

# Using CNN to diagnose Parkinson's Disease from spiral drawings and PET images

Shantala Totada

Received June 17, 2024

Accepted September 13, 2024

Electronic access October 15, 2024

Parkinson's disease (PD) is a neurodegenerative disorder associated with motor movements and affects the substantia nigra, the location for dopamine production. With the prevalence of PD increasing, diagnosing PD at an early stage is vital to prevent the onset and progression of symptoms. Many health professionals have started using technological devices such as positron emission tomography (PET) scans to diagnose patients with certain disorders. While these devices are used frequently, many are invasive and time-consuming, generating discomfort in patients and long-term side effects. With the growth of artificial intelligence (AI), new developments have been made for diagnosing PD using convolutional neural networking (CNN). CNN has helped easily recognize and diagnose PD and can be used to replace other methods of diagnosing PD, such as PET or DaTscans. Spiral drawings have become commonly used to diagnose PD due to it being less invasive and easier. In this study, the diagnostic accuracy is compared between PET scans and spiral drawings to determine the better method for diagnosing PD. Using two spiral drawing datasets and one PET scan database, all of which included both healthy and Parkinson patients, the PET scans were found to be more accurate than the spiral drawings. However, the accuracy of the spiral drawings was found to be relatively high, deeming spiral drawings a non-invasive alternative option for diagnosing PD. The study provides evidence that the use of CNN can help in the early recognition of PD, which can help prevent the onset of severe symptoms.

## Introduction

Parkinson's Disease (PD) is a chronic neurodegenerative disorder that results in uncontrolled movements. It is the second most common neurological condition, after Alzheimer's Disease, and it has no cure. PD was originally described by James Parkinson in 1817<sup>1</sup>. PD is characterized by the loss of nigrostriatal dopaminergic neurons<sup>1</sup>. The primary risk factor for PD is age; patients are typically over the age of 65 when they start to develop symptoms such as uncontrolled movements including tremors, stiffness, bradykinesia, imbalance, and coordination difficulties. Along with physical symptoms, PD can also lead to psychological symptoms such as depression, mood disorders, and cognitive dysfunction<sup>2</sup>.

According to the Parkinson's Foundation, a study conducted in 2022 estimated that 90,000 people are being diagnosed with PD in the U.S. every year, making PD a rising concern<sup>3</sup>. Much research is occurring to develop treatments as early as possible, but developing treatments for neurodegenerative diseases is not easy. Neurodegeneration is the progressive loss of neurons, where neurons degenerate and lose their function, leading to their death. Neurodegeneration affects a variety of issues within the brain including cognitive thinking and motor skills<sup>4</sup>. The main cause of neurodegeneration has been linked to protein aggregation extracellular to the synapse. Protein aggregation often occurs due to the misfolding of proteins<sup>5</sup>. Degeneration of

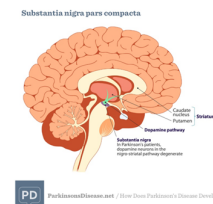


Figure 1. The location of neurodegeneration of PD is within the substantia nigra, located in the midbrain. The substantia nigra is responsible for the production of dopamine, which is a neurochemical important in PD. Image taken from ParkinsonsDisease.net.

neurons primarily affects the motor skills in PD patients. Most of the cells that die are found within the substantia nigra, an area within the brain that is responsible for movement. Figure 1 shows the exact region in the brain where the substantia nigra is found. The substantia nigra produces dopamine, a neurochemical responsible for a variety of functions within the nervous system such as movement control, cognitive executive functions, and emotional limbic activity<sup>6</sup>. Determining early-onset of PD becomes difficult as symptoms appear late, resulting in many cases of PD above the age of 65.

For a long time, PD was reported to be caused by the alpha-synuclein gene (SNCA), creating an autosomal dominant version. SNCA is a 14.5 kDa, 140 amino acid protein encoded

---

by 5 exons and  $\alpha$ -synuclein.  $\alpha$ -synuclein consists of an amino-terminal region containing seven imperfect repeats that are 11 amino acids long, a central hydrophobic domain, and a negatively charged carboxy-terminal that is acidic<sup>7,8,9</sup>.  $\alpha$ -synuclein makes up the components of Lewy bodies<sup>7,8,9</sup>. Lewy bodies are considered the hallmark of PD and have been found to result in neuronal degeneration<sup>8,10</sup>. SNCA is involved in the regulation of synaptic vesicle trafficking and neurotransmitter release, including dopamine. Mutations of SNCA have also contributed to increased neuro degeneration through an increase in the production of  $\alpha$ -synuclein<sup>7</sup>.  $\alpha$ -synuclein protein can misfold and aggregate, leading to the formation of soluble oligomers which lead to neuronal damage<sup>11</sup>.  $\alpha$ -synuclein fibrils interact with the neuronal members that release A11 – reactive oligomers, which are quite like the oligomers found in  $\alpha$ -synuclein aggregation<sup>11</sup>.

Understanding the pathology and the mechanism for PD is crucial for developing treatment plans. The progression of PD is different for each patient, therefore, using AI and neural networks help researchers and medical professionals identify and analyze large databases and patterns and make predictions to the highest accuracy. This development will open doors for a better diagnosis, different treatment plans, and a better understanding of Parkinson's disease.

### **The Use of AI and Neural Networks:**

AI is growing exponentially and has been shown to be helpful, especially in the medical field, by allowing medical practitioners to diagnose a disease. As of today, many neurologists are relying on invasive imaging tools such as magnetic resonance imaging (MRI) or positron emission tomography (PET) scans, and the use of electronic health records (EHR). While using scans may result in accurate predictions and findings, they can make many patients feel uncomfortable, creating a sense of claustrophobia, and are often expensive.

In this paper, we will review another method used for early detection of PD. This early detection method involves using hand-drawn spirals and PET scans to train a Deep Neural Networking (DNN) model, or more specifically Convolutional Neural Networking (CNN). DNN refers to neural networks with many layers (typically more than two), including deep architectures for complex tasks<sup>12</sup>. CNN is a type of neural network that can be used to classify samples associated with audio or images and sort them into categories. Training a CNN to classify images between healthy vs. Parkinson's patients can help with the early detection of PD. It will help researchers find a pattern between these two groups of patients that can help with early detection and provide results in image-related learning<sup>13</sup>. CNN can include many layers such as Conv2D, Maxpooling2D, and Dropout. The layers used in this model were Input, Conv2D, MaxPooling2D, BatchNormalization,

Dropout, Flatten, and Dense. MaxPooling2D is responsible for downsampling, which reduces the size of the feature maps in a CNN, resulting in improved performance of the CNN model. Additionally, this results in reducing the number of parameters and the computational load used after each Conv2D layer is applied<sup>13</sup>. The Conv2D layer is responsible for creating the convolution kernel and helps with the production of a tensor of outputs. It helps process two-dimensional input data and applies convolutional filters to the input data. Conv2D can pick certain features of the image for easy detection<sup>13</sup>. Input is used for defining the shape of the input data and the type of data the model will process<sup>13</sup>. BatchNormalization uses a transformation that attempts to maintain a mean output near 0; this transforms the inputs so that they are standardized<sup>14</sup>.

BatchNormalization normalizes the activations of the previous layer to improve training stability and performance. Dropout is responsible for preventing the model from overfitting while training<sup>15</sup>. The flattened layer helps transform multi-dimensional data into one-dimensional vectors for fully connected layers<sup>16</sup>. Finally, the Dense layer is used for classifying images based on output from the convolutional layers<sup>17</sup>. In addition to the layers used in the model, the Adam optimizer and regularization were used to help with the accuracy performance of the model through learning rates and weights<sup>18</sup>. Following the optimizers, initializers are also needed to help with the performance of the network. Initializers are used to set the values of the weights and biases in the CNN before the model starts training.

DNN has been used for recognizing a variety of diseases including PD. One benefit of CNNs is that they are suitable for specific tasks including capturing spatial hierarchies in images, object detection, and image classification. However, using CNN requires large datasets and training a model can take a long time. The purpose of this study is to use machine learning to classify images into two groups, healthy and Parkinson, making CNNs useful for this task.

## **Methods**

### **Dataset**

The spiral drawing images used for training the model were taken from the Kaggle database. Within the Kaggle database, the spiral drawings were split into two sections, testing and training. The testing contained 15 healthy images and 15 Parkinson's images. The training set contained 36 healthy images and 36 Parkinson's images<sup>19</sup>. One example of a user implementing this dataset into their research is Huang et al's paper, "Early Parkinson's Disease Diagnosis through Hand-Drawn Spiral and Wave Analysis Using Deep Learning Techniques"<sup>20</sup>. While the results of the model used depict overfitting, the researchers explain prevention measures for overfitting. The model did

---

manage to achieve a high training accuracy of 100%<sup>20</sup>. The Kaggle dataset comes directly from Zham et al research, “Distinguishing Different Stages of Parkinson’s Disease Using Composite Index of Speed and Pen-Pressure of Sketching a Spiral”<sup>21</sup>.

Another dataset used came from the Botucatu Medical School, Sao Paulo Brazil containing the NewHandPD dataset. The NewHandPD contained 72 control images and 296 patient images within training and 141 control images and 125 patient images within testing<sup>22</sup>. Images in the healthy group had six male and 12 female with ages ranging from 19 to 79 years. Additional factors to consider with the images are that two subjects are left-handed and 16 are right-handed.

Next, images were obtained from an online database from USC (University of Southern California) Stevens Neuroimaging and Informatics Institute comparing PET scans among PD patients<sup>23</sup>. Images were selected from the age range of 65 to 85 years and contains a mix of female and male scans. After gaining access to the database, the images were viewed in axial slices, mostly in the middle of the brain, and downloaded. While searching for images in the database to download, many PET scans were distorted, reducing the number of images available to download<sup>23</sup>. 10 healthy scans and 51 Parkinson scans were within the training folder and 10 healthy scans, and 41 Parkinson scans were within the testing folder. These images were hard to figure out if they PD was shown, which resulted in selection bias while collecting the images. The images were put into two folders, testing and training, was a task that had to be done manually since the two other datasets had already downloaded images and placed them into their respective folders, training and testing. Figure 2 shows examples of the spiral drawings from the Kaggle database, the NewHandPD database and the USC database.

## Building the model

Building three models for each of the three datasets started with downloading packages within Anaconda Navigator such as tensorflow, Keras, pandas, os, numpy, and matplotlib. The layers, optimizers, initializers, and regularizers were added to the model. Figure 3 depicts the layers split up which include Conv2D, MaxPooling2D, Flatten, and Dense<sup>15</sup>. The environment used was tensorflow with multiple packages downloaded including os, keras, and matplotlib.pyplot, load\_img, img\_to\_array, and opencv. The optimizers included in all three models were Adam and SGD. Both Adam and SGD help with updating the weight of the model during training to reduce the loss rate. Next were the initializers installed within the layers. Initializers such as gloriot\_uniform were also introduced. The gloriot\_uniform initializer helps maintain the variance of activations and gradients in the layers used. The model was created through a Python application within the Spyder program

found in the Anaconda Navigator. Within all three models, there certain parameters that remained constant such as the target size, which was (128,128), and the learning rate, which was 0.0001. However, the batch size and the number of epochs changed depending on the number of images within the dataset. For the Kaggle model, the batch size was 128 and the epoch number was 50. Within the NewHandPD dataset, the batch size and epochs were 60 and 70 respectively. The USC dataset had a batch size and epoch number of 30 and 70 respectively. The batch size is typically determined based on the number of images and the number of epochs typically remains between 50-100. 70 was a number chosen at random due to it being in between 50-100. The NewHandPD model consisted of three layers, with each layer consisting of Conv2D and MaxPooling2D.

The above code shows what all three model layers looked like. While all three models didn’t use the same layers or the same amount, the layer structure setup was quite like the code above. There were no biases while creating the model. Moving from the layers, the images in the datasets had to be augmented for the model to work. All three models required the use of Image Data Generator. Image Data Generator was used to help with augmenting the data for each of the three datasets. Augmenting the data included rotating, scaling, translating, and flipping the images. Table 1 shows the code used for data augmentation. All three datasets used a rotation\_range of 360 and a horizontal\_flip = True. Shear\_angle augments pictures in a way for the model to identify the image through different angles<sup>24</sup>. Zoom\_range zooms in on the images and horizontal\_flip and vertical\_flip is responsible for flipping half of the images horizontally and vertically respectively<sup>25</sup>. Finally, the width\_shift\_range and height\_shift\_range values are fractions of the total width or height to which they translate the images either horizontally or vertically<sup>25</sup>.

The three datasets were trained with the model, with slight changes made concerning the color scheme of the images and the file directory of the images. The training accuracy, validation accuracy, validation loss, and training loss of all three datasets were plotted. At first, the Kaggle and USC models were not performing well due to the imbalance of the images. class\_weights were introduced to help balance the images. By installing class\_weights, the model was able to assign different weights to different classes while the model was training. Additionally, class\_weights helped remove any bias towards the more frequent class images.

## Results

The accuracy between spiral drawings and PET scans was tested on Spyder to determine the accurate representation of diagnosing PD. The val\_accuracy, val\_loss, Training Loss, and Training Accuracy were produced on graphs following

```

classifier=Sequential()
classifier.add(Conv2D(128, (9, 9), padding='same', strides=(1, 1), name='conv1', activation='relu',
kernel_initializer='glorot_uniform', kernel_regularizer=regularizer))
classifier.add(MaxPooling2D((7, 7), strides=(3, 3)))
classifier.add(Dropout(0.4))
classifier.add(Conv2D(64, (7, 7), padding='same', strides=(1, 1), name='conv2', activation='relu',
kernel_initializer='glorot_uniform', kernel_regularizer=regularizer))
classifier.add(MaxPooling2D((5, 5), strides=(3, 3)))
classifier.add(Dropout(0.5))
#classifier.add(Conv2D(32, (3, 3), padding='same', strides=(1, 1), name='conv3', activation='relu',
#kernel_initializer='glorot_uniform', kernel_regularizer=regularizer))
#classifier.add(MaxPooling2D((1, 1), strides=(3, 3)))
classifier.add(Flatten())
classifier.add(Dense(64, activation='relu', kernel_initializer='glorot_uniform', name='fc1'))
classifier.add(Dense(2, activation='sigmoid', kernel_initializer='glorot_uniform', name='fc3'))
optimizer = Adam(3.15e-7)
classifier.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

```

Table 1. The data augmentation which includes rotation, scaling, translation, and flipping for each of the three models is listed below.

Dataset	Code for Data Augmentation
Kaggle	<pre> train_data_generator = ImageDataGenerator(rotation_range=360, width_shift_range=0.0, height_shift_range=0.0, horizontal_flip=True, vertical_flip=True)  test_data_generator = ImageDataGenerator(rotation_range=360, width_shift_range=0.0, height_shift_range=0.0, horizontal_flip=True, vertical_flip=True) </pre>
NewHandPD	<pre> train_datagen = ImageDataGenerator(rescale = 1./255, rotation_range = 360, shear_range = 0.2, zoom_range = 0.6, horizontal_flip=True)  test_datagen = ImageDataGenerator(rescale = 1./255) </pre>
USC	<pre> train_datagen = ImageDataGenerator(rescale = 1./255, rotation_range = 360, shear_range = 0.2, zoom_range = 0.6, horizontal_flip=True)  test_datagen = ImageDataGenerator(rescale = 1./255) </pre>

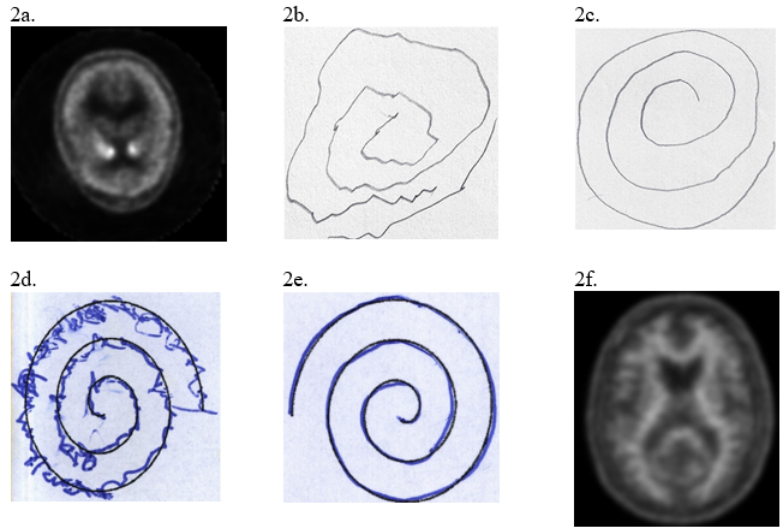


Figure 2. a. Parkinson PET Scan within the training set from the USC database<sup>23</sup>  
 b. Spiral drawing of a PD patient from the Kaggle database<sup>19</sup>  
 c. Spiral drawing of a healthy patient from the Kaggle database<sup>19</sup>  
 d. Spiral drawing of a PD patient from Sao Paulo Brazil database<sup>22</sup>  
 e. Spiral drawing of a healthy patient from Sao Paulo Brazil database<sup>22</sup>  
 f. PET scan of a healthy patient<sup>23</sup>

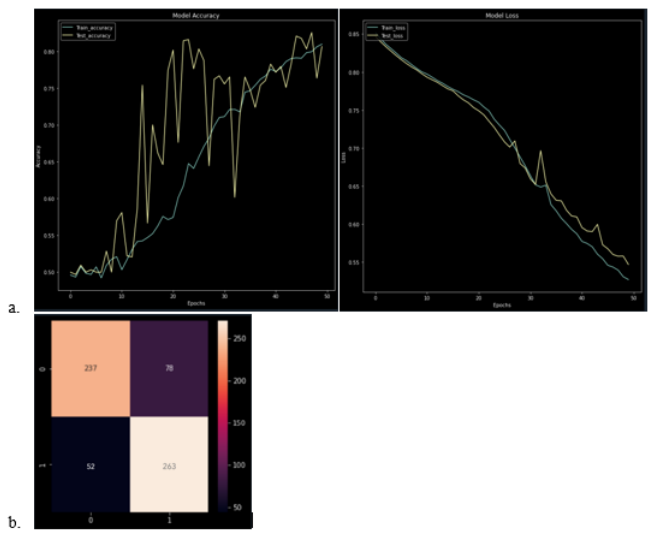


Figure 3. a. The training accuracy and test accuracy (validation accuracy) are depicted, going up to 80%. The training loss and test loss (validation loss) are depicted. b. The confusion matrix is produced from the model, showing the number of images labelled as either true positive, true negative, false positive, or false negative.

the run of the epochs. In Figure 3, the Kaggle dataset produced the training accuracy and the test\_accuracy (val\_accuracy). There is no difference or reason why test\_accuracy was written instead of val\_accuracy; both terms define the same idea. The test\_accuracy came

to 0.8063, a train\_accuracy of 0.8097, a test\_loss of 0.5468, and finally, a train\_loss of 0.5269. In this model, the number of images used was less compared to the NewHandPD dataset. In the NewHandPD dataset, the images were originally in RGB but then changed to greyscale

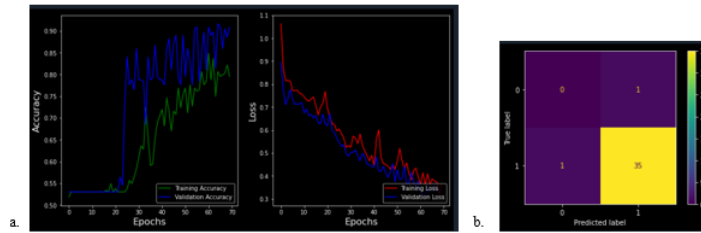


Figure 4. a. The training and validation accuracy and training and validation loss of spiral drawings from the NewHandPD dataset. b. The confusion matrix produced from the model shows the number of true positive, true negative, false positive, and false negatives images from the NewHandPD dataset.

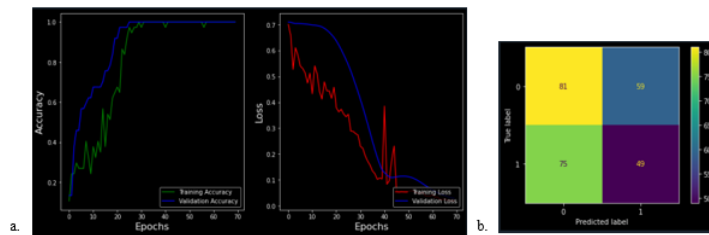


Figure 5. a. The training and validation accuracy and training and validation loss of PET scans from the USC dataset. b. The confusion matrix produced from the model shows the number of true positive, true negative, false positive, and false negative images from the USC dataset.

```

from sklearn.utils import compute_class_weight
labels = PET_train_generator.classes
class_weights =
compute_class_weight(class_weight='balanced',
classes=np.unique(labels), y=labels)
class_weights_dict =
dict(enumerate(class_weights))

```

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F_1Score = \frac{2 \times precision \times recall}{precision + recall}$$

by changing the color tone in one of the Conv2D layers. Figure 4 shows the next model for the NewHandPD, showing two plots of accuracy and loss. The `training_accuracy` came out to be 0.7983, `validation_accuracy` = 0.9073, `training_loss` = 0.3575, and `validation_loss` = 0.3624. The NewHandPD model successfully learned the difference between healthy and Parkinson's images.

Finally, the last model used PET scans from the USC database. The USC model consisted of a layer to change the color scheme from RGB to greyscale. Figure 5 shows the two plots of the accuracy and loss of the model. The `training_accuracy` came to be 1.000, `validation_accuracy` at 1.000, `training_loss` at 0.2103, and a `validation_loss` at 0.5246. There are no studies to compare the USC model performance; this creates challenges in determining the efficiency of this model and whether it is implementable in a healthcare setting.

Figure 6. The formula for sensitivity, specificity, accuracy, recall, and F1 score used to produce the values for each statistical calculation from the confusion matrixes. Image created by Shantala Totada.

## Discussion

The results of this study are like the results of other studies where the spiral images used produced a higher accuracy and learned well. One study like the results of the spiral drawings is by Huang et al<sup>20</sup>. In Huang et al's paper, the accuracy levels reached between 85% to 100%; however, their results set an example of overfitting, which is what happened in the USC model<sup>20</sup>. Nevertheless, this increased accuracy is due to the advanced computational level the model is at; the model implements many

Table 2. Calculation values of sensitivity, specificity, accuracy, precision, recall, and F1 score from the confusion matrix of each of the three models. Each calculation varies per dataset.

Dataset	Sensitivity	Specificity	Accuracy	Precision	Recall	F1 Score
Kaggle	0.8349	0.7524	0.7936	0.7713	0.8349	0.8018
NewHandPD	0.3952	0.5786	0.4924	0.4537	0.3769	0.4224
USC	0.9722	0	0.9459	0.9722	1	0.9722

layers including Conv2D, MaxPooling, BatchNormalization, Dropout, Dense, and Flatten. The use of the layers was one factor that helped the model create a distinction between the healthy images and the Parkinson images. The plot that was created from the model resulted in placing the figure legend closer to the bottom right corner making it hard to determine if the loss is decreasing or increasing; this is something that can be fixed in the future, but for the purposes of this study, the loss on all three models were decreasing. Additionally, the model was complex enough to add a confusion matrix to help with the performance of the model. Using the confusion matrix produced by all three models, the sensitivity, specificity, accuracy, precision, recall, and F1 score were all produced and shown in Table 2. Sensitivity measures how effectively a test detects true positives. Specificity measures the ability to identify true negative cases. High specificity will help minimize the number of false positives and correctly determine who is healthy in this case. Accuracy represents how correct a test is in classifying positive and negative cases. It involves the number of true positive and true negative cases over the total number of cases. Precision represents the total number of true positive results among all the positive results<sup>26</sup>. Finally, the last value calculated is the F1 score which assesses the predictive ability of a model through examining each individual class rather than looking at the overall performance<sup>27</sup>. Figure 6 shows the formula for each of the statistical calculations. In Table 2, it shows that the Kaggle model, in Figure 3, produced a sensitivity value of 83%, meaning that out of 100 positive images, about 83 of those images will be positive. The specificity value of the Kaggle model was around 75%. The accuracy and precision were around 79% and 77% respectively. These statistical values are also a good indication that the Kaggle model successfully learned the data. Additionally, the F1 score of 80% supports this model's precision and recall level.

On the NewHandPD model shown in Figure 4, Table 2 shows the sensitivity and specificity values being on the lower side, 39% and 58% respectively. However, the level of accuracy the NewHandPD model reached was high. The accuracy, precision, recall, and F1 score were 49%, 45%, 38%, and 42% respectively. This is a great indication that this model did not work out well and did not identify the clear difference between the healthy and Parkinson spiral drawings. The number of false positive and false negative images remained at 1, and the number of true negative images was 0, suggesting that this model wasn't able to

identify any negative cases and that the model is likely biased towards the positive cases since there aren't any true negative cases. A part of this error could be due to the NewHandPD images having a previously perfectly drawn spiral on the image and patients having to trace over the spiral. This could have caused an imbalance in the model in determining the positives and negatives of the model. Therefore, this does affect the reliability of the model in differentiating Parkinson and healthy spiral drawings. However, this is not a means of intentional manipulation. The Kaggle and NewHandPD dataset that was downloaded was put into the specific folders of healthy and Parkinson.

The final model, USC, experienced an increasing validation accuracy and training accuracy as shown in Figure 5; however, both accuracy levels reached 1.000, remaining constant after 40 epochs, an indication of overfitting. One issue with this dataset was the dimension of the images. One issue that came across while collecting the images was downloading them; there was no proper way to selectively download the PET scans, so screenshots were taken to create the selective dataset. Screenshots of the images could have affected the quality of the images and the statistical values. Additionally, in Table 2, the statistical values calculated from the confusion matrix resulted in some values being either 0 or being close to 100%, an indication something went wrong with the model or the dataset. Sensitivity precision, and F1 score were at 97% and accuracy was at 95%. The specificity was at 0% while the recall was at 100%. These numbers reflect the overfitting in the plot due to the high number of true positive images which must be an error in the model or the dataset. After multiple attempts of running the model, the model plot kept showing a constant line of overfitting. This issue was not resolved and remains an error of this model that could be solved using early stopping or cross-validation.

The accuracy levels for spiral drawing models, Kaggle and NewHandPD, were high, which must be due to the data augmentation and the class\_weights used on the models; this allowed for those two datasets to be evenly balanced. In the USC model, data augmentation and class\_weights were used; however, the data augmentation was not that helpful as the network wasn't able to clearly differentiate between a negative and positive case as shown in the confusion matrix in figure 5. Regardless, the increased accuracy in all three models, Kaggle, NewHandPD, and USC must be due to the nature of the images

---

and the layers used in the network. Even though the accuracy level of the spiral drawings was higher compared to the PET scans, this offers a new perspective on diagnosing PD patients; spiral drawings can be improved and utilized as a mechanism for detecting PD. Spiral drawings could be compared to PET scans and other types of scans such as DaTscans to determine if the spiral drawings reveal anything significant regarding motor impairments or any characteristics of PD just like PET scans and DaTscans. The use of CNN in determining PD suggests how the future of machine learning can be used for detection and analyzation of other severe neurological diseases, advancements in genetics, and for clinical practice. Within the field of clinical practice, machine learning has been used for patient monitoring, diagnosis, prognosis, creating personalized treatment, medical imaging, drug discovery, genomics, and other administrative tasks including billing and scheduling<sup>28</sup>.

## Conclusion

The use of the convolutional neural networks (CNNs) in determining the difference between a healthy patient and a Parkinson patient has been found to be a useful tool in determining PD. The three models created exhibited a complex computational structure with many layers, with Conv2D and MaxPooling2D being the common ones, optimizers and initializers. The three models created show a high accuracy and low error rate for the spiral drawings; however, the USC model posed several errors, including overfitting and image quality. All three models faced the error of not considering age or disease severity into factor, making the model attributable to any age or any level of disease severity. While the ages of the patients were described in the NewHandPD and USC datasets, nothing was done about it; age and disease severity was left ignored. This error would be something that would have to be considered if another study were to be conducted; the images would be split into different age groups, considering other factors. One error that could have caused errors in determining true positive and true negative images was regarding the drawing of the spiral drawings. In the Kaggle model, the spiral drawings were drawn by hand whereas in the spiral drawing dataset from Sao Paulo, Brazil, a spiral was already drawn, and patients had to trace their spiral drawing on top of the written spiral in blue ink. Hence, it would not be fair to compare spiral drawings to the PET scans because of the spiral trace that was made prior to the patients drawing their spiral. Another dataset should have been found to better compare the Kaggle dataset. Nevertheless, the models show that the slightest inaccuracies in a perfect spiral drawing can be detected as a PD case. One relevant use of this model for healthcare workers is determination of PD using spiral drawings like the ones in the Kaggle model and the NewHandPD model. Using CNNs is an efficient way to process and analyze medical images beyond spiral drawings and

PET images; examples include X-rays, MRIs, and CT scans. CNNs can ultimately enhance diagnostic accuracy and help increase performance levels within patient care. To provide more robustness testing, it is in the best interest to run all three models under different conditions, including different batch sizes or running different. Another option for testing robustness would be utilizing a different spiral dataset.

For future use, the model could be augmented to determine PD through a variety of images such as DaTscans and prevent the disease from staggering. Using the right technology, different images can properly determine PD and the characteristics of PD such as disease severity to the highest accuracy. The values for rotation range, transitions, flipping, and scaling could have been explored more to determine which values are best suited for each model. Additionally, the model layers should be adjusted for PET scans to help prevent overfitting. The PET scans must be downloaded properly and not screenshot to prevent any errors with the dimensions and image quality. The number of layers used in all three models were roughly similar which is not a suggested idea since each model consists of different images and each model requires different layers for accuracy and loss to be properly produced. Another fix for overfitting would be to use early stopping or cross-validation. It would be a future project to explore different images to see what type of images the model struggles with. The model learned the PET scans too well, but it would be interesting to see what type of images the model made doesn't learn well. These inconsistencies in the models provide a framework for the next development of CNNs for PD. This study shows that by using computational skills, CNNs can offer a promising diagnosis and progression for PD, leading to specialized treatment plans.

## References

- 1 A. Kouli, K. M. Torsney and W. Kuan, *Codon Publications eBooks*, 2018, pp. 3–26.
- 2 N. I. on Aging, *Parkinson's Disease: causes, symptoms, and treatments*, 2022, <https://www.nia.nih.gov/health/parkinsons-disease/parkinsons-disease-causes-symptoms-and-treatments>.
- 3 *Statistics*, n.d., Parkinson's Foundation.
- 4 S. Robertson, *What is Neurodegeneration?*, 2022, <https://www.news-medical.net/health/What-is-Neurodegeneration.aspx>.
- 5 G. Merlini, V. Bellotti, A. Andreola, G. Palladini, L. Obici, S. Casarini and V. Perfetti, *Protein aggregation*, 2001.
- 6 J. Sonne, V. Reddy and M. R. Beato, *Neuroanatomy, substantia nigra*, 2022.
- 7 I. J. Siddiqui, N. Pervaiz and A. A. Abbasi, *The Parkinson Disease gene SNCA: Evolutionary and structural insights with pathological implication*, 2016.
- 8 C. C. Pedersen, J. Lange, M. G. Førland, A. Macleod, G. Alves and J. Maple-Grødem, *A systematic review of associations between common SNCA variants and clinical heterogeneity in Parkinson's disease*, 2021.

- 
- 9 M. Funayama, K. Nishioka, Y. Li and N. Hattori, *Molecular genetics of Parkinson's disease: Contributions and global trends*, 2022.
  - 10 K. Wakabayashi, K. Tanji, F. Mori and H. Takahashi, *The Lewy body in Parkinson's disease: Molecules implicated in the formation and degradation of alpha-synuclein aggregates*, 2007.
  - 11 A. Bigi, R. Cascella and C. Cecchi, *-Synuclein oligomers and fibrils: partners in crime in synucleinopathies*, 2023.
  - 12 R. M. Cichy and D. Kaiser, *Deep neural networks as scientific models*, 2019.
  - 13 S. Shah, *Convolutional Neural Network: an overview*, 2022, <https://www.analyticsvidhya.com/blog/2022/01/convolutional-neural-network-an-overview/>.
  - 14 J. a. L. Marques, F. N. B. Gois, J. P. D. V. Madeiro, T. Li and S. Fong, *Elsevier eBooks*, 2022, pp. 79–99.
  - 15 I. Salehin and D. Kang, *A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain*, 2023.
  - 16 S. Yang, X. Li, X. Jia, Y. Wang, H. Zhao and J. Lee, *Deep Learning-Based Intelligent Defect Detection of Cutting Wheels with Industrial Images in Manufacturing*, 2020.
  - 17 V. L. H. Josephine, A. Nirmala and V. L. Alluri, *IOP Conference Series: Materials Science and Engineering*, 2021, p. 012007.
  - 18 E. Hassan, M. Y. Shams, N. A. Hikal and S. Elmougy, *The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study*, 2022.
  - 19 *Parkinson's drawings*, 2019, <https://www.kaggle.com/datasets/kmader/parkinsons-drawings>.
  - 20 Y. Huang, K. Chaturvedi, A. Nayan, M. H. Hesamian, A. Braytee and M. Prasad, *Early Parkinson's Disease Diagnosis through Hand-Drawn Spiral and Wave Analysis Using Deep Learning Techniques*, 2024.
  - 21 P. Zham, D. K. Kumar, P. Dabnichki, S. P. Arjunan and S. Raghav, *Distinguishing Different Stages of Parkinson's Disease Using Composite Index of Speed and Pen-Pressure of Sketching a Spiral*, 2017.
  - 22 *Welcome to the HandPD Dataset Home-page*, 2023, <https://wwp.fc.unesp.br/~papa/pub/datasets/Handpd/>, Accessed: 2023-09-30.
  - 23 Crawford, *IDA - Image and Data Archive*, 1997-2024, <https://ida.loni.usc.edu/pages/access/search.jsp>, Copyright Laboratory of Neuro Imaging, All Rights Reserved.
  - 24 *Shear | CloudFactory Computer Vision Wiki*, n.d., <https://wiki.cloudfactory.com/docs/mp-wiki/augmentations/shear>.
  - 25 B. F. Chollet, <https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>, *Building powerful image classification models using very little data*.
  - 26 *Accuracy, precision, specificity & sensitivity*, 2018, <https://labtestsonline.org.uk/articles/accuracy-precision-specificity-sensitivity>.
  - 27 I. Logunova, *A guide to F1 score*, 2023, <https://serokell.io/blog/a-guide-to-f1-score#>.
  - 28 S. Barth and S. Flam, *Machine Learning in Healthcare: Guide to Applications & benefits*, 2024, <https://www.foreseemed.com/blog/machine-learning-in-healthcare>.