The Parkinson Pathfinder

Aditya & Mishra
1009807327, adits.mishra@mail.utoronto.ca
Link to Project.

Project Description

Parkinson's is a complex disease that challenges the lives of many in the way that it works. A big part of detecting Parkinson's and detecting it early is to look at MRI scans of individuals and identify abnormalities that indicate Parkinson's. The rationale behind this project is to create a model which can utilize the brain scans of individuals and determine whether they have Parkinson's disease or are healthy.

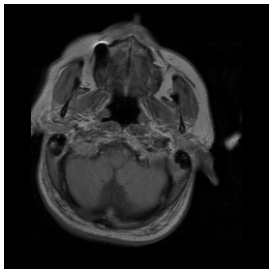

Figure 1: Example of a Parkinson Brain Scan

In the labs we've explored more basic tasks, here we have class imbalances, custom models that implement dropout, and a small dataset making this a challenging task. The rationale behind using a deep learning approach is because CNNs are state-of-the-art image classification models that can understand subtle patterns that radiologists might miss.

This report introduces the background and key related work, then goes through the main steps in data processing. Next, it presents the final model architecture and the baseline we compared against, leading into the results on the test set with a discussion of their implications and, finally, the ethical considerations.

Background & Related Work

Several previous studies have had similar objectives in using deep learning models to detect Parkinson's disease using magnetic resonance imaging (MRI). The first paper by Shokrpour [5] conducted a comprehensive review of the various datasets and algorithms used by previous researchers. They investigated the work of acoustic data, biomarkers, medical imaging, movement data, and multi modal datasets. While the paper by Manal Alrawis [1] used a deep learning approach as well, but went beyond the traditional CNNs. Using their model across three data sets and achieving an accuracy over 95 percent.

The paper published by Yan Chang[3] decided to omit MRI models and take a different method for detection. Upon finding two other papers written by Megan Courtman[4], and Milton Camacho[2], I saw that they used MRI scans and third-dimensional CNNs, respectively, to detect Parkinson's early.

As a whole, it is clear that Parkinson's detection is doable by Deep Learning Networks, and CNN's could be considered a reasonable approach. For the benchmark, I know that prior models also exist such as ResNet which is actively used in medical imaging; however, these are a benchmark, not a baseline.

Data Processing

I got my data set from Kaggle where I found 831 images of brain scans. Both of patients with Parkinson, and those without it. While data cleaning wasn't necessary, I had to emphasize data augmentation, resizing and normalization as the two main mechanics for my portion of data processing. Another important acknowledgement is class imbalances. Roughly 2/3rds of my data was Normal MRIs, and 1/3 was Parkinson's.

The first step of my data augmentation was rotating images. One of the problems with my data set is that I only have 831 pieces of total data. I realized that by slightly tilting images, I could double the size of the data set. So I performed a random angle tilt on all the images and increased my dataset size. However, this was done in a limited capacity between -30 and 30 degrees. Obviously patient brain scans wouldn't be completely flipped or rotated 90 degrees. Hence I decided on 30 degrees.

The next step was to ensure my data was a reasonable pixel value, and to normalize it. This would be done so that my model can be trained efficiently and capture appropriate features. For this I chose an input size of 128x128 pixels as mentioned in my proposal. The reason I normalized is because I want the scanner to focus on features, not intensity, and this will enable the learning of disease patterns. Once they were normalized, I used an 80-20 split. The test is the 20%. From there I'm also splitting the 80% into training and validation data sets. With me keeping about 20% once again for validation. This led to the following amounts: 1064 for training, 266 for validation, and 332 for final testing.

Baseline Model

For my baseline model, I'm using an ANN. The ANN is two-layer neuron network which utilizes a hidden layer, has a Relu activation function and one fully connected output layer. While ANN's are decent they can lead to excessive parameters, overfitting and poor generalization. Even still they're simple enough to generate a reasonable prediction on a smaller dataset. We have 16,384 features going into our model, 32 units within the hidden layer, and 2 logits for classes.

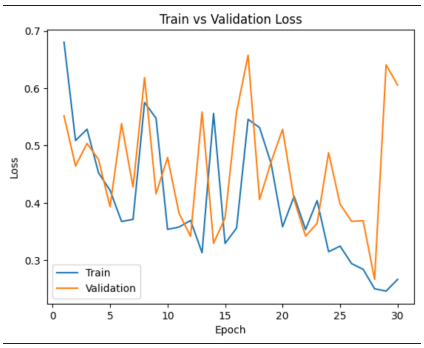

Figure 2: Train vs Validation Error of the ANN

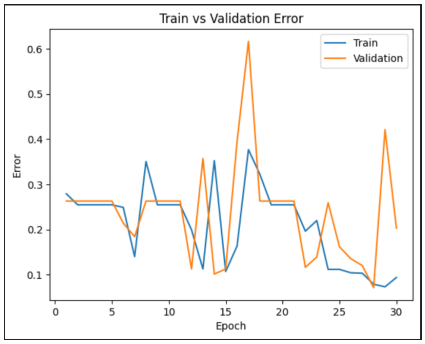

Figure 3: Train vs Validation Loss of the ANN

It can be visually seen that the ANN model is very erratic in both training/validation errors and loss. It is possible our data and model were limited in their parameters and number, but ultimately we know the ANN isn't ideal for medical imaging. As we predicted it is not exceptional at being able to do classification tasks. Ultimately, the baseline generated an accuracy of 74% on the test dataset which isn't bad, but we can definitely do better.

Evaluation and Model Results

The primary model I constructed was a CNN. As talked about in my proposal the reason I decided to go with a CNN was because they can address class imbalances, and are highly accurate. Especially at image recognition. Initially, I used an architecture of three convolutional layers, two pooling layers, and three fully connected layers. We had three main iterations to tune the parameters. I concluded on the batchsize of 32, a learning rate of 0.001 and 30 Epochs.

I wasn't entirely satisfied due to the fluctuations, so I decided to change the architecture. Our final CNN takes a 1×128×128 greyscale MRI input and applies three convolutional blocks. The first filter uses 32 filters, with a 3x3 kernel, a stride of 2 and padding of 1. The second uses 64 filters, (kernel 3×3, stride 2, padding 1) to produce 64×32×32 features. The third layer utilizes 128 filters, the same stride, padding and kernel. After flattening to 131,072 features, we use fully connected layers of sizes 128 and 64 with ReLU and dropout $p = 0.4$, followed by a final linear layer to 2 logits.

Illustration

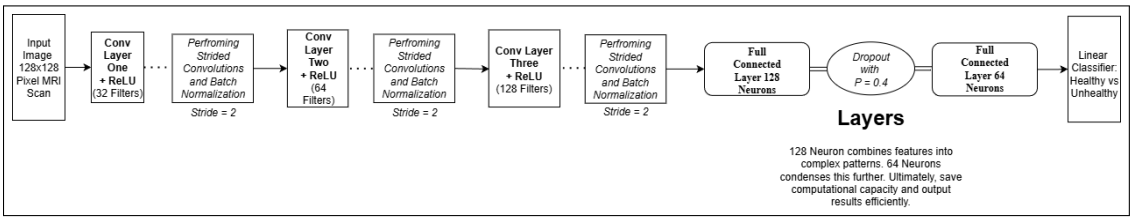

Figure 4: CNN architecture with three convolutional layers, strided convolutions, dropout, fully connected layers, and multi-class output.

The changes are used because I wanted to ensure that my model could learn more robust paths and distributed representations. Instead of the initial sigmoid I proposed, I decided to use a multi-classification system in the form of a linear classifier.

We can start off by looking at the Training and Validation curves to derive the qualitative results. These were chosen, because they are a good indicator of model performance, and let us see whether the model is following the general trend.

Figure 5: Train vs Validation Error of the CNN

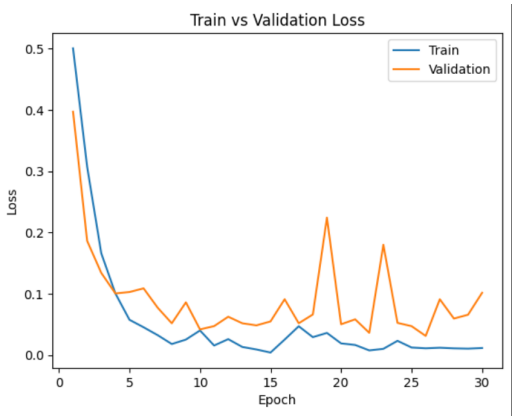

Figure 6: Train vs Validation Loss of the CNN

While not perfectly smooth, we can see that the model was able to fit the training and validation data reasonably well. While the loss did have some spikes which could indicate some overfitting, we could visually conclude that the model was performing well due to the steady decrease, and lower fluctuations. As well as the plateau at the end. We ended up concluding with a train error of 0.0009 and val error of 0.0263.

As explained in the data processing bit we set aside about 20% of the data and ensured that our model never looked at it prior to the final test. These images were never used for training or hyperparameter tuning, so they approximate performance on unseen data. We used a batchsize of 32, a learning rate of 0.001 and 30 epochs. As we had found earlier in the Parkinson Net initial model hyperparameter tuning. I wasn't able to source a dataset external to these Kaggle data set, so I kept these test samples as completely unseen and used them. Leading us to the following results: Here we can see that the model was very highly accurate on the test dataset, we ultimately generated an accuracy of 96%

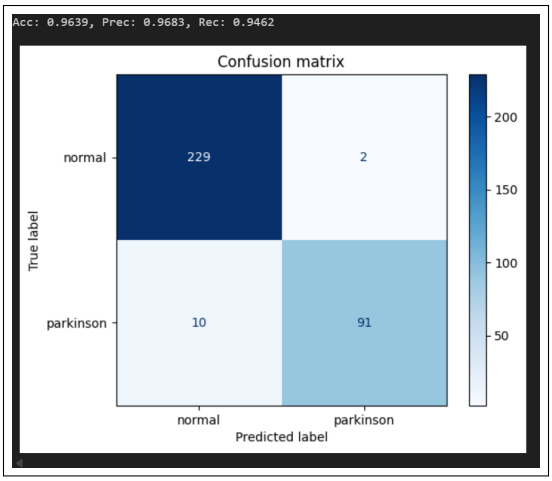

Figure 7: CNN Results with Confusion Matrix

which is a 22% increase over our baseline model. We also achieved a precision score of 97% which indicates a high degree of confidence in being able to determine true positive cases of Parkinson's with few errors. Finally, our recall at 96% was also quite high, indicating a strong ability to correctly identify people with Parkinson's disease.

Our final model has performed decently, it did meet my expectations, and I'm satisfied with the performance. Even still I learned that small architectural changes such as strided convolutions and dropout can significantly stabilize training and improve generalization on this limited medical imaging dataset.

Discussion

While the model did have a relatively high values of accuracy, recall, and precision. However, its important for us to consider the context of the problem, and within our dataset. Firstly, the data we have was very limited, we started off with 831 data points. Although, we did some processing, and were able to increase our dataset to 1662 it still isn't a large enough amount to generalize our models conclusions. There is concern whether our model is memorizing the examples within our data set. At the end of the day we only have 332 images left for our test data, which might allow our model to apply memorized patterns rather than fully learn. Another consideration for our model was the number of hidden layers, while three is decent, if we had more hidden layers, we could potentially increase the ability for us to extract features. Three layers allow us to extract more basic features within a MRI for Parkinson's disease, but the reality is that there might be some higher level features which we missed.

Overall the results were favorable, but it's important to remember the main concerns of the model. Both data validity and model architecture are things to consider in future iterations.

Ethical Considerations

While models like this can be useful, as we acknowledged the data validity was something to question. If this was to be implemented within a hospital, there are some serious considerations which would need to be taken. Firstly, due to the limits we can't generalize our conclusions. A 96% accuracy in the model is good, but it means there are cases we may miss, and this can be a real problem.

Our model being implemented in a hospital could be useful for radiologists, but at the end of the day we would need to make sure that a radiologist confirms the results with their own expertise and experience. Trusting the model blindly could lead to false negatives, and that is a scary thought for a serious disease.

References

[1] Manal Alrawis, Farah Mohammad, Saad Al-Ahmadi, and Jalal Al-Muhtadi. Fcn-pd: An advanced deep learning framework for parkinson's disease diagnosis using mri data. *Diagnostics*, 15(8):992, 2025.

[2] Milton Camacho, Matthias Wilms, Pauline Mouches, Hannes Almgren, Raissa Souza, Richard Camicioli, Zahinoor Ismail, Oury Monchi, and Nils D. Forkert. Explainable classification of parkinson's disease using deep learning trained on a large multi-center database of t1-weighted mri datasets. *NeuroImage: Clinical*, 38:103405, 2023.

[3] Y. Chang et al. Deep learning for parkinson's disease classification using multimodal imaging to differentiate parkinson's disease from multiple system atrophy. *Frontiers in Neuroscience*.

[4] Megan Courtman, Mark Thurston, Hongrui Wang, Sube Banerjee, Adam Streeter, Lucy McGavin, Stephen Hall, Lingfen Sun, Emmanuel Ifeachor, and Stephen Mullin. Deep learning classification of mri differentiates brain changes in genetic and idiopathic parkinson's disease. *medRxiv*, 2024.

[5] Sahar Shokrpour, AmirMehdi MoghadamFarid, Sepideh Bazzaz Abkenar, Mostafa Haghi Kashani, Mohammad Akbari, and Mostafa Sarvizadeh. Machine learning for parkinson's disease: a comprehensive review of datasets, algorithms, and challenges. *npj Parkinson's Disease*, 11(1):187, 2025.