LINKING EARLY DETECTION/TREATMENT OF PARKINSON'S DISEASE USING DEEP LEARNING TECHNIQUES

Sarah Fan[1] and Yu Sun[2]

[1]Sage Hill School, 20402 Newport Coast Dr, Newport Beach, CA 92657
[2]California State Polytechnic University, Pomona, CA, 91768, Irvine, CA 92620

ABSTRACT

Parkinson's Disease (PD) is a progressive neurodegenerative disorder that causes uncontrollable movements and difficulty with balance and coordination. It is highly important for early detection of Parkinson's Disease for patients to receive proper treatment. This paper aims to present a preliminary data mining procedure that help Parkinson's Disease patients slow down their progression of the disease while helping early detection of the disease. For early non-invasive treatment, our research first analyses the early symptoms of Parkinson's Disease, designs/selects a proper demo video, let the user follow the demo to exercise and upload his exercise video to our deep learning APP: LaBelle. LaBelle utilizing MediaPipe Pose to identify, analyze, and store data about the poses and movements of both demo and the user, calculates the angles created between different joints and major body parts. LaBelle's AI model uses a K-means clustering algorithm to create a group of clusters for both demo and the user dataset. Using the two sets of clusters, LaBelle identifies the key frames in the user video and searches the demo cluster set for a matching set of properties and frames. It evaluates the differences between the paired frames and produces a final score as well as feedback on the poses that need improving. Meanwhile, if the user is willing to donate their exercise data, he can simply input his age, whether he is a PD patient (maybe for how long) anonymously. Then his data can be stored into our customized dataset, used in data mining for Parkinson's Disease prediction, which involves building/training our deep learning CNN model and help early detection of Parkinson's Disease.

KEYWORDS

Deep Learning, K-means Clustering, Computer Vision, Parkinson's Disease, Data Mining.

1. INTRODUCTION

Parkinson's Disease (PD) is a progressive disorder of the nervous system marked by tremors, muscular rigidity, and slow, imprecise movement, chiefly affecting middle-aged and elderly people [1]. It is associated with degeneration of the brain's basal ganglia and a deficiency of the neurotransmitter dopamine. Worldwide, around 7-10 million people have Parkinson's Disease [2], making it highly important to diagnose PD accurately in the early stage so that patients can receive proper treatment [3]. Parkinson's disease (PD) is difficult to diagnose, particularly in its early stages, because the symptoms of other neurologic disorders can be similar to those found in PD. Meanwhile, early non-motor symptoms of PD may be mild and can be caused by many other conditions. Therefore, these symptoms are often overlooked, making the diagnosis of PD at an early stage more challenging [4]. To address these difficulties and refine the early detection of PD, different neuroimaging techniques (such as magnetic resonance imaging (MRI), computed tomography (CT) and positron emission tomography (PET)) and deep learning-based analysis methods have been developed [5].

DOI:10.5121/ijdkp.2022.12601

Data mining techniques[6] have taken a significant role in the early diagnosis and prognosis of many health diseases, including Parkinson’s Disease. Data mining is the process of extracting previously unknown knowledge from a large collection of data. The knowledge mined is often referred to as patterns or models, such as clusters, classification rules, linear models, etc. Data mining involves many stages including data collection, data processing, classifier build, prediction etc. For all different stages work together in a cohesive way, it needs huge resource and funds. The progressive in nature gives data mining for PD even more difficulties because all the tasks are time-consuming.

Meanwhile, once the patient is detected with PD, early non-invasive exercises/treatment is very helpful. Movement, especially exercises that encourage balance and reciprocal patterns (movements that require coordination of both sides of the body), can slow down the progression of the disease. Deep learning methods, especially in computer vision, are beginning to gain more and more popularity as technology advances their ability to produce accurate models. Human Pose Estimation (HPE)[7] addresses this particularly interesting area of deep learning research. It has versatile real-world application in healthcare.

In this research, we are trying to link the early detection and treatment for PD using deep learning techniques because these two stages can help each other. The early detection can let us provide the uses with some proper exercises to slow down the progression of the disease while PD patient’s exercise progress can help to build/train our AI model to make early detection more precise.

The rest of the paper is organized as follows: Section 2 describes our research background, direction, and the crucial elements needed in our research in detail; Section 3 presents our approach to solve the problem and relevant details about the experiment we did; Section 4 presents the results and analysis; Section 5 gives a brief summary of other work that tackles a similar problem; finally, Section 6 gives the conclusion remarks and discusses the future work of this project.

2. CHALLENGES

Data mining is the process of analyzing large blocks of information to identify patterns, to extract valuable information. The knowledge mined is often referred to as patterns or models, such as clusters, classification rules, linear models, and time-trends. Figure 1 shows the steps of data mining. Successful data mining takes all the resources, cooperation, and funds to work together and takes many iterations. This is extremely difficult because not only so many different parties are involved but also a huge amount of data is needed.

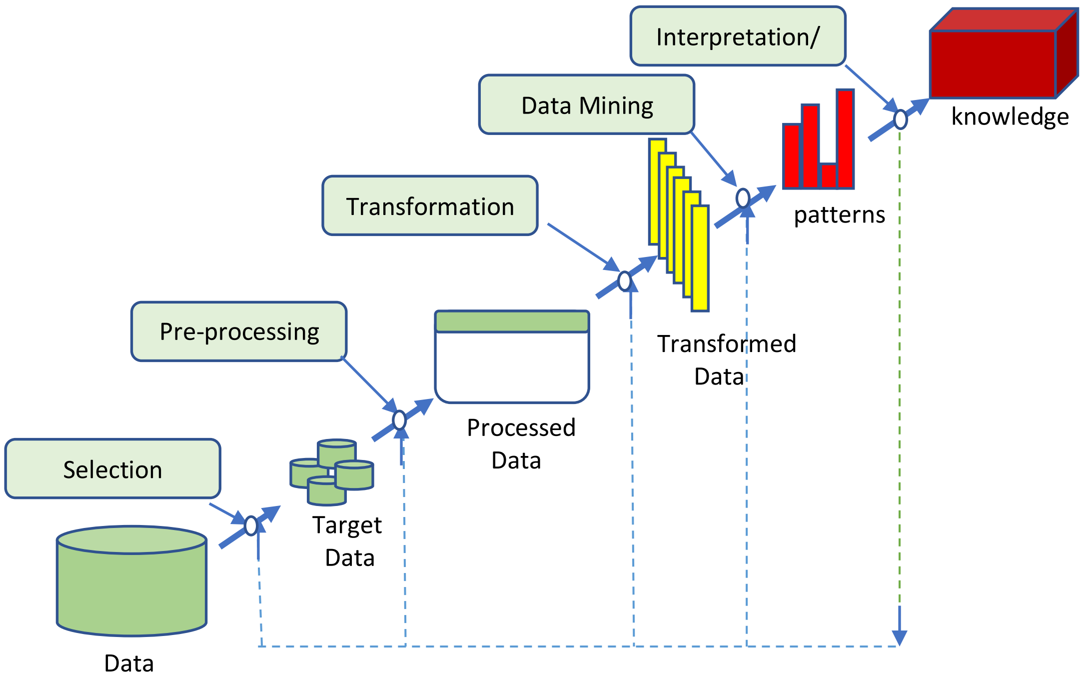

Figure 1. Steps of Data Mining (Source: http://www2.uregina.ca/).

2.1. Machine learning

Machine learning is a subset of data mining that encompasses supervised and unsupervised learning techniques. It is the technology of developing systems that can learn and draw inferences from patterns in data which can be applied to many different fields, from data analytics to predictive analytics, from service personalization to natural language processing, and so on [8].
According to Shalev-Shwartz, S. et al. [9], Machine learning can be defined as "using the experience to gain expertise." The learning could be supervised learning, unsupervised learning, etc. Supervised learning is the most common approach and is the approach we utilize in our research.
Supervised learning algorithms try to model relationships between the target prediction output and input features to predict output values for new data based on the relationships learned from the prior data sets. This type of learning is normally related to classification tasks, which is the process of teaching a classifier the relationship between the model's input and output to use this expertise later for un-seen input [10].

2.2. Deep Learning

Because machine learning is unable to meet the requirements due to the complexity of the problems in certain areas, Deep Learning (DL) is gaining popularity due to its supremacy in terms of accuracy. It is an advanced level of machine learning which includes a hierarchical function that enables machines to process data with a nonlinear approach. The deep learning networks are built with neuron nodes connected like the human brain and have many layers, each layer receiving information from the previous layer, trained to perform the desired tasks, and then passing on the information to the next layer [11].

2.3. Convolutional Neural Network

A convolutional neural network (CNN) can be made up of many layers of models, where each layer takes input from the previous layer, applies a filter to the data, and outputs it to the next layer. CNNs run much faster on GPU, and the huge stockpiles of data that have been collected can improve the accuracy of computer vision and NLP algorithms. A CNN consists of several convolutional layers, with each layer including three major stages: convolution, nonlinear activation (non linearity transform), and pooling (sub-sampling) [12].

2.4. Datasets

Datasets are fundamental in a deep learning system. An extensive and diverse dataset is a crucial requirement for the successful training of a deep neural network. In our research, we explore different CNNs using datasets we downloaded from HandPD [13].

3. SOLUTION

Figure 2 shows our solution. By study early "Parkinsonism", also known as "Parkinsonian Syndrome", we plan demo exercises through data collection preparation. Then using our mobile APP: LaBelle, user can mimic/learn basic movements that will help calm the shakiness/tremor, relax the stiffness, mitigate the slow down movement and improve user's ability to maintain balance. By doing the exercises, the user can record the history scores of their exercise to track their progress. Meanwhile, if they are willing to share their information anonymously, they can input their age, whether they are Parkinson's Disease patients, this information can be collected and further used to build/update/train deep learning model to help early detection of Parkinson's Disease.

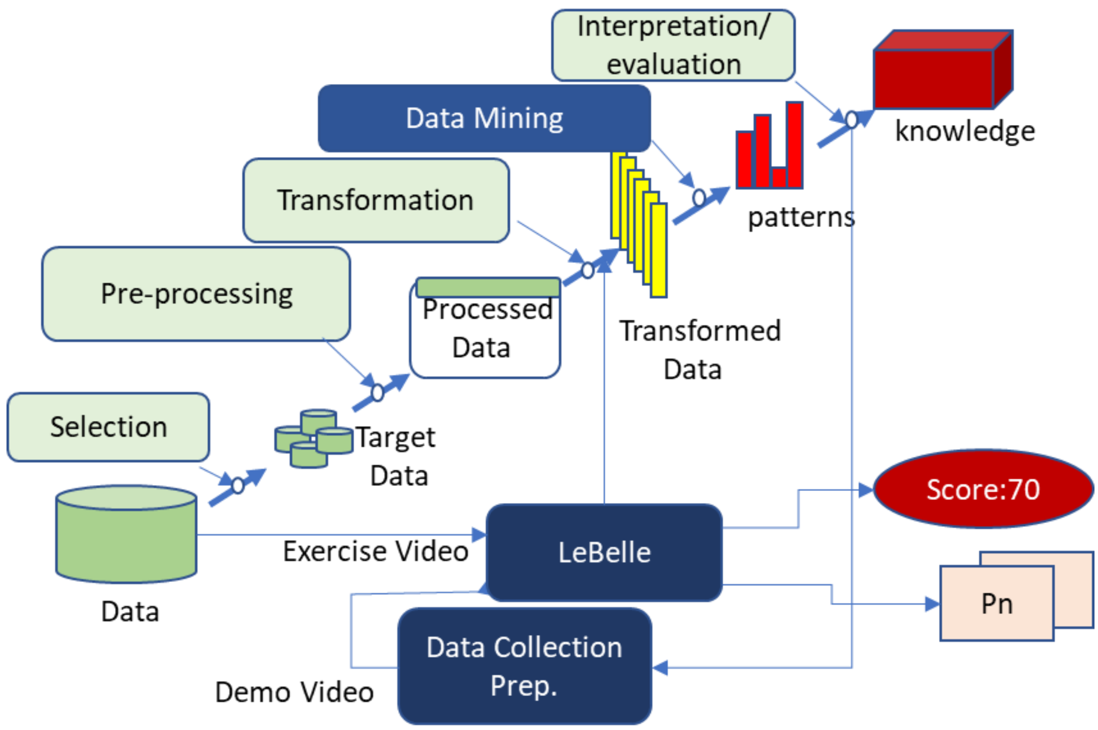

Figure 2. Our solution

3.1. Data Collection Planning

Parkinson's Disease is a progressive and neurodegenerative disease. It involves the loss of dopamine neuron, which make a little signalling chemical. When there is a reduction in dopamine, we will see the many movement signs of Parkinson's Disease.

MediaPipe developed by Google offers cross-platform, customizable machine learning solutions for live and streaming media. It has a lot of real time detection features. These features are examples of machine learning can be helpful in data collection. Table 1 shows some of early symptoms of PD. Not all the symptoms, but sooner or later, one or more symptoms will appear. For this research, we pick whole body related symptoms and use MediaPipe Pose as our main real time detection solution.

Symptom	Example	Real Time Detection	Demo Exercises
Shakiness	Hard to pick up small objects	MediaPipe Hands	Hand exercises for PD
Tremor	Hand tremor	MediaPipe Hands	Hand exercises for PD
Stiffness	Difficult to bend arm/leg	MediaPipe Pose	Sit down, get up, pick up plate from desk
Shuffling Steps	Small step length	MediaPipe Pose	Turn around exercises
Stooped posture	Difficult to have good posture	MediaPipe Pose	Posture Exercises
Slow movement	Slow Down Movement	MediaPipe Pose	Straight Walking
Masked Faces	diminished facial expressivity	MediaPipe Face Mesh	Facial Exercises
Asymmetric	Arms swing differently while walking	MediaPipe Pose	Walking Exercises
Freeze	Sudden stop while walking	MediaPipe Pose	Walking Exercises
Balance Trouble	Difficult to keep balance	MediaPipe Pose	Walking Exercises

Table 1. Data Collection Preparation

3.2. A Deep Learning APP: LaBelle

Our solution to the data collection is a deep learning App: LaBelle. LaBelle gives one demo exercise clip, takes in one of a user. LaBelle is structured into seven layers of processing. The first layer (L1) consists of two Mediapipe Pose models (M1 and M2) to process the demo and input video clip. The second layer (L2) consists of two blocks (B1 and B2) that calculate the major angles for important joints in the skeleton-based demo and exerciser model. The third layer(L3) consists of two blocks (C1 and C2) that group the angle-based properties into clusters for both demo and APP user. The fourth layer (L4) consists of one block (D11) to detect the key frames for APP user's video clip. The fifth layer (L5) consists of one block (D12) to find the centroids' feature property for each cluster of user's video clip based on the starting and ending frame. The sixth layer (L6) has one block F2 to find the matching property in the demo video clip for each centroid's feature property generated from the user's video. The last layer (L7) has one block S12 to compare matching properties of demo and user, generating a score to send to the user along with the frames corresponding to the centroids bearing lowest score to improve on.

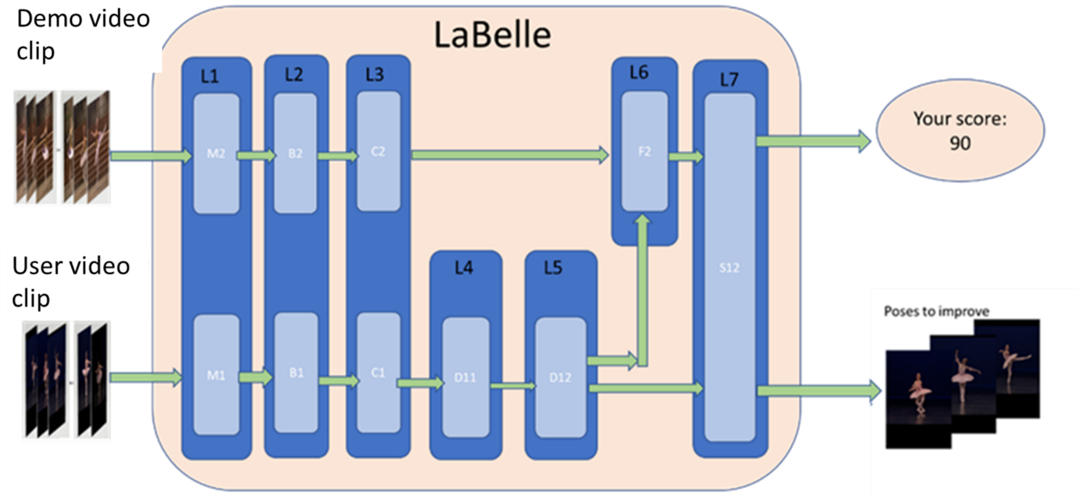

Figure 3. Data Collection: LaBelle

3.2.1. MediaPipe Pose (L1 Layer: M1 and M2)

Mediapipe is a cross-platform library developed by Google. It provides high quality ready-to-use Machine Learning solutions for computer vision tasks. Mediapipe Pose is one of the solutions we use for high-accuracy body pose detection and tracking [15]. The Pose Landmark Model (BlazePose GHUM 3D) allows us to predict the location of 33 pose landmarks.

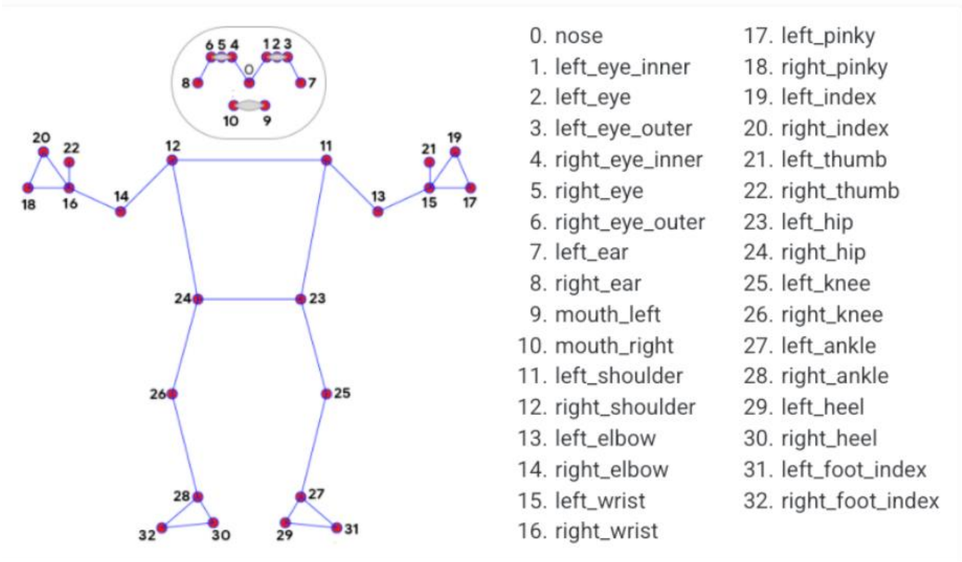

Figure 4. MediaPipe Pose Landmarks [6]

3.2.2. Property Extraction: Angle Calculation (L2 Layer: B2 and B1)

Mediapipe Pose detects/tracks 33 landmarks, outputs 33 landmark connection. However, to improve the performance, we define a set of key angles between two landmark connections. For example, we use (∠16, 14 ,12) for the right elbow. We use (∠14, 12, 24) for the right shoulder. We use (∠24, 26, 28) for the right knee. Same for the left side of the body. As shown in Figure 5 and Figure 6, we use Mediapipe's built in libraries to plot the human's pose on a 3D graph.

The coordinates of 3 adjacent joints are called into the plot_angle function of our algorithm, which calculates and returns the angle between the three landmarks. This process is repeated until all desired joint angles are returned. This process is carried out for each video frame (or every other, depending on how many frames are present), resulting in a final tuple array consisting of all major body angles for individual video frames.

```
angle, image = plot_angle(mp_pose.PoseLandmark.LEFT_SHOULDER.value,
                          mp_pose.PoseLandmark.LEFT_ELBOW.value,
                          mp_pose.PoseLandmark.LEFT_WRIST.value, landmarks, image, h, w + val)
```

Figure 5. MediaPipe code implementing key points into angle calculation

```
def plot_angle(p1, p2, p3, landmarks, image, h, w):
    # Get coordinates
    a = [landmarks[p1].x,
         landmarks[p1].y]
    b = [landmarks[p2].x, landmarks[p2].y]
    c = [landmarks[p3].x, landmarks[p3].y]

    # Calculate angle
    angle = calculate_angle(a, b, c)
    # print(angle)
    draw_angle(tuple(np.multiply(b, [w, h]).astype(int)), image, round(angle))
    return angle, image
```

Figure 6. MediaPipe code to generate angles given key joints

3.2.3. K-means Clustering (L3 Layer: C2 and C1)

We limit our demo video length to about 3 minutes. For each second, we can generalize about 30 frames. That's approximately 5400 frames in total. For the users' video, 30 second long, that's approximately 900 frames. We want to cluster our large number of frames into similar groups based on their properties, using our key angles as properties. The videos uploaded by the user will change each time, so it's impossible to preset cluster/centroid properties. To solve this problem, we need an unsupervised and autonomous clustering algorithm.

K-means (Macqueen, 1967)[15] is widely used in unsupervised learning algorithms that tackle clustering tasks. It uses "centroids", different randomly initiated points in the data, and assigns every data point to the nearest centroid (distance is dependent on similarity between data point and centroid), Once every data point has been assigned, the centroid is adjusted to the average of all the points assigned to it. Sklearn.cluster.KMeans works with KMeans using Lloyd's or Elkan's algorithm.

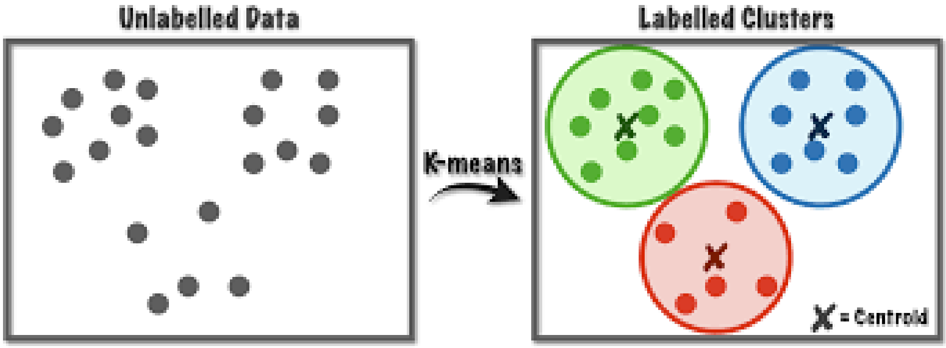

Figure 7. K-means Clustering: unsupervised clustering algorithm

3.2.4. K-mean Hyper Parameter Tuning using Silhouette Score

Both video clips need to be processed using clustering. To determine the optimal number of clusters for both video clips, we use silhouette scoring to tune the hyper parameters. One way we can do this is by assigning a random number of clusters ranging from 3 to 30 together with random initial centroid values. Alternatively, we can assign meaningful initial centroid values: by studying demo exercise and movements.

3.2.5. Key Frame Identification (L4 and L5 Layer: D11 & D12)

Based on the clusters generated from the user's video clip, each cluster will determine its own key frame, then the middle frame is identified and used as key frame.

3.2.6. Compare and Generate Final Score (L6 and L7 Layer: F2 & C12)

To compare the two videos, we go through the label 11 of all the centroids in the user video clip and find the modified label 12 by fitting the centroids within the demo clusters [16]. Then we find of the most similar property from demo video clip, producing the most similar poses for both user and demo, allowing us to calculate the score. Show in Figure 8 is part of the algorithm we use to find the most similar video frame, given a predetermined video frame. We use matrix norms from linear algebra to help us find the smallest distance (least difference between frames). We conclude the final score and output the pose that generates the lowest score, which means least matching, for user to improve in the future practice.

```
def get_nearest_neighbor(image, indexes, frames):
    a = np.array(image)
    min_dist = sys.maxsize
    nearest = indexes[0]
    for idx in indexes:
        b = np.array(frames[idx])
        dist = np.linalg.norm(a - b)
        if min_dist > dist:
            nearest = idx
            min_dist = dist
            # print(min_dist, nearest)
    return nearest
```

Figure 8. Portion of algorithm used to find the two most similar frames

3.3. Data Mining

The problem this research trying to solve can be summarized as the following

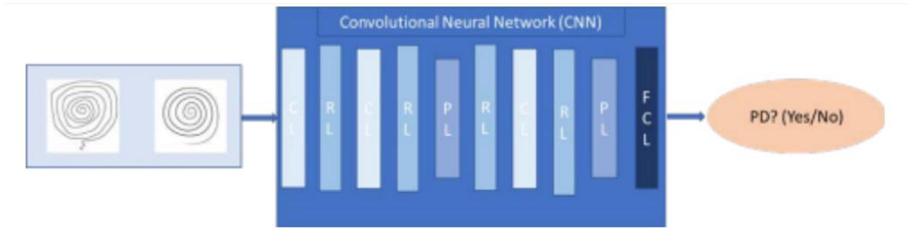

Figure 9. Data Mining CL: Convolutional Layer, FCL: Fully Connected Layer, PL: Pooling Layer, RL: ReLU Layer

Our model consists of 2 convolutional layers, 2 max-pooling layers, 2 dropout layers, 2 dense layers, and one flattened layer. All the activation functions are ReLU. Figure 10 shows our model. We chose this particular CNN architecture since it gives good results [17] [25].

The dropout layer is a technique introduced by Srivastava et al. [20]. This layer aims to avoid overfitting by randomly ignoring randomly some neurons from the previous layer. We inserted the dropout layers to improve the performance of our model.

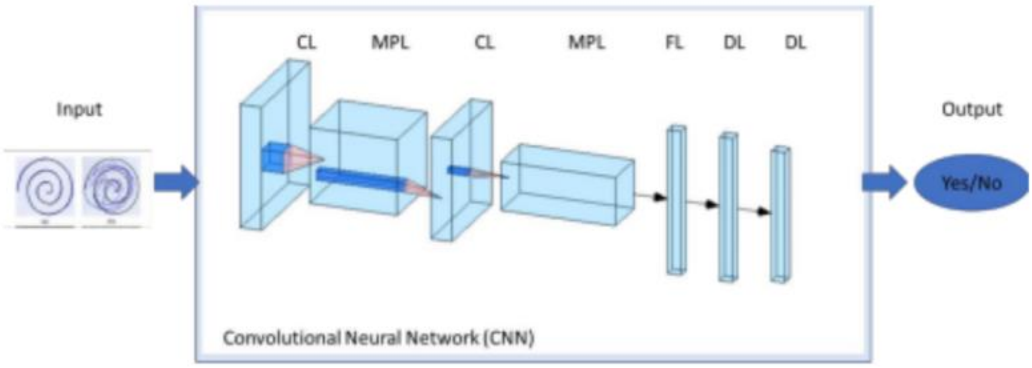

Figure 10. The Proposed CNN Architecture

The data Even though we published our deep learning APP LaBelle at Google Play, at this time, we have the algorithm and flow ready, but we don't have not collected enough data to build and train a CNN model. Instead, to verify our methodology, we choose to use dataset from the open library.

The dataset consists of hand drawn images (spiral/meander/wave) drawn by healthy people and Parkinson's disease patients. The model learns through training and uses a CNN to predict whether the person who drew the image has Parkinson's disease or not. The CNN model has one convolutional layer in front, a fully connected layer at the end, and a variable number of convolutional layers, max-pooling layers, and ReLU layers in between.

We downloaded HandPD dataset from [13]. The dataset contains 92 individuals, divided into 18 healthy people (Healthy Group) and 74 patients (Patients Group). Some examples are shown below. The brief description is the following:

- Healthy Group: 6 male and 12 female individuals with ages ranging from 19 to 79 years old (average age of 44.22±16.53 years). Among those individuals, 2 are left-handed and 16 are right-handed.
- Patient Group: 59 male and 15 female individuals with ages ranging from 38 to 78 years old (average age of 58.75±7.51 years). Among those individuals, 5 are left-handed and 69 are right-handed.

Therefore, each spiral and meander dataset is labeled in two groups: the healthy group containing 72 images, and the patient group containing 296 images. The images are labeled as follows: ID_EXAM-ID_IMAGE.jpg, in which ID_EXAM stands for the exam's identifier, and ID_IMAGE denotes the number of the image of the exam.

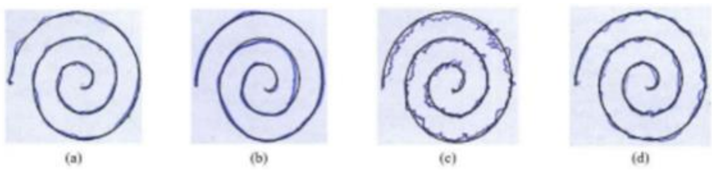

Figure 11 Some examples of spirals extracted from the HandPD dataset [14]

Figure 11 shows (a) 58-year-old males (b) 28-year-old female individuals of the control group, (c) 56-year-old males, and (d) 65 -year old female individuals of the patient group.

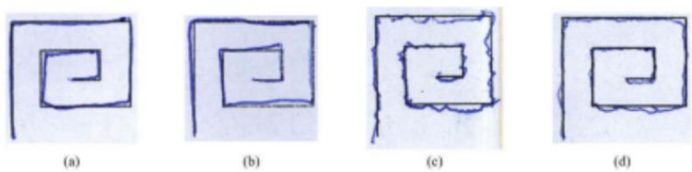

Figure 12. Some examples of meanders extracted from HandPD dataset [14]

Figure 12 shows (a) 58-years old male (b) 28-years old female individuals of a control group and (c) 56-years old mail and (d) 65-years old female individuals of a patient group.

All the data in HandPD dataset is in *.jpg format. For the exploration, we did some pre-processing including resizing, blurring, eroding, diluting, and color space converting (cv2.cvtColor() method).

4. EXPERIMENT

4.1. Experiment For Data COLLECTION: Labelle

In this section we give some examples with numerical and real data sets to demonstrate the performance of the proposed k-means algorithm. We show these unsupervised learning behaviours to get the best number of clusters for our algorithm to matching the video between demo and user. The dataset we are using is from the real data we collected through the training. To find the best K to using in the production level we try to train the dataset for both 40 k-means random state and 80 random states, run several k-means, increment k with each iteration, and record the SSE.

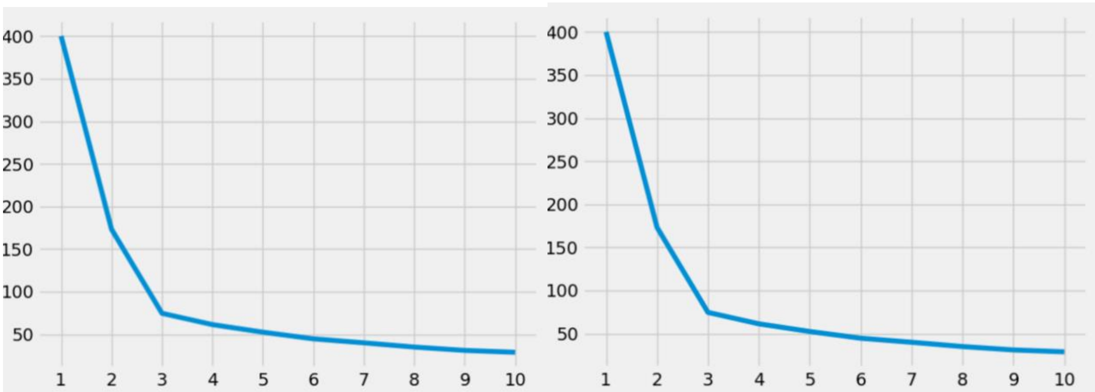

Figure 13. SSE curve with N Cluster changes for tested with 40 and 80 random states

The diagram shows K bigger than 3 has the acceptable SSE control. The silhouette coefficient is a measure of cluster cohesion and separation. It quantifies how well a data point fits into its assigned cluster. instead of computing SSE, compute the silhouette coefficient, to determine the best k for our algorithm:

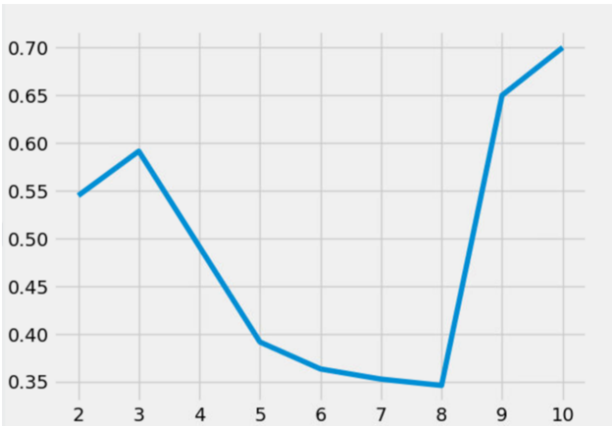

Figure 14. Silhouette scores for k shows that the best k is 10 since it has the maximum score

Ground truth labels categorize data points into group. These types of metrics do their best to suggest the correct number of clusters. To determine the best number of clusters, we fitted our dataset and get the plot.

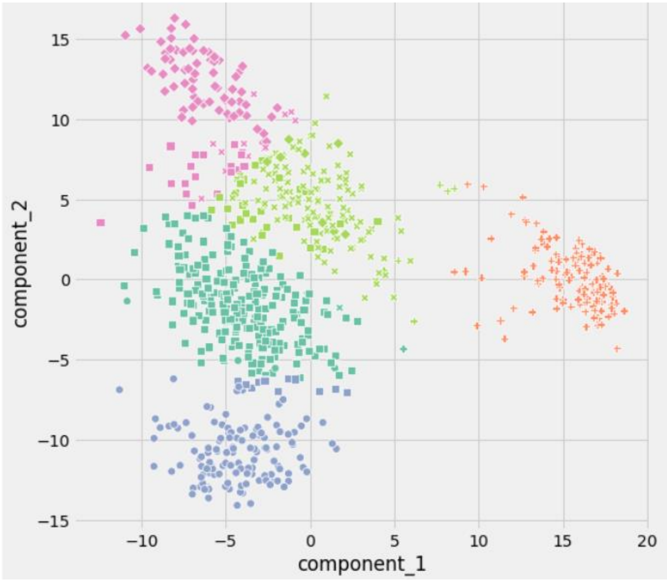

Figure 15. Ground truth labels for clusters 2, 4, 6, 8, 10

Ground truth labels also shows that cluster with 10 (orange points) has the best error value. The comparison shows that clusters with 8 has the best result to matching the user video and demo video.

4.2. Experiment for Data Mining

Figure 16 shows the training and validation accuracy and loss. The dataset we used is the spiral dataset downloaded from HandPD [13] without pre-processing. The CNN we used is the one shown in Figure 10. As we can see, it has a severe overfitting problem. To resolve this issue, we added dropout [20] after max-pooling. Figure 17 shows the validation accuracy and loss after dropout was added, preventing the model from overfitting and minimizing validation loss.

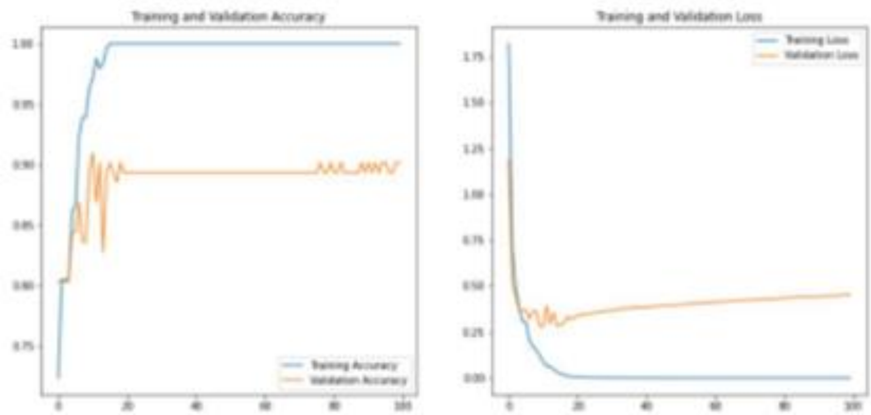

Figure 16. Pre-processing spiral data from HandPD using the model shown in Figure 10

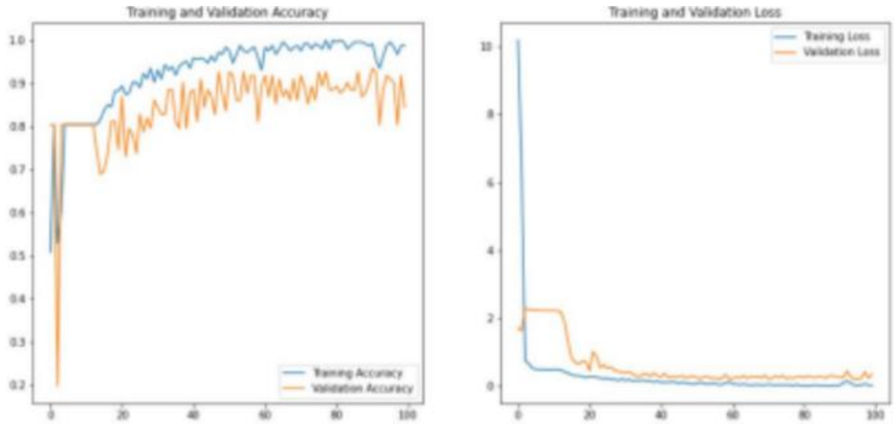

Figure 17. Spiral data from HandPD with two dropout layers added after max-pooling

To see the effects of different drawing patterns, we used two different patterns from the same dataset with the same CNN, with dropout added after max-pooling. Figure 17 uses spiral data from HandPD while Figure 18 uses meander data from HandPD. Comparing Figure 17 and Figure 18 we can see that both patterns generate similar validation accuracy and validation loss results, with the spiral slightly more accurate.

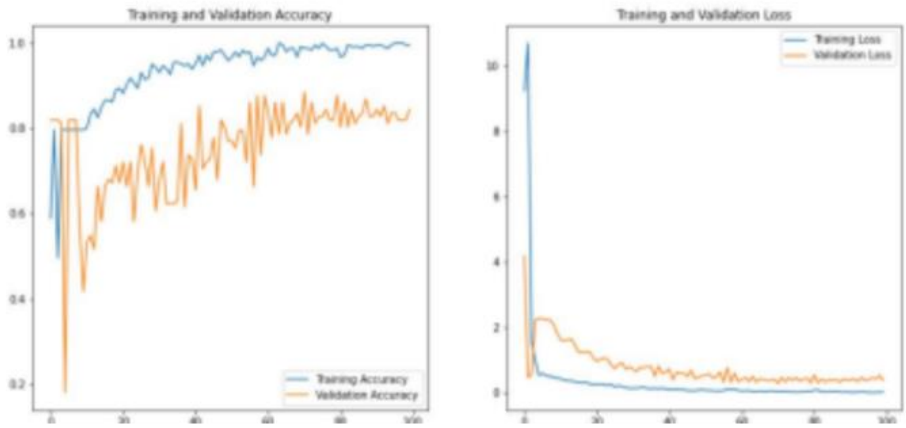

Figure 18. Meander data from HandPD with two dropout layers added after max-pooling

5. Related Work

Several researchers have worked on the K-means algorithm. In the paper “Unsupervised K-means clustering algorithm.", Sinaga, Kristina P., and Miin-Shen Yang construct an unsupervised learning schema for the k-means algorithm so that it is free of initialization without parameter selection [21]. They also simultaneously find an optimal number of clusters. In the paper The k-means Algorithm: A Comprehensive Survey and Performance Evaluation, Syed Islam explain the advance of k-means algorithm and discuss the using of comparison among different k-means clustering algorithms.

In the paper "Real-time comparison of movement of Bulgarian folk dances.", Uzunova, Zlatka. explain a method recording the dancing with two Kinetic sensors and analyze the data with machine learning algorithm. To understand how to applying and picking the data, we also check the research paper related to the machine learning filed, In the paper " Machine learning algorithms-a review" Mahesh, Batta published, he explained how different machine learning algorithm applying in data analyzing.

Several researchers have worked on the diagnosis of Parkinson's Disease by using machine learning methods, e.g. diagnosis using voice, diagnosis using brain scan images, diagnosis drawings such as meander patterns, spirals, waves, etc.

J. Mei et al. [22] did a review of literature on machine learning for the diagnosis of Parkinson's disease, using sound, MRI images, and hand-drawn images. It searches IEEE Xplore and PubMed. It reviewed research articles published from the year 2009 onwards and summarized data sources and sample size.

The public repositories and databases include HandPD [13], Kaggle dataset [23], the University of California at Irvine (UCI) Machine Learning Repository [24], Parkinson's Progression Markers Initiative (PPMI) [25], PhysioNet [26], etc.

Quite a few researchers use magnetic resonance images (MRI) or their variations as their research dataset. Noor et al. [5] surveyed the application of deep learning in detecting neurological disorders from magnetic resonance images (MRI) in the detection of Parkinson's disease, Alzheimer's disease, and schizophrenia.

E. Huseyn et al. [27] [28] used MRI images as their dataset. S. Chakraborty [29] and X. Zhang [30] has used a dataset from Parkinson's Progression Markers Initiative (PPMI). Z.Cai et al. [31] used an enhanced fuzzy k-nearest neighbor (FKNN) method for the early detection of Parkinson's Disease based on vocal measurements. L. Badea et al. [32] explored the reproducibility of functional connectivity alterations in Parkinson's Disease based on resting-state fMRI scans images.

Pereira et al. did a series of research on automatic detecting Classify Parkinson's disease for many years. At first, they used non-deep learning algorithms in diagnosing PD [33]. They collected/constructed a public dataset called "HandPD" [13]. Based on this dataset, they compared the efficiency of different hand drawn patterns in the diagnosis of PD [11]. Their results show that the meander pattern generates more accurate results compared to the spiral pattern. However, in our research, the spiral and meander patterns generate similar results when they are trained and tested through the same CNN.

Later, they explored the use of CNN on the images extracted from time-series signals and used three different CNN architectures, ImageNet, CIFAR-10, and LeNet as baseline approach [18].

In her master project, M. Alissa [19] used non-public datasets (spiral pentagon dataset) to evaluate the efficiency of two different neural networks (Recursive Neural Networks(RNN) and Convolutional Neural Networks (CNN)). We built a CNN similar to hers and used the dataset from HandPD [13] and Kaggle [23] to evaluate different CNNs and different datasets.

Gil-Martin et al. [34] presented a method to detect Parkinson's Disease from drawing movements using Convolutional Neural Networks. He used the dataset from the UCI machine learning repository as input data, applied signal-processing (sampling with 100Hz and 140 Hz, resampling with 110 Hz, perform Hamming windowing and FFT) to generate pre-processed data, and used this data to train/validate the CNNs.

M.E. Isenkul et al. [35] designed an improved spiral test dataset using a digitized graphics tablet for monitoring Parkinson's Disease. Digitized graphics have more information, including timestamps, grip angles, and hand pressure, etc. The significance of that can be investigated in future work.

P.Zham [36] presented a dataset at Kaggle [23] with waves and spirals. He used a composite index of speed and pen-pressure to distinguish different stages of Parkinson's Disease.

6. CONCLUSIONS

In this paper, we present a preliminary data mining procedure that help Parkinson's Disease patients slow down their progression of the disease while helping early detection of the disease. By linking the early detection and treatment together, the two stages can help each other, provide information back and forth, enlarge our database, improve the accuracy of deep learning models because deep learning is all about data.

Future work can be cooperated with therapists, provide their patients free APP and ask those patients to donate their exercise video clips.

REFERENCES

[1] M. Bhat, "Parkinson's Disease prediction based on hand tremor analysis," IEEE 2017 International Conference, 2017.

[2] "Parkinson's Disease Statistics," Parkinson's News Today, 2021. [Online]. Available: https://parkinsonsnewstoday.com/parkinsons-disease-statistics/.

[3] Editorial Team, "Diagnosis-Early Symptoms & Early Diagnosis," parkinsonsdisease.net, 8 March 2017. [Online]. Available: https://parkinsonsdisease.net/diagnosis/early-symptoms-signs.

[4] Blog post, "You could have Parkinson's disease symptoms in your 30s or 40s and not know it," health direct, 11 April 2019. [Online]. Available: https://www.healthdirect.gov.au/blog/parkinsons-disease-symptoms-in-your-30s-40s. [Accessed 3 September 2022].

[5] M. Noor, "Application of deep learning in detecting neurological disorders from magnetic resonance images: a survey on the detection of Alzheimer's disease, Parkinson's disease and schizophrenia," Brain Informatics, 9th October 2020.

[6] AK.Kumar Srivastava, "A systematic review on data mining application n Parkinson's disease," Neuroscience Informatics, Dec. 2022.

[7] Andriluka, Mykhaylo, et al. "2d human pose estimation: New benchmark and state of the art analysis." Proceedings of the IEEE Conference on computer Vision and Pattern Recognition. 2014.

[8] T. blog, "Guide to machine learning applications: 7 Major Fields," The APP solution, 2021. [Online]. Available: https://theappsolutions.com/blog/development/machine-learning-applications-guide/.

[9] S. Shalev-Shwartz, "Understanding machine learning: From theory to algorithms," Cambridge University Press, 2014.

[10] I. H. Witten, "Data Mining: Practical Machine Learning Tools and Techniques, Third Edition," Morgan Kaufmann Series in Data Management Systems, 2011.

[11] Trending Blog, "What is Deep Learning? and What are its Significance Deep Learning Trends," ALPHA Information Systems INDIA PVT LTD, 15 September 2019. [Online]. Available: https://www.aalpha.net/blog/what-is-deep-learning-and-what-are-its-significance/.

[12] H. Wang, "On the origin of deep learning," arXiv preprint arXiv, 2017.

[13] C. Pereira, "Welcome to the HandPD dataset," HandPD, 2017. [Online]. Available: https://wwwp.fc.unesp.br/~papa/pub/datasets/Handpd/.

[14] Zhang, Fan, et al. "MediaPipe hands: On-device real-time hand tracking." arXiv preprint arXiv:2006.10214 (2020).

[15] Likas, Aristidis, Nikos Vlassis, and Jakob J. Verbeek. "The global k-means clustering algorithm." Pattern recognition 36.2 (2003): 451-461.

[16] Yuan, Fang, et al. "A new algorithm to get the initial centroids." Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826). Vol. 2. IEEE, 2004.

[17] C. Pereira, "Deep Learning-Aided Parkinson's Disease Diagnosis from Handwritten Dynamics.," Graphics. Patterns and Images (SIBGRAPI) 2016 29th SIBGRAPI Conference IEEE, 2016.

[18] C. Pereira, "Convolutional neural networks applied for Parkinson's disease identification," Machine Learning for Health Informatics, 2017.

[19] M. Alisssa, "Master's Project: Parkinson's Disease Diagnosis Using Deep Learning," Heriot Watt University, 2018.

[20] N. Srivastava, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting.," The Journal of Machine Learning Research., 2014.

[21] Sinaga, Kristina P., and Miin-Shen Yang. "Unsupervised K-means clustering algorithm." IEEE access 8 (2020): 80716-80727.

[22] J. Mei, "Machine Learning for the Diagnosis of Parkinson’s disease: a review of literature," Frontiers in Aging Neuroscience, 2021.

[23] K. Mader, "Parkinson's Drawings," Kaggle.com, 2019. [Online]. Available: https://www.kaggle.com/kmader/parkinsons-drawings.

[24] D. Graff, "UCI Machine Learning Repository," University of California, Irvine.

[25] K. Marek, "The Parkinson Progression Marker Initiative (PPMI)," Progress Neurobiol, 2011.

[26] A. L. Goldberger, "PhysioBank, Physio Toolkit and PhysioNet: Components of a new research resource for complex physiologic signals," Circulation, 2000.

[27] E. Huseyn, "Deep Learning Based Early Diagnostis of Parkinsons Disease," Cornell University arXiv.org, Auguest 2020.

[28] S. Shinde, "Predictive markers for Parkinson's disease using deep neural nets on neuromelanin sensitive MRI," bioRxiv the preprint server for biology, 2019.

[29] S. Chakraborty, "Detection of Parkinson's Disease from 3T T1 weighted MRI scans using 3D convolutional neural network," diagnostics (MDPI), 2020.

[30] X. Zhang, "Multi-View graph convolutional network and its applications on neuroimage analysis for Parkinson's disease," Amia annual symposium proceedings archive, 2018.

[31] Z. Cai, "An Intelligent Parkinson's Disease Diagnostic," Hindawi.com, Computational and Mathematical Methods in Medicine, 2018.

[32] L. Badea, "Exploring the reproducibility of functional connectivity alterations in Parkinson’s disease," National Institute for Research and Development in Informatics, Bucharest, Romania.

[33] C. Pereira, "A step towards the automated diagnosis of parkinson's disease: Analyzing handwriting movements," in 28th International Symposium on Computer-Based Medical Systems (Sao Carlos: IEEE), 2015.

[34] M. Gil-Martin, "Parkinson's Disease Detection from Drawing movements using convolutional neural networks," in Electronics (MDPI), 2019.

[35] M Isenkul, "Improved Spiral Test using a Digitized Graphics Tablet for Monitoring Parkinson's Disease," in International Conference on e-Health and Telemedicine, 2014.

[36] P. Zham, "Distinguishing Different Stages of Parkinson's Disease Using Composite Index of Speed and Pen-Pressure of Sketching a Spiral," in Frontiers in Neurology, 2017.