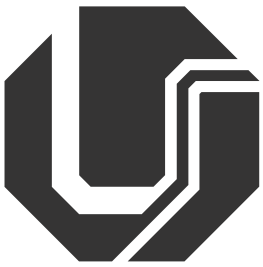
FEDERAL UNIVERSITY OF UBERLÂNDIA

FACULTY OF ELECTRICAL ENGINEERING

POSTGRADUATE PROGRAM IN BIOMEDICAL ENGINEERING



José Renato Munari Nardo

DeepLabCut Applied to the assessment of spiral and sinusoidal patterns in individuals with Parkinson's disease

Uberlândia, MG

2025

José Renato Munari Nardo

DeepLabCut Applied to the assessment of spiral and sinusoidal patterns in individuals with Parkinson's disease

Master's Dissertation submitted to the Postgraduate Program in Biomedical Engineering at the Federal University of Uberlândia, as part of the requirements for obtaining the degree of Master of Science.

Research area: Rehabilitation Engineering and Biomechanics.

Supervisor: Prof. Dr. Adriano de Oliveira Andrade

Uberlândia, MG

2025

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

N224 Nardo, José Renato Munári, 1998-
2025 DeepLabCut Applied to the assessment of spiral and sinusoidal patterns in individuals with Parkinson's disease [recurso eletrônico] / José Renato Munári Nardo. - 2025.

Orientadora: Adriano de Oliveira Andrade.
Dissertação (Mestrado) - Universidade Federal de Uberlândia, Pós-graduação em Engenharia Biomédica.
Modo de acesso: Internet.
DOI http://doi.org/10.14393/ufu.di.2025.644
Inclui bibliografia.
Inclui ilustrações.

1. Engenharia biomédica. I. Andrade, Adriano de Oliveira,1975-, (Orient.). II. Universidade Federal de Uberlândia. Pós-graduação em Engenharia Biomédica. III. Título.

CDU: 62:61

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Engenharia Biomédica
Av. João Naves de Ávila, 2121, Bloco 3N, Sala 115 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4761 - www.ppgeb.feelt.ufu.br - ppegb@feelt.ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Engenharia Biomédica				
Defesa de:	Dissertação de Mestrado Acadêmico, 125, PPGEB				
Data:	sete de novembro de dois mil e vinte e cinco	Hora de início:	08:00	Hora de encerramento:	10:51
Matrícula do Discente:	12222EBI001				
Nome do Discente:	José Renato Munári Nardo				
Título do Trabalho:	DeepLabCut Applied to the assessment of spiral and sinusoidal patterns in individuals with Parkinson's disease				
Área de concentração:	Engenharia Biomédica				
Linha de pesquisa:	Engenharia de Reabilitação e Biomecânica				
Projeto de Pesquisa de vinculação:	Avaliação multidimensional de sinais motores e não motores da doença de Parkinson				

Reuniu-se via ConferênciaWeb, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Engenharia Biomédica, assim composta: Professores Doutores: Luanne Cardoso Mendes - FEELT/UFU; César Ferreira Amorim - UNICID; Adriano de Oliveira Andrade - PPGEB/UFU orientador do candidato.

Iniciando os trabalhos o(a) presidente da mesa, Dr. Adriano de Oliveira Andrade, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado.

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente

ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Adriano de Oliveira Andrade**, **Professor(a) do Magistério Superior**, em 07/11/2025, às 10:52, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Luanne Cardoso Mendes**, **Professor(a) do Magistério Superior**, em 07/11/2025, às 10:53, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Cesar Ferreira Amorim**, **Usuário Externo**, em 07/11/2025, às 10:58, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **6834307** e o código CRC **2D5DA2AD**.

Referência: Processo nº 23117.079440/2025-44 SEI nº 6834307

ACKNOWLEDGEMENTS

I am grateful to God for the gift of life, for all the blessings and protection throughout my journey thus far and in the path ahead.

I am grateful to my grandfather, José Munari, for being my source of inspiration and the reason behind my pursuit of excellence and continuous growth, as well as for accompanying me throughout my journey and guiding me with wisdom.

I thank my family members who closely followed my master's program, offering me support and encouragement.

I am thankful to all the faculty members and fellow students who were part of my journey during the master's program, especially my advisor Adriando de Oliveira Andrade, and professors Adriano Alves Pereira, Edgard Afonso Lamounier Jr., Eduardo Lázaro Martins Naves, João Batista Destro Filho, and Selma Terezinha Milagre. I also thank my colleagues and friends: Caio Tonus Ribeiro, Camille Marques Alves, Daniel Hilário da Silva, Eduardo de Moura Neto, Lígia Nobrega dos Reis, and Luanne Cardoso Mendes. I am grateful for the knowledge you shared and for the support you provided along this path.

Finally, I thank the participants who volunteered in the research and the examination committee for their willingness to collaborate in the improvement of this work.

The present work was carried out with the support of the National Council for Scientific and Technological Development (444437/2024-0, 442150/2023-7, and 405365/2023-3), Coordination for the Improvement of Higher Education Personnel (CAPES), Foundation for Research Support of the State of Minas Gerais (FAPEMIG APQ-05015-24). Adriano de Oliveira Andrade is a CNPq fellow (302942/2022-0).

"Viva de tal maneira a não querer sair de onde está. Viva de tal maneira a absolutamente se encantar com o mundo que se apresenta diante de você. E cada segundo de vida vivido assim terá sido uma vida que você honrou, dignificou, sagrada que é, privilégio que é cada segundo de vida que você quis eterno." Clóvis de Barros Filho.

RESUMO

A doença de Parkinson (DP) é um distúrbio neurodegenerativo que afeta funções motoras e não motoras. É causada pela deficiência de dopamina no cérebro, particularmente na substantia nigra. Essa deficiência leva a sintomas como bradicinesia, tremores e rigidez, além de sintomas não motores, como perda de memória, ansiedade e depressão. Um dos principais desafios no diagnóstico da DP é a ausência de um método diagnóstico preciso, objetivo e em estágio precoce. Os métodos atualmente disponíveis são, em sua maioria, subjetivos e caros, e exames de imagem, como a ressonância magnética, geralmente só apresentam alterações em estágios mais avançados da doença. Este estudo propõe um protocolo de baixo custo e fácil replicação, apoiado por algoritmos de aprendizado de máquina (ML), com o objetivo de auxiliar no diagnóstico da DP. O método consiste na realização de tarefas gráficas (espirais e ondas senoidais) gravadas por vídeo com smartphone. As gravações foram processadas com o uso do DeepLabCut (ferramenta de deeplearning para treinar uma rede neural específica) para extração de dados posicionais, que foram posteriormente analisados por meio da biblioteca PyCaret, utilizando diferentes algoritmos de ML para a classificação dos indivíduos. Foram realizados diversos testes variando o número de pastas e combinando técnicas de seleção de atributos (FS) e função de otimização de hiperparâmetros (TF). O protocolo desenvolvido foi capaz de avaliar sinais motores relevantes para o diagnóstico da DP de forma simples e acessível. O DeepLabCut, quando treinado, alcançou uma média de precisão de 99%, apresentando dificuldades apenas em movimentos abruptos, devido ao desfoque de imagem decorrente de movimento. Os 14 algoritmos de ML treinados apresentaram bom desempenho, com os melhores resultados obtidos nos testes com 3, 7 e 9 pastas, sendo o de 7 pastas o mais equilibrado. Os três melhores classificadores — Extra Trees (ET), K-Nearest Neighbors (KNN) e Gradient Boosting Classifier (GBC) — atingiram taxas de acurácia superiores a 90%. As técnicas de FS e TF geralmente contribuíram para a melhora do desempenho, embora alguns cenários apresentassem compensações entre os parâmetros. Os modelos ET e GBC apresentaram sobreajuste devido ao tamanho reduzido do conjunto de dados, enquanto o KNN, embora com desempenho ligeiramente inferior, foi o mais equilibrado dos três. Em suma, o estudo apresenta uma abordagem promissora, de baixo custo e reprodutível, para auxiliar o diagnóstico da doença de Parkinson com o uso de ferramentas simples e técnicas de ML low-code (plataformas de baixo código).

Palavras-chave: doença de Parkinson; aprendizado de máquina; DeepLabCut; visão computacional; algoritimos de classificação.

ABSTRACT

Parkinson's disease (PD) is a neurodegenerative disorder that affects both motor and non-motor functions. It is caused by dopamine deficiency in the brain, particularly in the substantia nigra. This shortage leads to symptoms such as bradykinesia, tremor, and rigidity, as well as non-motor symptoms like memory loss, anxiety, and depression. One of the major challenges in diagnosing PD is the lack of a precise, objective, and early-stage diagnostic method. Most current diagnostic techniques are subjective and expensive, and imaging methods such as magnetic resonance imaging only become informative at later disease stages. This study proposes a low-cost and replicable protocol supported by Machine Learning (ML) to aid in the diagnosis of PD. The method involves drawing tasks (spirals and sinusoidal waves) recorded via smartphone video. These recordings were processed using DeepLabCut (DLC) (Deep learning tool to train a specific neural network to extract positional data), which was then analyzed using PyCaret to classify individuals with different ML algorithms. Multiple tests were conducted, varying the number of folds and combining feature selection (FS) and tune function (TF) techniques. The protocol successfully simulated symptoms relevant to PD diagnosis in a simple, accessible format. DeepLabCut, when trained, achieved an average detection precision of 99%, struggling only with abrupt movements due to motion blur. The ML models performed well, with the best results occurring in 3, 7, and 9-fold trials. The 7-fold setup was found to be the most balanced. The top three classifiers — Extra Trees, Gradient Boosting Classifier, and K-Nearest Neighbors — each achieved accuracy rates above 90%. FS and tf generally improved model performance, although some trade-offs were observed. Extra Trees and Gradient Boosting Classifier experienced overfitting due to the small dataset, whereas K-Nearest Neighbors, despite having slightly lower performance, remained more balanced. Overall, the study presents a promising, low-cost, and reproducible approach to assist PD diagnosis using simple tools and low-code ML pipelines.

Keywords: Parkinson's disease; machine learning; DeepLabCut; computer vision; classification algorithms.

Figure 1.0 – Graphical abstract.

List of Figures

List of Tables

List of Abbreviations and Acronyms

ANN Artificial Neural Network

CM Confusion Matrix

CNN Convolutional Neural Network

CPU Central Processing Unit

CR Classification Report

DBS Deep Brain Stimulation

DLC DeepLabCut

DNN Deep Neural Network

DT Decision Tree

EL Ensemble Learning

ET Extra Trees Classifier

FS feature selection

GBC Gradient Boost Classifier

GPU Graphics Processing Unit

HC Healthy Control

IBL Instance Based Learning

ID Identification Number

KNN K-Nearest Neighbors

LC Learning Curve

LR Logistic Regression

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

ML Machine Learning

MRI Magnetic Resonance Imaging

MTL Multi-task Learning

NB Naive Bayes

NN Neural Network

PD Parkinson's disease

RA Regression Algorithm

RAM Random Access Memory

RL Reinforced Learning

SL Supervised Learning

SLL Semi-supervised Learning

SVM Support Vector Machine

TF tune function

UN Unsupervised Learning

VC Validation Curve

Contents

1 Introdução

1.1 Problem formulation

Parkinson's disease has a difficult process to achieve a precise diagnosis. It is known that the disease starts before the patient notices any symptoms. Treatment is more effective at the beginning of the disease. Due to this, it is crucial to have an early diagnosis or a method that can indicate to the patient that he should look further for more information and tests for the disease [1, 2, 3, 4, 5]. With early treatment, the disease can be better controlled and advance slowly. Moving to the tests available, it is possible to use Magnetic Resonance Imaging (MRI), Radio Tomography, and ultrasound image tests. Those tests have a high price to execute, and some have risks, like tomography. In this scenario, it is possible to use alternative methods to achieve a diagnosis with less risk and lower cost.

An example of a test is the Movement Disorder Society-Sponsored Unified Parkinson's Disease Rating Scale (MDS-UPDRS), which measures some aspects of the patient and evaluates the answers to assess the patient. This test is subjective because it depends on the person's judgment, ability, and experience to evaluate the patient.

An alternative studied method is to analyze patients' drawings through machine learning algorithms. It is known that PD patients have motor abilities affected, and most of them have difficulties executing drawings or writing. Despite these motor difficulties, these changes are valuable for analysis, as they directly reflect the motor symptoms of the disease. Slower movements, tremors, and errors in tracing can be accurately quantified by computational tools, allowing the detection of patterns characteristic of PD. These imperfections are sensitive indicators of the disease and can be captured and analysed automatically by machine learning techniques. Thus, rather than being a limitation, this motor challenge becomes an opportunity to develop more accessible and objective diagnostic tools. The analysis of drawings is not new, but the use of Machine Learning ML can be a new tool to help studying the disease and be a support tool to diagnose PD. Some studies have great results by analyzing PD patients through the Archimedes Spiral

and ML simultaneously.

Despite the results of those studies, technology has evolved and new tools are available. Then it is possible to do tests with videos instead of just images, so it is a new method to approach and try to achieve accurate results.

1.2 Relevance of Thesis

This work proposes a low-cost, replicable framework for supporting the diagnosis of Parkinson's Disease using drawing tasks analyzed with machine learning. It leverages widely available tools—smartphones, DeepLabCut, and PyCaret—to extract and classify movement patterns. The proposed classification method achieved high accuracy (over 90%) with multiple algorithms, even on a small dataset. It demonstrates that accessible technologies can aid early, objective diagnosis of PD. This approach contributes to biomedical innovation by reducing reliance on expensive, specialized exams.

1.3 Objectives

To use the Archimedes Spiral and Sinusoidal Wave drawings allied with a new ML tool to extract features from video records of the drawings and use a second ML tool to identify and distinguish PD patients and healthy people from the data extracted.

The following objectives are secondary and aim to support the main goal of the study. They include developing a method to replicate video records efficiently and ensure a rapid setup process. Training a neural network to detect a specific object in a video recording and extract data about the object from the video. Training a second neural network to identify and distinguish between PD patients and healthy individuals.

1.4 Published papers

Sections and partial results of this research were published in 4 scientific publications and other 3 publications involving PD and machine learning.

1.4.1 Papers in reviewed scientific journals

1. José Renato Munari Nardo, Daniel Hilário Silva, Caio Tonus Ribeiro, Adriano Alves Pereira, Luanne Cardoso Mendes, and Adriano de Oliveira Andrade. *On the application of DeepLabCut for the assessment of spiral and sinusoidal patterns in individuals with Parkinson's Disease.* Tech. rep. International Journal of Online and Biomedical Engineering, 2025. (Accepted for publication and currently in the publishing process).

2. Daniel Hilário da Silva, Leandro Rodrigues da Silva Souza, Caio Tonus Ribeiro, Simone Hilário da Silva Brasileiro, José Renato Munari Nardo, Adriano Alves Pereira, and Adriano de Oliveira Andrade. "A Web Application for exploratory data analysis and classification of Parkinson's Disease patients using machine learning models on different datasets". In: *Software Impacts* 23 (Mar. 2025), p. 100737. ISSN: 2665-9638. 10.1016/j.simpa.2024.100737

3. Daniel Hilário da Silva, Caio Tonus Ribeiro, Leandro Rodrigues da Silva Souza, José Renato Munari Nardo, and Adriano Alves Pereira. "Exploring Essential Acoustic Features for Early Parkinson's Disease Classification: A Machine Learning Study". In: 21 (2 Feb. 2025), pp. 98–120. ISSN: 2626-8493. 10.3991/ijoe.v21i02.50503

4. Caio Tonus Ribeiro, Daniel Hilário da Silva, Leandro Rodrigues da Silva Souza, José Renato Munari Nardo, and Adriano Alves Pereira. "A Novel Validation Study of a Wrist Orthosis for the Objective Evaluation of Rigidity in Parkinson's Disease". In: *International Journal of Online and Biomedical Engineering (iJOE)* 20 (2024), pp. 90–108. ISSN: 2626-8493. 10.3991/ijoe.v20i12.50429

1.4.2 Conference Papers

1. José Renato Munari Nardo, Caio Tonus Ribeiro, Camille Marques Alves, Daniel Hilário da Silva, Eduardo Moura Neto, Luanne Cardoso Mendes, Adriano Alves Pereira, and Adriano de Oliveira Andrade. "Patterns in Drawings of Parkinson's Disease Patients Versus Healthy People Utilizing Markerless Object Tracking with Machine Learning". In: *XXIX Brazilian Congress on Biomedical Engineering - Volume 3: Biomedical Informatics, and Biomedical Signal and Image Processing*. Ed. by Alcimar Barbosa Soares, Renata Ferranti Leoni, and George Cunha Cardoso. Vol. 3. Oct. 2025, pp. 232–242. ISBN: 9783031949333. 10.1007/978-3-031-94934-0_24

2. José Renato Munari Nardo, Caio Tonus Ribeiro, Daniel Hilário da Silva, Adriano Alves Pereira, and Leandro Rodrigues da Silva Souza. "An experimental protocol for video recording with a smartphone for use in a pre-trained neural network". In: (2023). 10.5281/ZENODO.10157450

3. Daniel Hilário da Silva, Caio Tonus Ribeiro, Adriano Alves Pereira, José Renato Munari Nardo, and Leandro Rodrigues Silva Souza. "A low-code machine learning library in Python applied to classify and interpret data of patients with Parkinson's disease using voice records". In: (2023), p. 1. 10.5281/ZENODO.10162024

1.5 Work Structure

The dissertation consists of an introduction, 4 chapters and a general discussion. The document volume is 92 pages, in which are given 42 figures, and 12 tables.

2

Parkinson's disease

2.1 The disease

Parkinson's disease was first described in 1817 by James Parkinson in his paper "Essay on Shaking Palsy". His paper was based on his experience with six previous patients. Jean-Martin Charcot, who suggested the name "La Maladies de Parkinson". The disease was described as the patient having involuntary tremors, a tendency to bend the body forward, a lack of muscle strength while steady or performing movements, and difficulties starting to walk or move [13, 14].

Most of his discoveries have been sustained through time, but new observations have been added, like the non-motor symptoms. Nowadays, it is known that Parkinson's disease is a neurodegenerative disorder that has both motor and non-motor symptoms. The motor symptoms are described as bradykinesia, which is the slowness of movement, rigidity, and tremor while steady or while executing movement. Some examples of non-motor symptoms are dementia, dysautonomia, mood changes, pain, and sleep disturbances [15, 1].

The disease is the second most common neurodegenerative disease. It was estimated that in 2017, more than 1.2 million people had the disease, and the numbers for further years are increasing globally [16]. It is estimated that the incidence of the disease increased 0.61% per year from 1990 to 2019, and compared to female patients, male patients had more incidence and a higher increasing rate [16]. Parkinson's disease is associated with the progressive death of specific neurons in the "substantia nigra" brain area, among other structures. Another association is the presence of alpha-synuclein positive inclusions (Lewys bodies) in the substantia nigra and other brain structures. First, it was understood as a lack of dopamine in certain areas of the brain, but now there are studies that have evidence suggesting that PD is a multi-system brain disease [17, 2].

2.1.1 Motor and Non-motor Symptoms

Motor symptoms can be classified with subtypes, there are “tremor dominant”, “postural instability” and “gait difficulty”. The “tremor dominant” has been associated with a slower progression on the disease if compared to the other two subtypes. The most important diagnosis and the more common type of tremor is Essential Tremor. It has a frequency of 5 to 12 Hz which is more common in the hands, head, or voice. The resting tremor tends to increase during movement. Other types of tremor are “irregular tremor”, “jerky”, “dystonic tremor” and “action or intentional tremor” [18].

Rigidity is characterized by increased resistance to the skeletal muscles in a passive way. This symptom is less common than resting tremors, but frequently seen in PD patients. Rigidity is one of the most common symptoms in cases that are presenting with corticobasal degeneration [17].

Describing the symptoms with more detail, bradykinesia is the slowness of movement with several difficulties initiating or maintaining a body limb in motion. Findings on neurophysiological recordings suggest that this symptom is caused by an imbalance between different oscillatory rhythms. It is assumed that there is more antikinetic beta activity and less prokinetic gamma activity. Beta oscillations are more associated with the dopaminergic off state, and it decreases when patients are under dopaminergic medication effects. Studies suggest that these changes are caused by compensation brain activity in areas that are initially non-affected by the disease. These areas are commonly the anterior corticostriatal circuits and cortical regions less connected to the basal ganglia [4, 3].

Postural instability is a balance impairment that affects the patient’s ability to execute movements like walking, or standing, as well as static postures like being seated or on feet. This symptom is more common in patients with PD for a long period [13]. This symptom can result in several falls which represents a great risk to PD patients. Also, it can affect the gait, resulting in an even higher risk of falling [3].

Some of the non-motor symptoms are cognitive impairment, psychiatric disturbances, sleep dysfunction, autonomic dysfunction, and olfactory loss. It is known that PD patients have sensory symptoms and that they experience at least one sensory symptom [19].

Psychiatric disturbances are mostly common: anxiety, depression, and apathy. The anxiety affects up to 60% of the PD patients and this anxiety is generalized, meaning that it includes panic attacks, social phobia, fear, worries and apprehension. Commonly, anxiety is associated with depression. Anxiety symptoms are more often found in female patients, and this symptom can increase with motor fluctuations, which are attached to low levels of dopamine. Anxiety can appear before motor symptoms. Due to that, this symptom can be related to pathology outside the nigrostriatal pathways [20].

Depression is present in 35% of the population with PD. It is correlated with the disease progression, duration, and severity of motor symptoms, also it is affected by the occurrence of motor fluctuations, dosage of dopaminergic medication and also difficulties

with motion. Cognitive declines like dementia, anxiety, sleep disturbance and psychotic episodes are linked to increase the risk and severity of depression. The depression in PD patients is a complex phenomenon caused by PD pathology [20, 21].

Apathy is less studied than other disorders, but it is associated with depression and occurs in up to 60% of the PD population. This symptom also increases in advanced disease and is also associated with cognitive decline. Some studies have seen that PD patients with apathy had reductions on grey matter density in the cingulate gyrus and in the frontal gyrus. Another research suggests that apathy is more correlated to the ventral striatal and limbic brain areas, but also, apathy is associated with atrophy of the left nucleus accumbens [20, 21].

Sleep disturbance is one of the first symptoms that can appear in the early stages of the disease. Commonly sleep disturbance is connected to the rapid eye movement phase of sleep. This symptom is not just about the difficulty or quality in the Rapid Eye Movement phase or another; it is also excessive daytime sleep or even nightmares. This symptom can eventually trigger the anxiety and depression symptoms, and the brain region that causes this symptom is the hypothalamus and reticular formation and the neurotransmitters are dopamine and acetylcholine [20, 22].

2.1.2 Risk factors and probable causes of Parkinson's disease

Some causes and risk factors can trigger or lead to a higher rate of disease development. There are three important risk factors that impact directly on disease development, which are genetics, the environment, and interactions. Genetics can affect if the family has tendencies to develop the disease or if the person can develop the disease easily through life. Meanwhile, genetics can be an ally in treatment of the disease in the future because some specific gene-target treatments have been studied [4].

About the environmental risk factor, chemicals like pesticides, heavy metals, and toxic gases are mentioned. These examples are mostly found in industries and rural environments [18]. Substances like 1-methyl-4-phenyl tetrahydropyridine and annonacin can cause several cell death in the nigrostriatal region and then can trigger the development of the disease. Iron is a type of heavy metal that can be associated with the disease due to its role in creatine, an oxidative stress on brain cells. Manganese is also a heavy metal whose toxicity can result in impaired motor function and damage to the substantia nigra [23]. Risks associated with the environment also include falling due to head injuries, which can lead to traumatic brain injuries [10].

Illicit substances can also trigger the PD development. The abuse of illegal stimulants can increase the oxygen level in the brain which can contribute to oxidative stress leading to dopamine neuron toxicity. The three more studied stimulants that can be associated with the development of the disease are amphetamine, methamphetamine, and cocaine. The amphetamine is responsible for causing damage to neuron cells and its axon terminals,

while the methamphetamine decreases the integrity of dopamine neuron terminals in the basal ganglia region, this causes a reduction in dopamine levels and its transporters [23].

2.1.3 Parkinson's Disease Diagnosis

Although all the knowledge about the disease, the diagnosis is still uncertain and in need of more sophisticated tools and techniques. Diagnosis errors are common in practice, in clinical trials for early-stage PD patients, about 15% of people with the disease are diagnosed incorrectly, and this numbers can be increased if the diagnosis is made by a non-expert [4]. Nowadays, to achieve a diagnosis the patient can be conducted in several types of tests like genetic testing, challenges tests involving dopaminergic responsiveness, neurophysiological studies plus autonomic function testing, neuroimaging, and olfactory tests [1].

The genetic test is commonly looking for a list of genetic mutations that can cause monogenic types of PD. This list includes genes that can be related to the development of the disease. These genes are assigned as PARK loci, but there are other genes that can be related as well (GBA, GCH1, GLUD2, TBP) and others [3]. Luckily, the genetic factors do not represent a big percentage of PD cases in clinical practice, due to this, genetic test is not in the routine diagnosis process.

Another test that can be applied to identify the disease is the drug challenge test, which consists of measuring the level of responsiveness the patient has to dopaminergic drugs. This test is not taken as accurate due to its average results which are between 40 – 60 percent success. Due to this, a negative to the drug challenge test does not eliminate the possibility that the patient has the disease. Better results are achieved with this test when the patient has progressive supranuclear palsy, because it respond s favorably to Levodopa [1].

The imaging tests commonly look to visualize striatal dopamine depletion, also abnormal changes, such as mild-to-moderate degrees of cortical or subcortical atrophy. Added to that, cerebrovascular disease signs are also within the scope to look for. Computer tomography is widely used to observe those symptoms, but not limited to that, it is possible to detect pressure in the hydrocephalus. Nowadays the I-ioflupane single-photon emission is used too, this technique is called DaTscan, and another two similar types of tests are DAT-SPECT and SPECT imaging, which are techniques involving structures capable of binding with D2-dopamine receptors [1, 3].

Magnetic Resonance Imaging (MRI) has a significant role in the diagnosis of PD. This importance emerges when trying to distinguish between parkinsonian disorders and Parkinson's disease and other atypical parkinsonisms. MRI can identify several structures with atrophy, called multisystem atrophy, also identifies supranuclear palsy and corticobasal degeneration. The technique has great specificity and positive values for prediction, but its general sensitivity is around 60 to 80 percent. Only for striatal signal

intensity in the multisystem scenario, MRI can achieve sensitivity values higher than 80 percent when using a 3 to 5 mm slice thickness [1].

2.1.4 Treatment

Another challenge for PD patients and health professionals is their disease treatment. The disease is treatable but not curable yet. Treatment usually involves two or more symptoms to know, and mostly one of them is motor and the other is non-motor [4]. The difficulty is in finding a balance between those symptoms. Usually if the symptoms with more impact on the PD are a motor symptom, dopaminergic therapy is initiated to improve the quality of this symptom. On the other hand, this same therapy can develop psychosis and similar symptoms by decreasing the quality of non-motor symptoms [21].

Usually, treatment is associated with four main strategies. One of them is dopaminergic pharmacotherapy. The most common pharmacological is Levodopa. The problem with this type of therapy is that long-term use of it can cause oxidative stress which accelerates the progression of the disease. Most PD patients tend to decline or at least postpone the use of medication to treat the disease. Studies suggest that starting the use of Levodopa earlier can increase the quality of life of PD patients and manifest less symptoms than a PD patient that does not use the medication [4, 19].

The initial treatment of Parkinson's disease may benefit from Levodopa, a dopamine receptor agonist, and a monoamine oxidase type B inhibitor. Comparing these three treatments, Levodopa delivers better tolerance to treatment and better performance on mobility and movement scores [4]. Due to this, Levodopa is associated with significant motor improvement. On the other hand, it causes more dyskinesias than the other two strategies. The dopamine agonist has more side effects and they are less tolerant. Some side effects are sleep attacks or nightmares, and nausea. These effects are worse in elderly people and impulse control disorders may appear with long-term use of the medication [19].

With the long-term use of medications in PD patient treatment, it is common that the patient requires more and higher doses to control the symptoms. It is not a side effect of the medication, but the degeneration that the disease naturally causes. The older the patient gets, the less responsive it gets to dopaminergic medication, adding to this the brain loses of its capability to store extra dopamine [19]. Due to this, it is common for the patient to take more than one type of medication. This is usually used in treatment of the levodopa associated with MAO-B inhibitors and dopamine agonists.

An advanced option to treat PD is Deep Brain Stimulation (DBS) . This therapy is for prolonged use and consists of aiming a high-frequency electrical current at a determined target in the brain. Usually in Parkinson's disease the target is the globus pallidus internus and the thalamus region studies have shown that DBS can affect excitatory and inhibitory effects. The idea is that the technique can disrupt abnormal signals flowing through the

corticobasal ganglia loop. The DBS technique has positive results in controlling motor symptoms and, in many cases allows the patient to take a lower dose of medications [18].

Moving to non-pharmacological treatments, the most effective treatment is through exercise. The goal of this type of treatment is to retard disease evolution in both areas of motor and non-motor symptoms [24]. There are several physiotherapy strategies that bring benefits to the patient. In recent studies, it has shown that a physiotherapist that works with larger groups of PD patients has better outcomes and lower costs than a generic physiotherapist, many studies support the idea that practicing aerobic exercises can be an effective treatment for motor symptoms [4, 19].

Another finding is that exercises with a high intensity are more efficient than low intensity exercises to suppress symptoms. An example of this was a study with two groups. One group cycled on a stationary bike and the other group did stretching exercises. The outcome was better in the bike group which had done the exercise three times per week [4, 19].

3

Machine Learning

3.1 Machine Learning Algorithms

The concept of Machine Learning (ML) was introduced in 1959. The term was an idea of Arthur Samuel, who defined it as a field of study in which a computer could learn without being explicitly programmed. Years later, Tom Mitchell gave a new and more specific definition to the ML algorithms; he claimed that if the computer learns with some experience and respect to a task, then its performance is measured, and the performance should increase with more experience [25, 26]. In other words, the ML algorithms work by choosing options, and making predictions based on analyzed data.

Knowing that, we can affirm that a ML can have better chances to perform more accurately if it was given more reliable data to the ML to analyze and acquire more experience. So, the more robust the database you have the better the result can be.

Nowadays, machine learning algorithms are applied to a large variety of fields, like pattern recognition, natural language processing, robotics, virtual personal assistants, computer games, data mining, traffic prediction and online transportation network [27]. These algorithms help to avoid online frauds, can recommend a product, and can make a market prediction, medical diagnosis, agriculture advice, e-mail spam filtering, crime prediction, object recognition, and others.

The algorithms evolved exponentially through the years, but the processing power was a problem, memory is lacking, and velocity could be a problem. It was solved when ML algorithms started to be used on Graphics Processing Unit (GPU), which solved the Random Access Memory (RAM) memory and processing velocity problem. This solution allows us to use a significant amount of data to train the algorithms to achieve better performance. But the performance is not dependent on the amount of data and processing velocity. Nowadays we have a lot of machine learning algorithms, each one with their own method to learn and classify tasks [26, 28].

The number of algorithms we have today is so large that we need to classify them in groups and subgroups Supervised Learning (SL), Unsupervised Learning (UN), Semi-

supervised Learning (SLL), Reinforced Learning (RL), Multi-task Learning (MTL), Neural Network (NN), Ensemble Learning (EL), Instance Based Learning (IBL). It is possible to classify each method in one or more subdivisions [26, 29].

Even though we have a lot of types of machine learning algorithms, the first group of classifiers is composed of four types, which are: Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, and Reinforcement Learning [30].

Supervised Learning consists of learning a function that traces an input to an output. This is made by using an input-output pairs to serve as examples for the algorithm. In other words, the algorithm uses a collection of training examples to reach a function that leads to the desired result [29]. This type of ML needs external assistance to function. The most common SL tasks are the "classification" and "regression". The first one separates the data into groups and the second one fits the data into those groups [30]. Usually, the input is divided into two groups, the train and test dataset, and it is common to divide this in a ratio of 80/20, respectively. Figure 3.1 shows an example of how this ML algorithm works.

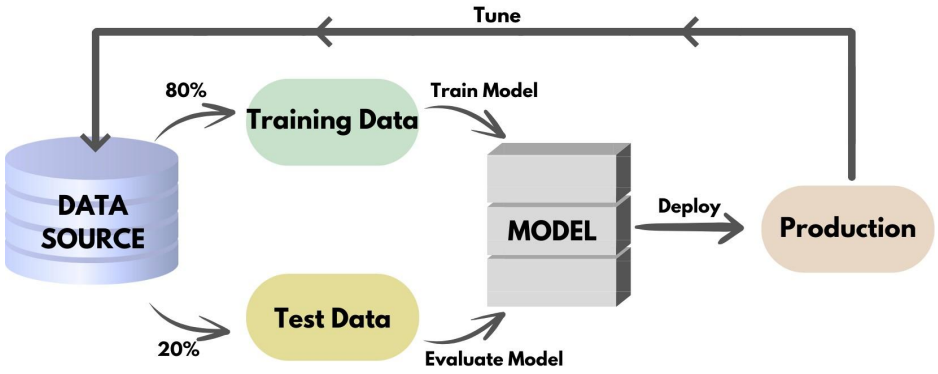


Figure 3.1 – Generic workflow of a Supervised Machine Learning Algorithm.

Moving to the UN, this type of ML relies on unlabeled datasets and does not need a human to work or interfere while working, because there is no correct answer. The model will use a few features from data training to further classify newly created data. These models are used to identify trends and structures, to cluster and reduce features. A common use for these models is clustering, anomaly detection, dimensional reduction, density estimation, and association rules [29, 30]. Figure 3.2 shows an example of a framework for an UN algorithm.

The SLL is a technique that merges the best of the other previous ML algorithms, Supervised and Unsupervised Learning. The main idea is to work with data that contains both labeled and unlabeled data mixed in the same group. This problem can occur due to the lack of human labor and expertise, and also because the amount of data can be vast [25, 29].

For the last technique, RL, it is possible to describe it as an algorithm based on rewards

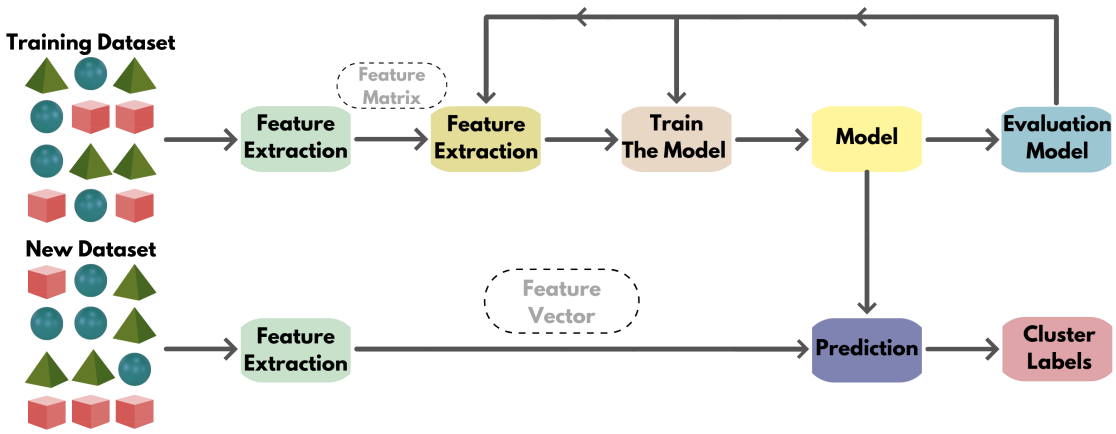


Figure 3.2 – Generic workflow of an Unsupervised Machine Learning Algorithm.

and penalties. The algorithm is responsible for taking actions based on the environment to deliver the most accurate output possible. If there's a wrong output, the algorithm gets a penalty, otherwise if the output is correct the algorithm gets a reward [25]. With this idea the ML keeps optimizing its results, looking to take fewer risks and increase the number of rewards. Figure 3.3 shows an example of a generic RL algorithm.

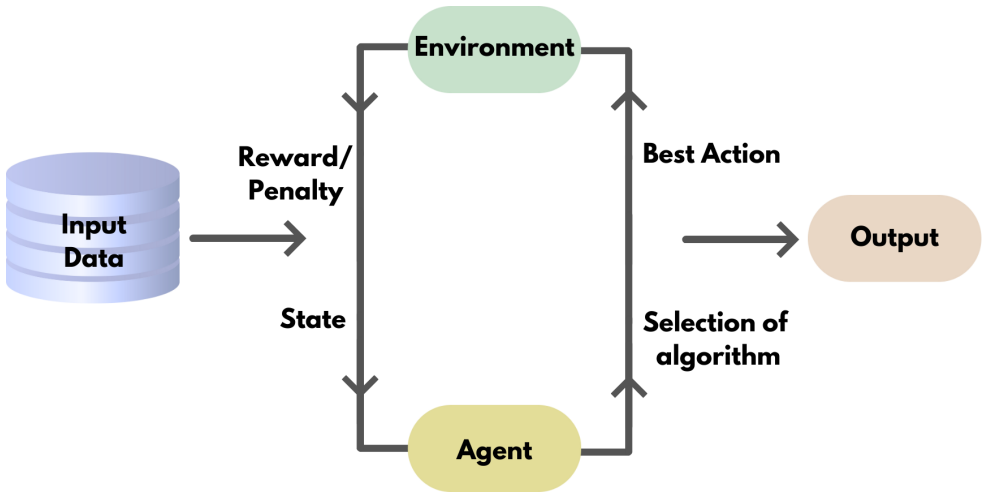


Figure 3.3 – Generic workflow of an reinforcement Machine Learning Algorithm.

With the base of ML algorithms, it is possible to dive into a new layer of classification and explore the most popular and used algorithms for a vast range of types of application.

3.1.1 Widely used machine learning algorithms

Starting with Support Vector Machine (SVM), which is the most popular algorithm for classification and regression problems. It is a supervised algorithm that works with the calculation of margins. The SVM plots every data as a point on a graph with n dimensions. The number of dimensions is the same as the number of features contained in the data set, and the coordinates on the graph represent the values of the features [25].

Decision Tree (DT) is another popular algorithm. It is also a supervised method, which involves mapping observations and conclusions. It tries to create many scenarios and their respective outcomes. Internally the DT has "nodes" referring to the feature from a specific object, and there is also the branch referring to the outcome of the specific node, then there is the "leaf" which represents the class labels. The DT allows working with both types of problems, classification and regression [31, 32].

Another well-known algorithm is the Naive Bayes (NB). This classifier uses Baye´s Theorem of Probability. The theorem calculates the posterior probability of an "x" event related to an event "y" [25]. The algorithm considers events independent of each other, meaning that a feature should not affect other features by any means. The NB is mostly used to classify text and detect spam, and the advantage of this algorithm is that is not necessary to have a large database to achieve a good result [32].

The K-Nearest Neighbors (KNN) is mostly used for classification and regression problems. The KNN works in a simple way, the algorithm makes prediction based on similar features among objects, in other words, the KNN works to find the nearest neighbors of an unknown feature whose class is about to be defined [31].

Regression Algorithm (RA) are also widely used and there is a subtype of RA that is more commonly used: Logistic Regression (LR). The idea of RA algorithms is to fit a curve on the training dataset to identify decisions for problems with continuous values. The main goal of the fitting is to minimize the distance of the features from the curve line [25, 31]. The model builds a relationship between the target and the predictor variables.

The Boosting Algorithms consist of many algorithms working with a similar idea, converting weak learning algorithms to strong learning algorithms. The Boosting works to decrease the bias and variance using multiple weak learners to enhance accuracy and efficiency [29].

On the one hand, with these examples, we covered several types of methods and logic processes used in most ML algorithms. On the other hand, according to [29] there is a type of machine learning that belongs to all groups of ML, Supervised, Unsupervised and Reinforced Learning, and deserves special attention. This type of ML is an Artificial Neural Network (ANN) that tries to imitate human brain functionalities. The idea of an ANN is to create connections between elements [33].

Although ANN is very effective and used in complex problems, this ML needs a large data base to work well. The main goal of ANN algorithms is to detect patterns, and these patterns are stored as numbers and contained in vectors. The ANN algorithms can deal with several types of data, which includes images, text, sound or voice records and time series [34].

Artificial Neural Networks has a subgroup to classify slightly different techniques and a few examples of these techniques are the Feed Forward Neural Network Artificial Neuron, Radial basis Function Neural Network, Kohonen Self Organizing Neural Network,

Recurrent Neural Network, Convolutional Neural Network, Modular Neural Network [33, 35]. Figure 3.4 shows an example of a generic workflow of a neural network workflow.

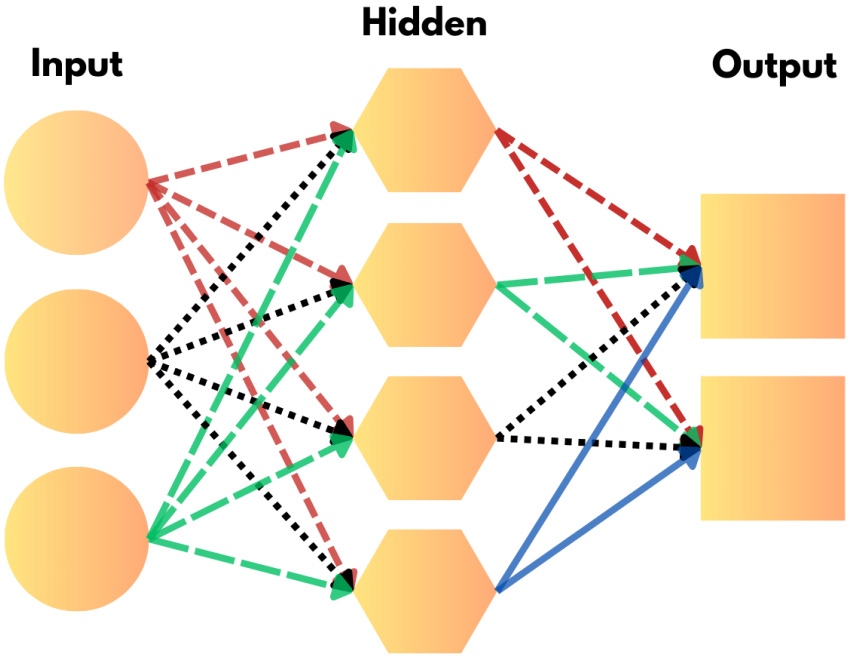


Figure 3.4 – Generic workflow of a Neural Network Machine Learning Algorithm.

3.1.2 Deep Neural Network and Convolutional Neural Network Algorithms

For this study it was necessary only to focus on Deep Neural Network (DNN) which was not cited yet and CNN technique which is used in the first software used to build the data collection process. Due to that, the other types of ANN will not be unraveled.

To understand the CNN idea, first it is interesting to understand what a DNN is. It is an ANN technique that was able to emerge when the back propagation algorithm was implemented into ANN structures. The concept was understood and applied in the middle 1980's. The DNN works with several layers of perceptrons, trained with multiple algorithms to detect data representations without manual design or feature extraction [35].

The idea is to imitate human brain functionality in its interconnection and neuron structures. A great point of advantage of the DNN structures is that the algorithm can both work with basic machine learning algorithms and with the deep learning process simultaneously. This means that human effort to organize the raw data is lower compared to basic ML algorithms, so DNN structures can work with unstructured data and feature extraction with minimal human effort [36].

As the DNN must have several numbers of layers of the process, the amount of data and training to keep it with acceptable results are bigger than most variations of ML and ANN. DNN's are usually applicable to big data problems and complex problems. The capability of DNN's is mostly dependent on the amount of data and the processing power supplied mostly by a GPU instead of a Central Processing Unit (CPU). An interesting point about DNN is that it is applicable to other ANN techniques. Due to that, other types of ANN could integrate with a DNN with no problems [35, 36].

Moving to the CNN technique, which is inspired by the human visual cortex is an option in computer vision identification, due to which a CNN can detect and recognize patterns in both images and videos. The structure consists of convolutional and sub sampling layers in series, then a fully connected layer and a normalized layer [35, 36]. About data processing, the data goes layer by layer (convolutional layers) to get more refined extracted features, then the connected layer does classification based on those refined extracted features.

Subsampling layers are commonly placed between convolutional layers. As the CNN has as input 2D images with different resolutions, each layer has its own 2D neurons which are called Kernels. A big difference between CNN and others is that the CNN architecture does not connect all neurons in a layer to every neuron in the adjacent layers. These neurons are connected to spatiality maps with fixed sizes and do partial overlaps with the neurons on the previous layer's input image or feature map [35]. As the architecture is complex, to have better performance and time of training, on the filter layer, each neuron in it is connected to the same number of neurons in the previous input layer and must have the same sequence of weights and biases [36].

To perform even quicker, the sub-sampling layers are responsible for reducing the size of the entire neural network. It works alongside the neuron with overlapping and shared weights to deliver faster learning, lower memory costs and less shifted or scaled distorted images. Then the final layers are responsible only for classifications [35, 36].

Merging the concept of a DNN to a CNN, it is possible to achieve incredible algorithms that can recognize patterns in complex images. Some examples [29] of this combination are the:

1. AlexNet, which was developed to run into Nvidia GPU's
2. Inception, created by Google
3. ResNet, which is a very deep CNN created by Microsoft, nowadays has several variations of the ResNet.
4. VGG, designed for large-scale image recognition

3.1.3 DeepLabCut and Pycaret tools

The first software used in this work to extract data from the video recording was DeepLabCut (DLC). This software is based on a CNN with DNN implemented to get the software more robust and capable of complex tasks, like processing videos with single and multi-animals or body-parts [37].

Going further on the DLC, the software works based on DeeperCut which is one of the best models to estimate positions. An advantage of DeeperCut is that it is not necessary a huge amount of data to train the algorithm to achieve a good result compared to other deep learning algorithms. The numbers are close to 25,000 labeled images for generic algorithms and around 200 for DeeperCut algorithms to achieve acceptable performance [37, 38]. This happens thanks to transfer learning based on extremely deep neural networks. This transfer learning was made using the ImageNet dataset, which is a large dataset dedicated to object detection.

The initial idea for the DLC was to track up animal's body part using a residual Neural Network (ResNet) through DeeperCut. At the beginning the DLC consisted of a pretrained ResNet and deconvolutional layers (originally from DeeperCut). To train the DLC it was used ImageNet, which is a popular large-scale object pose recognition dataset. In this scenario the deconvolutional layers will sampled the visual information and generate spatial probability densities [37, 38].

Like the other algorithms, this cited above is generic training, and for better tuning it is necessary to label the data and select the specific body part or characteristics that are desired to be tracked. This task consists of choosing frame by frame the interest point, then during the training session, the weights are adjusted to the highest and lowest probabilities of the body part position. At least this workflow combined with the pre-trained ResNet on the ImageNet data set, which leads to a robust algorithm both in performance and efficiency [37, 38].

To demonstrate the power of the tool, the researchers [37] used images of a single mouse to train the DLC tool, they used 1,080 frames from 7 mice and 2 different camera positions. Manually they labeled the interesting body parts. For the demonstration they used images of different mice, and the software worked well. Then they tried using frames with three mice at the same time, and the tool continued to detect correctly but not errorlessly the mice. It was found that the tool could perform even better if trained with multiple mice in the same frame. This finding is due to transfer learning, which allowed the tool to extend its learned features to an entirely different scenario where it had no training.

A significant advantage of the DLC is the speed at which it processes the images. Authors of [37] reach very good speeds with videos with the following presets: 682x540 pixel frames were processed at 30Hz on a Nvidia 1080Ti GPU. The videos are transformed into a low-dimensional time sequence by the software. In this case a 204x162 pixel image

could achieve 85 Hz process velocity.

To proceed to the second step of the study, it was used a library called PyCaret which is an open-source and low-code machine learning tool, built to run in the Python programming language. The main goal of the library is to reduce the time to run experiments. Instead of testing a single algorithm at a time, the library allows you to run several ML algorithms at once. By doing this, the library contributes to scientists performing experiments efficiently and quickly compared to the traditional method [39].

Another goal of PyCaret is to be ready to work with complex ML algorithms with just a few lines of code, allowing unexperienced users to test their ideas and hypotheses. This feature was developed considering the emerging amount of new data scientists daily. The pipeline of the library assures reproducibility and generates results. Furthermore the pipeline can be stored in binary format and transferred to other machines and works perfectly, and it is ready to integrate with other environments like Microsoft Power BI, KNIME, Tableau, Google Colab and others [39, 40].

PyCaret is a supervised Machine Learning method which is used to classify elements into a group. The prediction must occur in class labels both discrete and unordered, some common problems to classify are "Yes or No", "Leave or Stay" and "Positive or Negative" problems. Although the module can be used for multi-classing problems and other complex problems [39].

The steps necessary to start using the PyCaret are very simple. Following the information in [39], it is just necessary to build the setup and the other three simple functions. To exemplify, bellow it will be presented the steps to get the library ready and working according to [39].

First it is necessary to start the setup, it is responsible for initializing the training environment and creating the transformation pipeline. It is not possible to use any other function before the setup has been set, and for this step it is necessary to have the data and target ready to be used. The example of code is presented below:

```
# load sample dataset
from pycaret.datasets import get_data
data = get_data('diabetes')
```

Listing 3.1 – Standard Setup Function

Then it can move to the Functional API, which is part of the setup function. It will set anything that could be necessary to adjust, like the seed number, fold strategy, number of folds, the train size proportion and others. The generic code is presented below:

```
    from pycaret.classification import *
s = setup(data, target = 'Class variable', session_id = 123)
```

Listing 3.2 – Standard API Function

With the setup function determined, it is possible to move on to the compare models step. This section will train and evaluate the performance of all the estimators available

in the library, and the method used is cross-validation. In this section, it is possible to add another model or exclude all models if necessary, as well as metrics that can be added or removed. The output of this function is a table containing the average of the cross-validated scores. The code to this function is simple and listed below and figure 3.5 shows an outcome of the function related to this work:

```
# functional API
best = compare_models()
```

Listing 3.3 – Standard API Function

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
qda	Quadratic Discriminant Analysis	0.7802	0.8574	0.6973	0.8971	0.7384	0.5806	0.6296	0.1600
rf	Random Forest Classifier	0.7606	0.0000	0.7277	0.8027	0.7370	0.5172	0.5462	0.3329
dt	Decision Tree Classifier	0.7504	0.0000	0.7154	0.7894	0.7236	0.5028	0.5293	0.1029
lda	Linear Discriminant Analysis	0.7418	0.7532	0.7515	0.7438	0.7358	0.4814	0.4970	0.0771
gbc	Gradient Boosting Classifier	0.7394	0.8366	0.7528	0.7634	0.7364	0.4778	0.5054	0.2329
et	Extra Trees Classifier	0.7300	0.0000	0.6472	0.8102	0.7027	0.4464	0.4738	0.2000
lightgbm	Light Gradient Boosting Machine	0.7206	0.0000	0.7234	0.7636	0.7080	0.4436	0.4830	0.4600
xgboost	Extreme Gradient Boosting	0.7190	0.0000	0.7166	0.7524	0.7139	0.4381	0.4623	0.1943
ridge	Ridge Classifier	0.6766	0.7435	0.6744	0.6888	0.6510	0.3644	0.3934	0.1171
ada	Ada Boost Classifier	0.6750	0.7822	0.6268	0.7185	0.6428	0.3508	0.3711	0.1743
svm	SVM - Linear Kernel	0.6672	0.8152	0.6508	0.7053	0.6485	0.3480	0.3849	0.1329
knn	K Neighbors Classifier	0.6664	0.0000	0.6098	0.7072	0.6223	0.3352	0.3580	0.0843
lr	Logistic Regression	0.6546	0.7561	0.6472	0.6539	0.6329	0.3165	0.3269	0.8614
nb	Naive Bayes	0.5942	0.0000	0.7370	0.6421	0.6277	0.2241	0.2807	0.0743
dummy	Dummy Classifier	0.4482	0.0000	0.2857	0.1170	0.1654	0.0000	0.0000	0.0714

Figure 3.5 – Real Output of the Compare Models Function.

After the comparison is completed and unraveling can be done with the best model, the function allows you to plot graphs to evaluate the model's performance. This specific step is the Analyze Model function. Again, it is a simple line code, and it is shown in the sequence as well as an example of the outcome of this work. Represented in Figure 3.6.

After the function presented above is used, the pipeline moves to preparation for the tune function (TF). In this case, we select the most suitable ten models and store them in a variable to then use the 'Tune_model' function. This function will perform a retraining on the best ten models we had, achieving superior results. Figures 3.7 and 3.8 show the real a example of our work.

Finally, it is possible to predict the tuned models and compare the results with the non tuned models. Again, it is a simple line of code necessary to start the function. Figure 3.9 shows the line code for predicting the ten best models in this study.

With this theoretical basis it is possible to understand clearly all the work presented in this thesis. Based on the understanding of the disease, its challenges and progress through

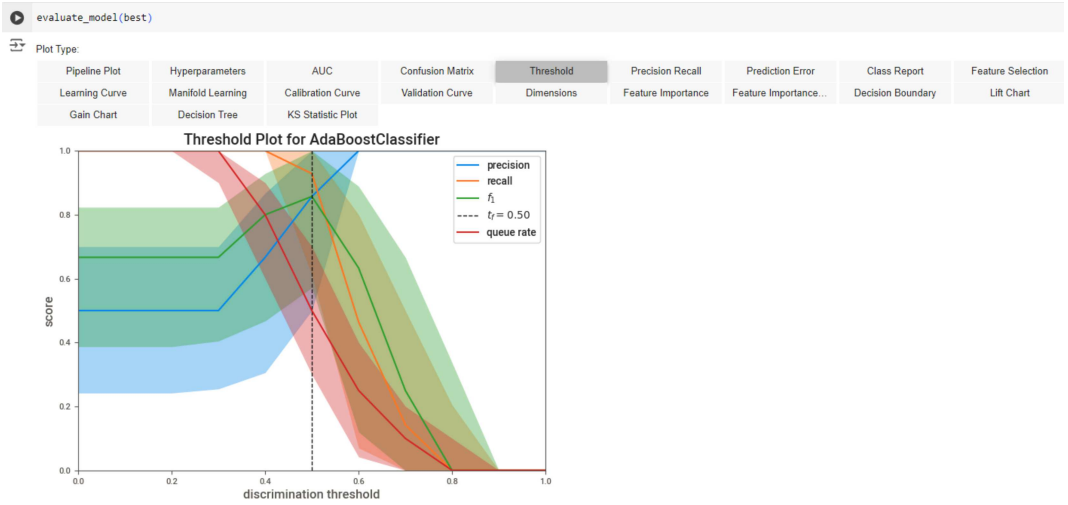


Figure 3.6 – Real Output of the Evaluate Best Model function.

```
########## Código para análise dos 10 melhores modelos.
## Criando os 10 melhores modelos

model_A = create_model('qda')
model_B = create_model('rf')
model_C = create_model('dt')
model_D = create_model('lda')
model_E = create_model('gbc')
model_F = create_model('et')
model_G = create_model('lightgbm')
model_H = create_model('xgboost')
model_I = create_model('ridge')
model_J = create_model('ada')
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9286	0.9796	0.8571	1.0000	0.9231	0.8571	0.8660
1	0.9286	1.0000	1.0000	0.8333	0.9091	0.8511	0.8607
2	0.7857	0.8776	0.8571	0.7500	0.8000	0.5714	0.5774
3	0.5714	0.6444	0.3333	1.0000	0.5000	0.2632	0.3892
4	0.6429	0.8367	0.2857	1.0000	0.4444	0.2857	0.4082
5	0.8462	0.8333	0.8333	0.8333	0.8333	0.6905	0.6905
6	0.6923	0.8333	0.4286	1.0000	0.6000	0.4091	0.5071
Mean	0.7708	0.8579	0.6565	0.9167	0.7157	0.5612	0.6142
Std	0.1298	0.1086	0.2735	0.0996	0.1832	0.2318	0.1835
	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC

Figure 3.7 – Real Output of the Evaluate Best Model function.

history, the same for machine learning. The next chapter will explain the method utilized to develop the data acquisition protocol, along with the tests and papers used to develop and inspire the actual used protocol.

```
[13] ## Otimizando os 10 melhores modelos
     tuned_A = tune_model(model_A)
     tuned_B = tune_model(model_B)
     tuned_C = tune_model(model_C)
     tuned_D = tune_model(model_D)
     tuned_E = tune_model(model_E)
     tuned_F = tune_model(model_F)
     tuned_G = tune_model(model_G)
     tuned_H = tune_model(model_H)
     tuned_I = tune_model(model_I)
     tuned_J = tune_model(model_J)
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9286	0.9796	1.0000	0.8750	0.9333	0.8571	0.8660
1	0.7857	0.9333	1.0000	0.6250	0.7692	0.5882	0.6455
2	0.8571	0.9592	0.7143	1.0000	0.8333	0.7143	0.7454
3	0.7857	0.9333	0.8889	0.8000	0.8421	0.5116	0.5185
4	0.7857	0.8980	0.5714	1.0000	0.7273	0.5714	0.6325
5	0.8462	0.8810	0.8333	0.8333	0.8333	0.6905	0.6905
6	0.9231	0.8810	0.8571	1.0000	0.9231	0.8471	0.8571
Mean	0.8446	0.9236	0.8379	0.8762	0.8374	0.6829	0.7079
Std	0.0583	0.0356	0.1423	0.1291	0.0690	0.1248	0.1161

```
Fitting 7 folds for each of 10 candidates, totalling 70 fits
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC

Figure 3.8 – Real Output of the Evaluate Best Model function.

```
## Predição para os 10 melhores modelos
predict_A = predict_model(tuned_A)
predict_B = predict_model(tuned_B)
predict_C = predict_model(tuned_C)
predict_D = predict_model(tuned_D)
predict_E = predict_model(tuned_E)
predict_F = predict_model(tuned_F)
predict_G = predict_model(tuned_G)
predict_H = predict_model(tuned_H)
predict_I = predict_model(tuned_I)
predict_J = predict_model(tuned_J)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Quadratic Discriminant Analysis	0.8333	0.8264	0.7500	0.9000	0.8182	0.6667	0.6761

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Random Forest Classifier	0.8333	0.9306	0.9167	0.7857	0.8462	0.6667	0.6761

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Decision Tree Classifier	0.7500	0.7500	1.0000	0.6667	0.8000	0.5000	0.5774

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Linear Discriminant Analysis	0.8750	0.9861	1.0000	0.8000	0.8889	0.7500	0.7746

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Gradient Boosting Classifier	0.9167	0.9722	0.9167	0.9167	0.9167	0.8333	0.8333

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Extra Trees Classifier	0.9167	1.0000	1.0000	0.8571	0.9231	0.8333	0.8452

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Light Gradient Boosting Machine	0.8750	0.9306	0.9167	0.8462	0.8800	0.7500	0.7526

Figure 3.9 – Real Output of the 10 best predicted models.

4

From Data Collection to Classification: The Proposed Methodology

4.1 Basis for the development of the experimental protocol

Having in mind the idea of the complexity of the PD it is certain that many researchers and students will try to develop some technology to achieve better and faster chemical treatments, ways to slow the disease progress, discover the trigger to develop the disease, better solutions to diagnose patients and others.

The diagnosis goals can be divided in sublayers, which can include, developing easier, faster, non-invasive and other mechanisms to diagnose someone with the disease. In this case, we focused on researching a mechanism capable of classifying patients between a Healthy Control (HC) and a PD group. The idea was to use simple materials to extract enough data to be processed in a machine learning algorithm and obtain an accurate result as some studies have done before [41, 42, 43, 44, 45].

The individuals demographic and clinical features are presented in Table 4.1 to better explain the conditions of each participant

Looking for something with this kind of idea, it was possible to reach some research involving drawings and writings not just to diagnose PD, but other diseases and disorders like dysgraphia and for transcribing handwritten texts to digital texts [42, 43, 44, 45, 46, 47]. It usually comes in a pen and paper test which is then photographed and processed by an ML algorithm, or using an apparatus developed to record data.

As is possible to see, many other authors [48, 49, 50, 51, 52, 53] used Archimedes Spiral test to extract some kind of data from voluntaries, using different method to achieve this data. Some used photographs of spirals after the shape were completed and some used tablets and extracted coordinates through the system itself.

In fact, both studies [48], and [50] used images of spirals already drawn as a data source. The idea consisted of extracting data from those images through different techniques but

Id	Age	Sex	UPDRS-III	H&Y
3	55	F	20	1
15	63	F	14	1
16	64	F	5	1
7	58	M	20	2
14	71	M	21	1
6	62	F	27	2
9	67	F	26	3
20	49	F	8	1
18	52	M	28	1
19	59	M	68	3
PPD	60 ± 6.78	- - -	20.5 ± 17.39	1 ± 0,84
30	58	F	- - -	- - -
33	53	F	- - -	- - -
35	66	F	- - -	- - -
36	55	F	- - -	- - -
37	65	F	- - -	- - -
40	62	F	- - -	- - -
31	56	F	- - -	- - -
32	56	F	- - -	- - -
34	52	F	- - -	- - -
38	72	F	- - -	- - -
PHC	57 ± 6.50	- - -	- - -	- - -
OAVG	58 ± 6,47	- - -	20.5 ± 17.39	1 ± 0,84

Table 4.1 – Demographic and clinical features of individuals. Id: Identification number, UPDRS-III: Movement Disorder Society-Unified Parkinson's Disease Rating Scale, H&Y: Hoehn And Yahr Scale, PPD: Partial average PD group, PHC: Partial average HC group, OAVG: Overal average.

trying to achieve a similar goal. The images were taken using a perpendicular angle from the camera to the paper containing the drawings. This is due to better recognition and feature extraction from the drawing; a picture taken from another angle must have the coordinates fixed or at least be considered. Therefore, to avoid more processing to acquire data, images are taken from this angle.

An advantage of this kind of process is the low cost of reproduction, and the velocity with which to and processing data. To acquire data, it is only necessary a paper sheet, a pen or pencil and a camera to take photos. Despite these advantages, there is some weakness in this kind of work compared to the others [49, 51, 52, 53]. As they used a tablet to capture the data, they can get data in real time and with a high precision of coordinates. Including that some tablets allow measuring the pressure exerted on the pen, and this information can be valuable.

Another advantage of using a tablet is that it is not necessary to convert an image into a coordinate system. This is because the tablet already takes the coordinates to create the drawing, and you can perform several drawings quickly. Despite that, the cost of a tablet can be high or inaccessible for some people, which can be a huge problem for acquiring data.

Trying to take advantage of both scenarios but focusing on affordability and speed to reproduce the experiment, the idea was to use a paper and pen process to keep affordability and a smartphone to capture the images. But instead of taking pictures of the drawings, the process would be recorded as a video. The video can provide us more than just the coordinates of the drawings. It could bring data from the transitions in the drawings.

Videos can provide data for other extraction feature processes and different experiments. Although videos have a big difference in time to processing in machine learning due to their length, archive size and complexity. Aiming to avoid an extreme scenario, the goal was to process videos with full HD resolution 1920 x 1080 pixels at 30 frame per second. This resolution was taken into consideration the capabilities and system requirements, and used by the software DLC. In addition, the machines available in the lab to process the data are mid-range machine in our country.

4.2 Drawing paper sheet development

With an idea of how to execute data acquisition, the next step was to develop a paper sheet containing all the necessary information to identify the patient, its session or number of collections made, and the specific spaces to draw the spirals. In this case, it was used an A4 sheet with gaps to be filled with Identification number, date and dominant hand to perform the drawing on top of the sheet, and 5 spaces to draw spirals containing references to the spirals on both sides of the gaps to draw.

After rethinking the drawing process and taking into consideration the work done in [48], it was decided to include sinusoidal waves in the process, to acquire more information about the drawing process. Then the paper sheet was changed to include five gaps to perform sinus wave drawings. Figure 4.1 shows how the paper sheet was defined.

The paper sheet was designed to allow the participants to have a minimum space to draw a good sequence of movements for each drawing and to serve as a parameter to

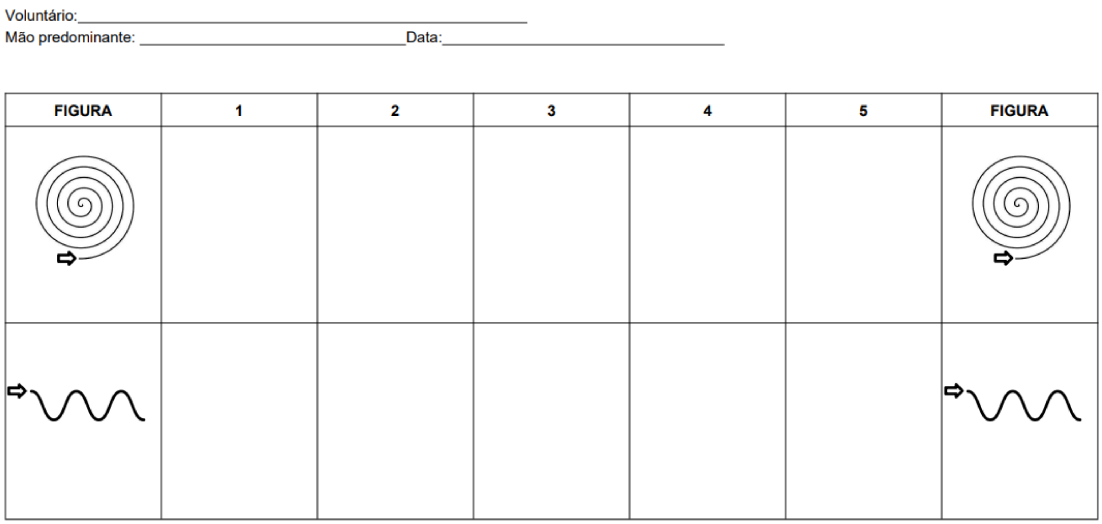


Figure 4.1 – Final version of the paper sheet designed to collect the data.

try to replicate drawings with better precision. The spaces were thought to allow the drawings to be centered on the paper and contain the reference drawings on both sides.

The reference drawings were included in case a patient lost the reference of the drawing to be performed and to serve as an incentive to perform more dense and good quality drawings. Both sides references were included to allow both right-handed and left-handed individuals to have the possibility to see the reference while drawing both tasks.

A sign was added to the reference drawings to indicate where the process must begin. The spiral drawing was decided to start from the outside to the inside to increase the difficulty to draw. This difficulty was observed while collecting samples to further train the DLC network. This difficulty might be caused by the automatic drawing process most people are used to, which is to draw the spirals inside out.

The decision for five drawings for each task was decided based on the camera view, which is limited, the pent tip size which is tiny and gets more challenging to detect at greater distances and considering the PD patient's fatigue. During the pilot test it was observed that some patients got worse drawings if they repeated the whole test twice in a row. Therefore, with all those considerations, the size and quantity of repetitions was decided on five repetitions each.

To maintain good storage of the data, it was included on the paper sheet some blank spots filled with important information, like the Identification Number (ID) of the patient, the dominant hand and the date that this test was executed. The ID number is crucial for storage and processing data, while the dominant hand is important for the matching individuals process, which matches age, sex and the dominant hand. The date of execution was important to verify the last session and maintain a follow-up with the patient. It was assumed to get this follow-up with the date to avoid confusion with the information from the other three studies which were conducted at the same time in the laboratory.

4.3 Image acquisition to train the neural network

The idea of using DeepLabCut as a tool to extract object positions without a marker was a challenge. Some studies tried to use different software to extract data from images without using a marker and achieved good results [54, 55, 56]. A markerless software can deliver good results, but it contains some difficulties, like soft tissue problems and mispositioning due to interferences or bad training. None of the laboratory members had previous experience with this kind of software. Due to that and its complexity, it was necessary to test and learn how the software worked.

To explore the potential of the tool and understand its functionality, tests were conducted using randomly hand-drawn images. In the first test, it was used video recordings of four volunteers. The video recording contained ten drawings of four different shapes. The shapes were a triangle, sinusoidal wave, a circle and a star. Firstly, those shapes were chosen randomly.

At total, it was possible to have four videos of each participant, summarizing sixteen videos to be labelled on the DLC software, and then train the neural network.

The first trial was done with just one video label. It was done to verify if the machine was capable of handling the DLC and the video size. With this first trial, it was discovered that a video with 4K resolution resulted in a very slow training and caused a few software crashes. Then the video was processed in full HD resolution to verify the speed and reliability of the training process.

The second trial with full HD resolution came back with good results. Faster processing time and no crashes. With the trial's success, it was possible to move forward and train the NN with more videos and check the possibility of tracking a pen tip while the patient draws.

To test the possibility cited above, all fifteen videos were used in the process. The process was then specifically labelled to track a blue pen of the brand "BIC Crystal" which has a blue tip. This pen was chosen for some reason. First, its cost is very low, which is one of the main goals of the experiment. It has a vibrant color that can help to be identified by the software, and this same color could help to avoid other similar colors in the environment and cause issues to detect the correct object. Then another reason is that it is a pen with a good flow which is good for Parkinson's disease participants due to their difficulty writing and drawing.

The decision about color and detecting the object was made by analyzing the difficulties found in another research. Difficulties rise along with complexity increases, like the number of parts to detect, and the different shapes or colors it can have, like a human hand or a group of people [57, 58, 59]. Another difficulty is based on object occlusion; a great example is when the software was designed to detect human hand joints. The software can track all the joints while the hand is open, but if the hand closes, some joints will be occluded, and software will not be able to detect them. Basically, what some

software tries to do is estimate the position based on other joints.

Researchers have tried to solve the problem of object occlusion and mispositioning mostly with two techniques. One of them is to capture images from more than one angle. Some studies tried to use more than 20 cameras simultaneously [60, 61, 62]. Other studies tried an alternative method for capturing images, they used a Microsoft Kinect camera. The idea is based on that this camera has both deep field vision and RGB vision. The advantage presented by this technique is supported by deep field vision which can deliver with some data processing a 3D spatial positioning of the object [56, 57, 63].

As no tests were conducted before in the lab, a test must be conducted to determine if the software would be able to detect the pen tip. Also discovering if the software considers the pen color or other characteristics. Another issue that was necessary to consider was object occlusion that might occur with some volunteers due to different styles of writing and different ways to hold the pen.

With the second test done it was possible to observe some results. That the number of iterations necessary was not too large for our objective, the video with 1920 x 1080 pixels of resolution delivered a good object tracking, and faster processing than the 3840 x 2160 pixels.

Due to these results, it was decided to conduct another trial, involving a larger number of images labeled from the fifteen original videos plus two videos of spirals and sinus waves from the other two individuals, and a higher iteration number. This time, it was used for 784 labelled images and four hundred-thousand iterations.

This test resulted in 1.88 pixels of error on the training session and 1.93 pixels of error on the test session. The result was made by a default configuration of the software that uses 95% of the data to train and 5% to test. To test in a real scenario, a video was recorded and processed using this NN to evaluate if the NN was able to properly track the pen tip. The video came out with a high precision like it will be presented in the results session on this work.

4.4 Development of a stand for the smartphone

Knowing that some challenges must have attention, like object occlusion. The pen tip goes from close range to long range in the field of view of the camera, which could cause problems with object detection due to the aimed object starting with more pixels in the first drawing and finishing the row of drawings with less pixels [61]. And the replicability depends on the reliability of the detection.

Due to that, it was necessary to develop a stand that could be positioned the same way for every data collection without a mark or reference point. Also allow the camera to be in the same position as well as the paper sheet it was desirable that the paper sheet was always at same distance and angle to the camera´s field of view.

Using an articulated stand for smartphones from "H'MASTON", several test images were done trying to figure out what angle should produce the best angle to capture the videos. No test was conducted using the DLC trained Neural Network, due to this being not the focus of the work. It was used just for comparison among several images by the author.

To choose images, it was necessary to find for some criteria. First, the angle should allow the camera to be close to the paper sheet to avoid undesired objects in the field of view. The angle should have all the information contained on the paper sheet, and the blank spaces to drawn must be in the center of the field of view, to avoid problems at the extremities of the field of view and to minimize object detection problems [61]. It was necessary that the angle was not too close to be perpendicular to the table, otherwise the drawings would not be formed on graphs later. Although the angle could not be parallel to the table, because the pen tip would be occluded by the hand and the pen itself. Figure 4.2 and Figure 4.3 show the smartphone stand and some possible angles.

Figure 4.2 – Hmaston phone stand used to perform the trials.

As this support was made to be used in high positions and to use the frontal camera of a smartphone, it was necessary to use the support in an inverted way. Although it was not the right way to use it, allowed us to find an appropriate angle to capture images.

After acquiring a probably good angle, it was necessary to use a Computer Aided System to develop a stand with a limited angle and as stable as possible. A second goal was to develop a template that could be attached to the stand to be a guide for paper sheet positioning. The idea of it being attached is due to the possibility that a volunteer could be right-handed or left-handed. With this part being attached, it was only necessary to invert the template.

Figure 4.3 – Smartphone stand with the smartphone positioned.

Firstly, a light prototype was designed to ensure the measurements and capabilities of the stand. This prototype was designed by the author using the software SolidWorks by Dassault Systems and it was later printed on a 3D printer (GtMax Core AB 400) in the laboratory using white PLA (plastic) from (GtMax Manufacturer). With this first prototype it was possible to verify that the structure needed to be reinforced in order to avoid shaking while the video recording was taken.

After seeing that the stand had the angle right and checking all the other requirements written above, it was time to develop a definitive stand with more stability, with the template sockets and the template itself. This definitive stand was also designed by the author using the same software and then 3D printed with the same 3D printer but using black PLA from (GtMax) as the material. Figures 4.4, 4.5, 4.6 and 4.7 show some measurements of the definitive stand, rendered image and the real setup ready for data collection to better represent the results.

The final design was built to support exactly the smartphone that would be used to record videos in the future. The model was an iPhone 13 Pro Max, and it was necessary to use without any smartphone case or accessory to keep the experiment more reliable and with less interference from different angles that could be caused by using a different model of case or a new accessory.

A future advance for this stand could be an adjustable slot for other models of smartphone. It was not designed due to the urgency to collect data and the necessity of a stable stand that could be printed simply and fast. As mobile parts can be an issue with stability due to the tolerance necessary to assemble the parts together and can be more complex to position and take more time to setup the environment to start collecting data.

After the 3D printing process was done, it was made a verification with the smartphone and the paper sheet to verify if everything went as planned in the test, it was possible to

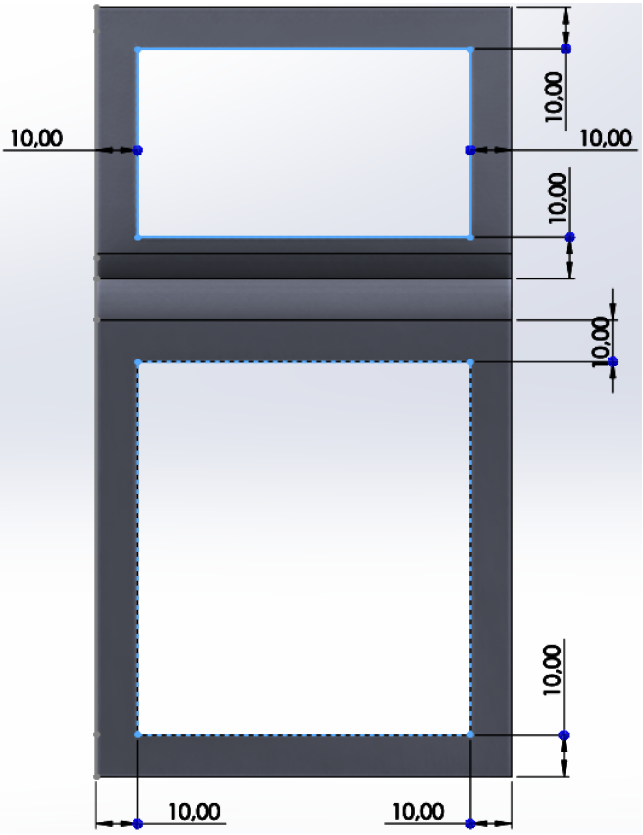


Figure 4.4 – Front view of the designed Smartphone stand.

check that the angle to capture the video was good, allowing a good and centered paper sheet on the field of view and an angle possible to capture from the pen tip while allowing the camera to detect the drawing being formed. Figure 4.8 shows the results in the field of view using the design stand.

As mentioned before, the angle could not be too high in order to capture the movement going forward and to further have an idea of the drawing in the graphs extracted from the videos. And it could not be too low to capture the pen tip without any occlusion caused by the hand or the pen itself. Figure 4.9 shows an example of how the pen and the volunteer's hand appear in the field of view.

Figure 4.9 shows how the hand and the pen tip appeared in the field of view and presented a good angle for both goals, object occlusion, and depth of the drawing. With these results, it is possible to move to the next step of the process, which was to develop a protocol to be followed during the process of video recording.

4.5 Development of the experimental protocol

The experimental protocol had the main goal of being as fast as possible to be setup and easy to replicate. Some other secondary goals were to minimize problems with pen tip

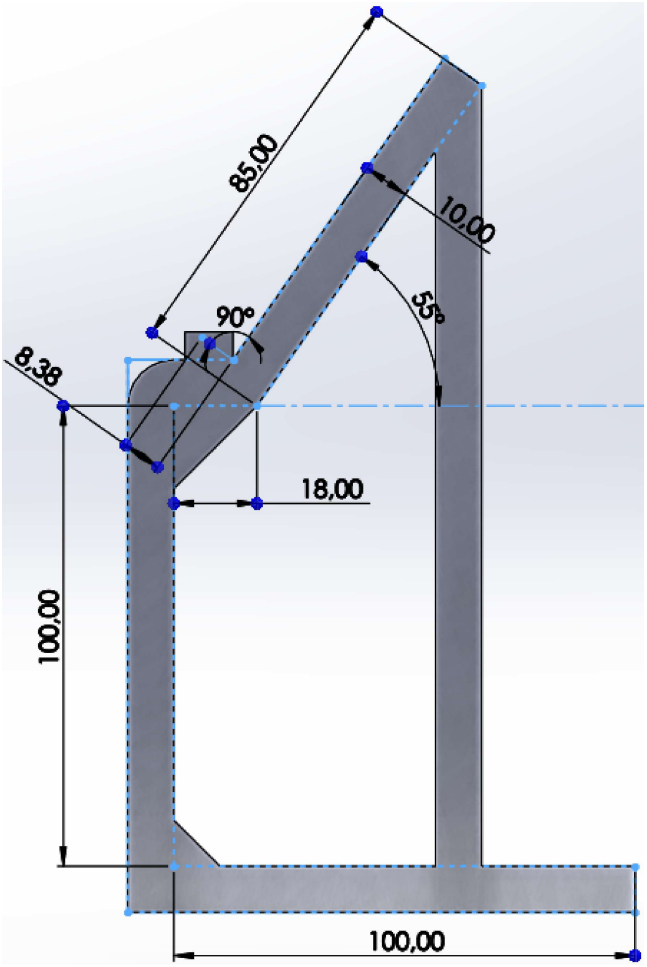


Figure 4.5 – Right view of the designed Smartphone stand.

detection, positioning problems and data loss. All the processes that will be explained in the next session were approved by CAAE number 43229921.8.0000.5152, and were entirely conducted on the specific site mentioned in the document.

To fulfill these objectives, they used a blue pen, the paper sheet, the stand, white tape, a large table and a chair without wheels. As the pen, the paper sheet and the stand have an extensive explanation, it will not be mentioned again in this section. Meanwhile, the other materials will be discussed.

To explain why using white tape for this process, it was thought about the difficulty that could be generated while drawing with one hand. In addition, there was no possibility of holding the paper sheet with the other hand. Holding the paper sheet on the other hand could result in accidents, like bumping into the smartphone stand or the paper sheet itself. This would cause data loss or misposition of all the equipment setup. To avoid this problem, tape was used to fix the paper sheet in place and allow the volunteers to draw without problems with the paper sheet moving around the table.

It was used tape in three places only, in the top right, bottom right and bottom left

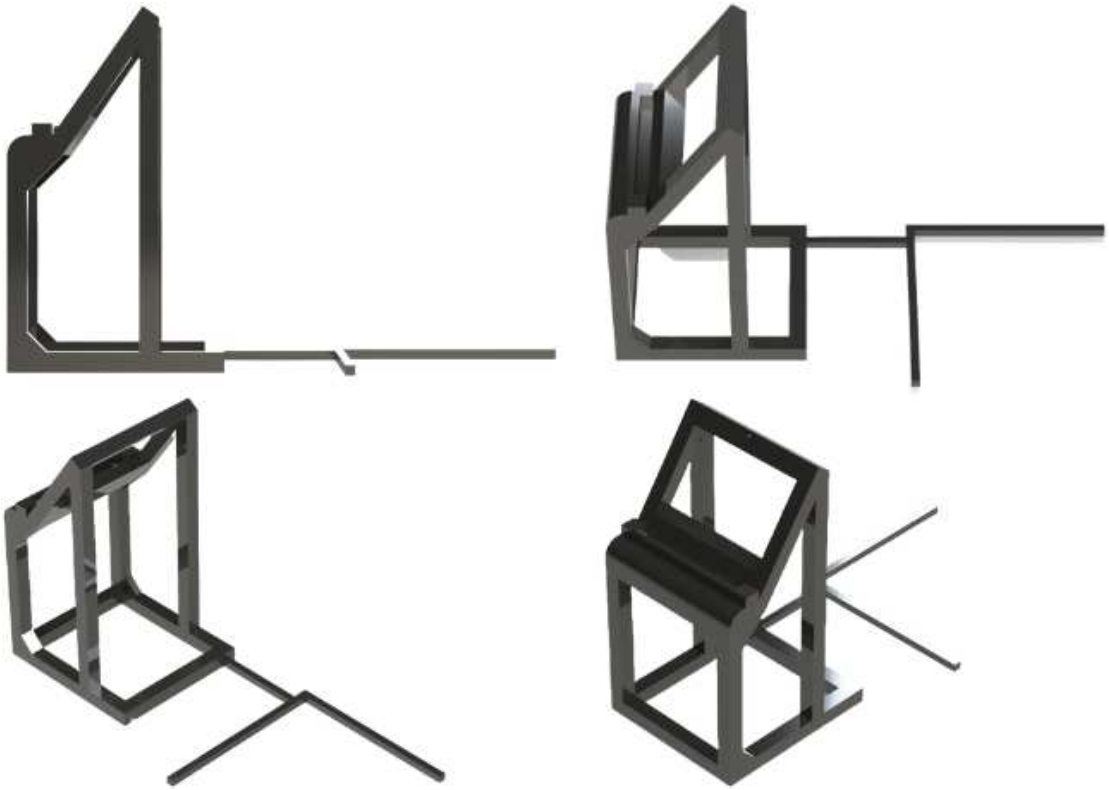


Figure 4.6 – Rendered view of the designed Smartphone stand.

corners of the paper sheet using the view of the volunteers as a reference. It was not used in the top left corner because in a simple test it causes problems with positioning. When it is used tape in this top left corner, when it is necessary to approach the template to the paper sheet, the tape can cause a mispositioning between both parts.

Later in the pilot test it was possible to verify that those three corners fixed with tape were able to hold the paper sheet in place during the drawing process. The color of the tape should be white or transparent to avoid problems later in detection using DLC software. But thinking about the speed of setup, the transparent tape could cause problems due to the difficulty to detach it from the paper, maybe causing an accident and ripping off the paper sheet. Due to that, white tape which has less adhesion was chosen to be used in the process.

The table needed to be very stable to avoid shaking during the drawing process. Thinking about the PD volunteers that might have bradykinesia and tremor, they could start to have symptoms and then the table would start to shake or move. Due to that, a large table was chosen to be used. Another characteristic necessary was to avoid a table with texture or dents, which may cause difficulties during the drawing process or even change the direction of the drawing, resulting in an unrealistic drawing. The table was also chosen by the color that should be closer to lighter colors than darker colors to have a homogeneous background considering the paper sheet and the tape. This way the

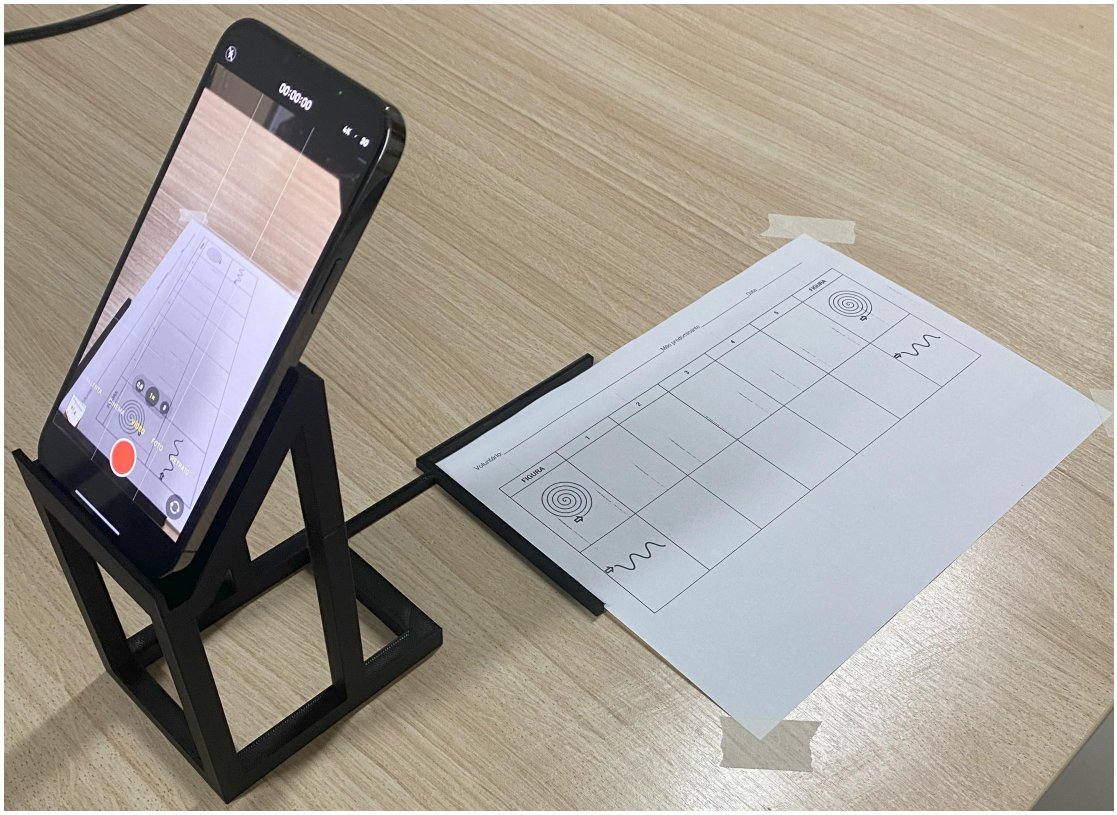


Figure 4.7 – Entire setup ready for data collection.

only two things should have colors that can distinguish in an abrupt way from the entire layout, which are the pen tip and the volunteer's hand.

To choose the chair, the main goal was to find a light chair that could be moved easily to allow the PD patients to adjust themselves into a comfortable position to start the drawing process and stay comfortable during the entire process. Other objectives were to avoid chairs with wheels, to avoid them moving without intention, or even causing an accident during the positioning process due to some difficulties the volunteers could present with mobility.

With all that it is possible to set some procedures that could be valuable to the protocol. Following this thought, it was developed a sequence to be followed with some instructions.

Before starting any positioning process, it is recommended to verify the storage available on the smartphone that will be used in the process. Verify if the video is set to capture images at resolution of 3840 x 2160 pixels at 30 frames per second. Then it is necessary to check the smartphone´s lens.

The lens can be an important part of the equipment due to problems that might occur with them. They can have scratch or dirt, which might cause trouble during the video recording process, like a blur on the image which can lead to a problem with object tracking in the future. With everything set on the smartphone, it is necessary to move

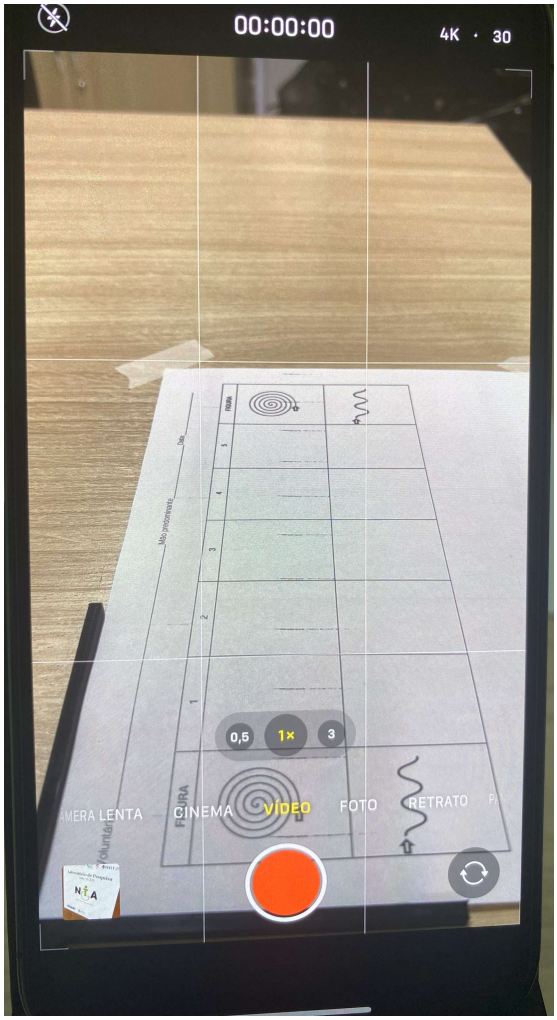


Figure 4.8 – Field of view of the camera using the designed stand.

forward with the process.

The next step is to position the paper sheet. The paper should be at a reasonable distance from the patient, but not too far from causing problems during the drawing process. It is desirable that in the camera's field of view only the table appears, avoiding other objects. After choosing the place that the paper sheet will be, it is just necessary to fix it with three pieces of tape as mentioned above.

After positioning the paper sheet, it is time to put the smartphone on the stand without any accessories or protection case. It is recommended that the smartphone is ready to record. After this, it is necessary to approach the stand with the template well connected to the stand and approach the template to the paper sheet in the top left corner of the paper sheet.

Once this is complete, almost the entire process is ready. Now it is just necessary to ask the patient to sit in the chair and position himself to start drawing. After the patient is sitting and ready to start, it is recommended for a smartphone that has a feature

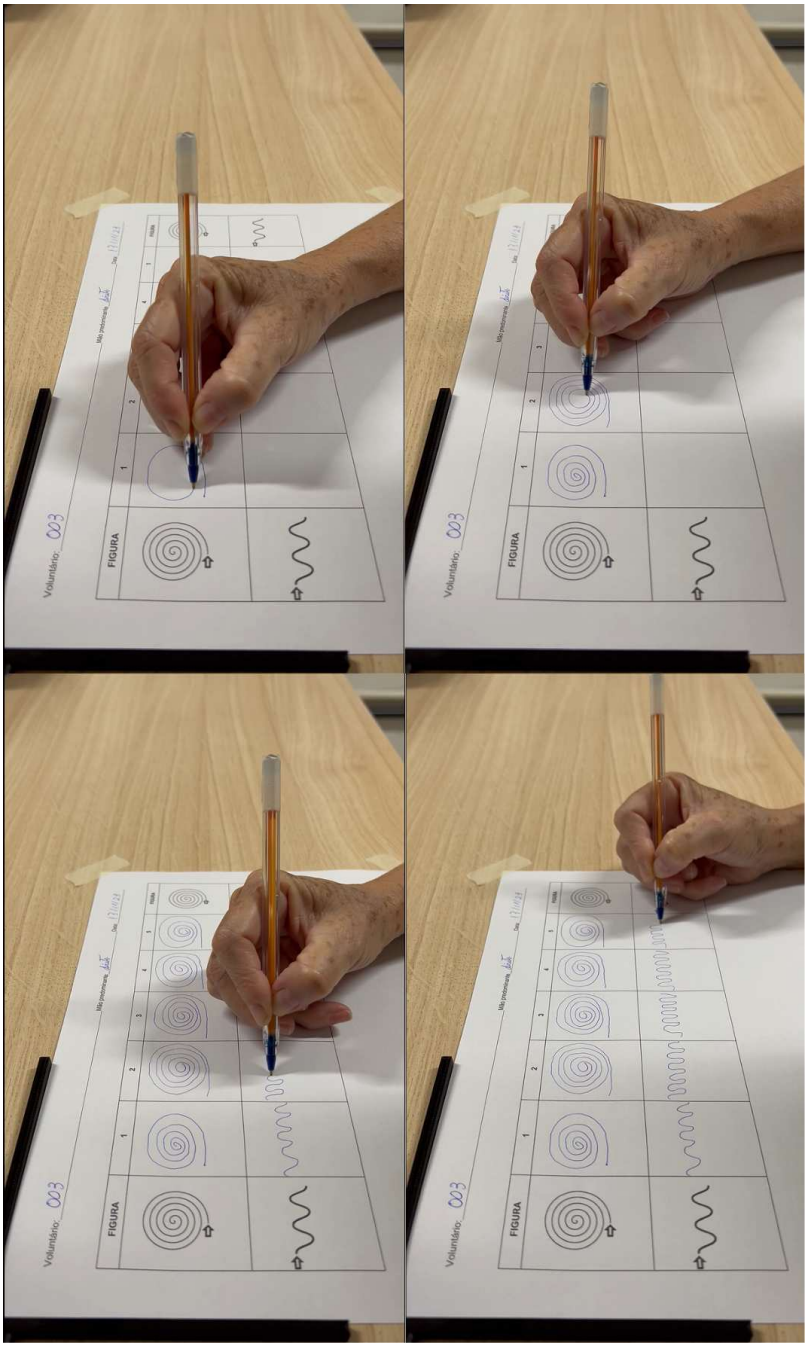


Figure 4.9 – Pen and volunteer's hand trough the field of view of the camera.

that blocks the focus of the camera, to use it in the center of the camera and to block this focus. This is desirable due to some features of smartphones that are constantly trying to recognize humans and animals during video recording. As the video just has a hand appearing in the field of view, the camera will focus on the hand, and it can cause quality loss during video recording. This problem can be caused by the size of the pen tip compared with the rest of the objects in the field of view including the hand of the volunteer.

To develop a more comprehensive and more accurate data set, the patients were asked to perform the test twelve times. This was chosen to avoid problems that could happen in the PD group. As they could have different responses over days or weeks, twelve trials could get information on many stages of the patient's development during a short period of time. A good point to pay attention to was that the patients did not draw more than two sessions in a week, more than two sessions could only be conducted under extreme necessity of the patients, like problems scheduling due to medical appointments or similar. As the study worked with ten volunteers with Parkinson's disease disease and ten healthy participants ages and sex were matched, we could achieve a total of two hundred and forty videos to analyze.

4.5.1 Basic instructions to the patient to perform the data collection

Some tips to perform better data collection are to first explain exactly what the patient must do. This includes how to hold the pen, what it is necessary to avoid, and what is not allowed to be done during the process.

Firstly, it is recommended to explain to the patient that they will perform ten drawings in total, of which five of them will be sinus waves and five will be spirals. The drawing must occur in sequence and in one row. It is not necessary to stop between drawings. It is necessary to explain that the spiral has the right way to draw, which is from the outside to the inside.

After that it is necessary to specify that the patient must draw everything as fast as possible and in the best quality they can. Some patients could take several minutes to draw interested in pursuing to do the best drawing they can. Due to that a time limit was set at two minutes for all the ten drawings and it must be explained to the volunteers.

To perform the drawing, volunteers need to be oriented so as not to move the hand in an abrupt way. Movements with high acceleration will be recorded but will result in blurred images which will cause problems with the detection using the DLC software. So the drawing process can be fast, but not have high acceleration. The only time that does not matter if the acceleration is high is during the transition between the last spiral and the first sinus wave.

Another request that must be made to the patient is to prevent removing the hand or the pen from the field of view of the camera. To avoid that, the instructions must be to avoid taking the pen too high off the paper sheet, or just to maintain the hand on the paper sheet during the process. It is necessary due to the loss of the pen tip when the patient removes it from the field of view. This will result in a blank spot on the graph and will cause a decrease in the accuracy of the process.

The last instruction for the volunteers is to hold the pen without covering the pen tip.

It is desirable that the patient holds the pen a little bit higher due to some movements during the drawing. These movements can make the finger be in front of the pen tip for a short time, or even long nails.

4.5.2 Basic instructions to ensure a good data collection

Always check if the lights in the room are proper, if there is a position that avoids shadows on the hands of the volunteers and the pen tip. It is necessary to have good light in the room to avoid problems with the object tracking later.

Another objective is to verify if in the room or at least in the field of view of the camera there is not an object or paint with a blue color or close to being blue. The whiter the ambient the better. This objective includes checking painted nails and accessories like rings and bracelets on a blue color scale.

To avoid problems with data loss, it is advisable to give a command to the volunteer to start the drawing process after 2 seconds of video or more. And ensuring the same after the drawing process is complete. It is also advisable that the ambient be quiet and just instructions must be given to the volunteers. This is necessary to avoid the volunteer getting distracted and taking longer to draw or even losing where they stop during the drawing process causing a necessity to restart the session.

It is necessary to pay attention during data collection to avoid the patient moving the hand with high acceleration, removing the hand and the pen from the field of view of the camera, if they occluded the pen tip, or if the volunteers lost the sequence or any other problem they might have. If many issues were presented in the same video recording, it is advisable to restart the sequence in order to obtain a better and more reliable result.

During the process the applicants must be aware of the volunteer using the free hand to hold the paper sheet or even getting their hand close to the stand, due to the risk of the volunteer bumping into the stand or the paper causing shaking which can lead to data loss. Not only that but also be aware that some patients pursuing better quality of the drawing might get their body closer to the paper sheet which usually approaches the head to the paper and might get their heads in front of the camera.

Special attention to the pen must be taken. This model of pen has a different color on the tip (blue), but it also has another part on the opposite side of it that has the same color. This tiny part must be removed or occluded by using white tape or something similar, in order to avoid problems with detection. It was observed that the pre-trained Neural Network assumed that color was one of the most important features to detect the desired object (pen tip). It was observed that in some frames of videos containing the blue part on the top caused confusion in the system that chose which part to be the result. It was not detected in all the video, but in a few frames. Figure 4.10 shows the model of the pen, and the part mentioned above.

Figure 4.10 – Blue pen used to train the Neural Network and collect data.

Putting all these information together it is possible to collect good videos which will lead to good results in processing using DLC software. Allowing then to use the video information in the second process to classify the volunteers between the two groups mentioned above.

4.6 Storing and editing the videos to be processed on DeepLabCut

After having the video on the smartphone which should be done using the steps mentioned above, it is necessary to get the video ready to be processed.

To process the video, there was used software that allowed the edition of the video track and the audio track in a simple way. The CapCup software was used because of its simplicity to work with and because it has all the necessary features for free. It is only used for the trim, separate and delete tools in the software.

To start the process, it is necessary to transfer the video from the smartphone to a laptop. To do this transfer it is necessary to use any means that does not lose data or compress the video. Due to that all the videos were transferred using the USB port and stored in specific folders.

A good way to store all the videos was creating a bunch of folders due to the creation of several archives later using the DLC. For that in this case, there was used a main fold

called “Data Collection”. Inside this main fold four more folds were created to divide the participants into four groups following the data collection process. Due to that it was available in folders named “Group 1,2,3 and 4”. Inside the group folders, other folders were created for each volunteer.

Then inside each volunteer’s folder another twelve folders were created for each volunteer’s video indicating what session was collected. This was done due to the process with the DeepLabCut creating three archives and a folder and they are created automatically and with the same name. To avoid problems and data loss, it was smarter to create these individual folders and have them all separated and easier to handle.

After having all the folders, the video was transferred to the specific volunteer’s folder with the specific session labelled and the names were changed to avoid data loss, to have better organization and to have an easier identification. The name follows the sequence of the identification number of the patient, then the identification of what session was done, and then the day it was conducted, and finally it was labelled “original”. It was done due to processing on CapCut that will create a second video that is labelled with all the same instructions mentioned before changing only the “original” to “edt” to certify that this video was the processed one.

To process the video in CapCut, it is only necessary to import the desired video and start trimming the video into the exact parts. Figure 4.11 shows how the layout must look like when it is ready to be edited.

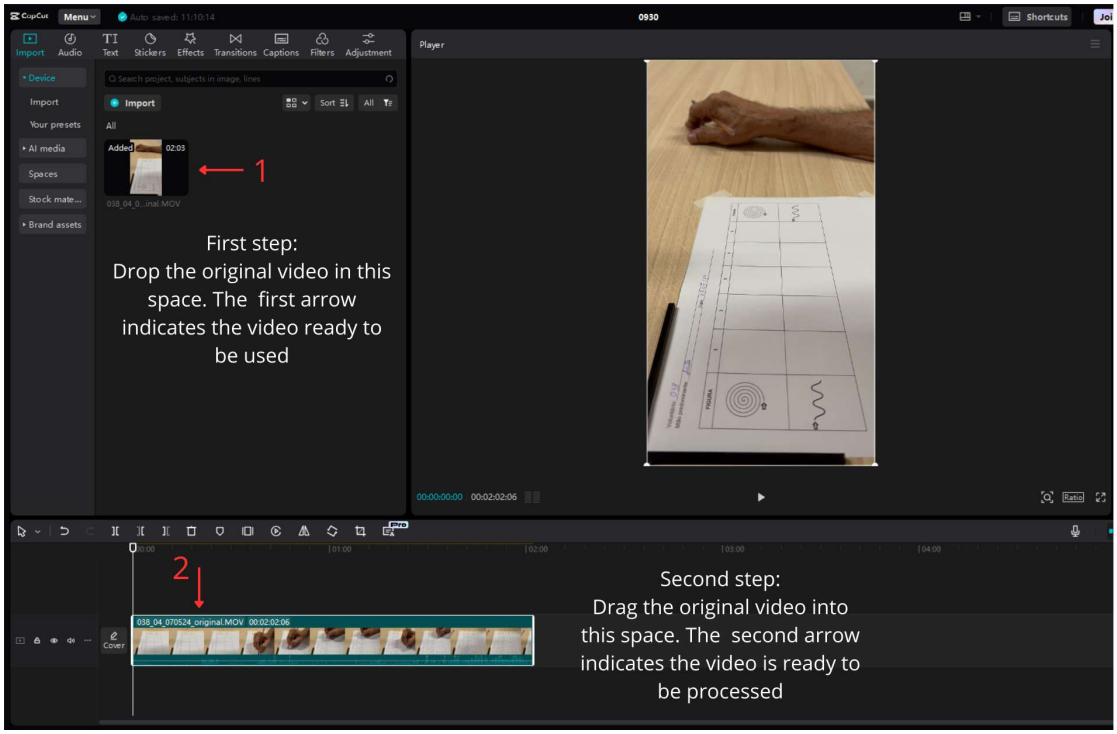


Figure 4.11 – CapCut layout example.

When the video is ready to be processed it is time to trim the necessary parts. To

avoid problems or forgetfulness, it is advisable to mute the soundtrack first. It is done to preserve the privacy of volunteers and to remove unnecessary data for the processing of DLC.

After muting the sound track it is necessary to trim the beginning of the video. The sync must be done between the first movement of the volunteer's hand to start the drawing and the first frame of the video. To do this, it is necessary to expand the video track to see frame by frame of the video and choose the exact frame that the drawing movement starts. Figure 4.12 shows an example of how the layout must be in this scenario.

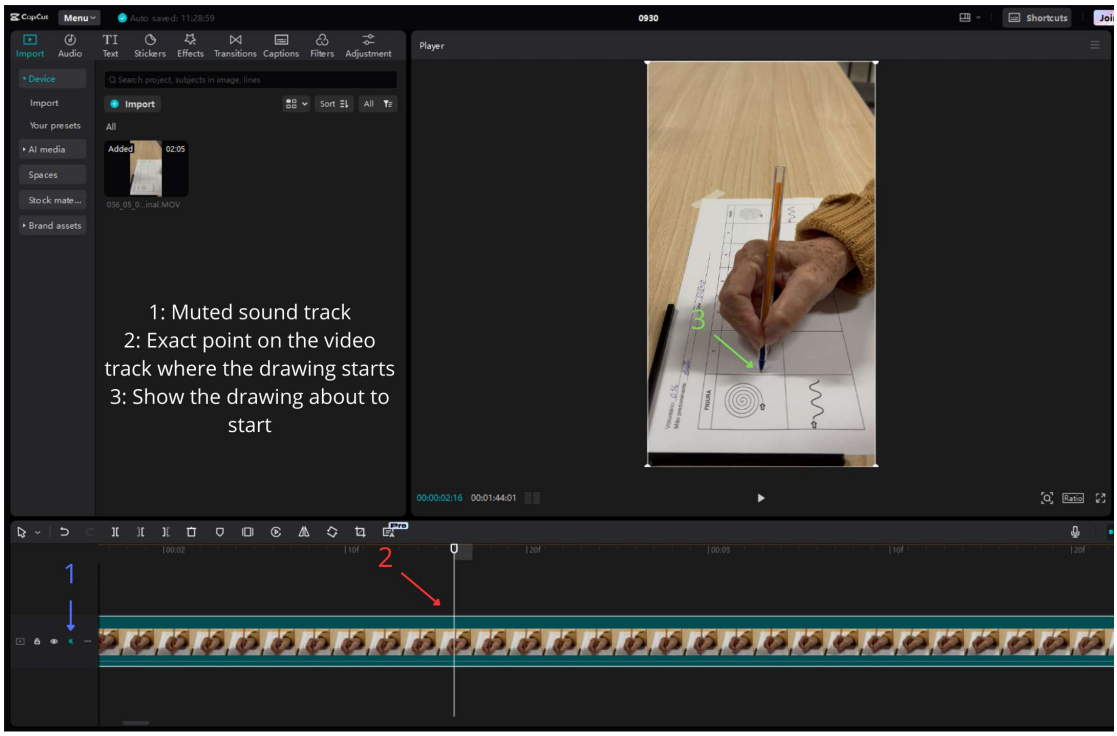


Figure 4.12 – Steps to synchronize the beginning of the video.

After removing the seconds before the drawing starts, it is necessary to remove the section between the last spiral drawing and the first sinus wave. This part is unnecessary for this study. This part was chosen to be deleted because each participant reacts in a different way to moving from a drawing to another drawing. It was noticed that between spirals the movement tends to be good, and the volunteers barely move the hand out of the field of view. Although, in this section of the process, most participants removed the hand from the field of view, which causes the loss of the pen tip in the tracking process, and most of them accelerated their hands too fast, resulting in blurring images.

Due to that, it is necessary to remove this part, and the process is similar to the previous process. This time, first it is necessary to split the video into two tracks. The split must be done at the last movement done in the last spiral drawing, then move the cursor to the first movement of the first sinus wave and delete the part that is behind the

cursor. This will result in two video tracks but without the middle section which was not necessary. Figure 4.13 shows what the process looks like.

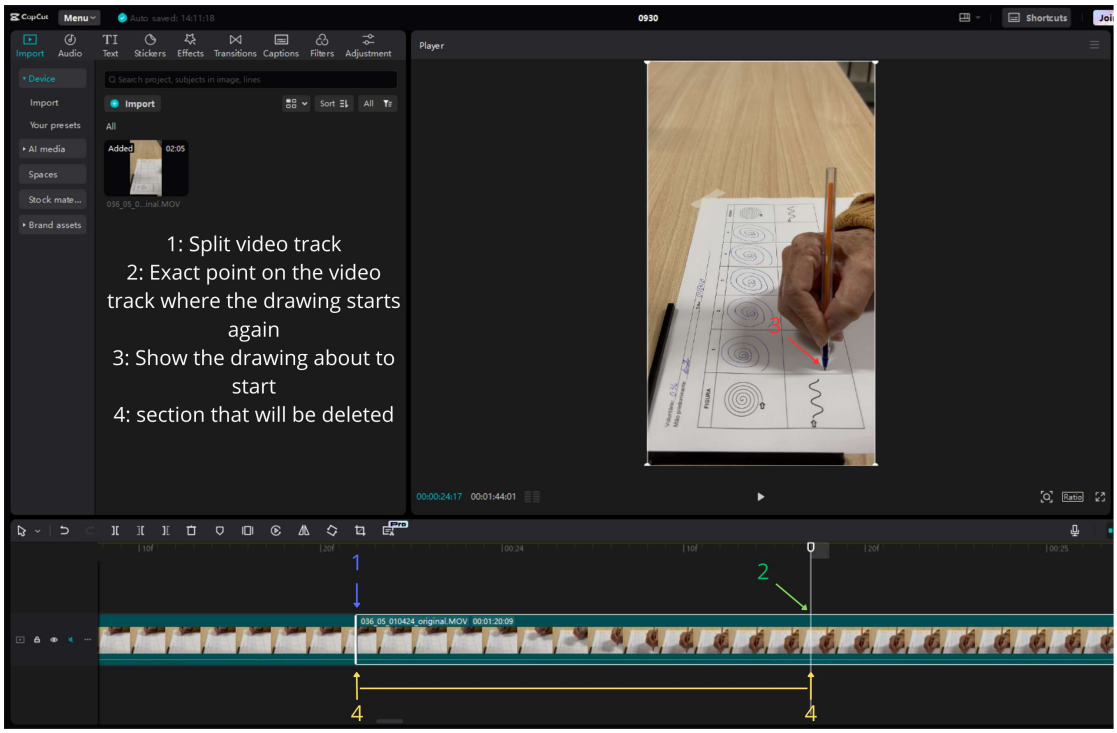


Figure 4.13 – Steps to synchronize the middle session of the video.

After removing the middle section, it is necessary to remove the last part of the video, like was done at the beginning of the video. This time the video must end together with the last movement of the pen while drawing the last sinus wave. Notice that using the software CapCut in the actual version 4.6.0 and some previous versions. While doing this process, it is necessary to find the exact frame with the last movement and move to the next frame. This is necessary due to some features in the software that removes the last frame before the cursor. Figure 4.14 shows how the layout should look like.

After doing all this, it is just necessary to select to export the video to the right folder of the patient's trial. It is necessary to choose the name of the video, which must be the same as the original video but changing the "original" to "edt" to certify that the video is ready to be processed on DLC. It is also necessary to select the resolution of the video.

In this case the resolution was downgraded to full HD quality which represents a resolution of 1920 x 1080 pixels. This was done as mentioned before to allow standard machines to process videos without problems or crashes. This also allowed the machines to process faster. A test was done trying to figure out if using a video in 2048 x 1080 instead of full HD quality would impact on the results of the precision in the object tracking process, and the result was the same for both resolutions. A test with 3840 x 2160 pixels could not be done due to the machines in the laboratory could not process such a dense video. In every trial at this resolution the entire system crashed.

Due to that the resolution is downgraded and the next configuration can be maintained as default. In this case, the “bit rate” was maintained as “recommended”, the “Codec” was maintained as “AV1”, “format” as “MP4” and “frame rate” as “30pfs” as well as the original video. With all of those configurations set, it is just necessary to confirm the exportation process and use this video in the DLC software.

An observation about why to record the video with 3840 x 2160 pixels of resolution is that by now the study was limited using a standard machine, but in the future, better machines could be acquired by the laboratory or even a collaboration with others could be done and use those videos again for the same reason or for other studies.

4.7 Processing the videos on DeepLabCut

Now it is time to use the first Machine Learning algorithm and extract the first amount of data from the videos recorded.

To use the DLC it is necessary to open navigator software called “Anaconda”. This software allows you to install and use the DLC with less effort and complications. It is advisable for unexperienced users of programming languages.

With the Anaconda software open, it is necessary to run the “CMD.exe prompt” and write two lines to activate the DLC. Those lines of code can vary due to the installation process and the names that were chosen during installation. Figure 4.15 shows the line

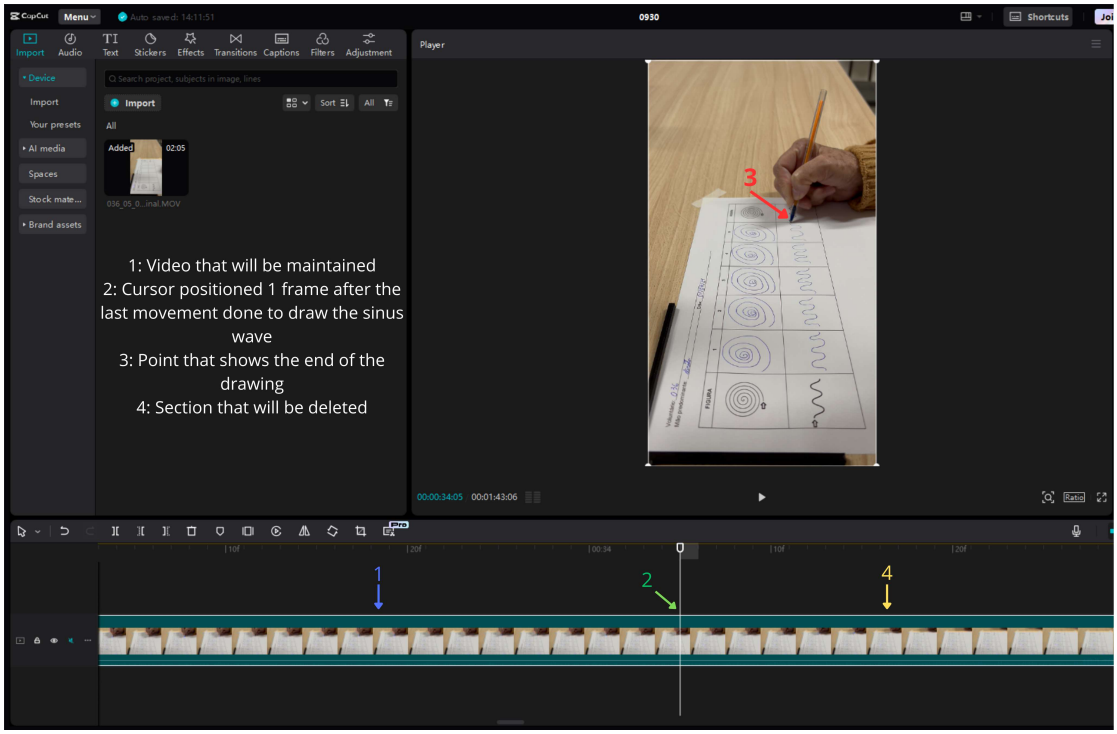


Figure 4.14 – Steps to synchronize the final session of the video.

codes necessary to run the DeepLabCut.

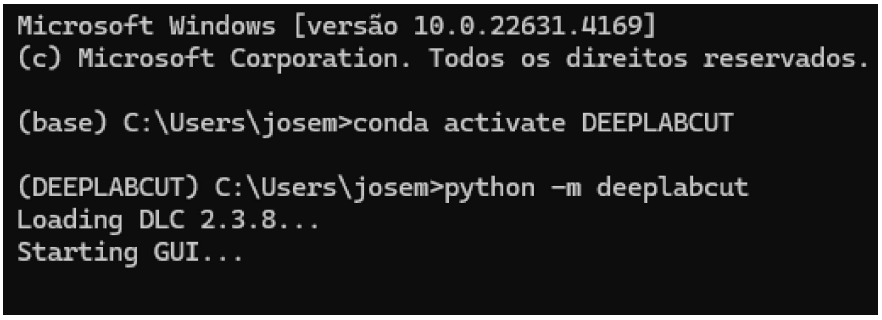


Figure 4.15 – DeepLabCut initializing Command Prompt layout.

The line that can change is the "conda activate DEEPLABCUT", the part that can change is the "DEEPLABCUT" which is the name of the environment created before the DLC installation.

After these two commands are given, an interface should appear on the screen allowing the user to load the pre-trained Neural Network. Figure 4.16 shows the first screen of the DLC software.

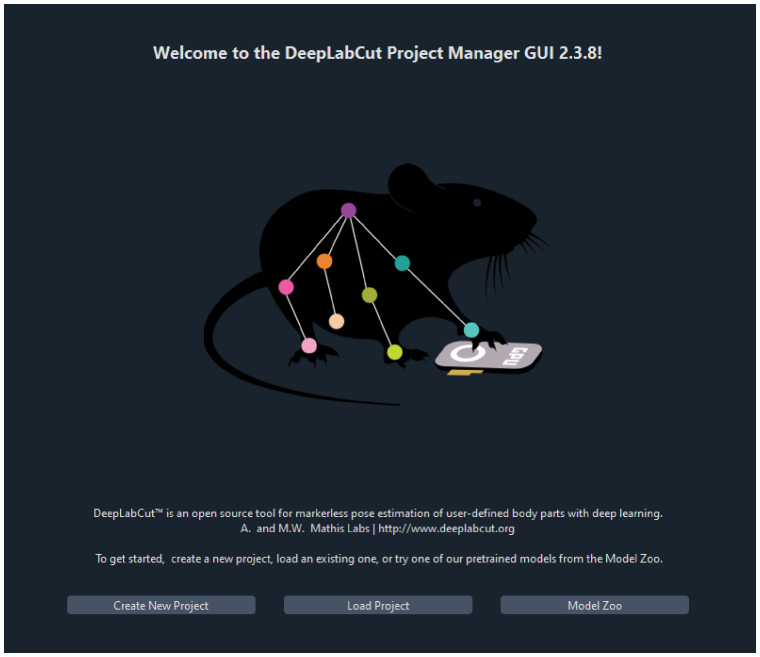


Figure 4.16 – DeepLabCut home screen and process.

After loading the project with the correct file which is named automatically by the DLC as "config.yaml" it is possible to move to the tab which contains the information needed to analyze the videos. In this tab it is possible to see a few configurations necessary to be set to start the process. Figure 4.17 shows the configurations mentioned above.

The following steps were mentioned in Figure 4.17 in the exact order that should be to maintain a good workflow. First it is necessary to select the video which will be processed.

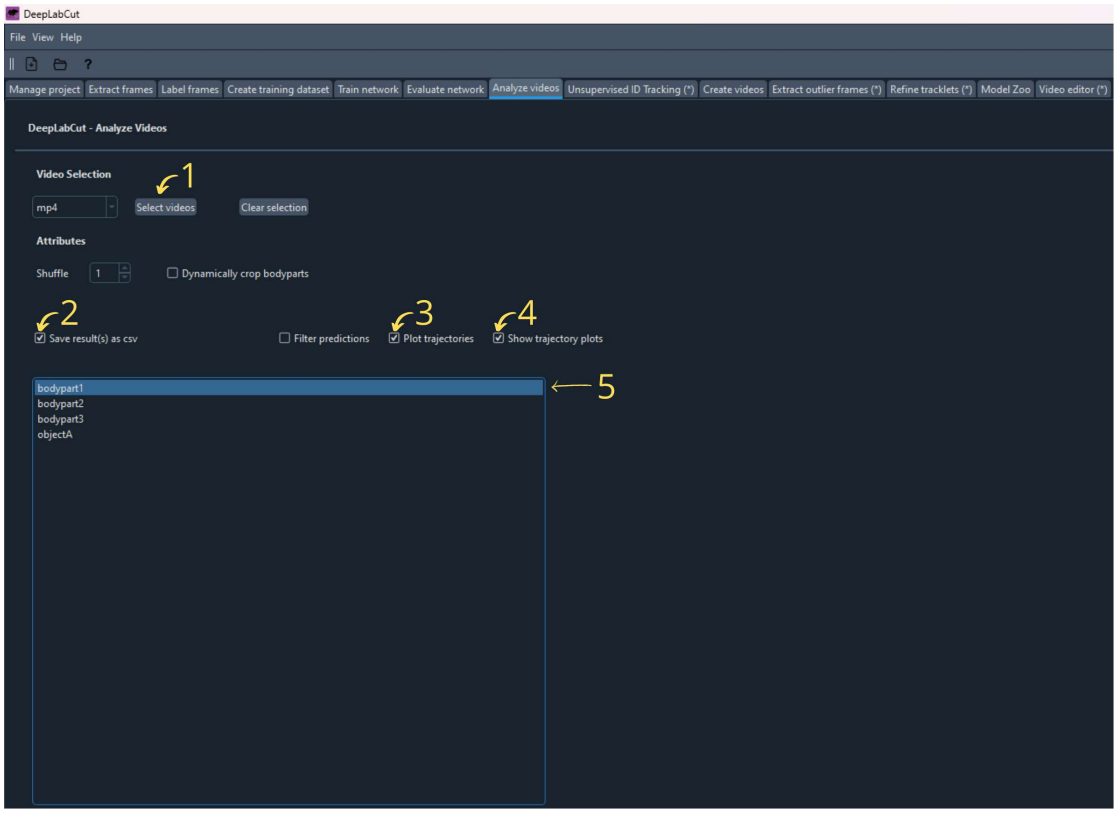


Figure 4.17 – DeepLabCut Analyze videos screen.

This is cited in the step first step “Select the video” button. It is advisable to process one video at a time, but it is possible to do more than one if they are stored in the same folder. As folders were created to generate individual results, the software allows only one video at a time. This is also good to check the results later and to be sure of what video those graphs belong to, avoiding mistakes while dealing with a big amount of data.

Then it is necessary to check the box to create a “.csv” file due to feature extraction that is done before the process of the DLC. The software automatically creates “.H5” and “.pickle” file, but due to the practicality of a “.csv” file it was chosen to work with it. Later it is necessary to select “Plot trajectories” to create graphs that show how object tracking is performed. This includes four graphs, but only three are interesting for checking what happens during the object tracking process. The first presents the precision the software achieved, the second with the “X” and “Y” pixels positions that the aimed part went through in the field of view of the camera and finally the graph containing these pixels plotted like the field of view of the camera, which will form a representation of the drawings. Figure 4.18 shows an example of each graph mentioned above.

After the graphs are plotted, it is necessary to look for some patterns that were observed during the first group of volunteers. To verify if the process had no error, the graphs with the “X” and “Y” positions cannot have a break in the lines. Those lines must be continuous, if any break appears on the graph, it can be due to some loss in the

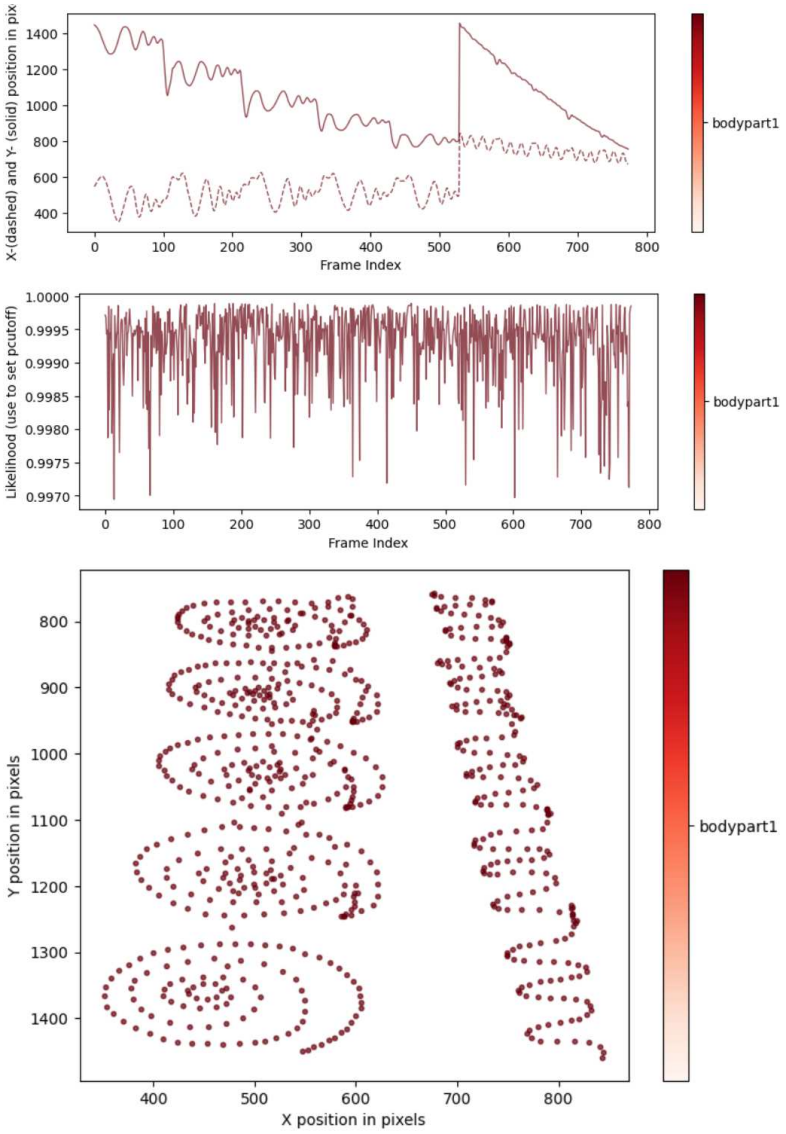


Figure 4.18 – Important graphs generated by DeepLabCut after analyzing the video.

process. This might be caused by multiple factors, but mainly factors associated with object occlusion, excessive accelerations, objects of the same color appearing in the field of view or the aimed object being removed from the field of view.

The other graph that must be analyzed is the one that contains the likelihood, which points to the precision that was achieved by the software. If a large peak in the downward direction occurred on the graph it might indicate that a loss occurred during the tracking process. Due to that those two graphs can affirm what happened during the process and in what part of the process it occurred. Taking this into account, it is necessary to always verify what happens in the process, in order to have the best data quality possible.

If some peaks occur like mentioned above, it is advisable to follow the process to the "create videos" tab which will create a video with a dot in any color desired to show where the software detected the aimed object. In this video, the dots on the object tracked, allow

inspection to be easier and to identify the cause of the data loss. In this scenario, it is advisable to use a video reproducer with capabilities that allow the video to be passed frame by frame or very slow due to some failures that occur just in a few frames of the video and might be difficult to be seen at normal speed. Figure 4.19 shows the "Create videos".

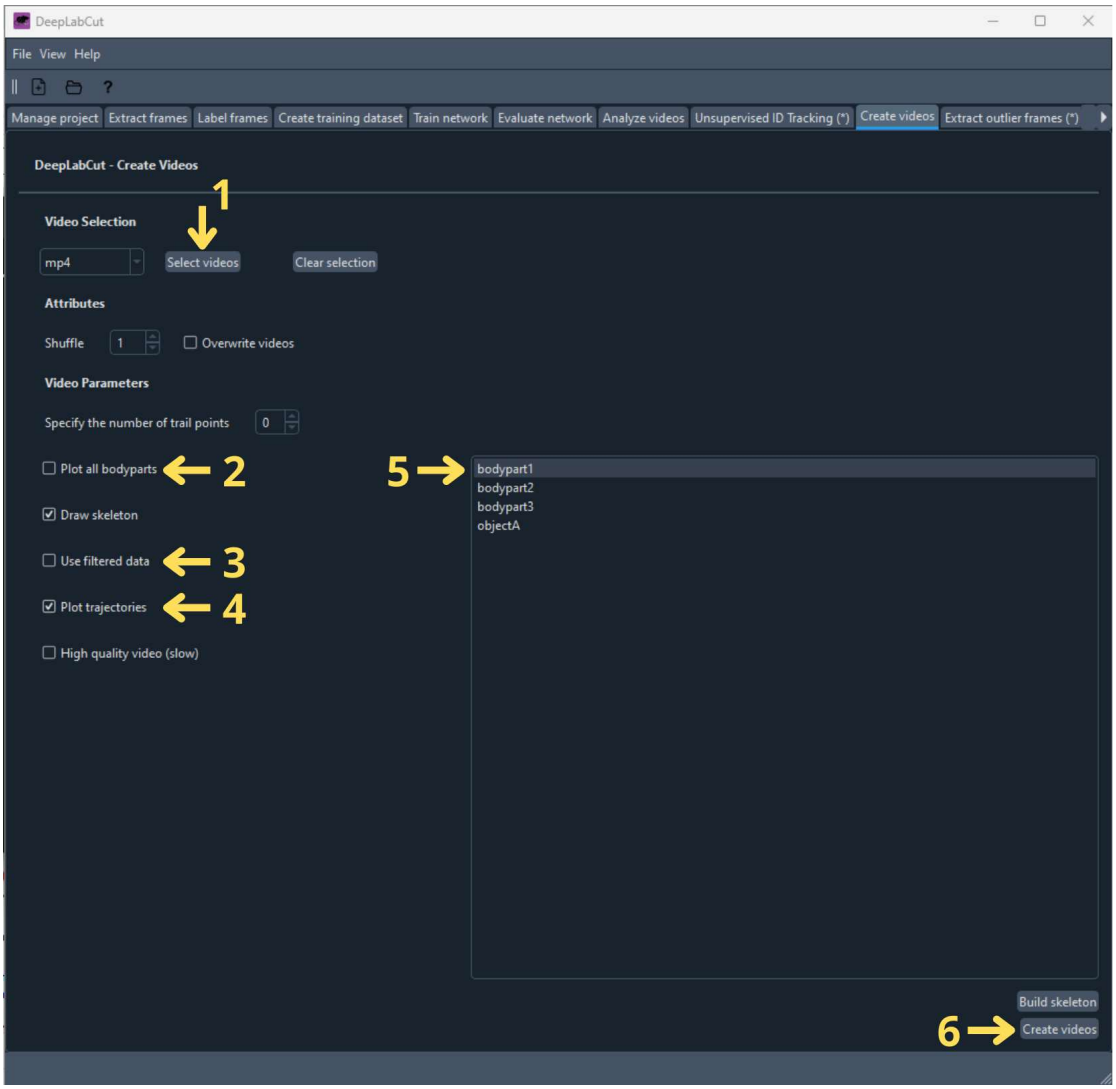


Figure 4.19 – DeepLabCut Create videos screen and process.

In this tab some configurations must be used in order to create a video. First step is to choose the video that was just analyzed and select the format of the video. Then it is necessary to mark some option boxes below. The box "Plot all body parts" in this study is necessary to be unmarked, because while training the neural network, the default parameters of body parts where maintained which creates three body parts possible and an object. None of the other body parts are a problem due to that it was not labeled in the training session. After unmarking the box, the "Use filtered data" must be unmarked, otherwise it will not work.

It will not work because in the analyze session, it was not a marker to avoid the system trying to smooth the pattern or something close to that, in our study the result must be raw to allow the other Machine Learning to capture what really happened. Using a smooth during the process could hide some characteristics of the patients.

After this box was unmarked, it is necessary to mark the box “Plot trajectories” to allow the system to mark the object with a dot, which is the goal of this step in creating the video. Then, it is necessary to mark the body part that must be plotted, in this case it is body part1. Finally, the button to create the video must be pressed. Figure 4.20 shows how the command prompt should look like while processing it.

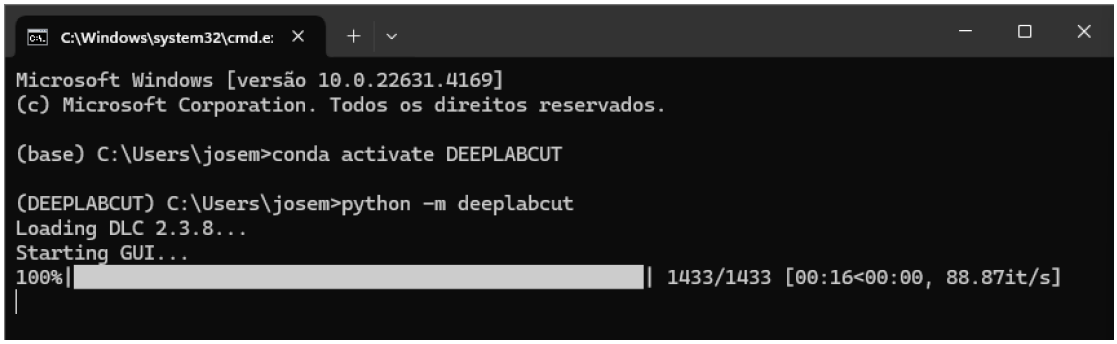


Figure 4.20 – DeepLabCut Command Prompt while creating a video.

The command to create the video is very fast, but unlikely on the “Analyze videos” this process does not exhibit output automatically. It is necessary to navigate to the folder manually and open the video. If the video reproducer is not able to reproduce the video as slow as necessary to detect the object tracking failure, the software CapCut could be used like it was done in the process to prepare the video to be analyzed in section 4.6. To use CapCut or other software, it is advisable to make a simple calculation to find the probable time that the error occurred. Using the approximate frame that the error was pointed at in the graph and dividing it by the quantity of frames per second used to record the video. Assuming the error occurred in 1.000 frames of the video and the video was recorded at 30 frames per second, the error occurred close to 33 seconds in the video.

The figure 4.21 shows an example of the dot while the tracking is working and the absence of the dot when the tracking fails.

In this case the pen tip could not be tracked due to object occlusion caused by the nails of the volunteer. And in this trial, the loss occurred for just one frame, which would be barely possible to be visible in normal speed reproduction. After doing an examination if the problem persists for too long, and classifying if it is necessary to do a retake or not, the next step in the workflow is to move to feature extraction.

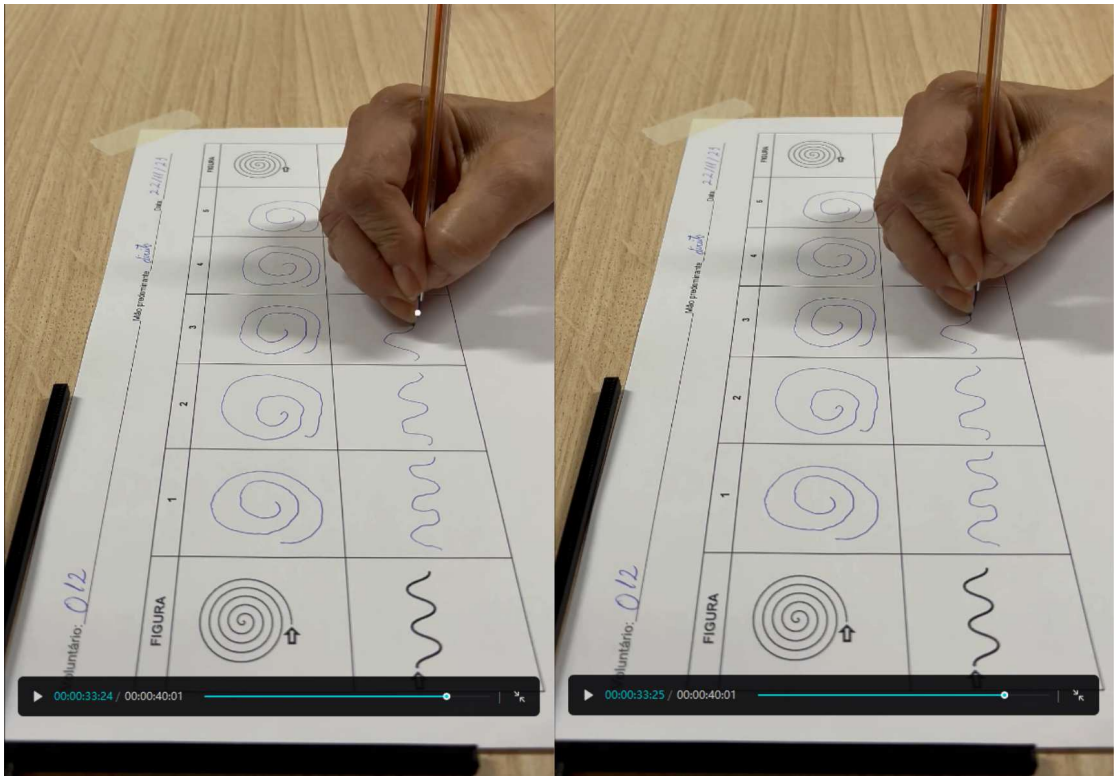


Figure 4.21 – DeepLabCut Create videos screen and process.

4.8 Signal Processing

Once the data had been processed in DLC and properly archived, the datasets were reorganized to optimize feature extraction. This involve converting CSV files into JSON format, editing them to retain only three essential columns (X and Y pixels positions and time in milliseconds), and removing redundant information. The operation was performed in Rstudio using pre-existing code crafted by one of the researchers and validated in earlier studies.

The displacement signals underwent a filtering process aimed at attenuating noise across all frequencies. A 4th-order Butterworth band-pass filter, configured with cutoff thresholds between 1 Hz and 16 Hz, was utilized. Parameters aligned with standard practices in studies examining kinematic movement. After applying the filtering methodology, signal features were systematically extracted from the X and Y directional axes [64, 65]. These included measurements pertaining to amplitude, entropy, frequency domain characteristics, displacement, and statistical attributes, all detailed in Table 4.2.

The amplitude features, computed within the time domain, represent the signal's magnitude and periodic fluctuations, which in turn mirror the extent of movement displayed by the object's position throughout the drawing activity. Frequency-related features quantify the recurrence of events within designated time intervals and are assessed through time-domain calculations. They also incorporate the analysis of energy distribution across

different frequencies in the frequency domain [65, 66].

Entropy-based features measure the level of unpredictability embedded in the dataset, providing essential information about the system's complexity and facilitating the assessment of uncertainty across sampled segments. Statistical descriptors were utilized to analyze the underlying distribution of the data points and to reveal the extent of variability present in the dataset. Displacement features were computed to capture and evaluate the overall motion trajectory of the object's (pen tip's) movement during the course of the experimental protocol [65, 66, 67].

The dataset was subsequently structured into a tabular format comprising 42 extracted features per axis, alongside additional columns representing participants' age and group classification (assigned as 1 for the Parkinson's disease (PD) group and 0 for the Healthy Control (HC)). This resulted in a total of 85 features utilized during the classification stage, in addition to the target variable—group—which indicated each participant's clinical condition within the dataset.

Following the data organization, the dataset was partitioned into two distinct subsets: a training set and a testing set, corresponding to about 80% and 20% of the entire data, respectively. This division is crucial to ensure that model training and evaluation occur on independent data, thereby mitigating potential biases inherited from the initial data processing stages.

The final step involved assessing the importance of each feature, aiming to determine the extent to which each variable influenced the model's classification outcomes and to highlight the key predictors driving the system's performance.

4.9 Pre-classification Data Processing

Subsequent to the initial evaluation of all models using the training dataset, a cross-validation procedure was implemented with varying numbers of folds to determine the configuration that yielded the most optimal performance. The models were subjected to systematic validation across multiple data partitions, ensuring a comprehensive assessment of their generalization capabilities under diverse conditions [11, 67].

Cross-validation is a widely used technique in data processing and Machine Learning for assessing a model's ability to generalize to unseen data. Rather than relying on a single train-test split, cross-validation partitions the dataset into multiple subsets to produce a more robust evaluation. The most common form, k-fold cross-validation, divides the data into k equal-sized folds. The model is trained on k–1 of these folds and validated on the remaining folds, with this process repeated k times so that each fold serves as the validation set once. The performance metrics from each iteration are then averaged to provide a comprehensive estimate of the model's accuracy. This approach helps reduce bias and variance, leading to a more reliable assessment of model performance, particularly

Group	Abreviation	Description
Amplitude	MAV	Mean Absolute Value
	MAVFD	Mean Absolute Value of the First Difference
	MAVSD	Mean Absolute Value of the Second Difference
	RMS	Root Mean Square
	PEAK	Maximum value of the vector, considering only positive values
	ZC	Zero Crossing
Frequency	FMean	Mean Frequency
	FPeak	Frequency with the maximum power
	F50	Median Frequency
	F80	Sum of the power of frequencies below 80 % of total energy
	P3.5-7.5	Power in frequency band 3.5–7.5 Hz
Entropy	EnFuzzy	Fuzzy Entropy
	EnAp	Approximate entropy
Statistical	VAR	Variance
	RANGE	Amplitude Range
	INTQ	Interquartile Range
	SKEW	Asymmetry
	KURTOSIS	Flattening of the curve
Displacement	VMean	Mean Velocity
	VMax	Max Velocity
	DTotal	Total displacement

Table 4.2 – Features extracted from the gathered data

when dealing with limited data [68, 69].

In accordance with the methodology described in [70] the dataset was partitioned into two distinct subsets, comprising 80% for model training and 20% for performance evaluation. This stratification mirrors the data division protocols employed in prior studies involving cohorts of patients with Parkinson's disease [70, 71]. Such a delineation is critical to guarantee that the algorithm is exposed to non-overlapping datasets during the training and testing phases, thereby mitigating any risk of data leakage and minimizing bias originating at earlier stages of the modeling pipeline.

The training dataset was utilized for both model development and performance assessment through cross validation. A k-fold cross-validation strategy was adopted, with the number of folds systematically varied from two to ten to ensure a thorough and robust evaluation. This iterative testing enabled a comprehensive analysis of model stability across different data partitions. Upon completion of these evaluations, it was determined that employing seven folds with the feature selection enable yielded the most optimal results for the training dataset. In contrast, the test dataset was reserved exclusively to assess the model's generalization capability on previously unseen data.

The feature selection (FS) procedure was initially applied to the dataset's 84 variables (one variable discarded "indivíduo") to construct a more streamlined and computationally efficient model. Two distinct FS methodologies were employed: the Remove_Multicollinearity technique, which discards features exhibiting strong inter correlations in order to reduce redundancy and enhance the interpretability of the model [70, 72];and the feature selection (FS) module provided by the PyCaret library, which implements a classical selection strategy utilizing the LightGBM estimator as its underlying mechanism [55, 73].

In this study, feature selection was employed as a fundamental preprocessing step within the data analysis pipeline, aimed at identifying the most informative and relevant features associated with the classification task. The integration of FS techniques into the classification workflows enabled the generation of refined feature subsets, each subsequently evaluated using a variety of classification algorithms. This approach facilitated a comparative analysis to determine the optimal number of features by assessing their individual and collective impact on improving classifier performance.

Beyond its role in model optimization, FS provides multiple practical and theoretical advantages. These include reducing the complexity and duration of future data collection processes, enhancing the interpretability of results—particularly in relation to underlying disease mechanisms—and minimizing computational overhead. Crucially, the appropriate selection of predictive variables through FS can substantially improve the accuracy and generalizability of classification models [74, 75].

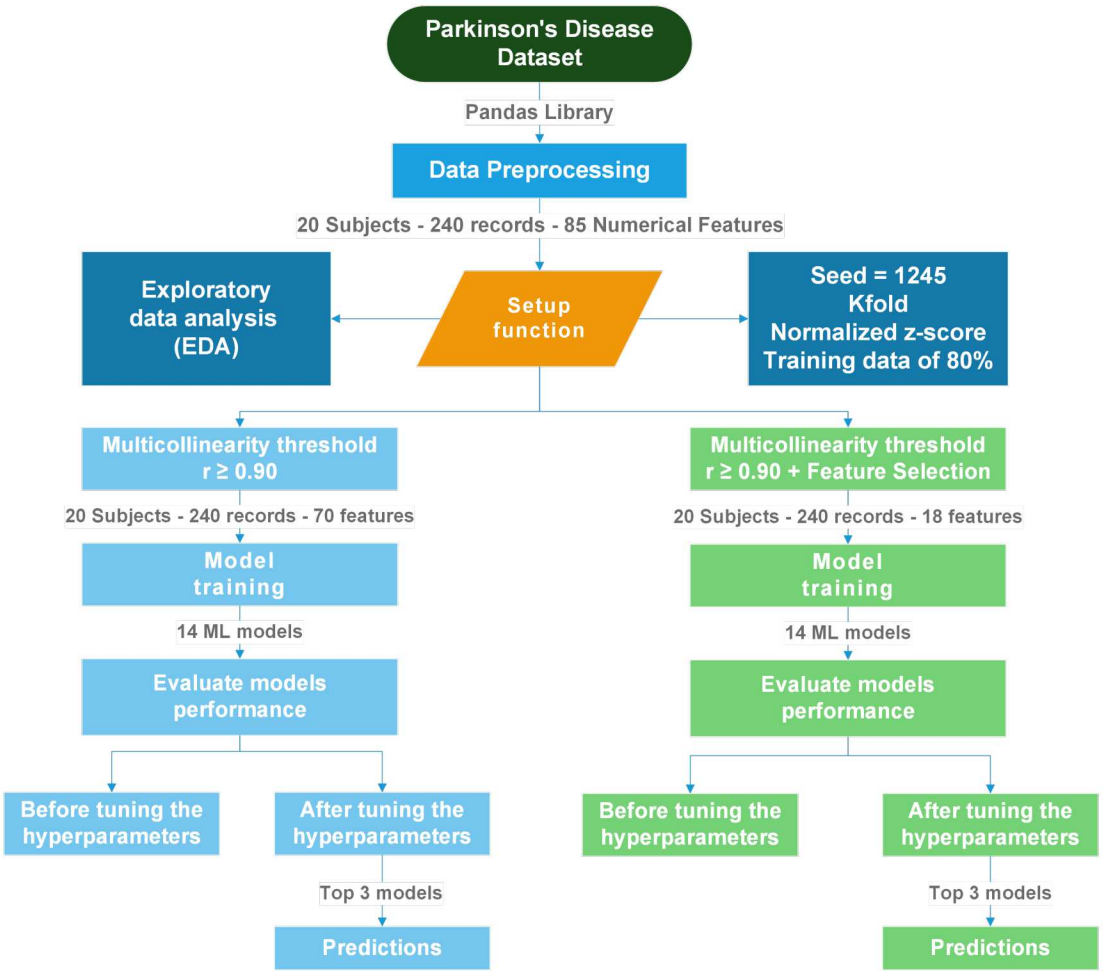


Figure 4.22 – Framework for ML modeling classification of Parkinson's disease using drawing patterns.

4.10 Classifying Framework

In this study, a total of 14 Support Vector Machine (SVM) algorithms—widely recognized and frequently referenced in the scientific literature [37] [18, 31]—were employed for the classification of Parkinson's disease using the PyCaret framework. The models were trained on 84 standardized features, each normalized to have zero mean and unit variance, with the binary disease status (Parkinson's disease vs. Healthy Control) serving as the target variable. The implementation was carried out in Python, utilizing robust libraries such as Scikit-learn and PyCaret to ensure an efficient and replicable analytical workflow. The complete methodological pipeline, including dataset preprocessing, data partitioning, model training, and results evaluation, is illustrated in Figure 4.22.

The training phase begins with a broad range of classification algorithms, ensuring a comprehensive assessment of model performance across diverse learning paradigms. A list of ML algorithms and their descriptions is listed in Table 4.3.

Table 4.3 – Machine Learning algorithms used

Type	Name	Abreviation	Description
Ensemble Methods	Extra Trees Classifier	ET	Ensemble, Tree-Based, Randomized Trees; Similar to Random Forest, but uses random thresholds for splitting, increasing variance reduction and speed.
	Random Forest Classifier	RF	Ensemble, Tree-Based; Combines multiple decision trees using bagging to improve accuracy and reduce overfitting.
	Gradient Boosting Classifier	GBC	Ensemble, Boosting; Builds trees sequentially where each new tree corrects the errors of the previous one.
	Light Gradient Boosting Machine	LightGBM	Ensemble, Boosting; A high-performance gradient boosting framework that uses histogram-based methods for speed and efficiency.
	AdaBoost Classifier	ADA	Ensemble, Boosting; Assigns weights to misclassified samples to focus subsequent models on the harder cases.
Instance-Based Learning	K-Neighbors Classifier	KNN	Instance-Based, Lazy Learning; Classifies based on the majority label of the k-nearest data points.
Probabilistic Models	Naive Bayes	NB	Probabilistic, Generative; Based on Bayes' theorem assuming feature independence; fast and simple.
	Quadratic Discriminant Analysis	QDA	Probabilistic, Generative; Assumes class-specific covariances to create quadratic decision boundaries.

Type	Name	Abreviation	Description
	Linear Discriminant Analysis	LDA	Probabilistic, Generative; Assumes shared covariance matrix across classes; forms linear decision boundaries.
Linear Models	Logistic Regression	LR	Linear, Discriminative; Models the probability of class membership using a logistic function.
	Ridge Classifier	RIDGE	Linear, Regularized; A linear classifier with L2 regularization to prevent overfitting.
Support Vector Machine	SVM - Linear Kernel	SVM	Linear, Margin-Based; Finds a hyperplane that best separates the classes with the widest margin.
Tree-Based Models	Decision Tree Classifier	DT	Tree-Based; Recursively splits data based on feature values to form a tree.
Baseline Model	Dummy Classifier	DUMMY	Rule-Based (Baseline); A control model that predicts using simple strategies like the most frequent class.

5

Results

This section presents comprehensive findings derived from the methodologies designed during this study. The results are organized into three main subsections, each corresponding to a core component of the proposed framework: (i) the protocol specifically designed to support the data acquisition and standardization process; (ii) the outcomes from the DLC-based pose estimation system; and (iii) the classification results generated through the application of supervised Machine Learning algorithms. Each part is presented with the goal of conveying the effectiveness, precision, and impact of the proposed approaches in a clear and structured manner. Emphasis is placed on both quantitative performance and practical implications, aiming at providing an accurate reflection of the contributions made at each stage of the research.

5.1 Protocol results and achievements

The motivation for creating a data collection protocol was to develop a step-by-step guide that ensures a reliable and efficient process for gathering data. This was particularly necessary due to the repetitive nature of the procedure and the potential difficulties Parkinson's disease patients might experience during the experiment. Furthermore, it was essential to establish a replicable methodology, enabling other research groups to either use the same tools to collect their own data or to validate the findings presented in this study.

The experimental protocol yielded promising results. The proposed objectives were successfully met, as outlined below.

1. Rapid setup (target completion time under 3 minutes)
2. Simplified assembly (minimal number of movable components)
3. Non-invasive procedure
4. Cost-effective construction (utilizing low-cost materials)

5. Capable of eliciting characteristic PD motor symptoms (rigidity, tremor, bradykinesia)

6. Materials readily available and easy to source

The first objective was to achieve a fast setup. The target was to complete the full setup—including positioning all necessary tools—in under three minutes. In practice, the average time recorded during setup tests was approximately 1 minute and 3 seconds, in some instances as fast as 45 seconds. These results clearly demonstrate that the goal of rapid setup was met.

The second objective focused on simplicity. The protocol involves only five movable components, significantly streamlining the setup process. Given this minimal number of elements, it is evident that assembling the data collection scenario is both fast and straightforward.

- Smartphone
- Smartphone holder
- Attachable smartphone holder paper guide
- Paper sheet
- Blue pen

An additional advantage, which reinforces the goal of efficiency, is that the entire setup does not need to be fully reassembled between sessions. Only the paper sheet needs to be replaced; all other components remain in place and ready for use. The smartphone stays mounted in the holder, the guide remains attached, and the pen is immediately available for the next participant. This significantly reduces the setup time between sessions and contributes to the overall ease of use of the protocol.

Upon achieving the third objective outlined in this study, the proposed protocol demonstrated its potential to generate actionable features for subsequent analyses, as previously discussed in this work. Consequently, the protocol can be regarded as non-invasive and capable of yielding meaningful results, enabling data and feature extraction within a viable and practical framework.

In order to assess cost-efficiency, it was necessary to research the prices of the materials employed. The cost of a smartphone was excluded from the analysis due to the wide variation in brands and models. Moreover, it was assumed that most individuals today have access to a smartphone capable of recording in Full HD resolution, or that such devices are readily available in clinical or hospital settings. Therefore, the analysis focused on the costs of the pen, paper, adhesive tape, and the plastic filament required to 3D-print the holder. These material costs are presented in Table 5.1.

Item	Cost	Quantity	Units Used	240 Sessions Cost
A4 paper	21,10 R$	300	240	16,88 R$
Blue pen	16,90 R$	5	1	3,38 R$
Adhesive Tape	8,29 R$	50 (m)	21,6 (m)	3,58 R$
ABS Filament	77,00	1000 (g)	129,02 (g)	9,93 R$
Total	123,29 R$	- - -	- - -	33,77 R$

Table 5.1 – Cost of experiment

As illustrated in Table 5.1, the experiment is economically feasible, with all the required materials being low-cost and easily accessible. The prices were updated and retrieved from Amazon.com.br on 01/05/2025. Given that the study involved 20 participants, each undergoing 12 sessions, and considering minor losses during execution, a 10% margin of error can be applied to the cost estimates. Despite this adjustment, the per-session cost remains low, confirming that financial constraints do not pose a significant barrier to implementing the proposed data collection protocol.

The protocol's ability to elicit certain Parkinson's disease characteristics was quantifiable through the use of the DeepLabCut tool. However, qualitative observations made during the sessions revealed that some patients exhibited more pronounced responses. Given that each session was supervised by at least two researchers with expertise in the Parkinson's disease field, it was evident that the protocol successfully triggered a few characteristic responses in most participants. Nonetheless, not all of these manifestations could be objectively be measured or quantified. To provide further clarity on these findings, selected results obtained with the DLC software are presented below.

Certain individuals, such as subject 012 (identified through anonymized numerical coding), exhibited pronounced tremor symptoms during the administration of the MDS-UPDRS Part III assessment. These symptoms were notably task-dependent, becoming more severe in specific motor exercises. The experimental protocol was effective in eliciting and capturing these tremor characteristics, particularly through the sinusoidal wave drawing tasks. In nearly all sessions, the subject consistently demonstrated this pathological feature. Furthermore, it was informally observed that the subject's motor performance deteriorated under fatigue.

Figure 5.1 illustrates the manifestation of tremor during voluntary motor tasks induced by the sinusoidal wave drawing exercise. A detailed examination of the figure reveals a well-defined and structured pattern in both the X and Y pixel coordinates prior to the black dashed line (left segment of the plot). This portion corresponds to the spiral drawing

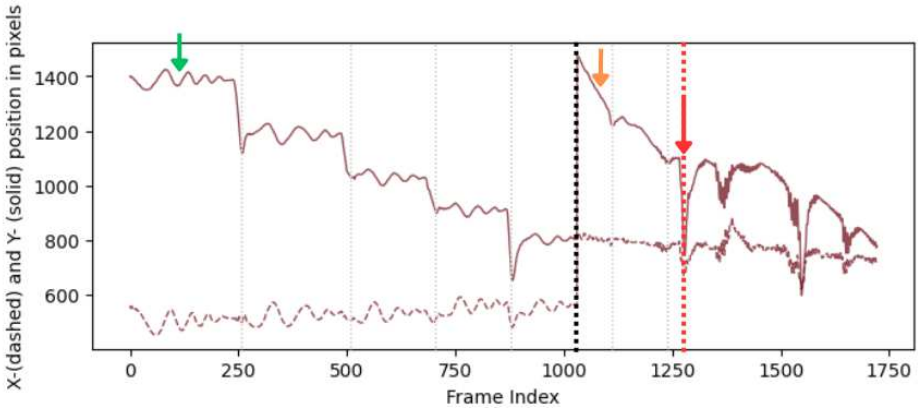


Figure 5.1 – X and Y pixel plot Ind. 012.

task. The Y-coordinate (solid line) in particular shows clear delineations, allowing one to easily identify the initiation and conclusion of individual drawings.

In contrast, the right segment of the plot—after the black dashed line—corresponds to the sinusoidal wave drawing. Here, a distinctly different pattern emerges. Initially, two identifiable wave patterns are observed; however, this structure gradually deteriorates. Both the X (dashed line) and Y (solid line) traces begin to exhibit high-frequency oscillations, resembling a signal waveform, indicative of involuntary tremor activity. This change strongly suggests that the experimental protocol successfully provoked a tremor response in the participant. The red dashed line marks the point at which this transition was most prominent.

On the one hand, patients such as subjects 006 and 014 exhibited pronounced bradykinesia (slowness of movement) consistently throughout the entire experimental protocol. Their performance remained stable across all sessions, with session duration variability limited to 2.15% and 3.35%, respectively. Notably, these individuals recorded the longest session durations in the Parkinson's disease group. This finding suggests that the proposed protocol may not only provoke but also reveal underlying rigidity—a core motor symptom of PD—during the drawing tasks.

On the other hand, Healthy Control participants, including subjects 037 and 038, demonstrated the longest session durations within the control group, with variation rates of 1.42% and 3.14%, respectively. This consistency might reflect their conscientious effort to complete the drawing tasks with precision, aiming for higher accuracy rather than speed.

These observations imply that additional features and parameters should be investigated to establish more robust and generalizable conclusions. Nevertheless, the preliminary data supports the hypothesis that the protocol is capable of capturing slowness of movement in individuals with Parkinson's disease.

The final objective of this study pertaining to the accessibility and ease of sourcing the materials required for the proposed protocol, aligning with the fourth specific aim

previously discussed. As outlined in the preceding sections, all components utilized in the experimental setup were selected with practicality and affordability in mind. Most materials are readily available through common retail outlets, such as stationery stores, or can be conveniently purchased online.

At the time the protocol was initially developed, the most challenging item to procure was the 3D printing filament, primarily due to the limited availability and adoption of 3D printing technology within the country. However, with the rapid advancement and dissemination of additive manufacturing technologies in recent years, both the raw materials and 3D printing services—including the production of custom smartphone holders—have become significantly more accessible and affordable.

5.2 Deep Lab Cut results and achievements

The results obtained using the DeepLabCut software are notably promising. When applied within the framework of the proposed protocol, the software demonstrated a high capacity to detect the target object without the use of physical markers, achieving an average precision of 99% across all participants in the study. In certain scenarios, the tracking accuracy was temporarily reduced, slightly affecting the overall mean precision. However, these fluctuations typically lasted only milliseconds to a few seconds and had minimal impact on the general performance.

It was observed that tracking precision declined particularly among participants who exhibited abrupt or erratic movements, especially during transitions between drawing tasks. In some cases, participants displayed involuntary motor tics at the conclusion of each drawing, often lifting their hands away from the paper as though to visually assess their work. Such behaviors frequently resulted in temporary object tracking losses. In more extreme cases, the object was entirely removed from the camera's field of view, further challenging the software's ability to maintain consistent tracking. To better explain that, figure 5.2 shows an example of a participant that had fewer losses on the tracking.

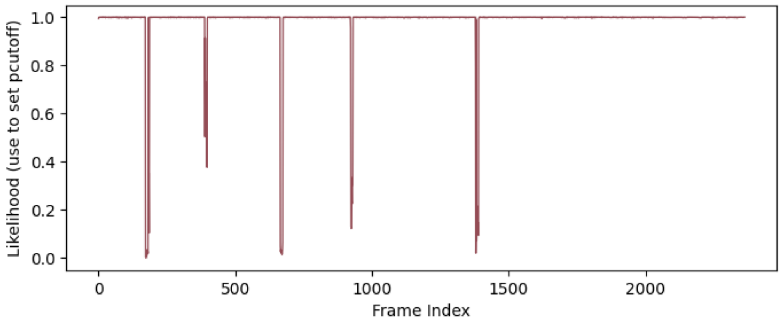


Figure 5.2 – Plot likelihood individual with high kinetic accelerations.

Despite these occasional issues, the DLC software consistently performed at a high

level. Upon completion of video processing, the software automatically generates a .csv file containing precise tracking data. To ensure the reliability of these outputs, a series of random manual verifications were conducted as part of a quality control procedure. These validations utilized a supplementary video generated by the DLC software, in which a clearly visible colored dot was overlaid on the tracked object. This allowed researchers to visually inspect whether the software accurately maintained object tracking throughout the entire duration of the recording.

An additional quality assurance step involves reviewing each X and Y coordinate graph immediately before conducting further data analyses. These graphs typically exhibit highly specific and recognizable patterns associated with drawing tasks. If any anomalous data points or irregular lines appeared—deviating from the expected drawing trajectories—the corresponding annotated video was reviewed to identify and evaluate the cause of the discrepancy, figure 5.3 shows a participant that has spikes on the graphs that should always be analyzed.

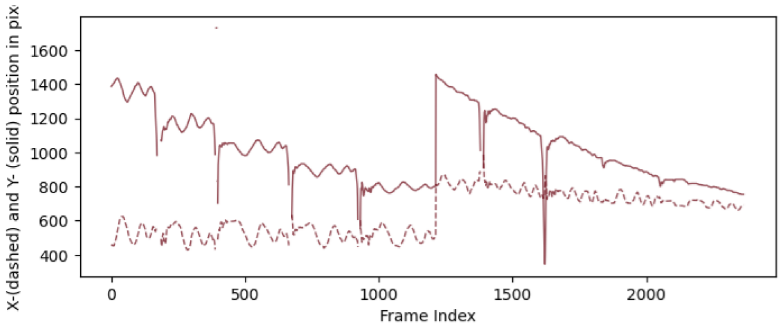


Figure 5.3 – Plot "X" and "Y" positional data of an individual with high kinetic accelerations.

As it is possible to see in figures 5.3 and 5.1, spikes occurred in both graphics, suggesting that some error occurred, and needed a video to confirm what caused the tracking failure, figures 5.4 and 5.5 shows how the confirmation is done with the video.

Below it will be presented the table 5.2 and table 5.3 basic features that can be extracted from the original .csv file originated from DLC. The table consists of showing the results from patients and bring other features not related to the DLC, like the MDS-UPDRS part III score and the most affected body side.

As it is possible to verify, in the table 5.2, the precision of the detection of the aimed object was above 99% in average for every participant. Which leads us to a very precise tracking system. Another point that is necessary to speak about is the standard variation on the tracking precision. It can indicates the difficulty of the participant in performing the task, and can also indicate some nervous tic, excessively rapid movements on the drawing change. But it is also possible to affirm that the variation was not high, reaching a peak of 8.66% for a healthy participant.

IND	SEX	P. AVG	P.STD	F.AVG	F.STD	G
3	F	99.9212%	0.3325%	1312.75	1.14%	1
6	F	99.4656%	6.4660%	2209.50	2.15%	3
7	M	99.9257%	0.0573%	1289.42	3.48%	2
9	F	99.7679%	3.8095%	1180.50	3.31%	3
14	M	99.7814%	3.5809%	2401.00	3.35%	2
15	F	99.8597%	2.3754%	2037.67	2.66%	1
16	F	99.9190%	0.5037%	1738.50	2.09%	1
18	M	99.9210%	0.0670%	1407.33	2.20%	3
19	M	99.9389%	0.0458%	940.67	1.74%	3
20	F	99.9187%	0.0665%	1331.25	2.78%	3
30	F	99.9235%	0.0609%	1245.92	4.67%	4
31	M	99.7782%	3.7227%	1340.25	2.01%	4
32	M	99.9246%	0.0612%	1367,67	1.71%	4
33	F	99.9193%	0.0593%	1397,58	3.08%	4
34	M	99.1525%	8.6617%	1634,25	3.31%	4
35	F	99.9273%	0.0579%	1772,67	2.84%	4
36	F	99.9011%	1.7790%	786,25	0.80%	4
37	F	99.9108%	1.4325%	2107,33	1.42%	4
38	M	99.9072%	1.0939%	2092,83	3.14%	4
40	F	99.9130%	0.7189%	1622,50	3.82%	4
Total Avg	—	99.8106%	2.0886%	1556.32	3.15%	—

Table 5.2 – DLC .csv results. IND: Individual, P.AVG: Precision average of DLC detection, P.STD: Precision standard deviation, F.AVG: Frames average, F.STD: Frames standard deviation, G: Group.

IND	SEX	G	AGE	UDPRS III	H&Y	R/L
3	F	1	55	20	1	8/10
6	F	3	62	27	2	13/7
7	M	2	58	20	2	9/7
9	F	3	67	26	3	11/4
14	M	2	71	21	1	11/8
15	F	1	63	14	1	10/2
16	F	1	64	52	1	4/1
18	M	3	52	28	1	20/2
19	M	3	59	68	3	15/32
20	F	3	49	8	1	5/2
30	F	4	58	—	—	—
31	M	4	56	—	—	—
32	M	4	56	—	—	—
33	F	4	53	—	—	—
34	M	4	52	—	—	—
35	F	4	66	—	—	—
36	F	4	55	—	—	—
37	F	4	65	—	—	—
38	M	4	72	—	—	—
40	F	4	62	—	—	—

Table 5.3 – Additional information of the individuals. IND: Individual, G: Group, UPDRS III: MDS-UPDRS Part III, H&Y: Honn & Yar Scale, R/L: Score for the Right/Left limb on the UPDRS Part III.

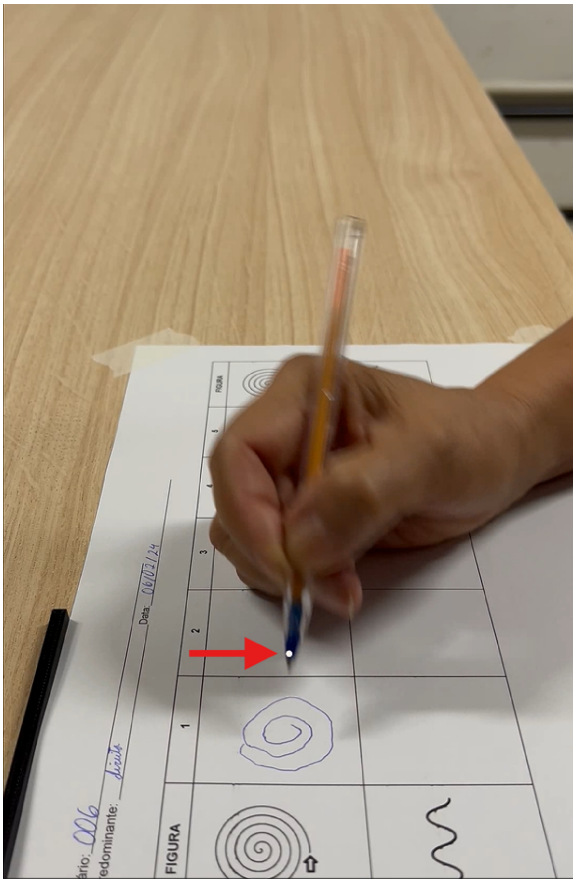


Figure 5.4 – Video verification with successful tracking.

The variation cited above was obtained by a healthy participant, what can lead to affirm the patient caused the variation with excessive acceleration on the movements, or the individual has a grip that is causing object occlusion. As cited earlier in the text, object occlusion is one of the worst scenarios that can happen with object tracking photo and video.

In this specific case, it is an estimation error caused by rapid movement, the individual has a pattern after concluding every drawing. The participant moves his hand very quickly out of the paper in order to make a movement to clearly see the drawing results. As the movement is fast, 30 frames per second in the video recording was not capable of capturing the pen without blurring the image. The blurred image caused difficulty for the DLC to process where is the aimed object.

Looking now at the highest values in the PD group, it is noticeable that the lowest value, reached only, is 6.47%. In this case the problem could be the same, but not only caused by a movement to check the drawing. In fact, there can be some PD characteristics caused by the protocol, which is desirable. In this specific case, the problem is exactly the same as the individual 34. The patient does the same rapid movement to check the drawing.

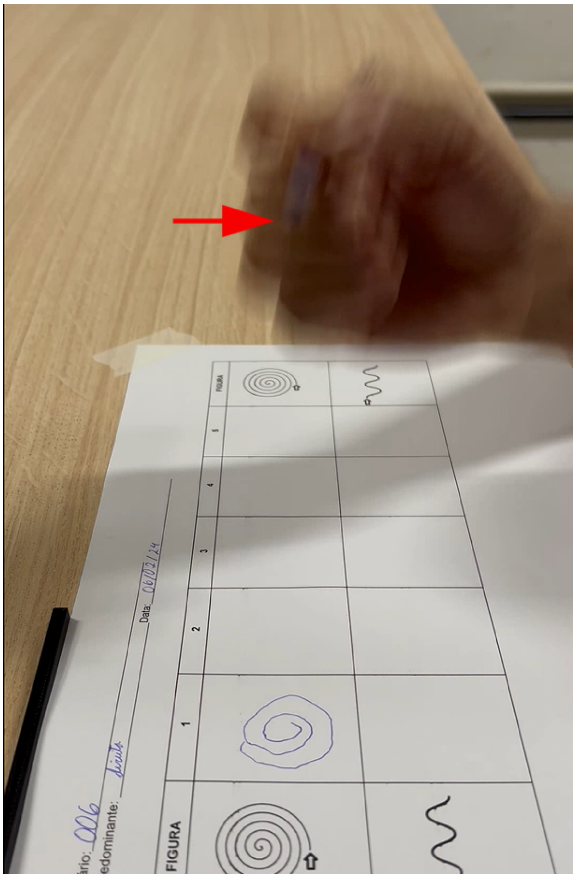


Figure 5.5 – Video Verification with miss tracking.

As it was mentioned before, the patients were instructed not to do these rapid movements, but sometimes they were not capable of avoiding them, even when they remembered not to do it. In a few cases, it was taken a second trial in order to void these little mistakes that could cause some bigger problems; in the end, almost every second trial the mistake persisted.

Despite that, it is possible to affirm that the DLC could performed very well when tracking the aimed object. Moving to the frames in the Table 5.2, the average for the entire experiment reached 1556.32 frames which means that the drawing task took around 51.88 (1556.32/30 FPS) seconds to conclude. This is a result that is desirable in a scenario where simplicity of execution is one of the main goals. The average standard variation indicates that the sessions are the a range of 1507.29 - 1605.34 frames which is 50.24 to 53.51 seconds.

The highest frame number was 2401.00 which can be converted to 80 seconds, which is a lot more than the average but it is a reasonable number due to the individual having a difficulty performing the task due to slowness in the movement. An important value was obtained with individual 19 which also belongs to the PD group, this individual reached an average of 940.67 frames with a low variance of 1.74%. This value is low considering the PD group and even the HC, which has only one volunteer that has done the task

faster than individual 19.

Looking at those numbers, it is not possible to assure you that the PD group performed a slower test, because the drawings varied a lot among individuals, some individuals drew larger spirals with more consistent lines and sinus waves with a higher frequency than others. So, it is not possible to detect if the patient has some slowness in movement by just looking at the time of the drawing process. This number needs to be associated with other features so it can be useful to classify individuals. For this session in specific, it is just a number reflecting the results obtained by the DLC video analyzing process.

The variation in frames for an individual on the other hand can be indicative of consistency, if the individual suffered from a specific factor that affected the execution of the task. There can be a variable number of factors that can affect the process, such as the patient's humor and exercise before the drawing session. Again, these numbers alone do not reflect a specific characteristic of PD, but aligned with other features it can be useful.

To briefly explain the groups, the groups are set up as 1, 2 and 3 representing PD groups, and the 4th group representing HC. The individuals were first separated into two groups, one contained only females and the second containing only males. This was done aimed at facilitating the schedule. As data collection was done involving the other three researches, the time available to the volunteers was short for every research. Due to that, the groups were divided into two and each contained only five individuals. Then the 3rd group emerged but with no separation between male and female. The 4th group had no need to be separated because not all research needed a long test with the HC, due to that the group did not needed to be separated.

In Table 5.3 it is shown some additional informations to help to understand the patterns and to evaluate the correlations between numbers presented in table 5.2.

Moving on to table 5.3, which contains some informations about the participants. The highlights that can be presented are that, the female in age average was 62 years old while the male was 57 years old. The standard deviation for both groups ages was 5.77 and 7.81 respectively (Both PD and HC participants). For the PD participants only, the MDS-UPDRS part III score reached the following values,:an average of 17 points with a standard deviation of 9.20 points for the female volunteers and 24.5 points with a standard deviation of 22.78 points for the male volunteers.

Unfortunately, all these numbers alone cannot represent much, but they are all useful to the Machine Learning algorithm to classify the groups. In the next session of the results it will be presented the results obtained by using the ML algorithms in VS Code by Microsoft.

5.3 Machine Learning results and achievements

The performance of the classification framework exceeded expectations. After testing several Machine Learning algorithms, the top 10 models achieved precision rates above 70%, which aligns with commonly accepted standards in the literature.

Tests were conducted both with and without the use of feature selection (FS), and the results showed some differences. However, the most significant variation was related to the number of folds used during cross-validation. All tests were completed and the data recorded both before and after the application of the tune function. Notably, the TF was not a decisive factor in improving precision; in some cases, the precision actually decreased following the tuning process.

Due to the small dataset, the number of folds used was also limited. A larger number of folds could increase the risk of overfitting or misclassification. The tests were conducted using fold values ranging from 2 to 10, with the best results using FS observed in the trials with 3, 7, and 9 folds. Among these, the 7-fold trial yielded the best overall performance.

Classification results were evaluated based on the average precision and the standard deviation of precision across all ML algorithms tested.

In the 3-fold trial, the results are presented in Table 5.4. The average precision exceeded 80%, indicating strong performance, and the standard deviation remained reasonably low. In this trial, the tuning process slightly worsened the accuracy. Although the cause of this decline is uncertain, the overall performance remained satisfactory. The standard deviation increased from 4.93% to 6.91%, indicating that while some algorithms improved, others performed worse after tuning.

The trial with 3 folds got the following results described on table 5.4

A closer examination reveals notable shifts in individual algorithm performance. Before the application of the TF, the top three algorithms achieved accuracies of 90.10%, 86.46%, and 85.94%. After tuning, the top three reached 93.75%, 93.75%, and 87.50%, respectively. This suggests that while some models improved, others did not, and the third-best tuned algorithm outperformed the second- and third-ranked models from the untuned trial.

Looking at the worst-performing algorithms, a similar trend can be observed. Prior to tuning, the lowest three models achieved accuracies of 77.60%, 75.52%, and 75.52%. After tuning, these values shifted to 77.08%, 77.08%, and 72.92%. In this case, the tuning process led to the lowest result in the table, reinforcing the observation that tuning did not consistently improve all models. When analyzing each ML algorithm individually, it becomes clear that the tuning function negatively affected the precision of some models.

In the 9-fold trial, which produced results similar to the 3-fold trial, the main variations were observed among the top-performing algorithms. While some models consistently performed well, others fluctuated between trials and across the tuned and untuned conditions. The results for this trial are presented in Table 5.5.

Algorithm	Acc B. (%)	Acc A. (%)
ET	90.10	93.75
RF	86.46	87.50
LIGHTGBM	85.94	83.33
GBC	84.90	85.42
KNN	84.38	93.75
ADA	82.29	83.33
DT	80.73	77.08
QDA	77.60	81.25
NB	75.52	77.08
RIDGE	75.52	72.92
Average	83.34	83.33
STD	4.93	6.91

Table 5.4 – Precision results 3-folds trial; Acc B.: Accuracy before tune function, Acc A.: Accuracy after tune function, Average: Accuracy precision, STD: Standard Deviation.

The 9-fold trial produced results similar to those of the 3-fold trial. The standard deviation was slightly higher compared to the other tests, while the precision—prior to applying the tune function—was also higher. After the tuning function was applied, the accuracy remained the same, but with a smaller standard deviation than in the previous trial.

When analyzing the ML algorithms individually, a pattern similar to that observed in Table 5.4 emerges. One of the main differences was in the models present in the top rankings. In this scenario, the most notable change was with the LIGHTGBM model, which dropped from third to sixth position in the "before tuning" column. The ET and RF algorithms remained among the top three performers, and the GBC model entered the top three rankings.

Looking at the right-hand side of Table 5.5, which contains the results after tuning, it is evident that the ET and KNN models performed exceptionally well, maintaining top-three positions in both the 3-fold and 9-fold trials. The algorithm that changed in the 9-fold trial was RF, which was replaced by the ADA model. Notably, in the 3-fold trial,

Algorithm	Acc B. (%)	Acc A. (%)
ET	92.11	93.75
GBC	85.86	85.42
RF	85.81	83.33
ADA	84.82	91.67
KNN	84.32	93.75
LIGHTGBM	84.32	83.33
QDA	78.11	81.25
NB	77.06	77.08
DT	76.00	79.17
LR	73.40	79.17
Average	84.32	83.33
STD	5.77	6.21

Table 5.5 – Precision results 9-folds trial; Acc B.: Accuracy before tune function, Acc A.: Accuracy after tune function, Average: Accuracy precision, STD: Standard Deviation.

the third-best model achieved an accuracy of 87.50%, whereas in the 9-fold trial, the third-best reached 91.67%, which is a significantly high result. It is also worth mentioning that the average accuracy before tuning was identical in both trials, but the standard deviation was slightly lower in the 9-fold trial.

Table 5.6 presents the results of the best-performing trial. Interestingly, the highlight of this trial is not the raw accuracy, but rather the improvement achieved through the application of the tune function.

From Table 5.6, it is evident that the tuning function led to a significant improvement, increasing the average accuracy from 81.75% to 85.42%. This represents the highest improvement observed across all three best trials. The standard deviation followed a similar trend: although the deviation was lower than in the other trials, it still increased slightly after tuning.

As with the previous trials, certain ML models performed very well and further improved after the tune function was applied. The highest accuracy reached in this study was 93.75%, and it was achieved in this trial. Additionally, the worst-performing ML in this

Algorithm	Acc B. (%)	Acc A. (%)
ET	90.06	93.75
RF	87.45	87.50
GBC	84.37	91.67
KNN	82.77	93.75
LIGHTGBM	82.26	85.42
ADA	81.24	83.33
QDA	79.72	85.42
DT	78.61	75.00
NB	76.53	81.77
LR	73.43	79.17
Average	81.75	85.42
STD	4.49	6.21

Table 5.6 – Precision results 7 folds trial; Acc B.: Accuracy before tune function, Acc A.: Accuracy after tune function, Average: Accuracy precision, STD: Standard Deviation.

trial still achieved an accuracy of 75.00%, which is not that far from the worst-performing models in the other trials.

All of this data was generated using Visual Studio Code and evaluated with feature selection enabled. An additional experiment was conducted by disabling FS to observe its effect on model performance. Surprisingly, the results were somewhat different from expectations. Although FS is designed to select the most informative features, the average model performance actually improved when this technique was not applied. It is important to note that another feature selection technique—'Remove_multicollinearity'— was already in place, which eliminates features with high correlation, potentially influencing the outcome.

Before discussing the results of the experiment without FS, it is useful to review which ML algorithms performed best across the three top trials. Table 5.7 presents these algorithms and their respective performances in each fold configuration.

Table 5.7 presents the best-performing models for each fold configuration, and several observations can be made. First, the tune function was effective across all of these models,

Algorithm	Folds	Acc B. (%)	Acc A. (%)
ET	3	90.10	93.75
	7	90.06	93.75
	9	92.11	93.75
KNN	3	84.38	93.75
	7	82.77	93.75
	9	84.32	93.75
RF	3	86.46	87.50
GBC	7	84.37	91.67
ADA	9	84.82	91.67

Table 5.7 – Best three models on the fold trials; Acc B.: Accuracy before tune function, Acc A.: Accuracy after tune function.

suggesting that they successfully identified important features and optimal parameter weights. The ET and KNN algorithms delivered identical accuracies after the application of the TF, while their pre-tuning results varied significantly.

Second, the KNN algorithm exhibited a remarkable improvement after tuning, increasing its accuracy from 82.77% to 93.75%. Third, the GBC and ADA models also demonstrated significant gains, achieving performance levels nearly identical to those of the ET and KNN models. This indicates that different algorithms can converge toward optimal weights and feature selections capable of accurately classifying the individual groups. Lastly, although the RF model performed well, its improvement was less pronounced compared to the others—yet it still delivered a high success rate.

These results help explain why the seven-fold trial was considered the best among all. It not only achieved the highest average accuracy and the lowest standard deviation, but also showed the most substantial improvement when the tuning function was applied —demonstrating that it best leveraged the model-boosting techniques implemented.

Next, the results of the same experiment conducted without the FS technique are presented. Table 5.8 summarizes the results of the 3-fold trial.

Now it will be presented the numbers acquired from the same test but excluding the FS option. Table 5.8 shows the numbers of the 3-fold trials.

In Table5.8, the first noteworthy result is that the average accuracy after the TF was higher than in the trials where FS was enabled. However, this came with a trade-off: the

Algorithm	Acc B. (%)	Acc A. (%)
GBC	86.98	93.75
ET	86.98	89.58
LIGHTGBM	86.46	85.42
ADA	84.90	87.50
RF	83.85	87.50
SVM	79.17	77.08
RIDGE	78.12	70.83
KNN	77.60	91.67
LR	77.08	70.83
LDA	76.04	70.83
Average	81.51	86.46
STD	4.50	9.17

Table 5.8 – Precision results 3-folds trial FS function disabled; Acc B.: Accuracy before tune function, Acc A.: Accuracy after tune function, Average: Accuracy precision, STD: Standard Deviation.

standard deviation was also higher than in the FS-enabled tests. Another point to consider is that the pre-tuning accuracy was lower compared to other trials, which supports the hypothesis that applying FS might negatively affect the results of the TF. In this test, the best-performing algorithm achieved an accuracy of 93.75%, with a 6.77% improvement after tuning. Notably, no other algorithm in this trial reached that performance level.

Moving on to Table 5.9, a similar pattern is observed. Once again, the best-performing model achieved 93.75% accuracy—identical to the result seen in the 3-fold trial. However, in this case, the pre-tuning accuracy was already high, at 90.06%, indicating that the model performed exceptionally well across all scenarios and was well-fitted to the training dataset. A meaningful comparison can be made with Table 5.6: the overall results are similar, with the primary difference being the model that reached the top performance. In the FS-enabled test, it was the ET model, while in this FS-disabled trial, the GBC model achieved the top result.

The average precision in this trial was also higher than in the FS-enabled version. However, as with the 3-fold FS-disabled test, the standard deviation was higher than in

Algorithm	Acc B. (%)	Acc A. (%)
GBC	90.06	93.75
ET	87.45	89.58
ADA	84.37	89.58
LIGHTGBM	82.77	85.42
RF	82.26	88.75
KNN	81.24	91.67
LR	79.72	70.83
RIDGE	78.61	72.92
LDA	76.53	70.83
DT	73.43	77.08
Average	81.75	87.09
STD	4.94	9.12

Table 5.9 – Precision results 7-folds trial FS function disabled; Acc B.: Accuracy before tune function, Acc A.: Accuracy after tune function, Average: Accuracy precision, STD: Standard Deviation.

the FS-enabled trials, reinforcing the earlier hypothesis. Furthermore, the variation in standard deviation increased more sharply compared to the tests with feature selection enabled. In summary, the best-performing model in this trial was again the GBC model, which achieved even better results than in the 3-fold FS-disabled trial.

Table 5.10 presents notable results when analyzing model accuracy. Before the application of the tune function, the accuracy was higher than in the other two fold-based tests. However, once the tuning was applied, this trial showed the smallest improvement in accuracy among all cases. Once again, the best-performing model was the GBC, achieving 88.48% accuracy before tuning and 93.75% after the TF was applied.

It is worth noting that the KNN algorithm performed consistently well across all fold-based trials, both with and without the application of the FS technique. The algorithm's minimum performance was 91.67% when tuning was applied, and it consistently demonstrated substantial improvements from the baseline values to the post-tuning results.

To further examine the behavior of the top three Machine Learning models, this section includes an analysis of their respective Confusion Matrix (CM). The CM allows

Algorithm	Acc B. (%)	Acc A. (%)
GBC	88.48	93.75
LIGHTGBM	87.42	87.50
ET	86.87	89.58
ADA	84.34	87.50
RF	84.32	83.33
KNN	81.70	91.67
DT	80.64	77.08
LR	77.06	70.83
RIDGE	76.05	70.83
QDA	75.08	81.25
Average	83.01	85.42
STD	4.89	8.22

Table 5.10 – Precision results 9-folds trial FS function disabled; Acc B.: Accuracy before tune function, Acc A.: Accuracy after tune function, Average: Accuracy precision, STD: Standard Deviation.

for a detailed view of each model's misclassifications. Figure 5.6 shows the confusion matrix for the ET model, both before and after the application of the TF.

From Figure 5.6, it becomes clear where the ET model improved or underperformed. The ET algorithm performed better after tuning, with fewer misclassifications. Specifically, the model exhibited greater difficulty in correctly classifying false positives in both the FS-enabled and FS-disabled trials.

Turning to the GBC model (Figure 5.7), the results display an interesting contrast. In this case, the GBC model performed better in the FS-disabled trial than in the FS-enabled one. Furthermore, compared to the ET model, the GBC model generated more false negatives, indicating a different pattern of classification error.

Finally, the results for the KNN model, shown in Figure 5.8, reveal a markedly different performance profile. Before the application of the TF, the KNN model performed worse than the ET and GBC models. The confusion matrix highlights this by showing significant difficulty in correctly classifying both false positives and false negatives.

It is clear that, prior to tuning, the KNN model struggled particularly with false

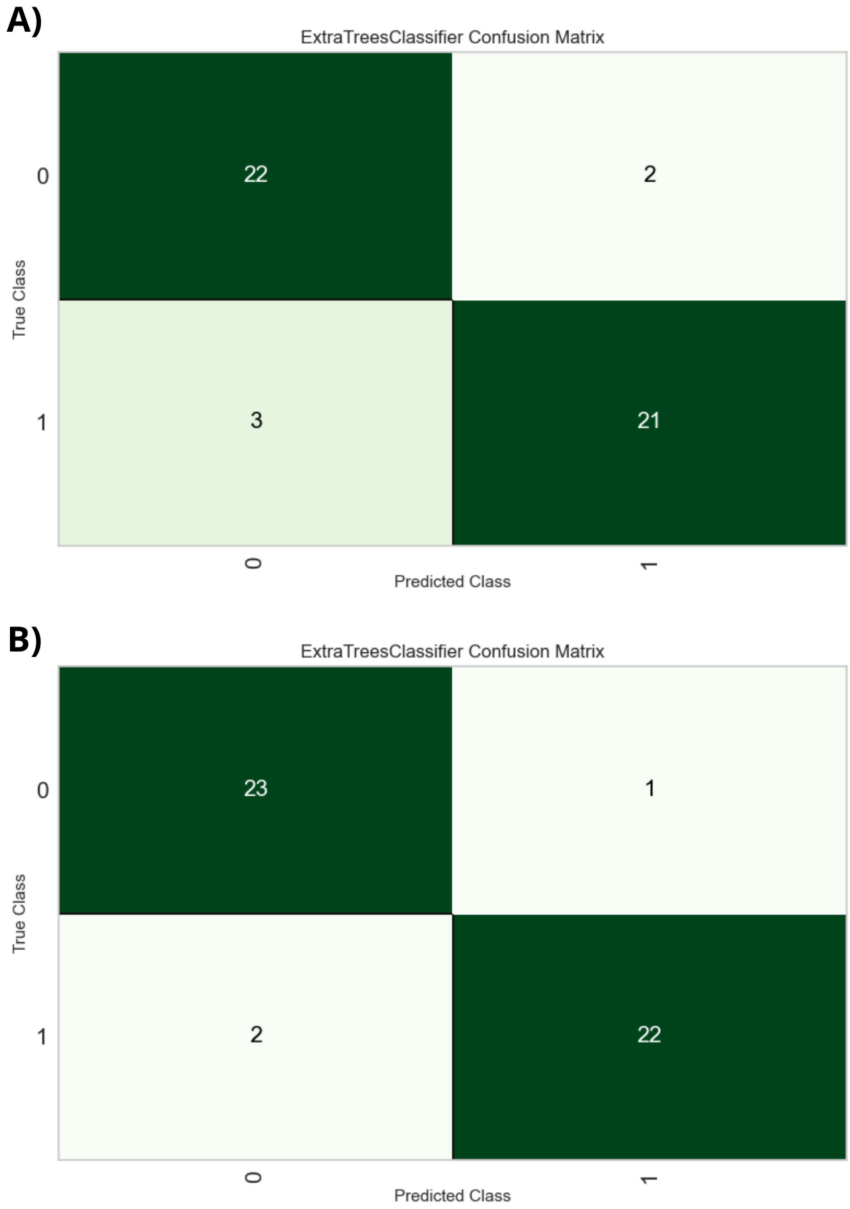


Figure 5.6 – 7-folds trial Extra Tree classifier Confusion Matrix; A): Test with feature selection disabled; B): test with feature selection enabled.

negatives and true negatives. However, after tuning (part B of the figure), its performance aligned more closely with that of the ET and GBC models—suggesting that the tuning process significantly enhanced KNN's predictive ability.

To deepen the analysis of the ML models' performance, the next section presents the Learning Curve (LC), Validation Curve (VC), and Classification Report (CR), which include additional metrics such as F1-score and recall.

Figure 5.9 displays the Learning Curve for the three models side by side, enabling a clearer comparison of their learning behavior in FS-enabled and FS-disabled scenarios.

The first important point to consider is how Learning Curve work. A Learning Curve

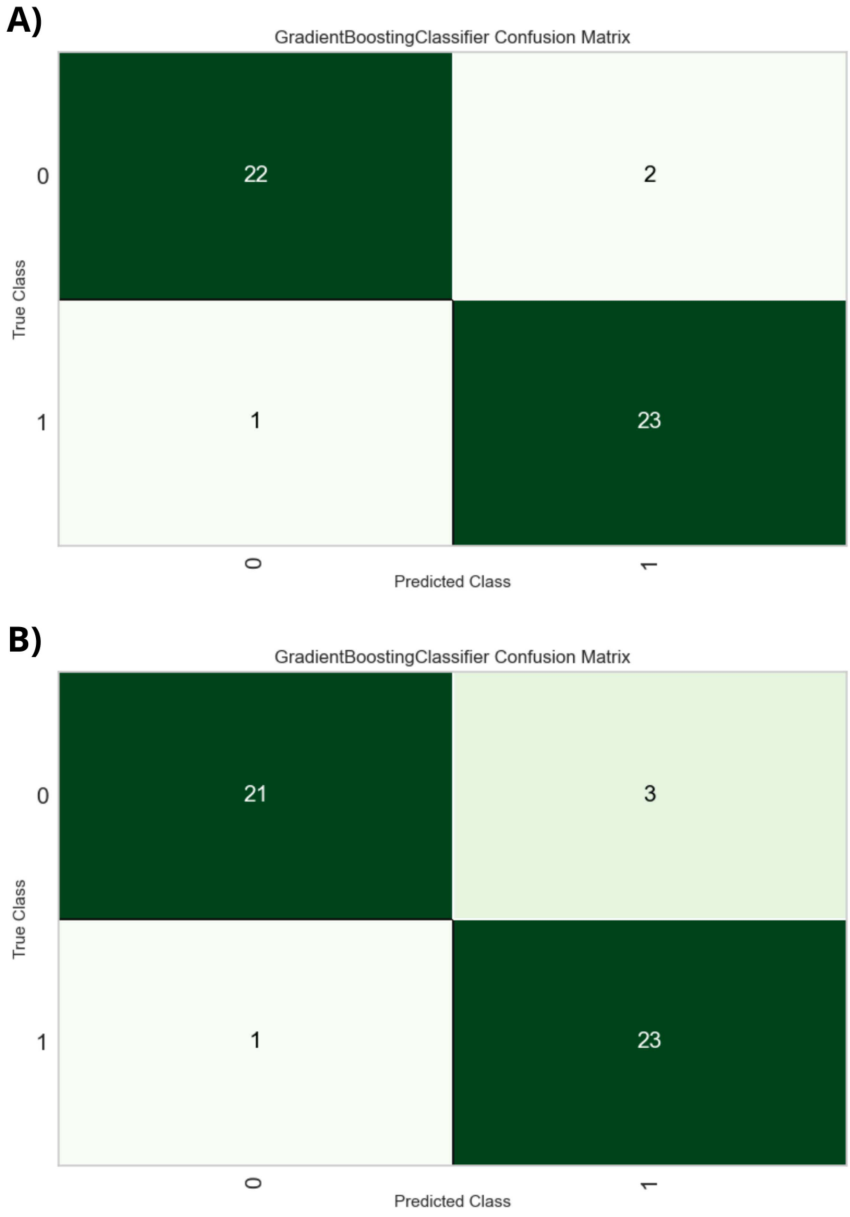


Figure 5.7 – 7-folds trial Gradient Boost classifier Confusion Matrix; A): Test with feature selection disabled; B): test with feature selection enabled.

illustrates model performance as a function of the training data size. A large gap between the training and validation scores typically indicates overfitting, whereas a small gap may suggest underfitting. In this study, due to the limited size of the dataset, some degree of overfitting and/or underfitting is expected across the evaluated models. With this context, we can better interpret the Learning Curve graphs.

The comparison between models with and without feature selection is particularly valuable, especially given the relatively high number of features compared to the dataset size. Figure 5.9 demonstrates that applying FS generally improves model performance across all three classifiers. It is evident that the Learning Curve with FS tend to start

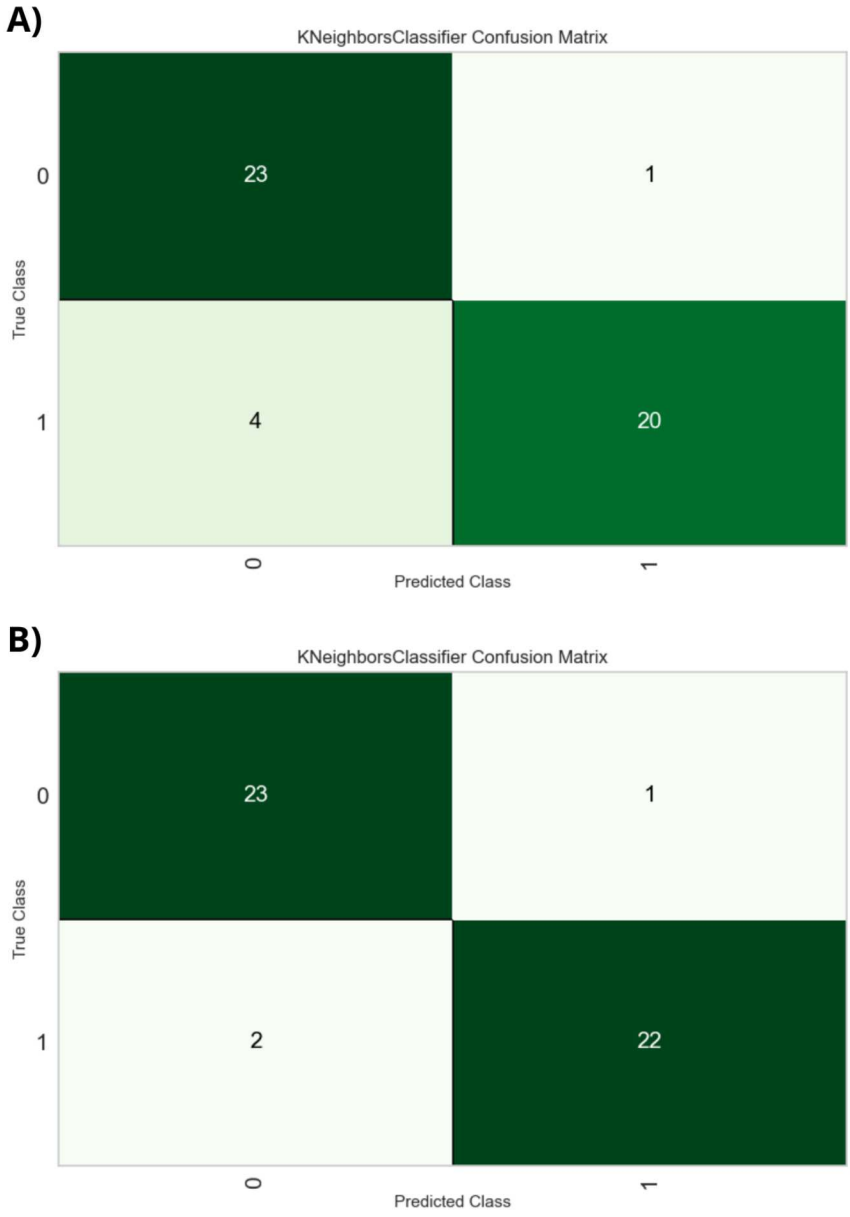


Figure 5.8 – 7-folds trial K-neighbors classifier Confusion Matrix; A): Test with feature selection disabled; B): test with feature selection enabled.

at and maintain higher validation scores, suggesting that FS is beneficial in small-data scenarios.

For the GBC model (Figures A.1 and A.2), the effect of FS is clear. Without FS, the cross-validation score starts just above 0.70 and gradually increases to approximately 0.85–0.90. In contrast, with FS enabled, the cross-validation score begins closer to 0.85

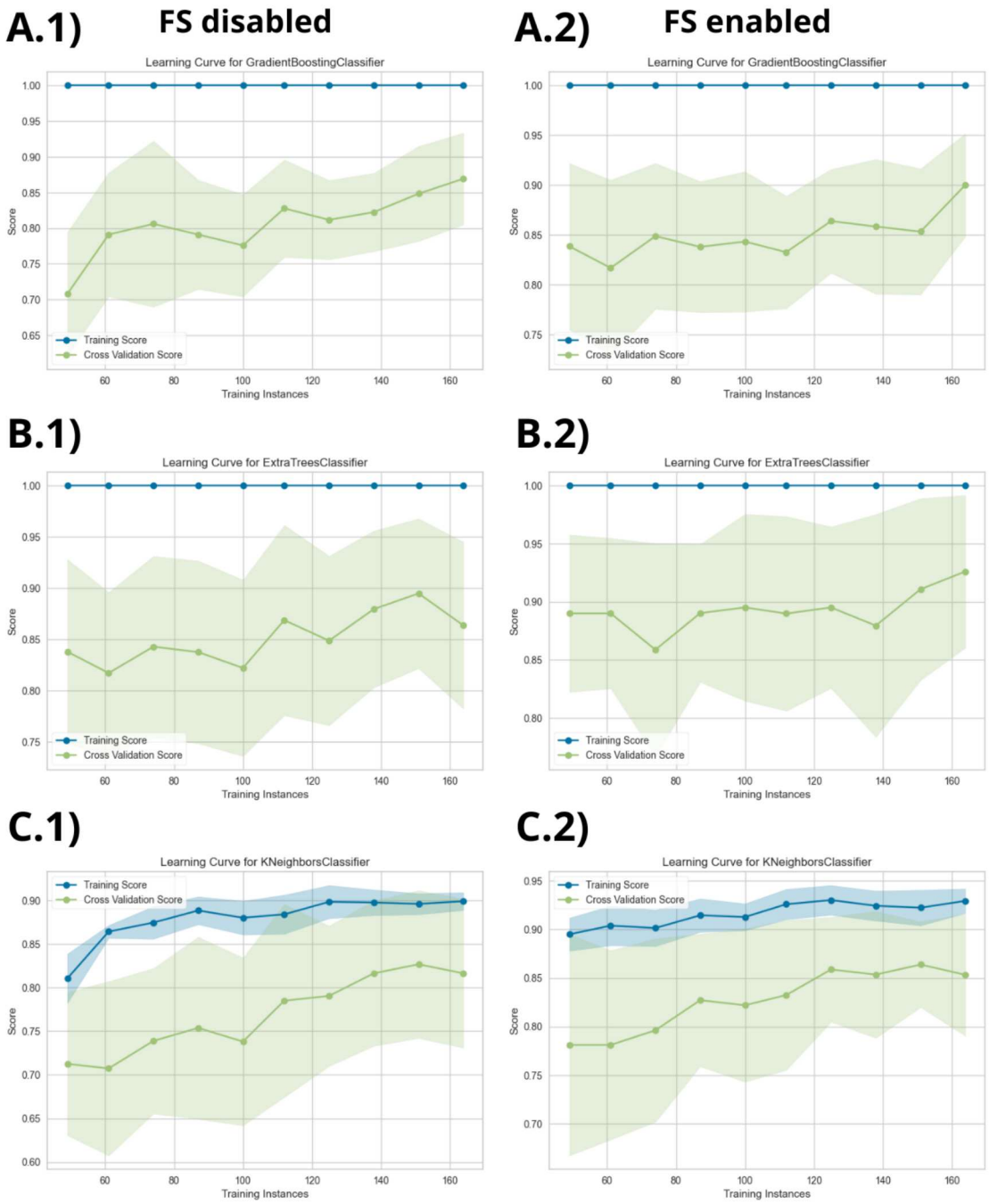


Figure 5.9 – Learning Curve for GBC, ET and KNN models; A.1): GBC model with FS disabled; A.2): GBC model with FS enabled; B.1): ET model with FS disabled; B.2): ET model with FS enabled; C.1): KNN model with FS disabled; C.2): KNN model with FS enabled.

and remains consistently high, peaking near 0.90. This suggests that FS contributes to better and more stable generalization performance.

The ET model (Figures B.1 and B.2) shows a similar trend. Without FS, cross-validation scores vary between 0.80 and 0.90, with about half falling below 0.85. With FS

enabled, the scores consistently start near 0.90 and often exceed it. The validation score is more stable, indicating that the algorithm benefits from FS both in terms of performance and consistency.

The KNN model (Figures C.1 and C.2) presents a different pattern. Unlike GBC and ET, its training score is not at 100%, indicating that it does not overfit the training data as strongly. Without FS, the model struggles to reach a cross-validation score of 0.85. However, with FS enabled, performance improves, with scores exceeding 0.85. The LC suggests that the model initially suffers from underfitting, but as more data is added, both training and validation scores improve—indicating better generalization.

Overall, FS proves advantageous for all models, particularly in improving cross-validation scores. Although the FS had a minor negative impact on the GBC model's validation variance, its effect on ET was negligible, and for KNN, it reduced variance as the dataset size increased, which is desirable. It is also worth noting how FS affects the gap between training and Validation Curve. For both GBC and ET, FS reduces this gap—though if the gap becomes too small, it may indicate underfitting. Therefore, when selecting a model, it is essential to consider multiple performance metrics to ensure a well-balanced result.

In conclusion, while the GBC and ET models demonstrate strong performance, their Learning Curve indicate a risk of overfitting due to near-perfect training scores. The KNN model, although initially weaker, shows a healthier learning dynamic, with gradual improvement and less overfitting. Adding more training data may help KNN catch up in terms of performance. Similarly, GBC and ET could also benefit from additional data to better balance memorization and generalization.

Moving forward to the Validation Curve, figure 5.10 will show the results obtained in the same way as figure 5.9

Validation curves are useful for evaluating a model's performance across different values of a specific hyperparameter, while keeping all other parameters fixed. These curves help detect overfitting and underfitting by plotting training and cross-validation scores. In this case, we assess the impact of the max_depth hyperparameter for the Gradient Boosting Classifier and Extra Trees Classifier, and the n_neighbors parameter for the K-Nearest Neighbors model. As in previous figures, we compare each model's behavior with and without feature selection. Figures A.1 (FS disabled) and A.2 (FS enabled) show the Validation Curve for GBC with varying max_depth. In both configurations, the training score remains consistently high, close to 1.0, indicating a strong tendency toward overfitting, especially as the depth increases.

Without FS (Figure A.1), the cross-validation score peaks around max_depth = 3, then drops steadily as depth increases. This pattern clearly shows overfitting at higher depths, as the model becomes too complex relative to the data size.

With FS enabled (Figure A.2), the model shows better generalization performance.

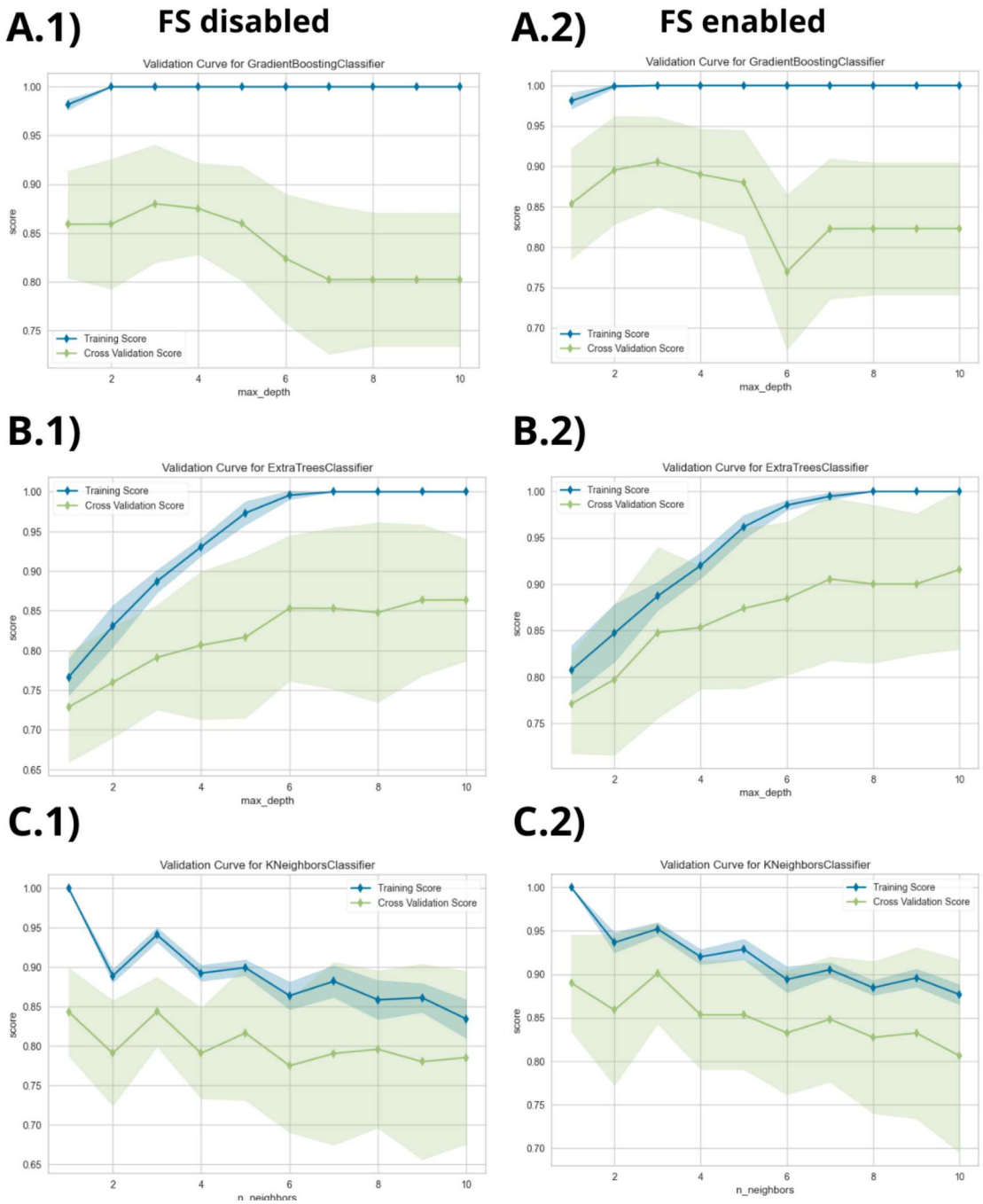


Figure 5.10 – Validation Curve for GBC, ET and KNN models; A.1): GBC model with FS disabled; A.2): GBC model with FS enabled; B.1): ET model with FS disabled; B.2): ET model with FS enabled; C.1): KNN model with FS disabled; C.2): KNN model with FS enabled.

The cross-validation score reaches its peak around max_depth = 3 to 4, and while it still drops with higher depths, the decline is less severe. The narrower gap between training and VC at lower depths suggests that FS reduces overfitting and improves generalization.

Figures B.1 and B.2 represent the ET model under the same max_depth tuning. In

both cases, the training score approaches 1.0, particularly for higher depth values. This again indicates overfitting when the trees become too deep. Without FS (Figure B.1), cross-validation scores increase with depth but plateau after max_depth = 6. The wide confidence interval and training-validation gap suggest some instability in generalization for deeper trees.

With FS enabled (Figure B.2), the cross-validation performance improves and becomes more stable. The model achieves a higher and flatter performance across depths 4–8, with a more consistent and narrower Validation Curve. This further supports the benefit of FS for this classifier, promoting both performance and robustness.

For the KNN model (Figures C.1 and C.2), VC are plotted over increasing values of n_neighbors. Here, the behavior is different due to the nature of the algorithm. A small number of neighbors (e.g., 1 or 2) leads to near-perfect training accuracy but poor generalization due to overfitting. As n_neighbors increases, both training and validation scores decline and gradually converge.

Without FS (Figure C.1), performance is less stable. Training scores are higher than validation scores across all values, and the gap remains large, indicating poor generalization. Cross-validation scores hover below 0.85 and show wide variance. With FS enabled (Figure C.2), both training and validation scores increase slightly and the gap between them narrows. While overall performance remains lower than the other two models, FS clearly reduces variance and improves generalization, especially for n_neighbors values between 5 and 8.

To further understand the impact of feature selection and evaluate model performance at a class level, Figure 5.11 presents the Classification Report (CR) for all three models with and without FS. These visualizations show the values for precision, recall, and F1-score for both classes (0 and 1), as well as the class support.

Starting with the GBC model (Figures A.1 and A.2), FS causes a trade-off between recall and precision for class 1: recall decreases from 0.958 to 0.875, while precision increases from 0.920 to 0.955. This implies FS improves the model's confidence in positive predictions but at the cost of missing more true positives.

The ET model (Figures B.1 and B.2) shows a slight increase across all metrics with FS enabled. For instance, the F1-score for class 0 remains high (0.898 $\rightarrow$ 0.939), suggesting FS brings more balanced improvements without compromising recall.

The KNN model (Figures C.1 and C.2), while overall less performant, suffers a drop in recall and F1-score for class 1 when FS is applied. This suggests that FS may have reduced KNN's capacity to detect class 1 instances, likely due to loss of important features for local distance computation. These observations confirm the earlier patterns seen in the learning and Validation Curve: GBC and ET benefit from FS, albeit with different trade-offs, while KNN shows inconsistent gains.

To complement the performance analyses, feature importance plots were generated

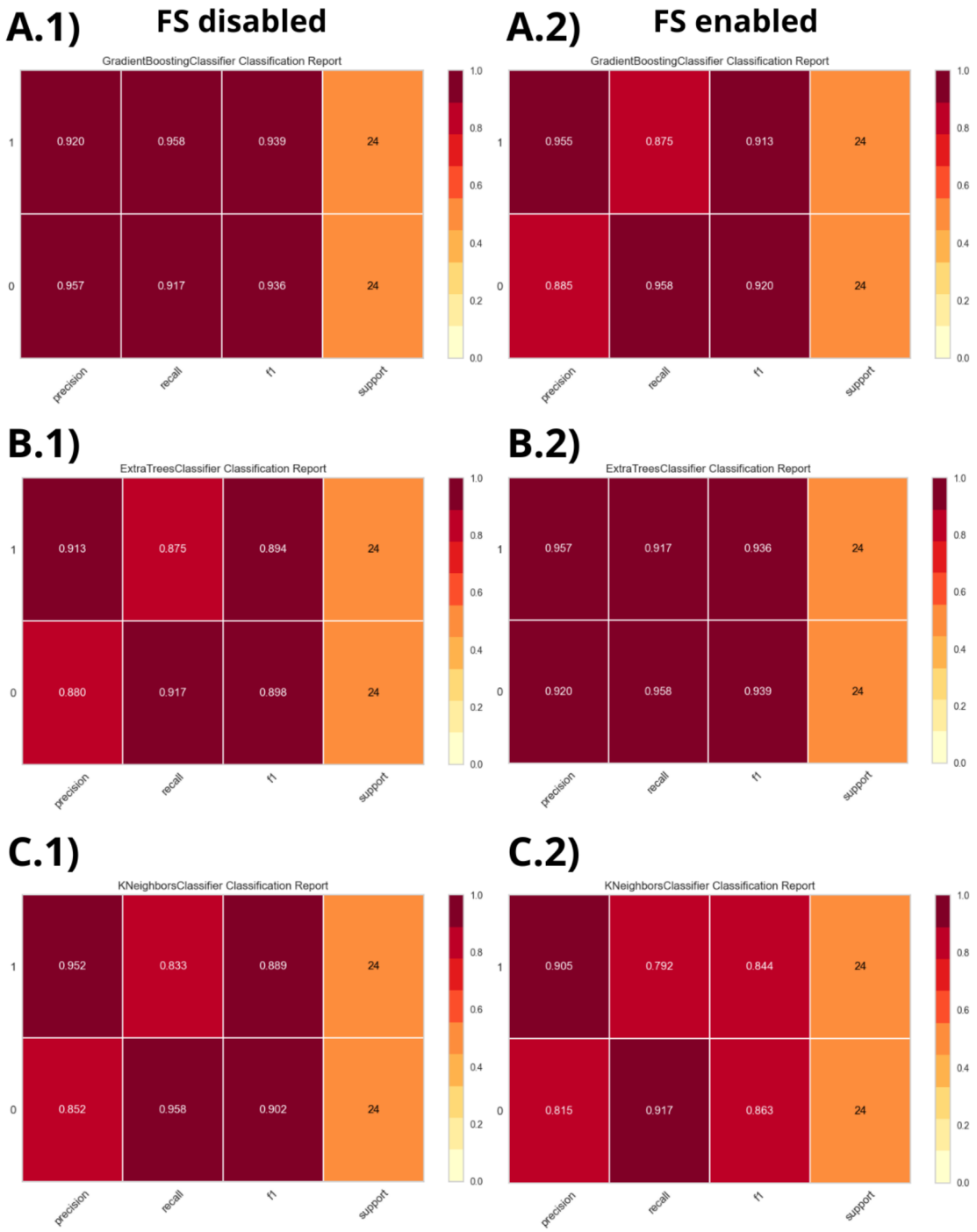


Figure 5.11 – Classification Report for GBC, ET and KNN models; A.1): GBC model with FS disabled; A.2): GBC model with FS enabled; B.1): ET model with FS disabled; B.2): ET model with FS enabled; C.1): KNN model with FS disabled; C.2): KNN model with FS enabled.

for the Gradient Boosting and Extra Trees classifiers after applying the feature selection step. These plots provide insight into which features were retained and how much they contributed to the model's decision-making. Figure 5.12 shows the plots about feature

importance.

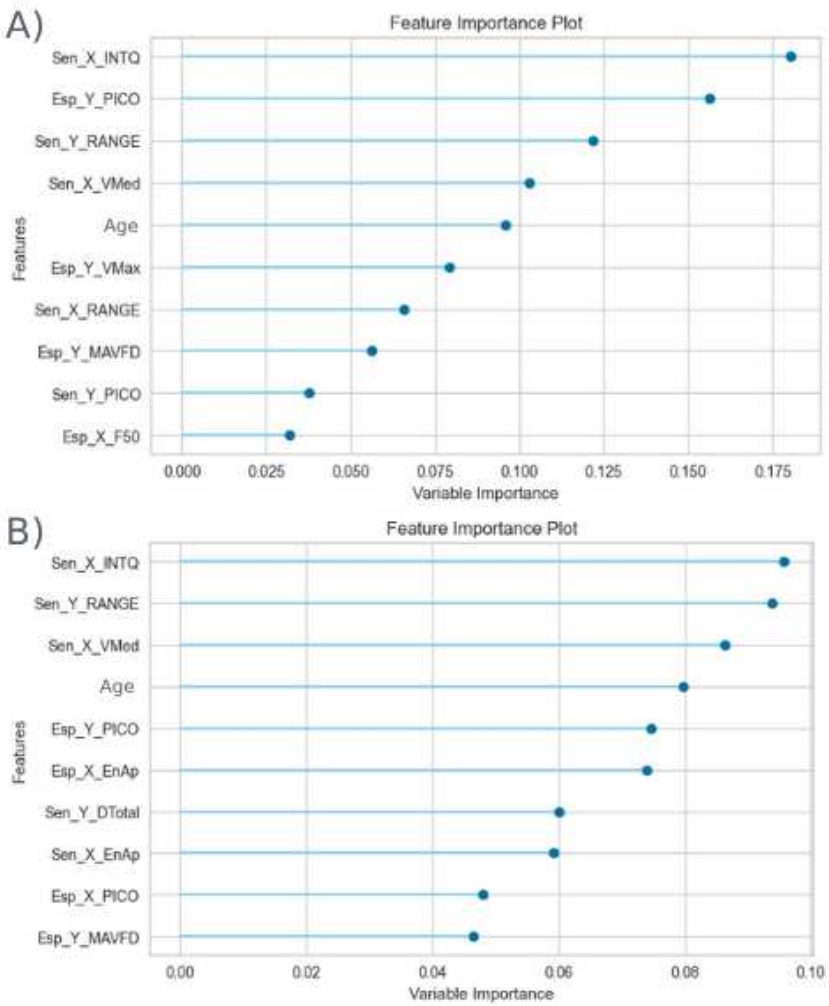


Figure 5.12 – Feature importance for GBC and ET model; A): GBC model with FS enabled; B): ET model with FS enabled.

Plot A illustrates the variable importance as determined by the Gradient Boosting model. The most influential features were first Sen_X_INTQ, then Esp_Y_PICO, and Sen_Y_RANGE, each contributing significantly to the final prediction. This distribution suggests that the FS algorithm successfully retained a compact set of discriminative features while filtering out those with low predictive value. The GBC model, being based on decision trees, naturally enables the extraction of importance scores through how often and how effectively features split the data.

In plot B, the Extra Trees classifier also indicates a concentration of importance in a few select features, such as Sen_X_INTQ, Sen_Y_RANGE, and Sen_X_VMed. This aligns with the FS goal of simplifying the feature space while preserving or enhancing classification performance. Compared to GBC, the ETC importance distribution appears more uniform, although it still highlights a handful of dominant variables. The slight

variation in feature rankings between GBC and ETC underscores model-specific perspectives on feature utility.

Unlike GBC and ETC, the KNeighborsClassifier does not support native feature importance measures. Since KNN is a distance-based, instance-level learner (i.e., a non-parametric model), it does not build an internal representation of feature contributions. As a result, importance scores cannot be extracted directly, and KNN is excluded from this figure. However, the improvement in its cross-validation performance post-FS (see previous validation curves) indirectly supports that FS helped remove irrelevant or noisy features, enhancing neighborhood similarity in the feature space.

6

Discussion

Starting the discussion with the development of the data collection protocol, which was validated in this study presents a significant step toward the creation of accessible, replicable and cost-effective tools for motor symptom analysis in PD. The results demonstrated that the proposed setup not only meets the predefined operational objectives but also facilitates the extraction of meaningful behavioral and kinematic data in a controlled, non-invasive setting.

The effectiveness and the usability of the protocol reached all the predefined operational goals with an average of less than one minute to get all the test setup, which confirms its efficiency. While the lower number of components and the non-invasive nature of the experiment, contributes to a user-friendly experience for both participants and researchers. The replicability and the re-usability of most components across multiple sessions further emphasize the protocol's suitability for clinical environments or large-scale trials.

The cost of a per-person session is around R$0.15 which ensures the affordability of the protocol. Given the current expansion of 3D printing technologies and widespread access to smartphones, this protocol can be implemented without significant financial burden, leading to a crucial advantage in low-resource clinical setting or for remote monitoring applications.

The last point that needs a special attention was that the protocol have the ability to elicit observable motor symptoms characteristics of PD. Through both quantitative (e.g., DeepLabCut data) and qualitative (clinician-observed) insights. It is clear that drawing task such as spirals and sinusoidal waves were successful in provoking tremor and bradykinesia. In particular, the tremor manifestation in individuals 012, and the slowness in subjects 006 and 014 provide evidence that the task design was appropriate for capturing pathological motor patterns.

Due to that, it is considered that the protocol has potential to serve as a standarized tool for home-based monitoring, telemedicine consultations, or as a pre-screening instrument in clinical trials. Due to the low cost and the simplicity of the protocol, it can

be attractive to longitudinal studies or integration into mobile health plataforms, where remote and scalable solutions are in need.

The reproducibility and low technical barrier presented in the protocol allows possibilities for cross-lab comparisons or even collaborative datasets, advancing the development of a machine learning model trained on more heterogeneous population.

The use of DeepLabCut in this study played a pivotal role in the extraction of kinematic features from video recordings without the need for physical markers, which is a substantial advantage for non-invasive biomedical research. The system demonstrated an exceptional capacity for precise object detection, achieving an average tracking precision above 99.0% across all participants. This performance reflects not only the robustness of the pre-trained neural network model but also the effectiveness of the data collection protocol and preprocessing procedures applied to the raw videos.

Despite the consistently high precision, the standard deviation in tracking accuracy varied across individuals, indicating that participant behavior influenced DLC's performance. In particular, volunteers who exhibited erratic or abrupt hand movements—often during the transitions between different drawing patterns;tended to show brief periods of tracking loss. These issues were typically visualized as downward peaks in the likelihood and plot x and y axis graphs generated by DLC, with some errors lasting only a few frames. In more extreme cases, temporary occlusions or the complete removal of the pen tip from the camera's field of view challenged the model's ability to maintain continuous detection. An important note is that if the software lost the aimed object due to the participant removing the object of the field of view, the software did not point an erratic object to maintain a random tracking.

However, it is important to highlight that even the largest precision deviation observed (8.66% in a healthy participant) remained within acceptable limits for the scope of this study. Furthermore, such deviations offered indirect insight into patient behavior. For instance, high standard deviation values might correlate with involuntary motor tics or hesitation movements, and therefore carry potential diagnostic value when interpreted alongside other clinical indicators.

A rigorous quality control workflow was implemented to mitigate the effects of tracking artifacts and ensure the reliability of the extracted data. This included reviewing DLC's coordinate plots and likelihood curves for anomalies, followed by video verification using the software's "create videos" function. These annotated videos featured a colored dot tracking the object of interest, facilitating frame-by-frame inspection of suspected errors.

This process proved essential in identifying cases where tracking failures were caused by factors such as occlusion from participant nails, confusing background colors, or rapid accelerations that introduced motion blur. By visually confirming these issues and, when necessary, excluding unreliable sequences, the study maintained high data integrity.

The choice to avoid smoothing filters in DLC was deliberate, aiming to preserve raw

movement patterns that might be diagnostically relevant. Although this increased the visibility of some noise in the data, it allowed subsequent Machine Learning models to capture subtle variations in motor behavior. Moreover, the dataset processing pipeline — exporting tracking data to .csv, cleaning and restructuring the files using RStudio scripts, and converting to JSON—ensured that essential features were calculated and retained for analysis, optimizing the performance of downstream classifiers.

While the current setup utilized Full HD (1920×1080) resolution for practical processing reasons, preliminary tests indicated no significant loss in detection accuracy compared to higher resolutions. Nonetheless, future studies could explore the benefits of ultra-high-definition recordings and high PFS if hardware capabilities allow, potentially improving tracking stability during high-speed motion, avoiding the motion blur effect.

The classification framework applied in this study was designed to explore the performance of a wide range of Machine Learning algorithms in the detection of Parkinson's disease using standardized kinematic features extracted from drawing tasks. Fourteen classifiers were evaluated, with and without feature selection, across multiple validation strategies (2 to 10 folds). This thorough experimental design allowed a robust comparative analysis, uncovering both general trends and model-specific behaviors.

Moving to classification framework, one of the most notable findings was the overall high performance achieved across multiple classifiers. In particular, the top 10 algorithms reached precision rates above 70%, with some exceeding 90%, aligning with or surpassing benchmarks found in the literature. This suggests that the features extracted from the drawing-based task hold considerable discriminative power for differentiating PD from Healthy Control.

Feature selection played a central role in enhancing the model performance and generalization capacity. When FS was applied, many models demonstrated improved validation scores and reduced overfitting, as evidenced by learning and validation curves. For example, Gradient Boosting Classifier (GBC) and Extra Trees (ET) models showed higher and more stable cross-validation scores after FS was enabled. In the case of GBC, the cross-validation score improved from 0.70–0.85 to consistently above 0.85 with FS. Similarly, ET reached validation scores around or above 0.90 after FS.

These improvements are further supported by the classification reports, where recall, precision, and F1-score metrics showed gains in class-wise performance for both models. In contrast, the K-Nearest Neighbors (KNN) algorithm displayed inconsistent behavior; its performance initially improved post-FS but declined in recall and F1-score for class 1. This likely reflects the model's sensitivity to local feature distances, which can be disrupted if relevant features are removed during FS.

The feature importance analysis confirms that FS effectively identified a reduced subset of high-impact features, e.g.: Sen_X_INTQ, Sen_Y_RANGE, and Esp_Y_PICO, which were consistently ranked as top contributors in tree-based models like GBC and

ET. The ability of FS to simplify the input space without significant loss of predictive performance is a major advantage, particularly in small-data scenarios typical of biomedical studies.

Another relevant insight is the nuanced impact of the tune function. While tuning was expected to enhance model performance, its benefits were not universal. In some trials, the tuning process led to a slight decrease in precision or increased standard deviation, suggesting that TF may introduce variance when the dataset is small or when models are already optimized by FS. Nonetheless, for certain models—especially ensemble-based ones like ET—the TF helped reduce misclassifications by refining decision boundaries.

Regarding fold strategy, trials with 3, 7, and 9 folds were particularly successful when FS was applied. The 7-fold setup emerged as the most consistent, balancing training and validation performance. This aligns with expectations in small datasets, where too many folds may lead to overfitting due to reduced training data per fold, and too few may result in unstable estimates.

A comparative analysis of the Confusion Matrix and learning dynamics further revealed model-specific strengths and weaknesses. GBC, despite strong precision, experienced a trade-off in recall after FS, indicating a tendency to miss some positive cases. ET, in contrast, showed more balanced improvements without sacrificing recall. KNN, while less accurate, demonstrated lower overfitting and a more gradual learning pattern, which may suggest a better generalization potential if more data were available.

These patterns were also visible in the Validation Curve, where GBC and ET benefited from FS through more stable performance across hyperparameter variations. KNN, although improved slightly with FS, remained more sensitive and less consistent.

Taken together, the findings of this study provide compelling evidence for the feasibility and effectiveness of combining low-cost, non-invasive data acquisition protocols with advanced computational techniques such as feature selection and Machine Learning classifiers for Parkinson's disease assessment.

Each component: protocol design, kinematic feature extraction, and classification — demonstrated not only technical robustness but also clinical potential. The ability to elicit, detect, and quantify motor symptoms with minimal resources positions this framework as a promising foundation for scalable, accessible solutions in remote healthcare, early screening, and long-term disease monitoring.

Nevertheless, future research with larger and more heterogeneous populations is essential to validate the generalization of the results and to further optimize model performance. Additionally, the integration of this approach into user-friendly mobile or web-based platforms could unlock practical deployment in real-world clinical settings. These promising outcomes set the stage for the next phase of research and development, reinforcing the value of interdisciplinary strategies in biomedical innovation.

7

Limitations of the study and future work

While the results of this study are encouraging, several limitations must be acknowledged, offering directions for future improvement and expansion.

First, the use of standard Full HD video resolution (1920×1080 during analysis) and a frame rate limited to 30 FPS was adopted for practical processing and hardware constraints. Although this setup yielded sufficient precision for the current analysis, higher video quality—such as ultra-high-resolution recording combined with frame rates above 60 FPS—could enhance the temporal and spatial resolution of motion trackings. This would be particularly beneficial in detecting smaller aimed object and rapid micro-movements or subtle motor anomalies that may go unnoticed under current settings due to motion blur.

Second, the tracking in this study was restricted to a single aimed object (typically the pen tip), limiting the range of extractable motion features. Future iterations of the protocol could incorporate multiple aimed objects on different parts of the hand or pen (e.g., knuckles, wrist, pen base), potentially enabling a more comprehensive analysis of motor control and coordination.

Another technical development that would significantly enhance accessibility and scalability is the creation of a web-based application. Such a platform would allow remote data upload, automatic DeepLabCut analysis, and Machine Learning-based classification, facilitating usage in decentralized settings such as telemedicine consultations or large-scale screening programs in resource-limited environments.

Regarding the dataset, two primary limitations concern its size and heterogeneity. While the number of participants provided valuable insights for a proof-of-concept study, it remains insufficient for robust generalization across the broader population. Expanding the volunteer pool and ensuring greater demographic and clinical diversity—such as varying disease stages, ages, and comorbidities—would strengthen the model's reliability and external validity.

In the classification phase, although multiple Machine Learning algorithms were tested with and without feature selection and tune function, the hyperparameter optimization

process was not exhaustively explored. Future studies should systematically investigate hyperparameter configurations using techniques like grid search or Bayesian optimization to identify the best-performing configurations for each model. Additionally, focusing on fine-tuning the most balanced model—identified through comparative performance and generalization metrics—could yield a highly reliable classifier tailored for real-world applications.

Finally, while the current study relied solely on video-based kinematic data, integrating additional data sources such as inertial measurement units, pressure sensors, or electromyography could provide a richer, multimodal representation of motor function. This fusion of data types may improve classification accuracy and offer more comprehensive insights into motor symptomatology.

In conclusion, addressing these limitations and implementing the proposed enhancements in future work can significantly advance the development of a robust, cost-effective, and clinically valuable tool for Parkinson's disease monitoring and diagnosis.

References

[1] Spain Tolosa, W Scholz, Eduardo Tolosa, Alicia Garrido, Sonja W Scholz, and Werner Poewe. *Challenges in the diagnosis of Parkinson's disease*. Tech. rep. 2021, pp. 385–97. URL: www.thelancet.com/neurology.

[2] Eduardo Tolosa, Gregor Wenning, and Werner Poewe. *Review Clinical differential diagnosis Essential tremor*. Tech. rep. Jan. 2006, pp. 75–86. URL: http://neurology.thelancet.com.

[3] Werner Poewe, Klaus Seppi, Caroline M. Tanner, Glenda M. Halliday, Patrik Brundin, Jens Volkmann, Anette Eleonore Schrag, and Anthony E. Lang. "Parkinson disease". In: *Nature Reviews Disease Primers* 3 (Mar. 2017), pp. 1–21. ISSN: 2056676X. DOI: 10.1038/nrdp.2017.13.

[4] Bastiaan R. Bloem, Michael S. Okun, and Christine Klein. "Parkinson's disease". In: *The Lancet* 397 (10291 June 2021), pp. 2284–2303. ISSN: 1474547X. DOI: 10.1016/S0140-6736(21)00218-X.

[5] Hussaini Adam. "An update on pathogenesis and clinical scenario for Parkinson's disease: diagnosis and treatment". In: 2023. DOI: .org/10.1007/s13205-023-03553-8.

[6] José Renato Munari Nardo, Daniel Hilário Silva, Caio Tonus Ribeiro, Adriano Alves Pereira, Luanne Cardoso Mendes, and Adriano de Oliveira Andrade. *On the application of DeepLabCut for the assessment of spiral and sinusoidal patterns in individuals with Parkinson's Disease*. Tech. rep. International Journal of Online and Biomedical Engineering, 2025.

[7] Daniel Hilário da Silva, Leandro Rodrigues da Silva Souza, Caio Tonus Ribeiro, Simone Hilário da Silva Brasileiro, José Renato Munari Nardo, Adriano Alves Pereira, and Adriano de Oliveira Andrade. "A Web Application for exploratory data analysis and classification of Parkinson's Disease patients using machine learning models on different datasets". In: *Software Impacts* 23 (Mar. 2025), p. 100737. ISSN: 2665-9638. DOI: 10.1016/j.simpa.2024.100737.

[8] Daniel Hilário da Silva, Caio Tonus Ribeiro, Leandro Rodrigues da Silva Souza, José Renato Munari Nardo, and Adriano Alves Pereira. "Exploring Essential Acoustic Features for Early Parkinson's Disease Classification: A Machine Learning Study". In: 21 (2 Feb. 2025), pp. 98–120. ISSN: 2626-8493. DOI: 10.3991/ijoe.v21i02.50503.

[9] Caio Tonus Ribeiro, Daniel Hilário da Silva, Leandro Rodrigues da Silva Souza, José Renato Munari Nardo, and Adriano Alves Pereira. "A Novel Validation Study of a Wrist Orthosis for the Objective Evaluation of Rigidity in Parkinson's Disease". In: *International Journal of Online and Biomedical Engineering (iJOE)* 20 (2024), pp. 90–108. ISSN: 2626-8493. DOI: 10.3991/ijoe.v20i12.50429.

[10] José Renato Munari Nardo, Caio Tonus Ribeiro, Camille Marques Alves, Daniel Hilário da Silva, Eduardo Moura Neto, Luanne Cardoso Mendes, Adriano Alves Pereira, and Adriano de Oliveira Andrade. "Patterns in Drawings of Parkinson's Disease Patients Versus Healthy People Utilizing Markerless Object Tracking with Machine Learning". In: *XXIX Brazilian Congress on Biomedical Engineering - Volume 3: Biomedical Informatics, and Biomedical Signal and Image Processing*. Ed. by Alcimar Barbosa Soares, Renata Ferranti Leoni, and George Cunha Cardoso. Vol. 3. Oct. 2025, pp. 232–242. ISBN: 9783031949333. DOI: 10.1007/978-3-031-94934-0_24.

[11] José Renato Munari Nardo, Caio Tonus Ribeiro, Daniel Hilário da Silva, Adriano Alves Pereira, and Leandro Rodrigues da Silva Souza. "An experimental protocol for video recording with a smartphone for use in a pre-trained neural network". In: (2023). DOI: 10.5281/ZENODO.10157450.

[12] Daniel Hilário da Silva, Caio Tonus Ribeiro, Adriano Alves Pereira, José Renato Munari Nardo, and Leandro Rodrigues Silva Souza. "A low-code machine learning library in Python applied to classify and interpret data of patients with Parkinson's disease using voice records". In: (2023), p. 1. DOI: 10.5281/ZENODO.10162024.

[13] Leonardo Palacios-Sánchez, Martha Torres Nupan, and Juan Sebastián Botero-Meneses. "James Parkinson and his essay on "shaking palsy", two hundred years later". In: *Arquivos de Neuro-Psiquiatria* 75 (9 Sept. 2017), pp. 671–672. ISSN: 16784227. DOI: 10.1590/0004-282X20170108.

[14] James Parkinson. *NEUROPSYCHIATRY CLASSICS An Essay on the Shaking Palsy Member of the Royal College of Surgeons PREFACE*. Tech. rep. 2002. DOI: 10.1176/jnp.14.2.223.

[15] Marios Politis, Kit Wu, Sophie Molloy, Peter G. Bain, K. Ray Chaudhuri, and Paola Piccini. "Parkinson's disease symptoms: The patient's perspective". In: *Movement Disorders* 25 (11 Aug. 2010), pp. 1646–1651. ISSN: 08853185. DOI: 10.1002/mds.23135.

[16] Zejin Ou, Jing Pan, Shihao Tang, Danping Duan, Danfeng Yu, Huiqi Nong, and Zhi Wang. "Global Trends in the Incidence, Prevalence, and Years Lived With Disability of Parkinson's Disease in 204 Countries/Territories From 1990 to 2019". In: *Frontiers in Public Health* 9 (Dec. 2021). ISSN: 22962565. DOI: 10.3389/fpubh.2021.776847.

[17] Guido Alves, Elin Bjelland Forsaa, Kenn Freddy Pedersen, Michaela Dreetz Gjerstad, and Jan Petter Larsen. "Epidemiology of Parkinson's disease". In: *Journal of Neurology*. Vol. 255. Sept. 2008, pp. 18–32. DOI: 10.1007/s00415-008-5004-3.

[18] R. Balestrino and A. H.V. Schapira. "Parkinson disease". In: *European Journal of Neurology* 27 (1 Jan. 2020), pp. 27–42. ISSN: 14681331. DOI: 10.1111/ene.14108.

[19] Melissa J. Armstrong and Michael S. Okun. "Diagnosis and Treatment of Parkinson Disease: A Review". In: *JAMA - Journal of the American Medical Association* 323 (6 Feb. 2020), pp. 548–560. ISSN: 15383598. DOI: 10.1001/jama.2019.22360.

[20] Anthony H.V. Schapira, K. Ray Chaudhuri, and Peter Jenner. "Non-motor features of Parkinson disease". In: *Nature Reviews Neuroscience* 18 (7 July 2017), pp. 435–450. ISSN: 14710048. DOI: 10.1038/nrn.2017.62.

[21] Daniel Weintraub, Dag Aarsland, Kallol Ray Chaudhuri, Roseanne D Dobkin, Albert Fg Leentjens, Mayela Rodriguez-Violante, and Anette Schrag. *The neuropsychiatry of Parkinson's disease: advances and challenges*. Tech. rep. 2022, p. 89. URL: www.thelancet.com/neurology.

[22] Houyam Tibar, Khalil El Bayad, Ahmed Bouhouche, El Hachmia Ait Ben Haddou, Ali Benomar, Mohamed Yahyaoui, Abdelhamid Benazzouz, and Wafa Regragui. "Non-motor symptoms of Parkinson's Disease and their impact on quality of life in a cohort of Moroccan patients". In: *Frontiers in Neurology* 9 (APR Apr. 2018). ISSN: 16642295. DOI: 10.3389/fneur.2018.00170.

[23] Nicole Ball, Wei Peng Teo, Shaneel Chandra, and James Chapman. "Parkinson's disease and the environment". In: *Frontiers in Neurology* 10 (Mar. 2019). ISSN: 16642295. DOI: 10.3389/fneur.2019.00218.

[24] Frank C. Church. "Review treatment options for motor and non-motor symptoms of parkinson's disease". In: *Biomolecules* 11 (4 Apr. 2021). ISSN: 2218273X. DOI: 10.3390/biom11040612.

[25] Jafar Alzubi, Anand Nayyar, and Akshi Kumar. "Machine Learning from Theory to Algorithms: An Overview". In: *Journal of Physics: Conference Series*. Vol. 1142. Institute of Physics Publishing, Nov. 2018. DOI: 10.1088/1742-6596/1142/1/012012.

[26] Pramila P. Shinde and Dr. Seema Shah. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. Literaturangaben. Piscataway, NJ: IEEE, Aug. 2018. ISBN: 9781538652572.

[27] Susmita Ray. *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing : trends, prespectives and prospects : COMITCON-2019 : 14th-16th February, 2019*. Literaturangaben. [Piscataway, NJ]: [IEEE], Feb. 2019. ISBN: 9781728102115.

[28] Alexander L. Fradkov. "Early history of machine learning". In: *IFAC-PapersOnLine*. Vol. 53. Elsevier B.V., 2020, pp. 1385–1390. DOI: 10.1016/j.ifacol.2020.12.1888.

[29] Batta Mahesh. "Machine Learning Algorithms - A Review". In: *International Journal of Science and Research (IJSR)* 9 (1 Jan. 2020), pp. 381–386. DOI: 10.21275/art20203995.

[30] Iqbal H. Sarker. "Machine Learning: Algorithms, Real-World Applications and Research Directions". In: *SN Computer Science* 2 (3 May 2021). ISSN: 26618907. DOI: 10.1007/s42979-021-00592-x.

[31] Darpan Pandey, Kamal Niwaria, and Bharti Chourasia. "Machine Learning Algorithms: A Review". In: *International Research Journal of Engineering and Technology* (2008). ISSN: 2395-0072. URL: www.irjet.net.

[32] S Rao, Kondaiah, G Chandra, and K Kiran Kumar. "International Conference on Innovative Research in Computer and Communication Engineering: A Survey on Machine Learning: Concept, Algorithms and Applications". In: vol. 5. Feb. 2017, pp. 1301–1309.

[33] Mohaiminul Islam, Guorong Chen, and Shangzhu Jin. "An Overview of Neural Network". In: *American Journal of Neural Networks and Applications* 5 (1 2019), p. 7. ISSN: 2469-7400. DOI: 10.11648/j.ajnna.20190501.12.

[34] Yu chen Wu and Jun wen Feng. "Development and Application of Artificial Neural Network". In: *Wireless Personal Communications* 102 (2 Sept. 2018), pp. 1645–1656. ISSN: 1572834X. DOI: 10.1007/s11277-017-5224-x.

[35] Ajay Shrestha and Ausif Mahmood. "Review of deep learning algorithms and architectures". In: *IEEE Access* 7 (2019), pp. 53040–53065. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2912200.

[36] Nisha.C.M and N. Thangarasu. "Deep learning algorithms and their relevance: A review". In: *International Journal of Data Informatics and Intelligent Computing* 2 (4 Nov. 2023), pp. 1–10. DOI: 10.59461/ijdiic.v2i4.78.

[37] Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning". In: *Nature Neuroscience* 21 (9 Sept. 2018), pp. 1281–1289. ISSN: 15461726. DOI: 10.1038/s41593-018-0209-y.

[38] Tanmay Nath, Alexander Mathis, An Chi Chen, Amir Patel, Matthias Bethge, and Mackenzie Weygandt Mathis. "Using DeepLabCut for 3D markerless pose estimation across species and behaviors". In: *Nature Protocols* 14 (7 July 2019), pp. 2152–2176. ISSN: 17502799. DOI: 10.1038/s41596-019-0176-0.

[39] PyCaret Contributors. *Quickstart | Docs*. June 2025. URL: https://github.com/pycaret/pycaret.

[40] Ulla Gain and Virpi Hotti. "Low-code AutoML-augmented data pipeline – A review and experiments". In: *Journal of Physics: Conference Series*. Vol. 1828. IOP Publishing Ltd, Mar. 2021. DOI: 10.1088/1742-6596/1828/1/012015.

[41] Sven Nõmm, Sergei Zarembo, Kadri Medijainen, Pille Taba, and Aaro Toomela. "Deep CNN Based Classification of the Archimedes Spiral Drawing Tests to Support Diagnostics of the Parkinson's Disease". In: *IFAC-PapersOnLine*. Vol. 53. Elsevier B.V., 2020, pp. 260–264. DOI: 10.1016/j.ifacol.2021.04.185.

[42] Md Ariful Islam, Md Ziaul Hasan Majumder, Md Alomgeer Hussein, Khondoker Murad Hossain, and Md Sohel Miah. "A review of machine learning and deep learning algorithms for Parkinson's disease detection using handwriting and voice datasets". In: *Heliyon* 10 (3 Feb. 2024). ISSN: 24058440. DOI: 10.1016/j.heliyon.2024.e25469.

[43] Eugenio Lomurno, Linda Greta Dui, Madhurii Gatto, Matteo Bollettino, Matteo Matteucci, and Simona Ferrante. "Deep Learning and Procrustes Analysis for Early Dysgraphia Risk Detection with a Tablet Application". In: *Life* 13 (3 Mar. 2023). ISSN: 20751729. DOI: 10.3390/life13030598.

[44] Zoltan Galaz, Peter Drotar, Jiri Mekyska, Matej Gazda, Jan Mucha, Vojtech Zvoncak, Zdenek Smekal, Marcos Faundez-Zanuy, Reinel Castrillon, Juan Rafael Orozco-Arroyave, Steven Rapcsak, Tamas Kincses, Lubos Brabenec, and Irena Rektorova. "Comparison of CNN-Learned vs. Handcrafted Features for Detection of Parkinson's Disease Dysgraphia in a Multilingual Dataset". In: *Frontiers in Neuroinformatics* 16 (May 2022). ISSN: 16625196. DOI: 10.3389/fninf.2022.877139.

[45] Riya Tyagi, Tanish Tyagi, Ming Wang, and Lujin Zhang. "Machine Learning for Real-Time, Automatic, and Early Diagnosis of Parkinson's Disease by Extracting Signs of Micrographia from Handwriting Images". In: (Nov. 2021). URL: http://arxiv.org/abs/2111.14781.

[46] Mohamad Wehbi, Daniel Luge, Tim Hamann, Jens Barth, Peter Kaempf, Dario Zanca, and Bjoern M. Eskofier. "Surface-Free Multi-Stroke Trajectory Reconstruction and Word Recognition Using an IMU-Enhanced Digital Pen". In: *Sensors* 22 (14 July 2022). ISSN: 14248220. DOI: 10.3390/s22145347.

[47] Maximilian Schrapel, Max Ludwig Stadler, and Michael Rohs. "Pentelligence: Combining pen tip motion and writing sounds for handwritten digit recognition". In: *Conference on Human Factors in Computing Systems - Proceedings*. Vol. 2018-April. Association for Computing Machinery, Apr. 2018. ISBN: 9781450356206. DOI: 10.1145/3173574.3173705.

[48] Sabyasachi Chakraborty, Satyabrata Aich, Jong-Seong-Sim, Eunyoung Han, Jinse Park, and Hee-Cheol Kim. *The 22nd International Conference on Advanced Communications Technology : "Digital Security Global Agenda for Safe Society!" : ICACT 2020 : Phoenix Park, Pyeongchang, Korea (South), Feb. 16 -19, 2020 : proceeding & jpurnal. International Conference on Advanced Communication Technology*. Literaturangaben. Piscataway, NJ, USA: IEEE, Feb. 2020. ISBN: 9791188428045.

[49] Muhammed Erdem Isenkul, Betul Erdogdu Sakar, and Betul Erdogdu Sakar Olcay Kursun. "Improved Spiral Test Using Digitized Graphics Tablet for Monitoring Parkinson's Disease". In: (2014). DOI: 10.13140/RG.2.1.1898.6005. URL: www.icehtm.net.

[50] Decho Surangsrirat and Chusak Thanawattano. "Android application for spiral analysis in Parkinson's Disease". In: *2012 Proceedings of IEEE Southeastcon*. IEEE, Mar. 2012, pp. 1–6. DOI: 10.1109/secon.2012.6196943.

[51] Marta San Luciano, Cuiling Wang, Roberto A. Ortega, Qiping Yu, Sarah Boschung, Jeannie Soto-Valencia, Susan B. Bressman, Richard B. Lipton, Seth Pullman, and Rachel Saunders-Pullman. "Digitized spiral drawing: A possible biomarker for early Parkinson's disease". In: *PLoS ONE* 11 (10 Oct. 2016). ISSN: 19326203. DOI: 10.1371/journal.pone.0162799.

[52] Megha Kamble, Prashant Shrivastava, and Megha Jain. "Digitized spiral drawing classification for Parkinson's disease diagnosis". In: *Measurement: Sensors* 16 (Aug. 2021). ISSN: 26659174. DOI: 10.1016/j.measen.2021.100047.

[53] Rachel Saunders-Pullman, Carol Derby, Kaili Stanley, ALicia Floyd, Susan Bressman, Richard B. Lipton, Amanda Deligtisch, Lawrence Severt, Qiping Yu, Mónica Kurtis, and Seth L. Pullman. "Validity of spiral analysis in Early Parkinson's disease". In: *Movement Disorders* 23 (4 Mar. 2008), pp. 531–537. ISSN: 08853185. DOI: 10.1002/mds.21874.

[54] Lulu Lv, Jiantao Yang, Fanbin Gu, Jingyuan Fan, Qingtang Zhu, and Xiaolin Liu. "Validity and Reliability of a Depth Camera–Based Quantitative Measurement for Joint Motion of the Hand". In: *Journal of Hand Surgery Global Online* 5 (1 Jan. 2023), pp. 39–47. ISSN: 25895141. DOI: 10.1016/j.jhsg.2022.08.011.

[55] Pin-Ling Liu and Chien-Chi Chang. "Simple method integrating OpenPose and RGB-D camera for identifying 3D body landmark locations in various postures". In: *International Journal of Industrial Ergonomics* 91 (Sept. 2022). Dentro da proposta, p. 103354. ISSN: 01698141. DOI: 10.1016/j.ergon.2022.103354. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169814122000956.

[56] Javad Abbasi, Hassan Salarieh, and Aria Alasty. "A motion capture algorithm based on inertia-Kinect sensors for lower body elements and step length estimation". In: *Biomedical Signal Processing and Control* 64 (Feb. 2021). ISSN: 17468108. DOI: 10.1016/j.bspc.2020.102290.

[57] Anne Schmitz, Mao Ye, Robert Shapiro, Ruigang Yang, and Brian Noehren. "Accuracy and repeatability of joint angles measured using a single camera markerless motion capture system". In: *Journal of Biomechanics* 47 (2 Jan. 2014), pp. 587–591. ISSN: 00219290. DOI: 10.1016/j.jbiomech.2013.11.031.

[58] Saman Vafadar, Wafa Skalli, Aurore Bonnet-Lebrun, Ayman Assi, and Laurent Gajny. "Assessment of a novel deep learning-based marker-less motion capture system for gait study". In: *Gait and Posture* 94 (May 2022), pp. 138–143. ISSN: 18792219. DOI: 10.1016/j.gaitpost.2022.03.008.

[59] Jinbao Wang, Shujie Tan, Xiantong Zhen, Shuo Xu, Feng Zheng, Zhenyu He, and Ling Shao. "Deep 3D human pose estimation: A review". In: *Computer Vision and Image Understanding* 210 (Sept. 2021). ISSN: 1090235X. DOI: 10.1016/j.cviu.2021.103225.

[60] Edward P. Washabaugh, Thanikai Adhithiyan Shanmugam, Rajiv Ranganathan, and Chandramouli Krishnan. "Comparing the accuracy of open-source pose estimation methods for measuring gait kinematics". In: *Gait and Posture* 97 (Sept. 2022), pp. 188–195. ISSN: 18792219. DOI: 10.1016/j.gaitpost.2022.08.008.

[61] Alexander M. Aurand, Jonathan S. Dufour, and William S. Marras. "Accuracy map of an optical motion capture system with 42 or 21 cameras in a large measurement volume". In: *Journal of Biomechanics* 58 (June 2017), pp. 237–240. ISSN: 18732380. DOI: 10.1016/j.jbiomech.2017.05.006.

[62] Saman Vafadar, Wafa Skalli, Aurore Bonnet-Lebrun, Marc Khalifé, Mathis Renaudin, Amine Hamza, and Laurent Gajny. "A novel dataset and deep learning-based approach for marker-less motion capture during gait". In: *Gait and Posture* 86 (May 2021), pp. 70–76. ISSN: 18792219. DOI: 10.1016/j.gaitpost.2021.03.003.

[63] Trent M. Guess, Rebecca Bliss, Jamie B. Hall, and Andrew M. Kiselica. "Comparison of Azure Kinect overground gait spatiotemporal parameters to marker based optical motion capture". In: *Gait and Posture* 96 (July 2022), pp. 130–136. ISSN: 18792219. DOI: 10.1016/j.gaitpost.2022.05.021.

[64] Veronica de Lima Gonçalves, Caio Tonus Ribeiro, Guilherme Lopes Cavalheiro, Maria José Ferreira Zaruz, Daniel Hilário da Silva, Selma Terezinha Milagre, Adriano de Oliveira Andrade, and Adriano Alves Pereira. "A hybrid linear discriminant analysis and genetic algorithm to create a linear model of aging when performing motor tasks through inertial sensors positioned on the hand and forearm". In: *BioMedical Engineering Online* 22 (1 Dec. 2023). ISSN: 1475925X. DOI: 10.1186/s12938-023-01161-4.

[65] Donato Impedovo. "Velocity-Based Signal Features for the Assessment of Parkinsonian Handwriting". In: *IEEE Signal Processing Letters* 26 (4 Apr. 2019), pp. 632–636. ISSN: 10709908. DOI: 10.1109/LSP.2019.2902936.

[66] Sascha Gruss, Roi Treister, Philipp Werner, Harald C. Traue, Stephen Crawcour, Adriano Andrade, and Steffen Walter. "Pain intensity recognition rates via biopotential feature patterns with support vector machines". In: *PLoS ONE* 10 (10 Oct. 2015). ISSN: 19326203. DOI: 10.1371/journal.pone.0140330.

[67] Günther Deuschl, Michael Lauk, and Jens Timmer. "Tremor classification and tremor time series analysis". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 5 (1 Mar. 1995), pp. 48–51. ISSN: 10541500. DOI: 10.1063/1.166084.

[68] Stephen Bates, Trevor Hastie, and Robert Tibshirani. "Cross-validation: what does it estimate and how well does it do it?" In: *Journal of the American Statistical Association* 119 (May 2023), pp. 1434–1445. ISSN: 1537-274X. DOI: 10.1080/01621459.2023.2197686.

[69] Benyamin Ghojogh, Mark Crowley, Bghojogh@uwaterloo Ca, and Mcrowley@uwaterloo Ca. "The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial". In: (May 2019). DOI: 10.48550/arXiv.1905.12787. URL: https://www.researchgate.net/publication/333505702.

[70] Rana Zia Ur Rehman, Silvia Del Din, Yu Guan, Alison J. Yarnall, Jian Qing Shi, and Lynn Rochester. "Selecting Clinically Relevant Gait Characteristics for Classification of Early Parkinson's Disease: A Comprehensive Machine Learning Approach". In: *Scientific Reports* 9 (1 Dec. 2019). ISSN: 20452322. DOI: 10.1038/s41598-019-53656-7.

[71] A. Yasar, I. Saritas, M. A. Sahman, and A. C. Cinar. "Classification of Parkinson disease data with artificial neural networks". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 675. Institute of Physics Publishing, Nov. 2019. DOI: 10.1088/1757-899X/675/1/012031.

[72] Akhil Garg and Kang Tai. *Comparison of statistical and machine learning methods in modelling of data with multicollinearity*. Tech. rep. 2013, pp. 295–312. DOI: 10.1504/ijmic.2013.053535.

[73] Moez Ali. *PyCaret: An open source, low-code machine learning library in Python*. Apr. 2024. URL: https://www.pycaret.org.

[74] Ibrahim Karabayir, Samuel M. Goldman, Suguna Pappu, and Oguz Akbilgic. "Gradient boosting for Parkinson's disease diagnosis from voice recordings". In: *BMC Medical Informatics and Decision Making* 20 (1 Sept. 2020). ISSN: 14726947. DOI: 10.1186/s12911-020-01250-7.

[75] Divya Jain and Vijendra Singh. "Feature selection and classification systems for chronic disease prediction: A review". In: *Egyptian Informatics Journal* 19 (3 Nov. 2018), pp. 179–189. ISSN: 11108665. DOI: 10.1016/j.eij.2018.03.002.