```
from keras.models import load_model
from tkinter import *
import tkinter as tk
import win32gui
from PIL import ImageGrab, Image
import numpy as np
model = load_model('mnist.h5')
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 to support our model input and normalizing
 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.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()
```

7.4 OUTPUT :

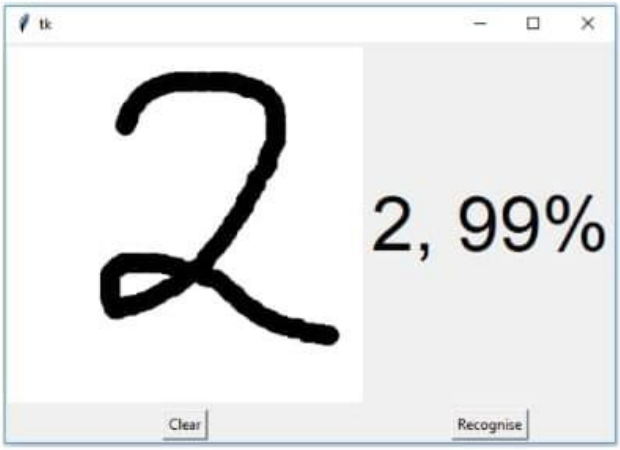

Figure 22 Output 'digit 2'

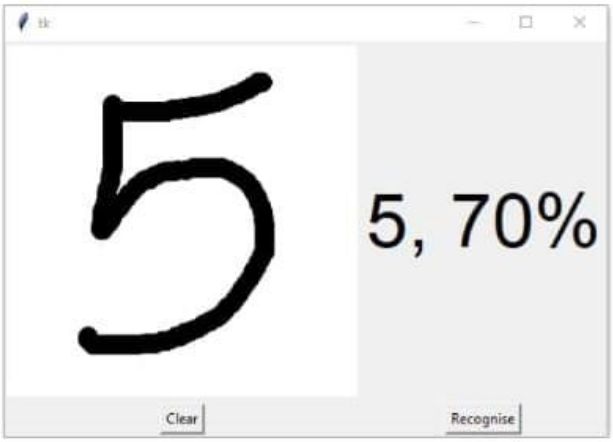

Figure 23 Output 'digit 5'