Chania, Greece, 2013.
- [30] P. Drotar, J. Mekyska, Z. Smekal, I. Rektorova, L. Masarova and M. Faundez-Zanuy, "Prediction potential of different handwriting tasks for diagnosis of Parkinson's," in *IEEE 2013 E-Health and Bioengineering Conference (EHB)*, IASI, Romania, 2013.
- [31] P. & M. J. & R. I. & M. L. & S. Z. & F.-Z. M. Drotar, "Analysis of In-Air Movement in Handwriting: A Novel Marker for Parkinson's Disease," *Computer methods and programs in biomedicine*, p. 117, 2014.
- [32] P. & M. J. & R. I. & M. L. & S. Z. & F.-Z. M. Drotar, "Decision Support Framework for Parkinson's Disease Based on Novel Handwriting Markers," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 11, no. 4, 12 2014.
- [33] P. a. M. J. a. S. Z. a. R. I. a. M. L. a. F.-Z. M. Drotár, "Contribution of different handwriting modalities to differential diagnosis of Parkinson's Disease," in *2015 IEEE International Symposium on Medical Measurements and Applications (MeMeA) Proceedings*, 2015.
- [34] C. R. Pereira, D. R. Pereira, F. A. Silva, J. P. Masieiro, S. A. Weber, C. Hook and J. P. Papa, "A new computer vision-based approach to aid the diagnosis of Parkinson's disease.," *Computer Methods and Programs in Biomedicine*, no. 10.1016/j.cmpb.2016.08.005, p. 79–88, 2016.
- [35] C. R. Pereira, S. A. T. Weber, C. Hook, G. H. Rosa and J. P. Papa, "Deep Learning-Aided Parkinson's Disease Diagnosis from Handwritten Dynamics," in *IEEE 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, Sao Paulo, Brazil, 2016.
- [36] C. a. K. M. a. M. C. a. L.-S. L. Taleb, Feature selection for an improved Parkinson's disease identification based on handwriting, 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), 2017, pp. 52-56.



- [37] L. Ali, C. Zhu, N. A. Golilarz, A. Javeed, M. Zhou and Y. Liu, "Reliable Parkinson's Disease Detection by Analyzing Handwritten Drawings: Construction of an Unbiased Cascaded Learning System based on Feature Selection and Adaptive Boosting Model," *IEEE Access*, pp. 1-1, 2019.
- [38] M. Diaz, M. A. Ferrer, D. Impedovo, G. Pirlo and G. Vessio, "Dynamically enhanced static handwriting representation for Parkinson's disease detection," *Pattern Recognition Letters*, p. 204–210, 2019.
- [39] C. Taleb, L. Likforman-Sulem, C. Mokbel and M. Khachab, "Detection of Parkinson's disease from handwriting using deep learning: a comparative study," *Evolutionary Intelligence*, 2020.
- [40] A. R. M. N. S. e. a. Naseer, "Refining Parkinson's neurological disorder identification through deep transfer learning," *Neural Comput & Applic*, p. 839–854, 2020.
- [41] M. L. M. C. J. e. a. Alissa, "Parkinson's disease diagnosis using convolutional neural networks and figure-copying tasks," *Neural Comput & Applic*, p. 1433–1453, 2022.
- [42] M. Gazda, M. Hires and P. Drotar, "Multiple-Fine-Tuned Convolutional Neural Networks for Parkinson's Disease Diagnosis From Offline Handwriting," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, no. doi:10.1109/tsmc.2020.3048892, 2022.
- [43] C. R. P. a. S. A. T. W. a. C. H. a. G. H. R. a. J. P. Papa, "Deep Learning-aided Parkinson's Disease Diagnosis," in *Proceedings of the SIBGRAPI 2016 - Conference on Graphics, Patterns and Images*, 2016.
- [44] N. D. Jyotismita Chaki, *A Beginner's Guide to Image Preprocessing Techniques*, 2019.
- [45] P. N. M. T. P. A. J. E. Tomonobu Senjyu, "Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition," in *Proceedings of ICTIS 2020*, 2020.
- [46] U. M. J. M. L. M. G. J. S. Dan C. Cireşan, "Flexible, High Performance Convolutional," in *the Twenty-Second International Joint Conference on Artificial Intelligence*, IDSIA, USI and SUPSI Galleria 2, 6928 Manno-Lugano, Switzerland.
- [47] C. François, "Quasi\_svm," Keras, 17 04 2020. [Online]. Available: [https://keras.io/examples/keras\\_recipes/quasi\\_svm/](https://keras.io/examples/keras_recipes/quasi_svm/). [Accessed 14 05 2022].
- [48] D. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," *Mach. Learn. Technol.*, vol. 2, 2008.

- [49] L. A. a. P. C. R. a. R. E. R. S. a. C. T. J. a. W. S. A. a. H. C. a. P. J. P. Passos, "Parkinson Disease Identification Using Residual Networks and Optimum-Path Forest," in *2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 2018, pp. 325-330.
- [50] S. T. H. a. B. A. a. H. M. H. A. a. K. H. Kieu, "A Survey of Deep Learning for Lung Disease Detection on Medical Images: State-of-the-Art, Taxonomy, Issues and Future Directions," *Journal of Imaging*, vol. 6, 2020.
- [51] A. Rastogi, "ResNet50," Medium, [Online]. Available: <https://blog.devgenius.io/resnet50-6b42934db431>. [Accessed 31 05 2022].
- [52] A. Patel, "Chapter-7 Under-fitting, over-fitting and its solution," Medium, [Online]. Available: <https://medium.com/ml-research-lab/under-fitting-over-fitting-and-its-solution-dc6191e34250>. [Accessed 31 05 2022].
- [53] IBM Cloud Education, "Overfitting," IBM, [Online]. Available: <https://www.ibm.com/cloud/learn/overfitting>. [Accessed 31 05 2022].
- [54] R. A. E. (RAX), "Introducing Transfer Learning as Your Next Engine to Drive Future Innovations," medium, [Online]. Available: <https://medium.datadriveninvestor.com/introducing-transfer-learning-as-your-next-engine-to-drive-future-innovations-5e81a15bb567>. [Accessed 04 06 2022].
- [55] "Comparison image neural networks," wikimedia commons, [Online]. Available: [https://commons.wikimedia.org/wiki/File:Comparison\\_image\\_neural\\_networks.svg](https://commons.wikimedia.org/wiki/File:Comparison_image_neural_networks.svg). [Accessed 05 06 2022].

# Table of figures

Table 1: Summary table of Data sets used.....	43
Table 2: Used VGG16 architecture.....	47
Table 3: VGG16 architecture used in this work.....	47
Table 4: proposed Park-net architecture.....	48
Table 5: data splits for all data used for training and validating.....	55
Table 6: training results for VGG16 on the HandPD dataset.....	56
Table 7: training results for VGG16 on the NewHandPD dataset.....	56
Table 8: training results for VGG16 on both the HandPD and NewHandPD datasets combined.....	57
Table 9: training results for Park-Net on the HandPD dataset.....	57
Table 10: training results for Park-Net on the NewHandPD dataset.....	58
Table 11: training results for Park-Net on both the HandPD and NewHandPD datasets combined.....	58
Table 12: training results for Park-Net + SVM on the HandPD dataset.....	59
Table 13: training results for Park-Net + SVM on the HandPD.....	59
Table 14: training results for Park-Net + SVM on the HandPD.....	60
Table 15: gobal results table (ACC%, TNR%, TPR%).....	62
Table 16: gobal results table with Balanced Accuracy ( $ACC_{bal}\%$ ).....	63
Table 17: Results Table of training done on augmented datasets (ACC, TNR, TPR).....	64
Table 18: Results Table of training done on augmented datasets (ACC, TNR, TPR).....	64
Table 19: HandPD state-of-the-art comparison, using the balanced accuracy metric ( $ACC_{bal}$ ).....	65
Table 20: NewHandPD state-of-the-art comparison, using the balanced accuracy metric ( $ACC_{bal}$ ).....	65
Figure 1: Basic 5 player CNN architecture outline [50].....	17
Figure 2: “max pooling” example.....	18
Figure 3: “average pooling” example.....	18
Figure 4: VGG16 architecture [17].....	20
Figure 5: VGG19 architecture [18].....	20
Figure 6: ResNet50 architecture [51].....	21
Figure 7: AlexNet architecture [55].....	21
Figure 8: difference between training from scratch and transfer learning [54].....	23
Figure 9: illustrative graphs displaying the different between Under-fitting, Optimal-fitting, and Over-fitting [52].....	24
Figure 10: example graph of an Early-stopping in the case of Training-loss vs Validation-loss [53].....	25

Figure 11: handwriting sample from in-air movements dataset [29].....	28
Figure 12: template proposed by "Prediction potential of different handwriting tasks for diagnosis of Parkinson's (2013)" [30] .....	29
Figure 13: HandPDMultiMC dataset sample, the seven tasks were segmented from a sheet filled by a PD subject [39] .....	34
Figure 14: Parkinson's disease diagnosis using convolutional neural networks and figure-copying tasks (2022) proposed CNN architecture [41] .....	35
Figure 15: Mohamad et al, pentagon/cube data set samples [41] .....	36
Figure 16: State of the art summary table .....	37
Figure 17: HandPD test template [34] .....	41
Figure 18: image resizing example (sample taken from Parkinson's drawings dataset) .....	44
Figure 19: example figures for image transformation techniques (sample taken from HandPD dataset) .....	46
Figure 20: proposed Park-net architecture .....	49
Figure 21: google colab logo .....	54