Team 3

Aya Eyad - Habiba Salama - Hadeer Sherif - Mariam Hatem - Yasmin Tariq

Clinical Decision Support System (SBE 3030)

01 January 2024

Parkinson's Detection

1. ABSTRACT

Parkinson's disease (PD) is an irreversible neurological disorder, and currently, there is no established medical protocol for Parkinson's disease diagnosis. This research focuses on investigating a specific fine motor symptom, namely sketching. The study involves conducting experiments on a substantial number of individuals with PD and a healthy control group without PD. The objective is to introduce a system capable of differentiating PD patients based on their sketching behavior. Utilizing deep learning algorithms, particularly Convolutional Neural Network (CNN), we aimed to classify sketched images and distinguish individuals affected by Parkinson's Disease from those in the healthy control group. The study employed various CNN models, incorporating transfer learning techniques and applying them to Spiral and Wave sketched data. The proposed system demonstrated promising results, achieving an accuracy of 86.67% on the VGG model specifically when analyzing spiral sketches

2. LITERATURE REVIEW

2.1. Parkinson's Disease Detection Using ResNet50 with Transfer Learning [1]. In this investigation, a computational system was developed for discerning Parkinson's disease through the analysis of sketches. The methodology employed Convolutional Neural Networks (CNNs), specifically utilizing the ResNet50 model, and focused on classifying spiral drawings generated by individuals with Parkinson's and healthy subjects. The outcomes demonstrated a commendable ***accuracy rate of 96.67%***, underscoring the efficiency of the system in distinguishing between sketches originating from healthy individuals and those affected by Parkinson's disease.

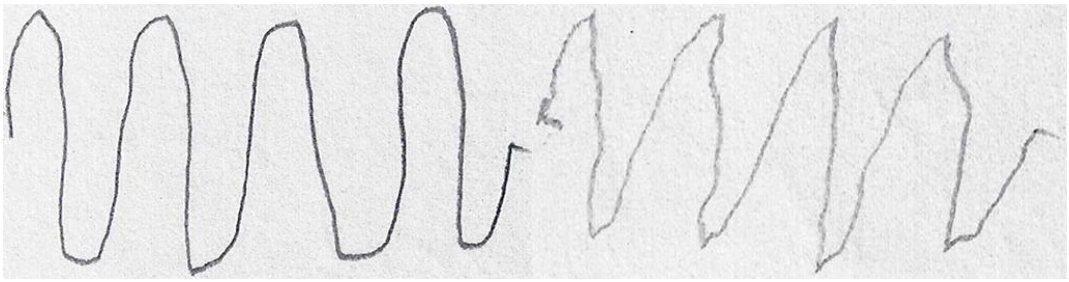
2.2. Deep Transfer Learning Based Parkinson's Disease Detection Using Optimized Feature Selection [2]. This paper introduces a new approach for accurately detecting Parkinson's disease using handwritten records from the standard NewHandPD dataset. The proposed method utilizes transfer learning models, including ResNet, VGG19, and InceptionV3, to mitigate training time requirements. The collective features extracted from these transfer learning models undergo an optimization process using a genetic algorithm. This optimization aims to obtain an optimized feature vector that enhances classification results. The key contribution of the presented model lies in its ability to identify the most optimal features, leading to enhanced performance accuracies. Feature optimization is achieved through a genetic algorithm in conjunction with the K-Nearest Neighbour technique. The innovative model yields a detection **accuracy exceeding 95%, a precision of 98%**, an area under the curve of 0.90, and a minimal loss of 0.12. To showcase its superior detection capability, the performance of the proposed model is compared

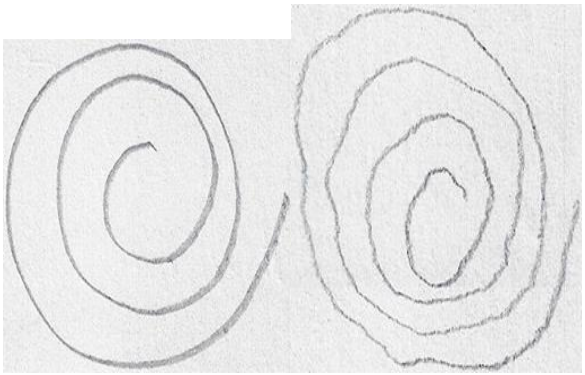
with several state-of-the-art machine learning and deep learning-based Parkinson's disease detection approaches.

3. DATASET

3.1. About the data. The used data is the Parkinson's Drawings on Kaggle [1] which came from the paper: Zham P, Kumar DK, Dabnichki P, Poosapadi Arjunan S and Raghav S (2017) Distinguishing Different Stages of Parkinson’s Disease Using Composite Index of Speed and Pen-Pressure of Sketching a Spiral. Front. Neurol. 8:435. doi: 10.3389/fneur.2017.00435 [2]. The dataset has a total of 204 sketches with two types of drawings; 102 spiral sketches and 102 wave sketches. Each of the two sketches is split into two categories; 30 testing sketches and 72 training sketches. The dataset is further split into 15 healthy sketches and 15 Parkinson's sketches for the testing category, and 36 healthy sketches and 36 Parkinson's sketches for the training category.

3.2. Visualization.





3.3. Preprocessing. Preprocessing is crucial for training deep learning models, as it helps the model generalize better to new data, and can significantly improve the model's performance. Four preprocessing methods and algorithms were performed on the dataset.

3.3.1. *Keras' ImageDataGenerator.* This tool was used for performing image augmentation and preprocessing. Image augmentation is a technique used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. The "preprocessing_function=preprocess_input" argument applies a model-specific preprocessing function. This function prepares the images in the way that the model was originally trained. It scales the pixel values and adjusts the colors. The "brightness_range=[0.5, 1.5]" argument adjusts the brightness of the images. This helps the model generalize better to different lighting conditions in the input images. The "validation_split=0.2" argument reserves 20% of the images for validation. This allows to evaluate the performance of the model on unseen data during training.

3.3.2. *Zhang-Suen.* The Zhang-Suen thinning algorithm was implemented to reduce the width of an object in an image to a single pixel width, which is useful in image processing tasks such as edge detection and image recognition. The algorithm works by iteratively removing pixels from the edges of objects in the image until no more pixels can be removed.

3.3.3. *Data blurring.* A Gaussian blur was also applied to some of the images as a data augmentation technique. The Gaussian blur is a type of image-blurring filter that uses a Gaussian function. This is a widely used effect in graphics software, typically to reduce image noise and detail.

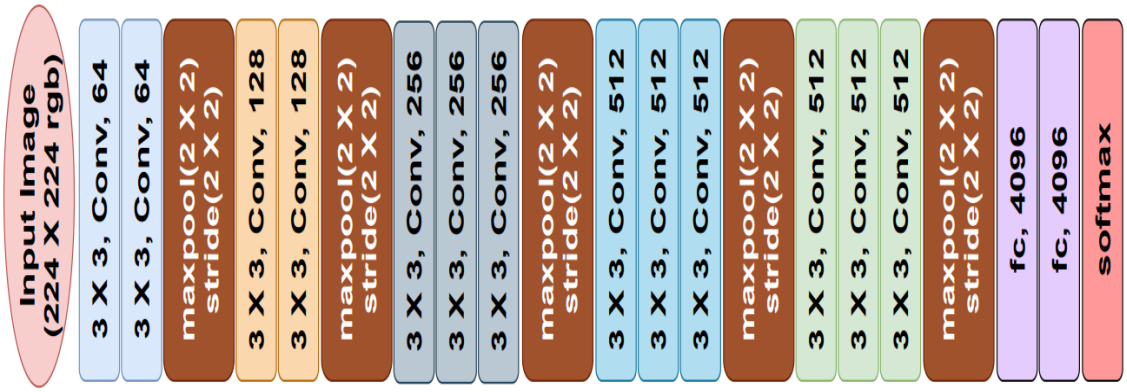
3.3.4. *Otsu thresholding.* Image thresholding was performed using Otsu's method, which is a way of automatically determining the best threshold value to separate the foreground and background of an image. The result is a binary image where the pixels of the objects of interest are white (255) and the background pixels are black (0). We then reversed them to make the background white and the foreground black as was the case in the original images.

4. TRANSFER LEARNING

Transfer learning is a machine learning technique that leverages knowledge gained from training a model on one task and applies it to a different, but related, task. It is particularly valuable when working with limited data for a new task, as the pre-trained model has already learned useful features from a different but related domain.

4.1. Used models.

4.1.1. VGG-16

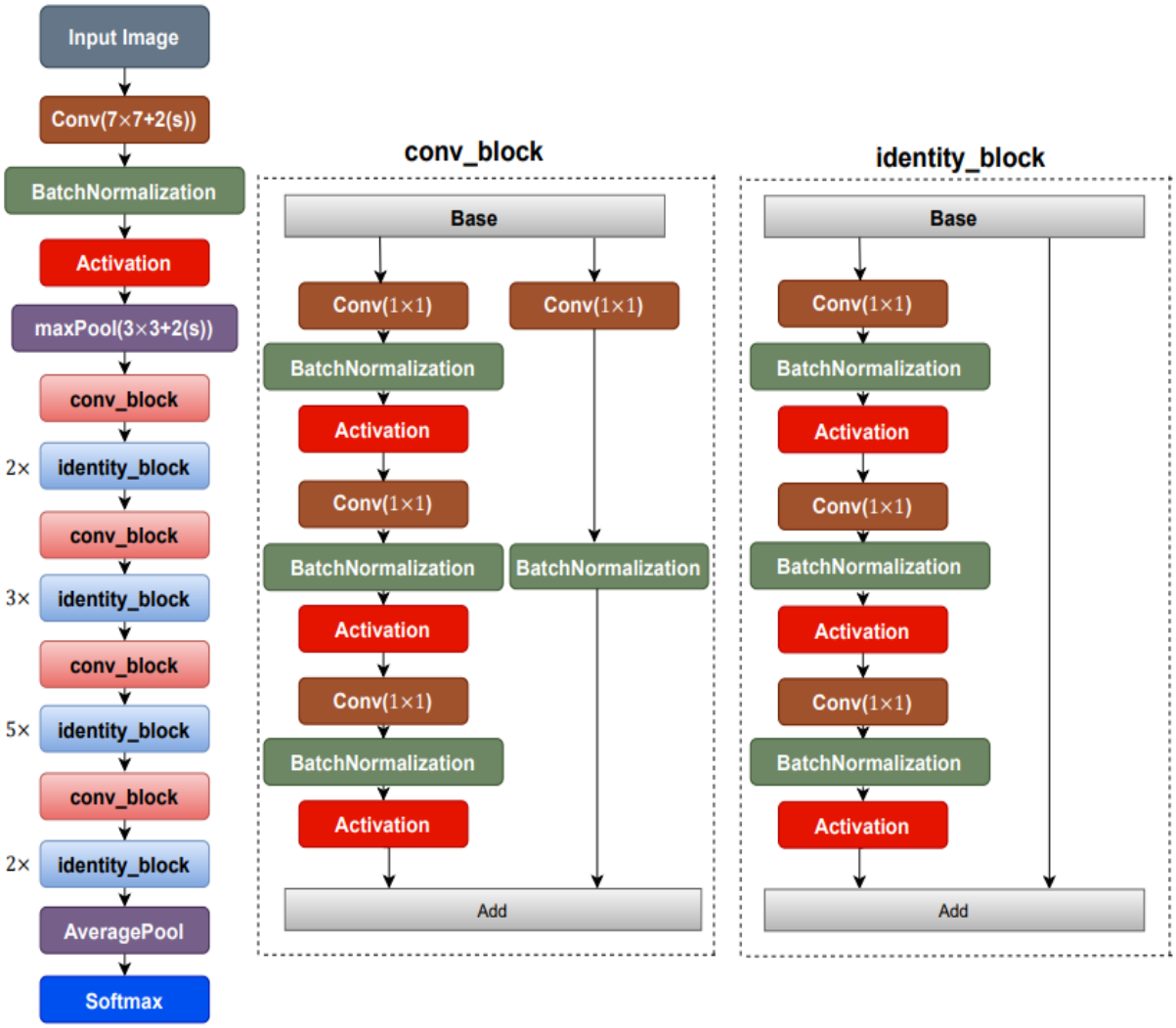


VGG-16 (Visual Geometry Group) is a convolutional neural network (CNN) architecture that was developed by Oxford and won the ILSVR (ImageNet) competition in 2014. It is also called OxfordNet. This architecture consists of 13 convolutional layers and 3 fully connected

layers. The input size is the default, and it is 224 × 224 RGB images. There is a fixed 3 × 3 filter size in convolutional layers [16]. The figure below shows VGG-16 architecture.

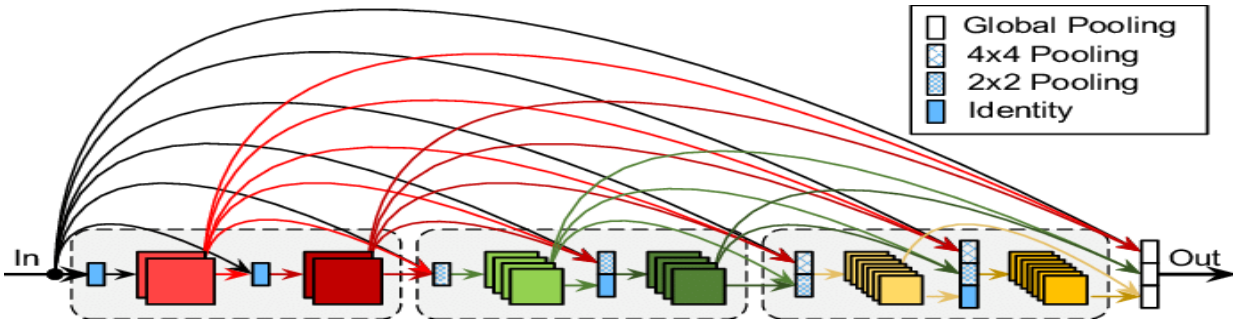
4.1.2. ResNet50

ResNet50, short for Residual Network with 50 layers, is a specific deep learning architecture within the ResNet family. ResNet architectures are known for introducing the concept of residual learning, which involves the use of residual blocks to facilitate the training of very deep neural networks. The "50" in ResNet50 refers to the number of layers in the network.



4.1.2. DenseNet

DenseNet architectures consist of densely connected blocks, which include convolutional, batch normalization, and activation layers. Each dense block is followed by a transition layer that reduces the spatial dimensions (width and height) of the feature maps.

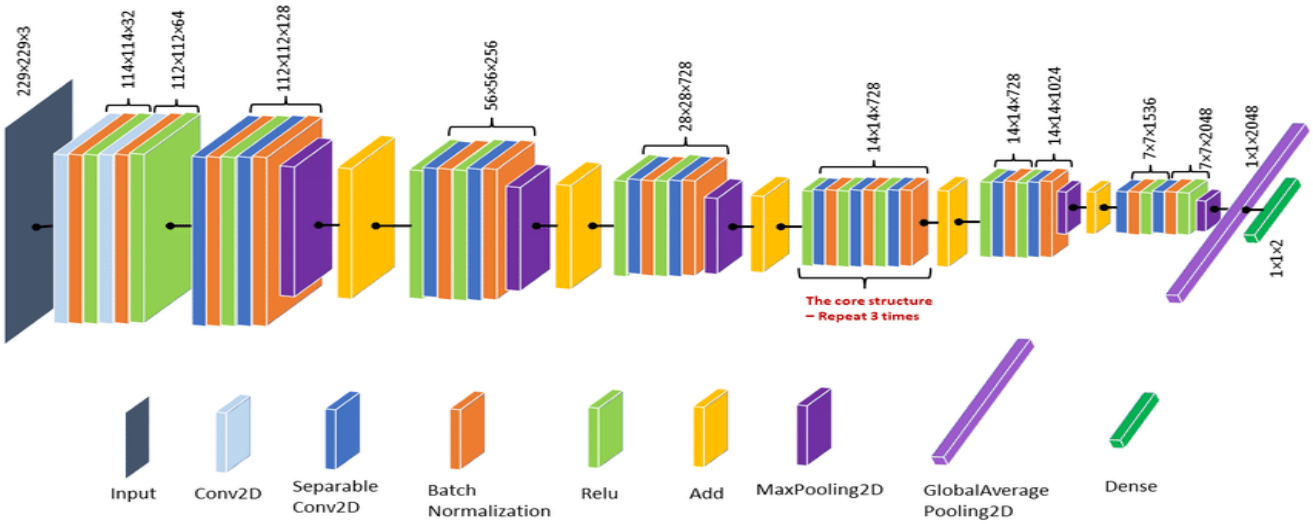


4.1.2. Xception

Xception is based on the concept of depthwise separable convolutions, which is a more factorized form of the standard convolutional operation. In a traditional convolutional layer, the operation involves convolving the input with a set of filters, leading to a large number of parameters. Depthwise separable convolutions, on the other hand, split the standard convolution into two separate operations: a depthwise convolution and a pointwise convolution.

4.2. Tuning the two best models.

4.2.1 VGG-16



1) Augmentation (photometric and geometric)

   Epchos: 5

   Test loss: 0.87

   Test accuracy: 70%

2) Augmentation (photometric and geometric)

   Epochs: 10

   Test loss: 0.653

   Test accuracy: 80%

3) Augmentation (photometric and geometric)

   Epochs:10

   Learning rate: Default

   A dense layer with 64 neurons and then another dense layer with 32 neurons were added

Test loss: 0.60

Test accuracy: 86.7%

4) Augmentation (photometric and geometric)

Two layers were added

Epochs:10

Adam(learning_rate=0.1)

Test loss: 0.69

Test accuracy: 50%

5) Augmentation (photometric and geometric)

Epochs: 10

Learning rate: 0.5e-6

Test accuracy: 81.67%

6) Augmentation (photometric and geometric)

Epochs: 10

Learning rate: 0.5e-5

Test accuracy: 78.3%

7) Augmentation (photometric and geometric)

Epochs: 20

Learning rate: 0.5e-5

Test accuracy: 80%

8) Augmentation (photometric and geometric)

Epochs: 35

Learning rate: 0.5e-5

Test accuracy: 80%

9) Augmentation (the same with fewer epochs and augmentation)

Epochs: 35

Learning rate: 0.5e-5

Test accuracy: 86.6%

10) Augmentation (photometric and non-geometric)

Epochs: 40

Learning rate: 1e-5

Test accuracy: 81%

11) First augmentation (non-geometric)

Epochs: 15

Learning rate: 1.0e-5

Test accuracy: 86.6% (Best)

4.2.2 ResNet

(photometric augmentation only)

1) Optimizer: Adam

Epochs: 10

Learning rate: 0.001

Test accuracy: 81.6%

2) Optimizer: Adam

Epochs: 20

Learning rate: 0.001

Test accuracy: 81.6%

3) Optimizer: Adam

   Epochs: 10

   Learning rate: 1e-5

   Test accuracy: 83%

4) Optimizer: Adam

   Epochs: 10

   Learning rate: 1e-4

   Test accuracy: 80%

5) Optimizer: RMSprop

   Epochs: 10

   Learning rate: 1e-5

   Test accuracy: 80%

(photometric and geometric augmentation)

6) Optimizer: PMSprop

   Epochs: 20

   Learning Rate: 1e-5

   Test accuracy: 85% (Best)

7) Optimizer: Adam

   Epochs: 20

   Learning rate: 1e-5

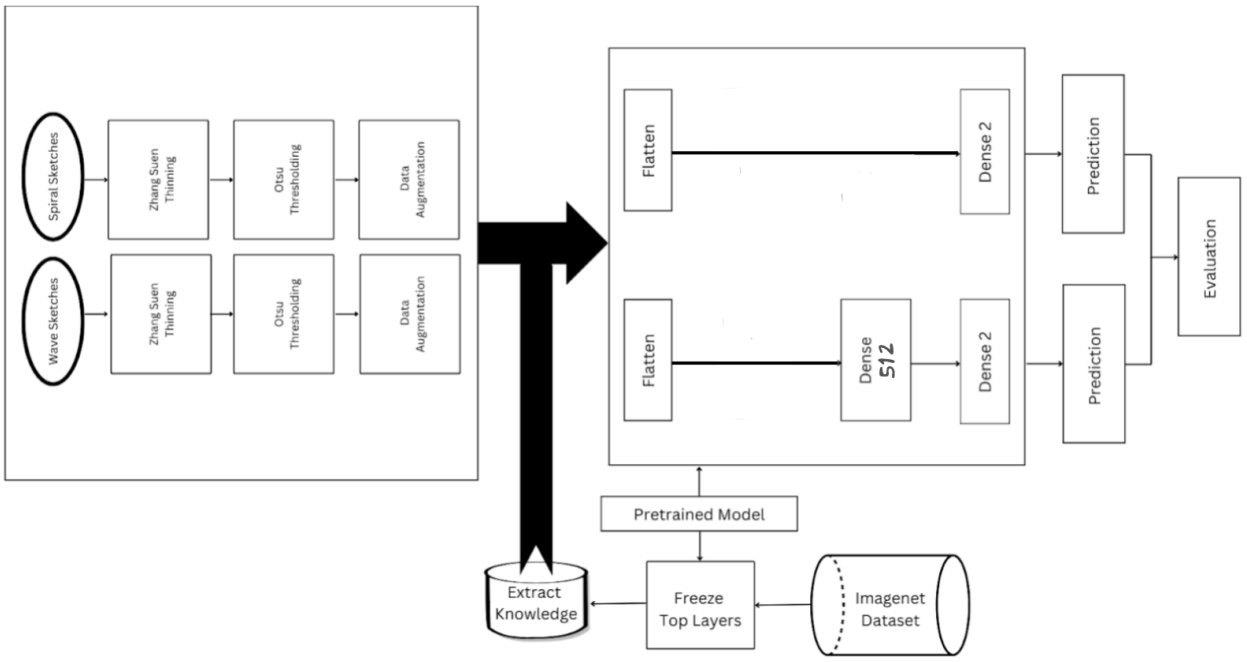
   Test accuracy: 73.3%

5. SCHEMATIC DIAGRAMS

The “plot_model” function from the “tensorflow.keras.utils” module was used to visualize the schematic diagrams of the four used architectures. The “tensorflow.keras.utils” is a module in

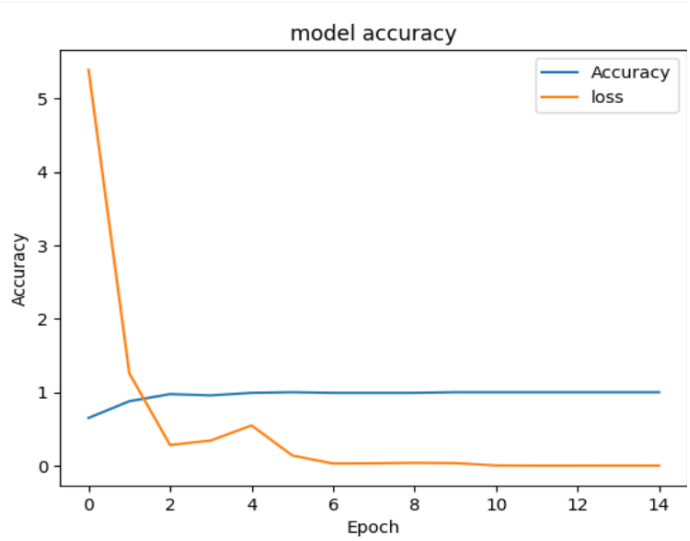
TensorFlow that provides various utility functions and classes to assist with data preprocessing, model visualization, and other tasks and can be found here.

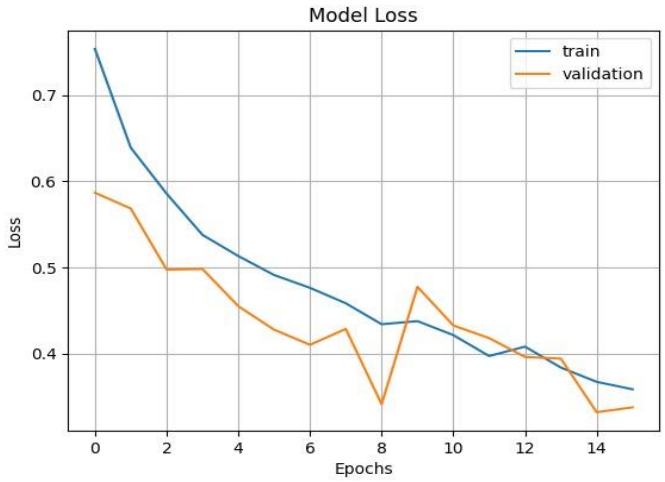
6. BLOCK DIAGRAM



7. RESULTS AND CONCLUSION

The accuracies of the models VGG-16, and ResNet50 were estimated by tuning different hyperparameters and figuring out various gains and losses of these models on spiral and wave sketching image data.





Observing the results shown in the above two figures for the VGG-16 (the one on the left) and ResNet50 (the one on the right), it is evident that these models exhibit superior performance when analyzing spiral sketches compared to wave sketches, particularly in the context of Parkinson's disease detection. Subsequently, a more in-depth analysis focused on spiral sketching. Upon comparing the accuracy of these models, it was observed that the VGG-16 model outperformed the ResNet model in terms of accuracy for spiral sketching.

So to sum up, this study found that the VGG-16 model for learning rate 1.0e-5 which provides 86.67% accuracy is the best transfer model for the problem of Parkinson's Detection.

Works Cited

[1] *http://cennser.org/IJCVSP/finalPaper/110103.pdf*

[2] *https://www.researchgate.net/publication/366857566_Deep_Transfer_Learning_Based_Parkinson%27s_Disease_Detection_Using_Optimized_Feature_Selection*

[3] *https://www.kaggle.com/datasets/kmader/parkinsons-drawings*

[4] *https://www.frontiersin.org/articles/10.3389/fneur.2017.00435/full*