Fake Signature detection

Code:

```
import tkinter as tk
from tkinter.filedialog import askopenfilename
from tkinter import messagebox
from PIL import Image, ImageTk
import os
import cv2
from signature import match

THRESHOLD = 85

# ---------- Image Processing ----------

def get_filtered_image(image_path):
 try:
 img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
 if img is None:
 return None
filtered = cv2.threshold(img, 128, 255, cv2.THRESH_BINARY_INV +
cv2.THRESH_OTSU)
return filtered
except Exception as e:
print(f"Filtering error: {e}")
return None
def cv2_to_tk(img_array, width=200, height=100):
try:
img = cv2.resize(img_array, (width, height))
im = Image.fromarray(img)
return ImageTk.PhotoImage(image=im)
except Exception as e:
print(f"Conversion error: {e}")
return None
def update_preview(image_path, preview_label, filtered_label):
try:
img = Image.open(image_path).resize((200, 100))
tk_img = ImageTk.PhotoImage(img)
preview_label.img = tk_img
preview_label.config(image=tk_img)
filtered = get_filtered_image(image_path)
```

```
if filtered is not None:
tk_filtered = cv2_to_tk(filtered)
filtered_label.img = tk_filtered
filtered_label.config(image=tk_filtered)
except Exception as e:
print(f"Error updating previews: {e}")

# ---------- File Handling ----------

def browsefunc(ent, preview, filtered):
filename = askopenfilename(filetypes=[("Image files", ".jpeg .png .jpg")])
if filename:
ent.delete(0, tk.END)
ent.insert(0, filename)
update_preview(filename, preview, filtered)
def capture_image_from_cam_into_temp(sign=1):
cam = cv2.VideoCapture(0, cv2.CAP_DSHOW)
cv2.namedWindow("Camera Preview")
while True:
ret, frame = cam.read()
if not ret:
print("failed to grab frame")
break
cv2.imshow("Camera Preview", frame)
k = cv2.waitKey(1)
if k % 256 == 27:
break
elif k % 256 == 32:
if not os.path.isdir('temp'):
os.mkdir('temp', mode=0o777)
img_name = f"./temp/test_img{sign}.png"
cv2.imwrite(img_name, frame)
print(f"{img_name} written!")
cam.release()
cv2.destroyAllWindows()
return True
```

```
def captureImage(ent, preview, filtered, sign=1):
filename = os.path.join(os.getcwd(), f'temp/test_img{sign}.png')
res = messagebox.askquestion('Click Picture', 'Press Space Bar to click picture and ESC to
exit')
if res == 'yes':
capture_image_from_cam_into_temp(sign)
ent.delete(0, tk.END)
ent.insert(tk.END, filename)
update_preview(filename, preview, filtered)
return True

# ----------------- Similarity Check ------------------

def checkSimilarity(window, path1, path2):
result = match(path1=path1, path2=path2)
if result <= THRESHOLD:
messagebox.showerror("Failure", f"Signatures are {result}% similar — No Match")
else:
messagebox.showinfo("Success", f"Signatures are {result}% similar — Match Found")
return True

# ----------------- GUI Setup ------------------

root = tk.Tk()
root.title("Fake Signature Detection")
root.geometry("600x700")
tk.Label(root, text="Compare Two Signatures").pack(pady=10)

# Signature 1 Section
tk.Label(root, text="Signature 1").pack()
img1_entry = tk.Entry(root, width=50)
img1_entry.pack()
img1_preview = tk.Label(root)
img1_preview.pack()
img1_filtered = tk.Label(root)
img1_filtered.pack()
```

```
tk.Button(root, text="Browse", command=lambda: browsefunc(img1_entry, img1_preview,
img1_filtered)).pack()
tk.Button(root, text="Capture", command=lambda: captureImage(img1_entry, img1_preview,
img1_filtered, sign=1)).pack(pady=5)

# Signature 2 Section
tk.Label(root, text="Signature 2").pack()
img2_entry = tk.Entry(root, width=50)
img2_entry.pack()
img2_preview = tk.Label(root)
img2_preview.pack()
img2_filtered = tk.Label(root)
img2_filtered.pack()
tk.Button(root, text="Browse", command=lambda: browsefunc(img2_entry, img2_preview,
img2_filtered)).pack()
tk.Button(root, text="Capture", command=lambda: captureImage(img2_entry, img2_preview,
img2_filtered, sign=2)).pack(pady=5)

# Compare Button
tk.Button(root, text="Compare Signatures", command=lambda: checkSimilarity(
root, img1_entry.get(), img2_entry.get())).pack(pady=20)
root.mainloop()
```

Screenshots:

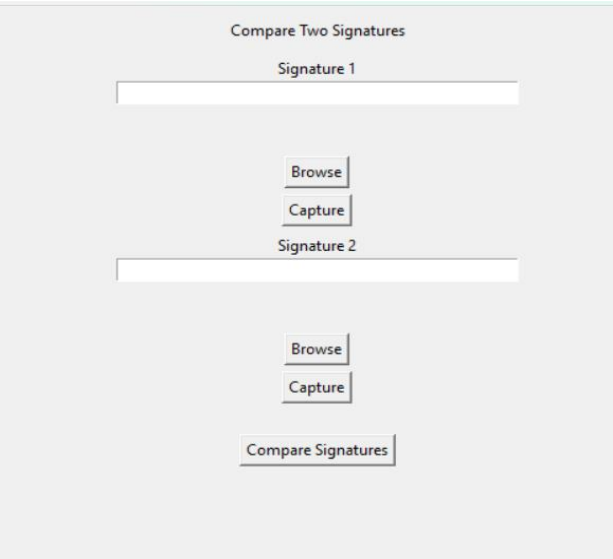

Figure A : Home Page

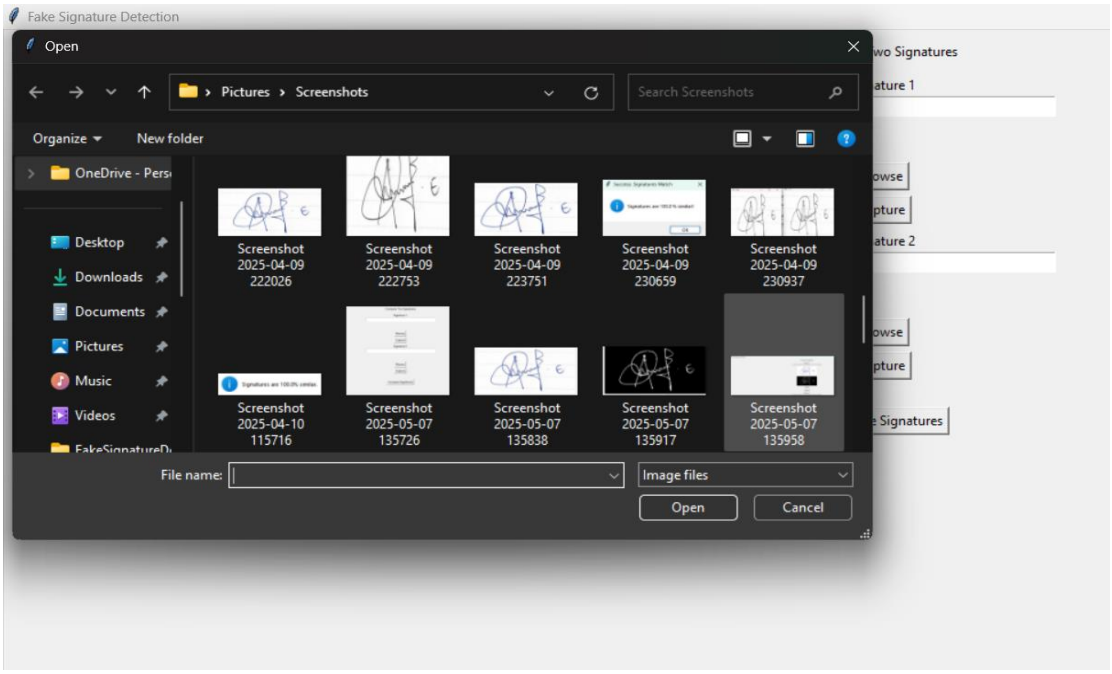

Figure B : Browse for the Saved Pictures

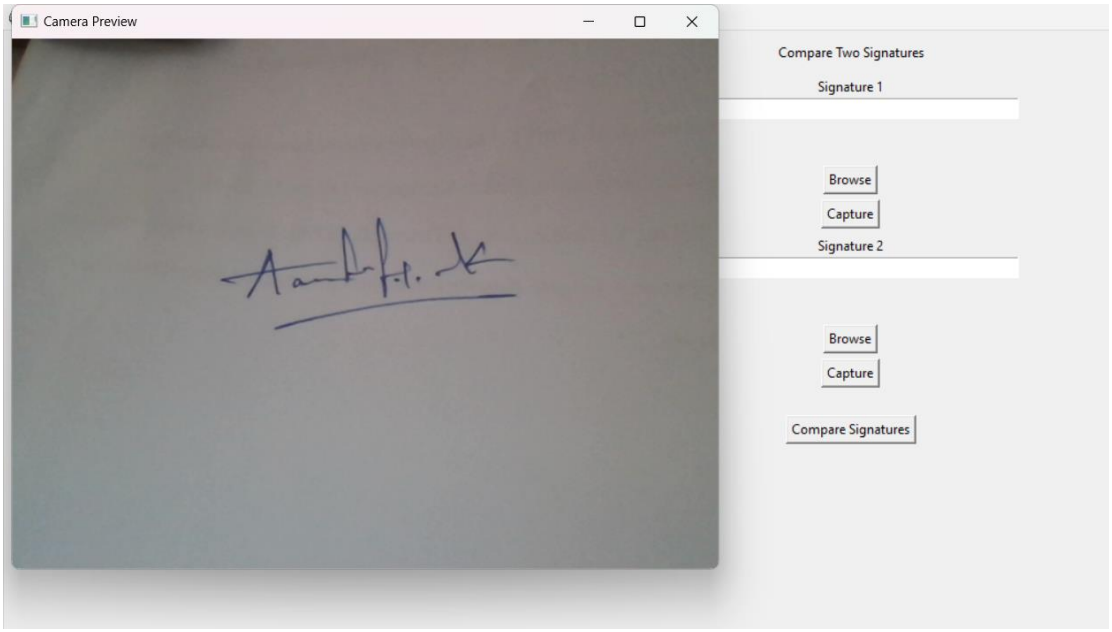

Figure C : Capturing Images using Webcam

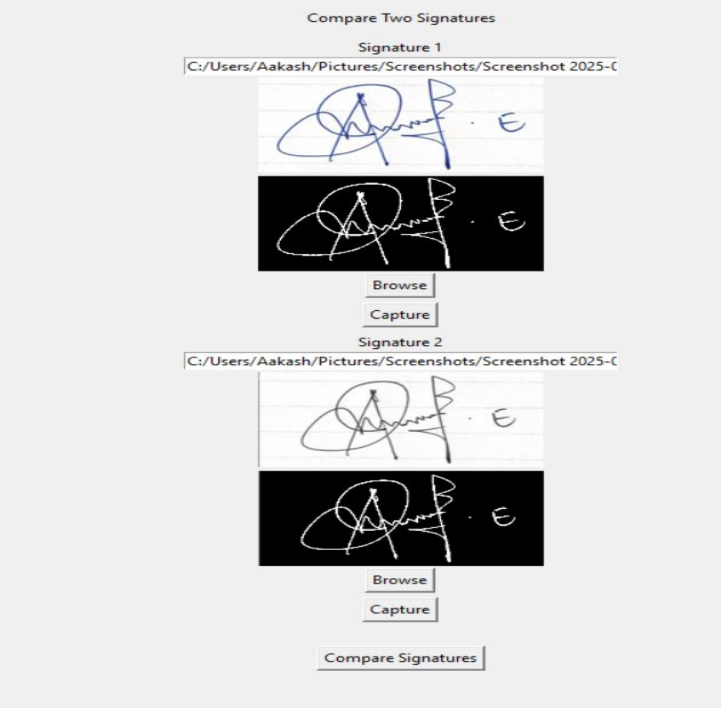

Figure D : Images have been Uploaded for the Comparison

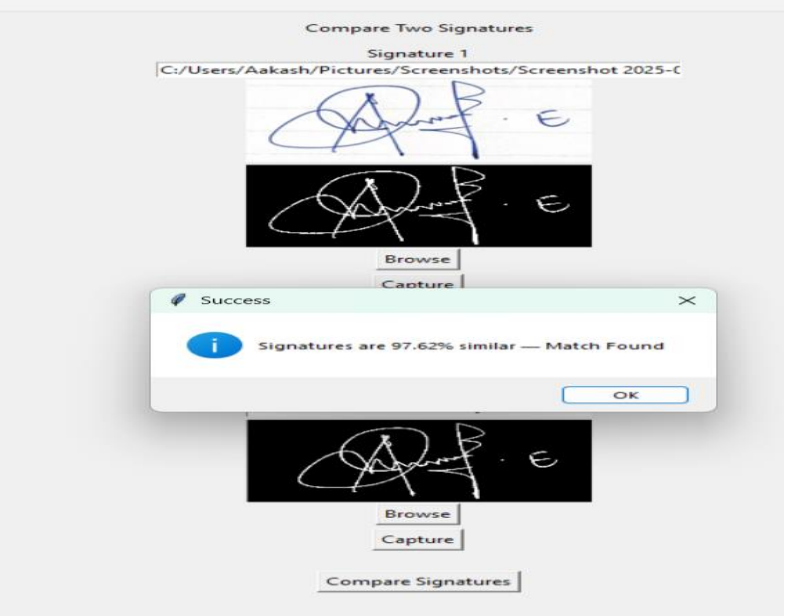

Figure E : Predicted Successfully

Figure F : Prediction Failed