Using Machine Learning to Detect Parkinson's Disease Through Drawing Data

Katie Yue
San Mateo, CA, USA

ABSTRACT

Parkinson's is a neurodegenerative disease that causes uncontrolled movements such as tremors or stiffness. Currently, there is no specific clinical test that can test for Parkinsons, therefore many cases go undiagnosed. Patients that do get diagnosed often must go through a lengthy and expensive process of a series of examinations and consultations with specialists. In this paper, we present a few machine learning models, including logistic regression, K-Nearest Neighbors, Decision Trees, Convolutional Neural Networks, and Transfer Learning, as a potential way to detect Parkinson's through patients' drawings of spirals and waves. We decided to use patients' drawings, as this data is much more accessible and inexpensive to obtain in comparison to imaging or lab results. Coupling this data with machine learning models could therefore serve to increase Parkinson's diagnoses, particularly in developing countries, and ultimately help patients receive better treatment options. Our results show that the logistic regression classifier performed the best, with an accuracy of 91%, precision score of 92%, and recall score of 89%. These results show that machine learning and spiral and wave drawings can be used as a viable tool for furthering Parkinson's diagnosis.

INTRODUCTION

Parkinson's disease is a central nervous system disorder which occurs when brain cells stop making dopamine, a chemical messenger which coordinates movement (National Institute on Aging 2022). More than 10 million people worldwide currently have Parkinson's, the real number likely being much higher given the large number of undiagnosed or misdiagnosed individuals (Parkinson's Foundation). This large number of misdiagnoses are a result of that fact that there is currently no specific test for Parkinsons, and the only way to diagnose the disease is through a series of analyses of the patient's medical history, symptoms, and a neurological and physical exam, a process which often takes regular follow up appointments with neurologists in order to diagnose the disease (Mayo Clinic 2023). Certain physical symptoms need to be observed in order to determine that one has Parkinson's, such as tremors, rigidity in the limbs, and bradykinesia, however because patients with an early onset of Parkinson's may not meet the clinical criteria for diagnosis, many individuals may go undiagnosed (Johns Hopkins Medicine). Furthermore, the diagnosis and monitoring of Parkinson's are very costly and sometimes invasive processes, which only increases the number of individuals who go undiagnosed and untreated in developing parts of the world (Walker 2012). One potential biomarker for detecting Parkinson's is handwriting and drawing analysis, as it reflects the motor impairments that may occur earlier than other typical symptoms. More specifically, drawings of Archimedes' spirals (Figure 2) and waves (Figure 3) can display the micrographia which occurs in Parkinson's disease patients. Using ML to analyze these drawings could provide a possibly more concrete way to diagnose Parkinson's which improves efficiency and accessibility while also reducing costs and ambiguity.

DATASET

Because Parkinson's has no conclusive diagnostic test, we used drawing data which reflects the tremors present in Parkinson's patients. The Kaggle dataset used in this project contains images of drawings of spirals and waves by both Parkinson's and non-Parkinson's patients in png format. The initial dataset, created by K. Scott Mader, contained 204 total images, but was increased by B. Anil Kumar to a total of 3264 images, with 1632 Healthy and 1632 Parkinson's images (Figure 1).

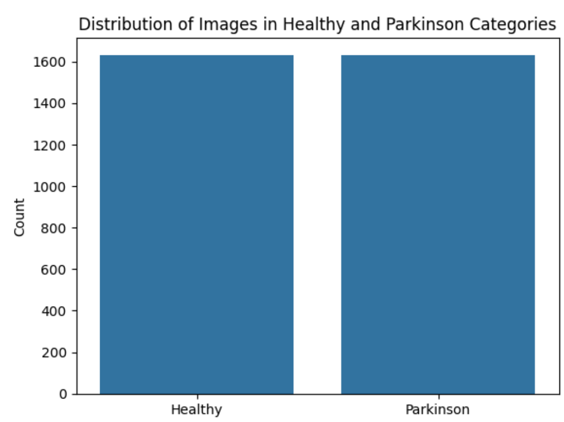

Figure 1: Bar chart of Healthy and Parkinson's images.

The additional images were created through augmentation processes such as rotating the images 90°, 180°, 270°, and 180° and colorizing the images (Kumar 2023). These processes allowed the dataset our model to have more training data and therefore be more accurate. We split the data so that 80% was used to train the model, while 20% was used to test it.

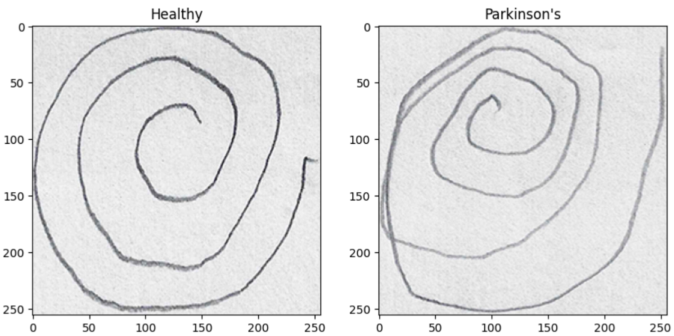

Figure 2: Side by side comparison of spiral drawing by healthy patient and Parkinson's patient. The micrographia, shown by the uneven spacing of the spiral, can be seen in the Parkinson's drawing.

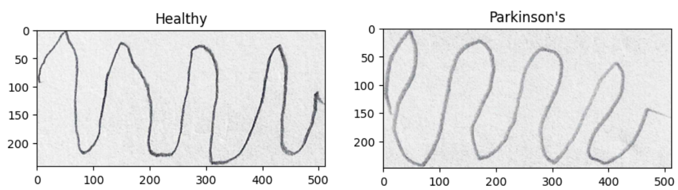

Figure 3: Side by side comparison of wave drawing by healthy patient and Parkinson's patient.

METHODOLOGY/MODELS

In order to classify between Parkinson's and Healthy images, we used a total of five different classification models, including Convolutional Neural Networks (CNN), Logistic Regression, K-Nearest Neighbors, Decision Trees, and Transfer Learning.

DATA PROCESSING

We also used the Numpy, Pandas, OpenCV, Seaborn, and Matplotlib libraries to visualize the data. We used OpenCV and NumPy to create a NumPy array containing the images as feature vectors. Feature vectors are a list of numerical properties that machine-learning models use as input for a prediction (Parashar 2023). Because machines can only process numeric information at this time, as opposed to qualitative data, feature vectors are instead used to represent the features. To ensure consistency, we also resized all the images to be 256x256 pixels, using OpenCV's cv2.resize() function. We created a second array containing labels, in this case with 0 and 1 corresponding to Healthy and Parkinson, respectively. After training the data with the 80/20 split, we also flattened the data from a 4D array into a 2D array by flattening the height, width, and channels into a vector length of 256x256x3 = 196608.

LOGISTIC REGRESSION

The first model we implemented was a logistic regression model. Logistic regression uses a logistic function to convert the numeric input values into probability values between 0 and 1. These values, which are obtained from the flattened vector data, are graphed in a sigmoid function defined by the formula $f(x) = (1 + e^{-x})$ (Kanade 2022). Loss functions, such as binary cross entropy, are used to determine the best weights that minimize loss. In our model, the only parameter we implemented was the number of iterations, for which we found 500 iterations to yield the highest accuracy.

K-NEAREST NEIGHBORS

The next model we implemented was a K-Nearest Neighbor, or KNN, model. KNNs measure the straight line distance between the data points, to determine which points are the "neighbors." The most commonly used metrics are Euclidean, Manhattan, and Minkowski distance. It then assigns the most common classification value amongst the neighboring data points to the new data point. The K in KNNs represents the number of nearest neighbors to consider (Jain 2024). We found that 5 neighbors worked the best for our model.

DECISION TREE

The last model we implemented that used flattened data was a Decision Tree model. The Decision Tree algorithm starts at a root node and selects the feature that splits the data into the most homogeneous branches. This process is repeated for each subset until the maximum depth is reached or if there is a minimum number of training inputs set for each leaf (Gupta 2017). In our model, we found that a maximum depth of 17 yielded the best results.

CNN MODEL

Having established the baseline models, we decided to implement a CNN, which is specialized in processing unflattened, grid-like data, such as images. The convolutional layers in CNNs use filters known as kernels to detect patterns or features such as edges or textures. In this case, because the dataset contains both spirals and waves, the model is able to extract more features that help differentiate between different types of drawings. Although CNNs take in unflattened data, the data is still downsized later on through Max Pooling, which moves a tiny window, generally of size 2x2 pixels, across the image and selects and saves only the maximum values. Lastly, fully connected, or Dense, layers connect each neuron from the previous layer to each neuron of the current layer, and use the features learned from previous layers to make predictions (IBM).

TRANSFER LEARNING MODEL

The final model we implemented was a Transfer Learning model, where we built a new "expert_model" using the pre-trained CNN models, VGG16, VGG19, ResNet50, and DenseNet121 from the Keras library as a base. These models had been pre-trained on other large scale image datasets such as ImageNet, so by implementing them to be used on this Parkinsons dataset, we could leverage the models' existing abilities to extract image features, without having to retrain the models or alter the weights. Instead, we added Dense layers and trained only those layers on the Parkinson's dataset (Donges and Powers 2022).

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	91%	92%	89%	91%
K-Nearest Neighbors	52%	51%	94%	66%
Decision Trees	64%	69%	52%	60%
CNN	52%	51%	80%	63%
Transfer Learning	86%	---	---	---

Table 1: Metrics for all models tested.

RESULTS AND DISCUSSION

Despite only taking in flattened data, the logistic regression model yielded the highest accuracy and precision scores of 91% and 92% (Figure 4). Because this dataset is not particularly large, a larger percentage of the dataset may contain bias. Therefore, the simplicity of the logistic regression model could have allowed it to avoid overfitting to those biases. Meanwhile, the other two baseline models were comparatively more complex. The KNN model yielded an accuracy score of 52% and precision score of 51%. The Decisions Tree model yielded an accuracy score of 64% and a precision score of 69%. Because these baseline models processed the flattened data, it was expected that the accuracy may not be very high, as the input data simply contained less information than in the actual dataset. Beyond the baseline models, the CNN model, when tested with a parameter of 3 hidden layers, yielded an accuracy score of 52% and a precision score of 51%. Because the CNN model processed the unflattened data, we expected it to perform better. We recognized that having too many layers could lower the accuracy, as the repeated pooling of the images each time would only decrease the size of the data more, and therefore only implemented 3 layers. However, it is possible that there was other hyperparameter tuning, such as changing the kernel size or learning rate, that could have been done to improve the model. Furthermore, deep learning classifiers such as CNNs require large datasets. Though 3265 images is not necessarily a small number, particularly for medical datasets, it still may not have been enough to prevent the CNN model from overfitting. Lastly, the Transfer Learning model, when set to 10 epochs, yielded an accuracy score of 86%. Although not higher than the logistic regression score, the Transfer Learning model did quite well, especially in comparison to the other CNN model. The Transfer Learning model may have been better able to avoid overfitting due to already being trained on very large datasets as opposed to one single dataset, and also been able to better extract features due to the pre-trained weights. Results for all models can be found in Table 1. Despite certain variance in the accuracy of our models, overall, our results show that using machine learning to analyze drawing data can be very valuable to providing more efficient and accessible Parkinson's diagnosis.

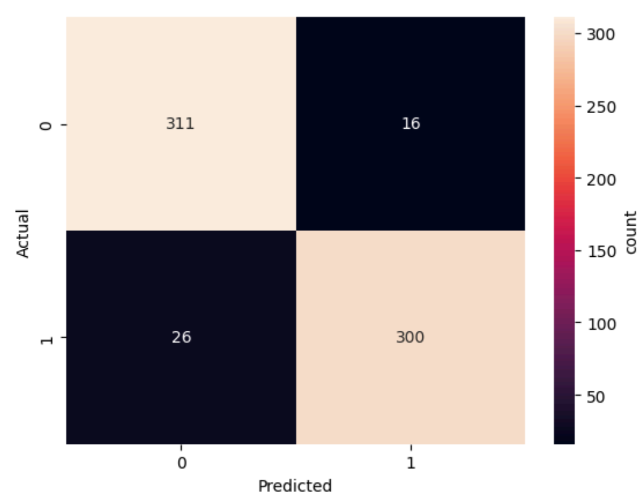

Figure 4: Confusion Matrix of predictions from Logistic Regression Classifier.

ACKNOWLEDGEMENTS

I would like to thank my mentor, Sriram Hathwar, for all his patient and thoughtful guidance throughout the course of this project. He helped me find a suitable dataset for my goals and taught me so much about building different machine learning models and how to write my first research paper.

REFERENCES

B. Anil Kumar. “Parkinson's Disease Augmented Data of Handdrawings.” Kaggle, 25 Nov. 2023, https://www.kaggle.com/datasets/banilkumar20phd7071/handwritten-parkinsons-disease-augmented-data/data?select=A+Transfer+Learning+Approach+with+MobileNetV2+for+Parkinsons+Disease+Detection+using+Hand+Drawings.pdf.

Donges, Niklas, and Jessica Powers. 2022. “What Is Transfer Learning? A Guide for Deep Learning.” Built In. https://builtin.com/data-science/transfer-learning.

“Figure 2: Archimedes spirals drawn by a normal volunteer and patients...” ResearchGate. Accessed January 30, 2024. https://www.researchgate.net/figure/Archimedes-spirals-drawn-by-a-normal-volunteer-and-patients-with-Parkinsons-disease_fig2_310666616.

Gupta, Prashant. 2017. “Decision Trees in Machine Learning | by Prashant Gupta.” Towards Data Science. https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052.

“How Parkinson's Disease Is Diagnosed.” Johns Hopkins Medicine. Accessed November 29, 2024. https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/how-parkinson-disease-is-diagnosed.

Jain, Sandeep. 2024. “K-Nearest Neighbor(KNN) Algorithm.” GeeksforGeeks. https://www.geeksforgeeks.org/k-nearest-neighbours/.

Jain, Sandeep. 2024. “Convolutional Neural Network (CNN) in Machine Learning.” GeeksforGeeks. https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machine-learning/.

Kamble, Megha, Prashant Shrivastava, and Megha Jain. 2021. “Digitized spiral drawing classification for Parkinson's disease diagnosis.” ScienceDirect. https://www.sciencedirect.com/science/article/pii/S266591742100009X.

Kanade, Vijay. 2022. “Logistic Regression: Equation, Assumptions, Types, and Best Practices.” Spiceworks. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/.

P, Varalakshmi, Tharani P. B, Anu R. B, Bhuvaneaswari R, and Sakthi J. Rajasekar. 2022. “Diagnosis of Parkinson's disease from hand drawing utilizing hybrid models.” ScienceDirect. https://www.sciencedirect.com/science/article/abs/pii/S1353802022003480.

Parashar, Nilesh. 2023. “A Beginner’s Guide to Understanding Feature Vector in Data Science.” Medium. https://medium.com/@niitwork0921/a-beginners-guide-to-understanding-feature-vector-in-data-science-3b6aa5fb1371.

“Parkinson's 101 | Parkinson's Disease.” Michael J. Fox Foundation. Accessed January 28, 2024. https://www.michaeljfox.org/parkinsons-101.

“Parkinson's Disease: Causes, Symptoms, and Treatments | National Institute on Aging.” 2022. National Institute on Aging. https://www.nia.nih.gov/health/parkinsons-disease/parkinsons-disease-causes-symptoms-and-treatments.

“Parkinson's disease - Diagnosis and treatment.” 2023. Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/parkinsons-disease/diagnosis-treatment/drc-20376062.

“Parkinson's Disease: What It Is, Causes, Symptoms & Treatment.” 2022. Cleveland Clinic. https://my.clevelandclinic.org/health/diseases/8525-parkinsons-disease-an-overview.

Rosebrock, Adrian. 2019. “Detecting Parkinson's Disease with OpenCV, Computer Vision, and the Spiral/Wave Test.” PyImageSearch. https://pyimagesearch.com/2019/04/29/detecting-parkinsons-disease-with-opencv-computer-vision-and-the-spiral-wave-test/.

Unzueta, Diego. 2022. "Fully Connected Layer vs Convolutional Layer: Explained." Built In. https://builtin.com/machine-learning/fully-connected-layer.

Walker, Richard. 2012. "The management of Parkinson's disease in sub-Saharan Africa." PubMed. https://pubmed.ncbi.nlm.nih.gov/22650168/.

"What are Convolutional Neural Networks?" IBM. Accessed April 1, 2024. https://www.ibm.com/topics/convolutional-neural-networks.

"What is Parkinson's?" Parkinson's Foundation. Accessed January 14, 2024. https://www.parkinson.org/understanding-parkinsons/what-is-parkinsons.

"What is Parkinson's Disease | APDA." American Parkinson Disease Association. Accessed January 16, 2024. https://www.apdaparkinson.org/what-is-parkinsons/.