**SOURCE CODE**

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

from tkinter.filedialog import askopenfilename

import cv2

import random

import numpy as np

from keras.utils.np\_utils import to\_categorical

from keras.layers import MaxPooling2D

from keras.layers import Dense, Dropout, Activation, Flatten

from keras.layers import Convolution2D

from keras.models import Sequential

from keras.models import model\_from\_json

import pickle

import os

import pandas as pd

import matplotlib.pyplot as plt

main = tkinter.Tk()

main.title("Convolutional Neural Networks Model for Parkinson Disease Detection from Images")

main.geometry("1300x1200")

global filename

global image\_classifier

def loadDLModel():

global image\_classifier

with open('model/images\_model.json', "r") as json\_file:

loaded\_model\_json = json\_file.read()

image\_classifier = model\_from\_json(loaded\_model\_json)

json\_file.close()

image\_classifier.load\_weights("model/images\_model\_weights.h5")

image\_classifier.\_make\_predict\_function()

print(image\_classifier.summary())

f = open('model/images\_history.pckl', 'rb')

data = pickle.load(f)

f.close()

acc = data['accuracy']

accuracy = acc[48]\*100

text.insert(END,"Deep CNN accuracy ")

text.insert(END,str(accuracy))

def imageDetection():

global image\_classifier

labels = ['Healthy','Parkinson']

filename = filedialog.askopenfilename(initialdir="testImages")

image = cv2.imread(filename)

img = cv2.resize(image, (64,64))

im2arr = np.array(img)

im2arr = im2arr.reshape(1,64,64,3)

img = np.asarray(im2arr)

img = img.astype('float32')

img = img/255

preds = image\_classifier.predict(img)

predict = np.argmax(preds)

img = cv2.imread(filename)

img = cv2.resize(img, (500,400))

cv2.putText(img, 'Image Data Predicted as : '+labels[predict], (10, 25), cv2.FONT\_HERSHEY\_SIMPLEX,0.7, (255, 0, 0), 2)

cv2.imshow('Image Data Predicted as : '+labels[predict], img)

cv2.waitKey(0)

def graph():

f = open('model/images\_history.pckl', 'rb')

image = pickle.load(f)

f.close()

img\_accuracy = image['accuracy']

img\_loss = image['loss']

plt.figure(figsize=(10,6))

plt.grid(True)

plt.xlabel('Epochs')

plt.ylabel('Accuracy/Loss')

plt.plot(img\_accuracy, 'ro-', color = 'red')

plt.plot(img\_loss, 'ro-', color = 'green')

plt.legend(['Training Accuracy', 'Training Loss'], loc='upper left')

plt.title('Performance Comparison')

plt.show()

def close():

main.destroy()

font = ('times', 16, 'bold')

title = Label(main, text='Convolutional Neural Networks Model for Parkinson Disease Detection from Images',anchor=W, justify=CENTER)

title.config(bg='yellow4', fg='white')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 13, 'bold')

loadButton = Button(main, text="Load Deep CNN model", command=loadDLModel)

loadButton.place(x=50,y=100)

loadButton.config(font=font1)

imageButton = Button(main, text="Detect Parkinson", command=imageDetection)

imageButton.place(x=50,y=150)

imageButton.config(font=font1)

graphButton = Button(main, text="Performance Graph", command=graph)

graphButton.place(x=50,y=200)

graphButton.config(font=font1)

exitButton = Button(main, text="Exit", command=close)

exitButton.place(x=50,y=250)

exitButton.config(font=font1)

font1 = ('times', 12, 'bold')

text=Text(main,height=20,width=78)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=450,y=100)

text.config(font=font1)

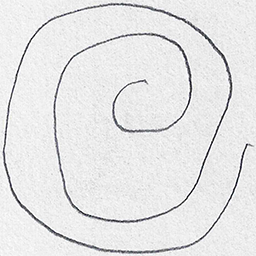
main.config(bg='Sky Blue')

main.mainloop()

**RESULTS AND DISCUSSION**

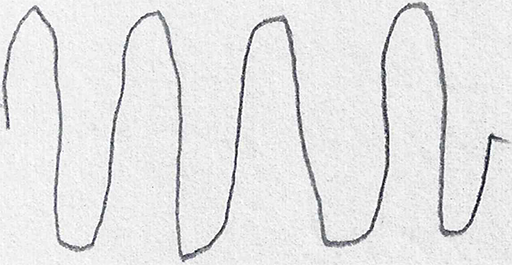
Figure 1 displays examples of spiral drawings made by individuals who are healthy. It shows a few samples of spiral patterns drawn by individuals without Parkinson's disease. These samples are used as a reference for comparison with the spiral drawings from individuals with Parkinson's disease. Figure 2 displays an examples of wave patterns drawn by healthy individuals. These patterns are used as a baseline for wave drawings created by individuals without Parkinson's disease.

A black spiral on a white background

Description automatically generated  A drawing of a spiral

Description automatically generated

Figure 1: Sample spiral drawings from healthy patients.



A drawing of a letter

Description automatically generated with medium confidence

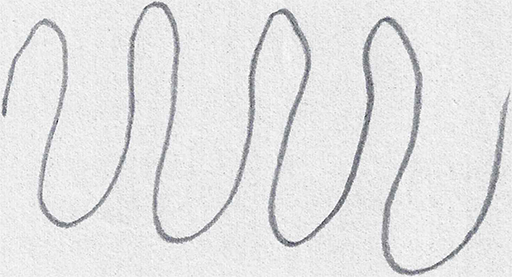


Figure 2: Sample wave drawings from healthy patients.

Figure 3 showcase examples of spiral drawings made by individuals diagnosed with Parkinson's disease. These drawings exhibit characteristic features associated with the motor control issues typically seen in Parkinson's patients. Figure 4 displays examples of wave patterns drawn by individuals with Parkinson's disease. The purpose is to demonstrate the differences between the wave patterns of healthy individuals and those with Parkinson's disease.

 A drawing of a spiral

Description automatically generated A black spiral on a white surface

Description automatically generated

Figure 3: Sample sprial drawings from parkinson patients.

A close-up of a drawing

Description automatically generated

A close-up of a line

Description automatically generated

A close up of a letter

Description automatically generated

Figure 4: Sample wave drawings from Parkinson patients.

A screenshot of a computer

Description automatically generated

Figure 5: User interface of proposed CNN-based Parkinson disease detection from images.

A screenshot of a computer

Description automatically generated

Figure 6: Illustration of UI with Deep CNN model accuracy after loading the pre-trained model.

Figure 5 illustrates the graphical user interface (GUI) of proposed CNN-based Parkinson disease detection application. It shows the layout of buttons, text fields, and other components that users interact with when using this application. Figure 6 shows the accuracy of the pre-trained CNN model after it has been loaded, which helps to understand the reliability of the model's predictions. Figure 7 depicts the architecture of deep CNN model layer by layer. It shows the convolutional layers, pooling layers, fully connected layers, and other components of model, along with the specifications.

A screenshot of a computer program

Description automatically generated

Figure 7: Layer-wise summary of proposed deep CNN model.

A picture containing graphical user interface

Description automatically generated

Figure 8: Sample test data used for evaluating the pre-trained proposed CNN model.

Figure 8 shows some examples of the test data used to evaluate the performance of proposed deep CNN model. It includes a few images that represent the input data the model is tested on. Figure 9 displays examples of images that have been input into trained CNN model, along with the model's predictions for each image. This is a visual representation of how well the model is performing in predicting whether an individual has Parkinson's disease based on their drawings.

A screenshot of a computer screen

Description automatically generated A picture containing diagram

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

Figure 9: Sample predicted images using proposed CNN model.

A graph showing a performance comparison

Description automatically generated with medium confidence

Figure 10: Performance analysis of obtained accuracy, and loss using proposed CNN model.

In Figure 10, x-axis represents training epoch and y-axis represents accuracy and loss values and with each increasing epoch, its accuracy getting increased, and loss is decreasing. At the final epoch, accuracy reached closer to 1 and loss reached closer to 0.