```
!pip install Pillow
```

```
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (9.4.0)
```

```
from keras.models import load_model
from tkinter import *
from PIL import ImageGrab, Image
import numpy as np


model = load_model('/content/Handwritten_Digit_Recognition_using_Neural_Network.ipynb')

def predict_digit(img):
    # Resize image to 28x28 pixels
    img = img.resize((28, 28))
    # Convert RGB to grayscale
    img = img.convert('L')
    img = np.array(img)
    # Reshaping for model normalization
    img = img.reshape(1, 28, 28, 1)
    img = img / 255.0
    # Predicting the class
    res = model.predict([img])[0]
    return np.argmax(res), max(res)

class App(Tk):
    def __init__(self):
        Tk.__init__(self)
        self.x = self.y = 0
        self.canvas = Canvas(self, width=200, height=200, bg="white", cursor="cross")
        self.label = Label(self, text="Draw a digit", font=("Helvetica", 48))
        self.classify_btn = Button(self, text="Classify", command=self.classify_handwriting)
        self.button_clear = Button(self, text="Clear", command=self.clear_all)

        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("<Button-1>", self.start_pos)
        self.canvas.bind("<B1-Motion>", self.draw_lines)

    def clear_all(self):
        self.canvas.delete("all")

    def classify_handwriting(self):
        x0 = self.canvas.winfo_rootx() + self.canvas.winfo_x()
        y0 = self.canvas.winfo_rooty() + self.canvas.winfo_y()
        x1 = x0 + self.canvas.winfo_width()
        y1 = y0 + self.canvas.winfo_height()
        rect = (x0, y0, x1, y1)
        im = ImageGrab.grab(bbox=rect)
        digit, acc = predict_digit(im)
        self.label.configure(text=str(digit) + ', ' + str(int(acc * 100)) + '%')

    def start_pos(self, event):
        self.x = event.x
        self.y = event.y

    def draw_lines(self, event):
        x = event.x
        y = event.y
        r = 8
        self.canvas.create_oval(x - r, y - r, x + r, y + r, fill='black')

app = App()
mainloop()
```