1. Definition

Project Overview

Parkinson’s is a neurological disease which specifically affects movement. One of the most distinguishing symptoms of the disease is tremor and rigidity faced by patients. The patient’s condition keeps worsening over time and being a neuro-degenerative disease, it’s incurable in nature.

In medical terms, nerve cell damage in the brain causes the dopamine levels to drop, causing tremor in hands, legs and other parts of the body. The tremor in hand causes the patient to produce messy, non-clear greatly impacting the visual appearance of diagrams and figures drawn by Parkinson’s patients. Parkinson’s disease is a disease in itself and other than that it can also act as a symptom in other forms of dementia, for eg: people suffering from Fronto-Temporal Dementia have a tendency to display signs of tremor although the disease is not Parkinson’s. It’s known as Parkinsonism. With that said, it’s not possible to detect Parkinson’s disease just by few tests or symptoms but Parkinsonism in itself can be detected by asking the patient to draw certain shapes.

I was inspired for this project because of personal challenges upfront.

In this project, I created a python application that sounds capable of detected Parkinsonism in patients through geometric drawing test. Deep Learning is used here and dataset used is from UCI ML repository along with customized dataset.

Problem Statement

Research has shown conclusive evidence that the drawing speed and pen pressure is affected the most among Parkinson’s patients. From stage 2, this is generally the case. The current procedure of identifying Parkinsonism includes offline based test taken by neurologists. There is no digitized form of detecting Parkinsonism from drawn shapes & geometric drawings with Deep Learning in high accuracy. Such a project would advance the research further and prove to be useful in understanding drawing semantics in detail. Summarizing, it can be used to distinguish between healthy people and people with Parkinsonism symptoms. Build a python application that would be able to distinguish between people suffering from Parkinsonism and healthy people through geometric drawing test.

Metrics

1) Accuracy is the metric used here, it takes into consideration True Negative and True Positive.

Accuracy=(TN+TP)/Total dataset size

2) Confusion Matrix

It is a table used to describe the performance of a classification model.

2. Analysis

Data Exploration

The dataset is taken from UCI ML repository. Along with that, samples are added from some real Parkinson's patients. Total, there are 80 samples of patients with Parkinson's and 20 samples of healthy patients.

Exploratory Data Visualization

The dataset of people with Parkinson's have the foll characteristics with respect to the geometric drawings:

a. The drawings are abnormally looking
b. The drawings are asymmetrical as opposed to how they should be
c. The drawings go out of the zone where they should be
d. The drawings are terribly shaken and there is clear visibility of something purely 'not normal'

These data points form the feature set of our concern.

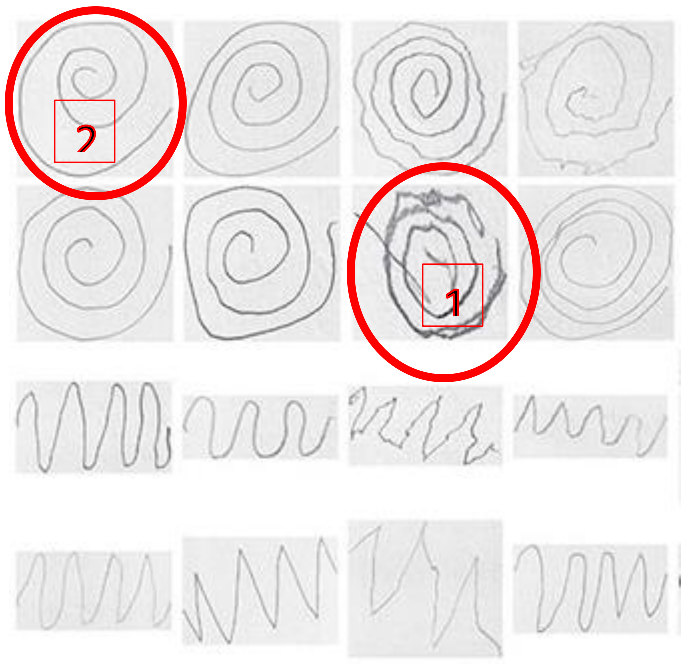


Figure 1. Initial dataset

1- Spiral drawn by a person with Parkinson
2- Spiral drawn by a healthy person

3. Algorithms and Techniques

The hig- level algorithm used is Computer Vision Learner, proven state-of-the-art technique for several image processing tasks. This is a part of Transfer Learning. In this technique, the existing model has already been trained on a very large dataset (ImageNet) and we can use the model even for a decent sized or small dataset as in my case. Backbone model in this is Convolutional Neural Network. How this is done is the last convolutional layer has analyzed features that went through the model and the task is to convert them into end predictions suitable to our classes (healthy and Parkinson in this case).

For optimizing the classifier, the parameters changed are:

- Length of training
- Size of batch (no of images given in a single training step)
- Learning rate

Benchmark Model

The benchmark model is the use of HOG (for Computer Vision) and Random Forest Classifier with an accuracy of 83.4%.

HOG: Histogram of Oriented Gradients is a feature descriptor specifically used for object detection in Computer Vision and Image processing. In projects such as these, HOG is used to quantify the input images. It performs really well when coupled with Random Forest even in limited amount of training data. It comes with scikit-image python package. Programmatically, HOG is a method used to define with a spiral or other diagram image. HOG is a structural descriptor which captures and quantifies changes in local gradient within the input image. HOG is naturally able to quantify the directions of a both spirals as well as waves gradient. Additionally, HOG is able to capture if these patient’s drawings have more of a tremor to them, as we might generally expect from a Parkinson’s patient.

Random Forest Classifier: The training is done using Random Forest classifier on top of extracted features. Random Forest fits several decision tree classifiers on various sub-samples of the dataset and leverages averaging to improve the predictive accuracy and particularly control over-fitting.

4. Methodology

a. Dataset creation and preprocessing

The dataset used by me is small but it the model used is from Transfer Learning so the original dataset used is ImageNet. Dataset creation was done by taking samples of patients with Parkinsonism after they were asked to draw geometric drawings.

I. Images were initially randomized
II. Created dataset was classified into training and testing set
III. Conversion of those images into grayscale

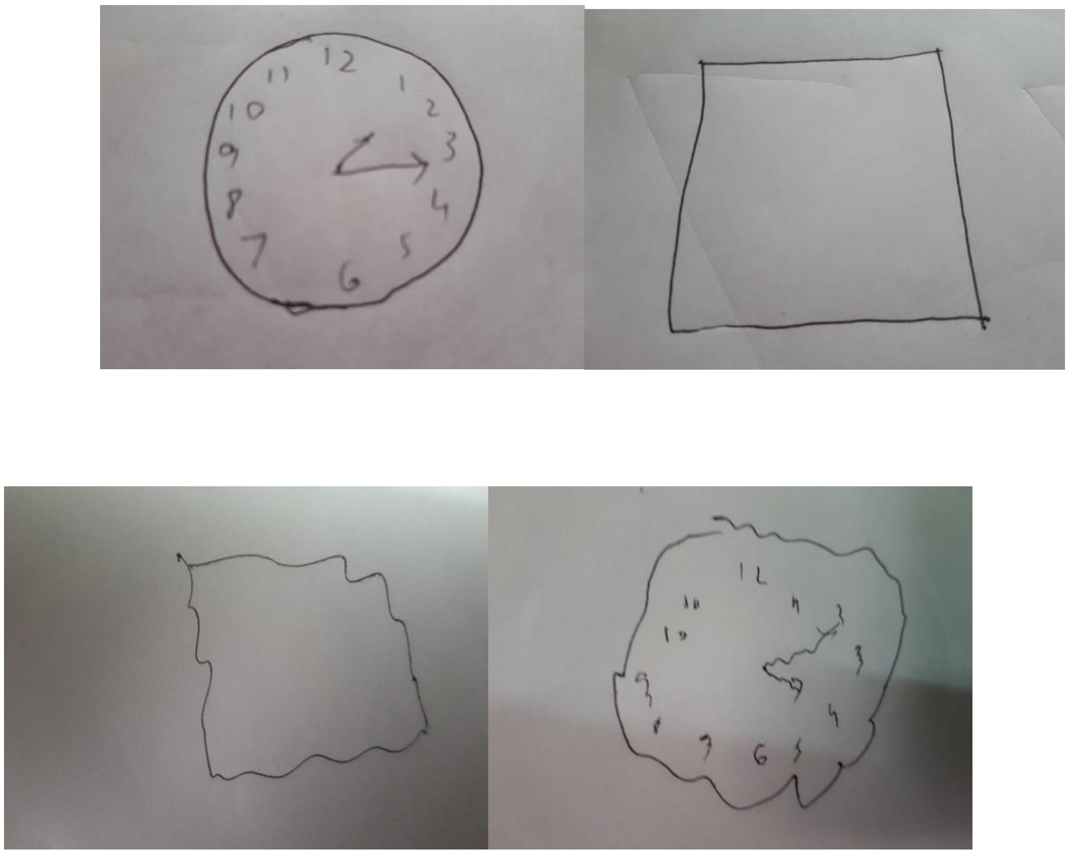


Figure 2: Dataset creation

Implementation

b. Convolutional Layers

   The backbone used is convolutional layer and it will have analyzed features in images that passed through the model and the job of the head (few of the linear layers) is to convert those in predictions specifically for each of our classes. The backbone is kept intact but the head is changed according to our model.

c. Model training details

   Initially, the body weights of the model are frozen. Then, the head is trained to get predictions from analyzed features.

d. Unfreezing backbone layers

   Then, all the necessary layers of backbone which are necessary are allowed. Since CNN is used as the backbone, not all the layers are used.

Refinements

e. Model fine-tuning

   Fine tuning is similar to inferencing where the results of one trained task is applied elsewhere. This method including testing with different learning rates until the one is found which delivers the best result.

5. Results

Model Evaluation

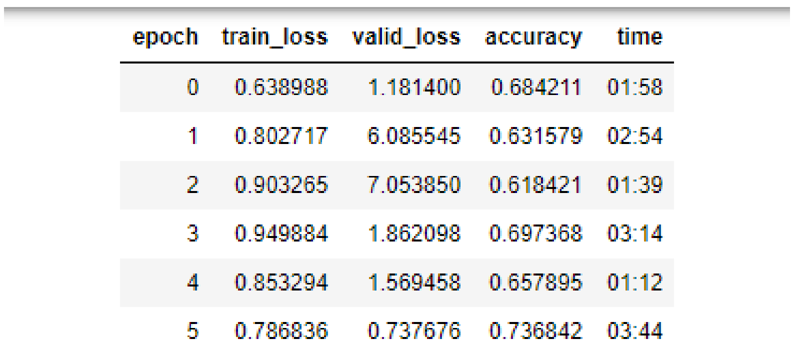
Initially, CNN was chosen but it felt like beating a pin with a hammer since the dataset isn't big enough (less than 100 images for training) to use Deep Learning directly. In examples such as these, Transfer Learning is the best bet because:

- A highly trained model on a huge dataset is needed
- That would serve as the base for model fine-tuning
- That would suffice for the lack of a huge dataset in this project

Going by sensitivity analysis, the model performs well even if the brightness is changed with regards to some of the images in the dataset. This is proven through the dataset which is in GitHub repo. In that way, the model is pretty robust.

- ➢ The final model is absolutely reasonable and aligns with the solution expectations.
- ➢ The model has been thoroughly tested.
- ➢ The model is robust enough and small changes in input space or training data do not greatly affect the training results.
- ➢ The results from the model can be found as attached Jupyter notebook. It can be run after installing dependent libraries.

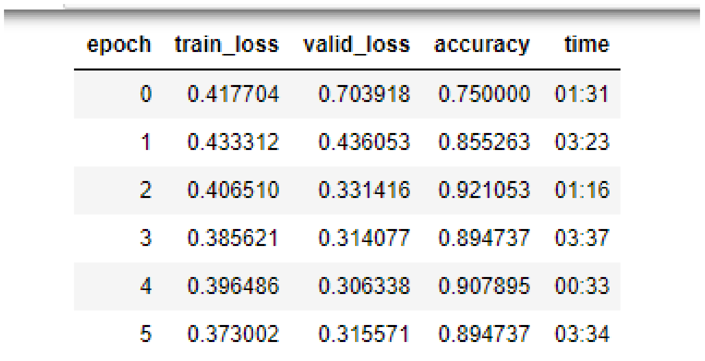
The testing dataset is used to evaluate the model.



epoch	train_loss	valid_loss	accuracy	time
0	0.638988	1.181400	0.684211	01:58
1	0.802717	6.085545	0.631579	02:54
2	0.903265	7.053850	0.618421	01:39
3	0.949884	1.862098	0.697368	03:14
4	0.853294	1.569458	0.657895	01:12
5	0.786836	0.737676	0.736842	03:44

Figure 3: Initial results

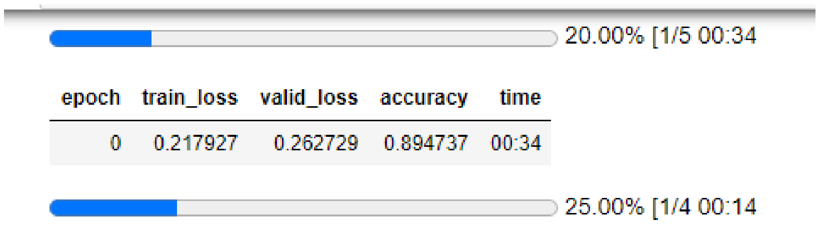
The results after unfreeze() function was found out to be 73%. References and literature survey suggested that the accuracy could be improved by a huge margin.



epoch	train_loss	valid_loss	accuracy	time
0	0.417704	0.703918	0.750000	01:31
1	0.433312	0.436053	0.855263	03:23
2	0.406510	0.331416	0.921053	01:16
3	0.385621	0.314077	0.894737	03:37
4	0.396486	0.306338	0.907895	00:33
5	0.373002	0.315571	0.894737	03:34

Figure 4: Results

After adjusting the learning rate and using fit_one_cycle() method, the accuracy jumped to 89.4%.



```
20.00% [1/5 00:34

epoch  train_loss  valid_loss  accuracy  time
0      0.217927    0.262729    0.894737  00:34

25.00% [1/4 00:14
```

Figure 5: Final results

This is the final result, after iterating the model over 7 epochs. The model was trained for 4-6 epochs as well, but it didn't show satisfactory results.

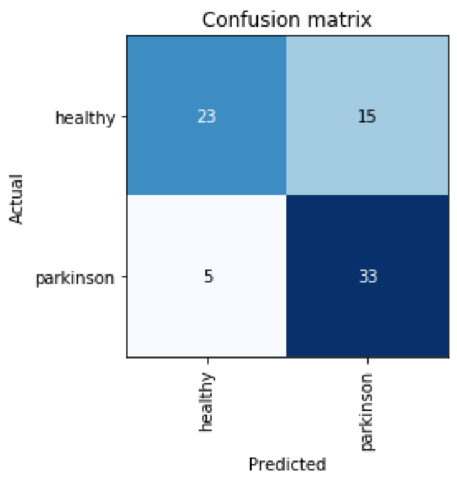


Figure 6: Confusion Matrix

A confusion matrix depicting the final result.

Justification

- The final results are stronger than the benchmark (83.4%). There is evidence in the result section of the Jupyter notebook.

- The model was trained on a much variant and bigger dataset as compared to the benchmark.

Uniqueness

I can say that Parkinson's disease comprises of a lot of other health conditions. The symptoms of tremor and rigidity can be at times a part of other dementia in the form of symptoms. Generally, spiral wave test is used to detect tremors displayed on the hand while drawing. But in my case, I'm using a bunch of geometric drawings to detect Parkinsonism. Mine is different in that manner as it's not related to just spiral wave test. I know of some of the other shapes & figures neurologists use to detect Parkinsonism as I've experienced the same.

- The final result is significant to have solved the problem since it tests correctly if the geometric drawings were drawn by a healthy being or a patient suffering from Parkinsonism.
- In other similar projects, attempt was being made to detect Parkinson's disease and it is wrong since technically, Parkinson's disease can't be detected by tremor alone, there are symptoms which need to be validated.
- On the other hand, Parkinsonism can be detected since Parkinsonism technically is a term generally used for detecting tremors as displayed through abnormal geometric drawings.

6. Conclusion

Free-Form Visualization

The unique aspect of the project as discussed above is that other than spiral, there are other geometric drawings (clock, triangle and rectangle) to detect Parkinsonism that are mentioned here, used in clinical practice, but hardly referenced in any of the projects.

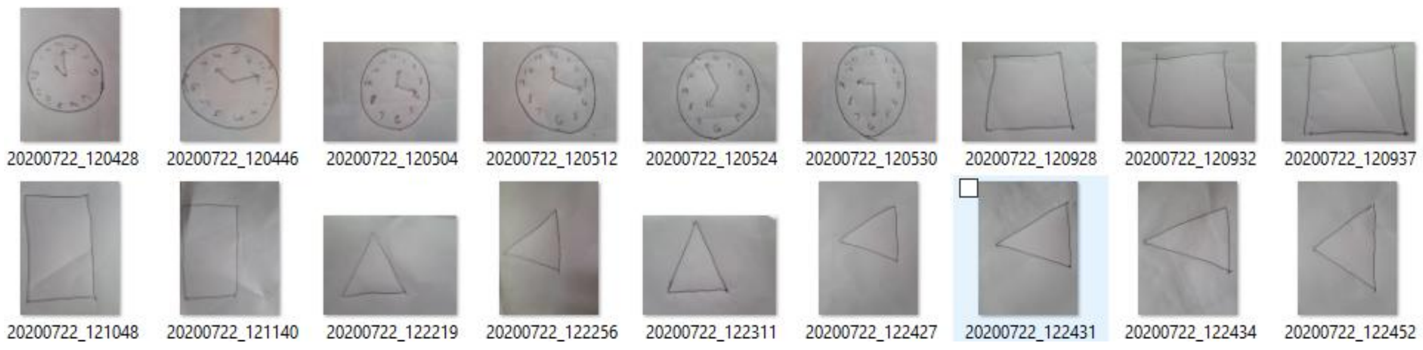


Figure 7: Training dataset

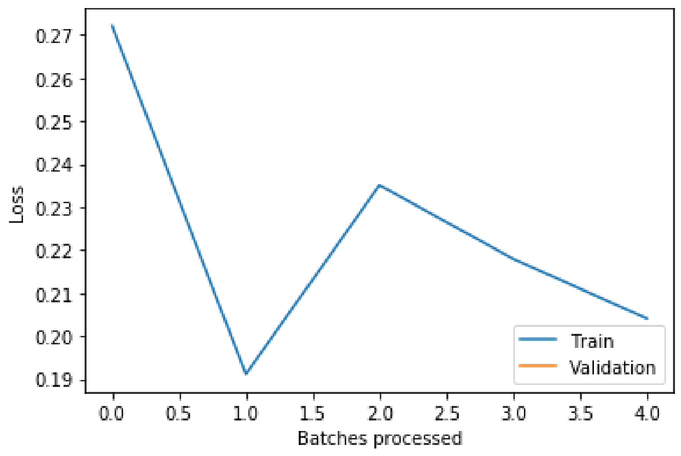


Figure 8: Plot of train loss

[Figure 8] depicts the plot of train loss with the number of batches processed. It is clearly visible that the loss decreases as and when more batches are processed.

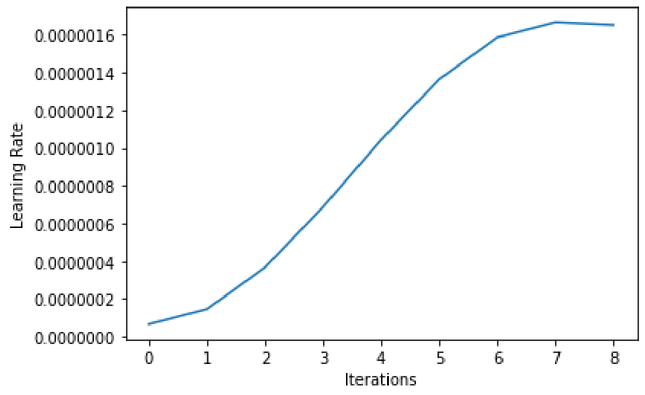


Figure 9: Plot of learning rate

This is an important plot since the learning rate goes a long way in defining how the model would turn out to be. It is observed that the learning rate gradually becomes near asymptotic when its value approaches 0.0000016.

Reflection

Some of the challenges I faced in this project:

I. Dataset creation: My firm target was to have images of geometric drawings and not just spiral or wave which were available almost everywhere. For that, I had to go and physically talk to people with Parkinsonism, then take their data. It took some efforts to get it done.
II. Model selection: I was stuck at selecting CNN for long after which I realized that if the dataset is not big enough, there would be no point in using CNN. I'd rather go with Transfer Learning as its statistically the best bet in situations similar to these.

My past experience: My past experience was really tough since **my father is suffering from Parkinsonism.** I'm very well aware of the various stages a patient goes through as well the nuances of it. I've spoken to 15+ neurologists and neuropsychologists in the country including some of the reputed and best ones in the country and I have a good perspective about Parkinsonism and Parkinson's disease. That's how I applied that knowledge in my project.

Summary: Overall, the problem fits my expectations for the problem and it should be used in a general settings to solve these kind of problems.

Improvement

These are some of the aspects of the project which could be improved:

1. Fp16 could be used to improve speed
2. A much larger dataset can be used to get more accurate and reliable results although current results are more than enough to draw observations from
3. There could be some kind of normalizer to understand, for example if someone’s hand is injured, it might tremble during drawing, it could be understood if there is some kind of normalizer, one of the ways for it to get to get done would be to add additional data drawn by the person in context

Overall, this final solution can be used as the new benchmark.