ASSIGNMENT 3

|  |  |
| --- | --- |
| **ASSIGNMENT DATE** | **12OCTOBER 2022** |
| **TEAM ID** | **PNT2022TMID17271** |
| **TITTLE** | **A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM** |
| **MAXIMUM MARKS** | **2 marks** |

**QUESTION :** WRITE A PYTHON CODE FOR A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

**CODE :**

class App(tk.Tk):

def \_init\_(self):

tk.Tk.\_init\_(self)

self.x = self.y = 0

# Creating elements

self.canvas = tk.Canvas(self, width=300, height=300, bg = "white", cursor="cross")

self.label = tk.Label(self, text="Thinking..", font=("Helvetica", 48))

self.classify\_btn = tk.Button(self, text = "Recognise", command = self.classify\_handwriting)

self.button\_clear = tk.Button(self, text = "Clear", command = self.clear\_all)

# Grid structure

self.canvas.grid(row=0, column=0, pady=2, sticky=W, )

self.label.grid(row=0, column=1,pady=2, padx=2)

self.classify\_btn.grid(row=1, column=1, pady=2, padx=2)

self.button\_clear.grid(row=1, column=0, pady=2)

#self.canvas.bind("<Motion>", self.start\_pos)

self.canvas.bind("<B1-Motion>", self.draw\_lines)

def clear\_all(self):

self.canvas.delete("all")

def classify\_handwriting(self):

HWND = self.canvas.winfo\_id() # get the handle of the canvas

rect = win32gui.GetWindowRect(HWND) # get the coordinate of the canvas

im = ImageGrab.grab(rect)

digit, acc = predict\_digit(im)

self.label.configure(text= str(digit)+', '+ str(int(acc\*100))+'%')

def draw\_lines(self, event):

self.x = event.x

self.y = event.y

r=8

self.canvas.create\_oval(self.x-r, self.y-r, self.x + r, self.y + r, fill='black')

app = App()

mainloop()