THE FACE WEB

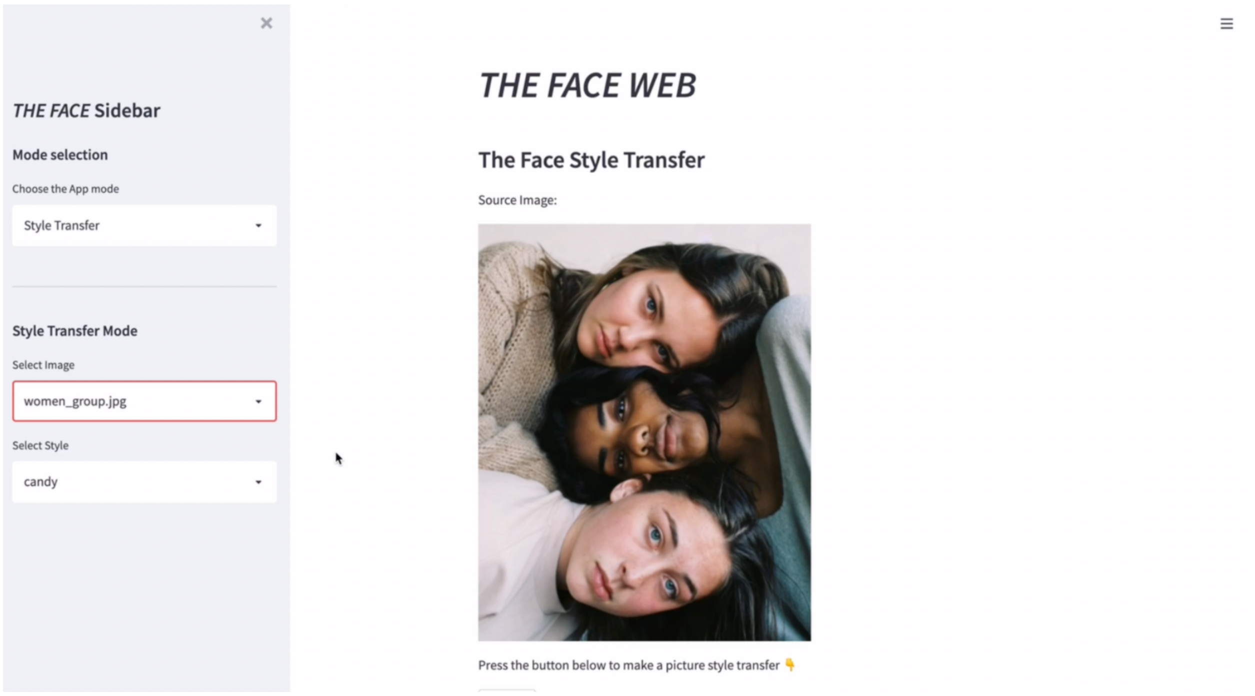
Bonus :

> **I used opencv, python, PyTorch, streamlit, numpy, etc. to create a local website called “THE FACE WEB" for users to use**

github link：https://github.com/orionmel/Coding2_Final_Artwork_Advanced_Framework

Video link ：https://vimeo.com/689315352

INTRODUCTION


THE FACE Sidebar

Mode selection

Choose the App mode: Style Transfer

Style Transfer Mode

Select Image: women_group.jpg

Select Style: candy

THE FACE WEB

The Face Style Transfer

Source Image:

Press the button below to make a picture style transfer 👇

Welcome to the website of THE FACE WEB, which was created using tools such as opencv, python, PyTorch, streamlit, numpy, and more.

On this Website, there are three different modes.

In Image Recognition mode: This is a picture recognition system.You can upload your own images to see the number of faces in the uploaded images

In Video Recognition Mode: This is a video recognition system where you can upload videos, and the website will recognize the number of faces of people in the video

In Style Transfer mode: You can select different pictures about "faces" provided by the system, and then select different styles to convert the pictures

---

The description of my code :

⚠️⚠️⚠️ Tips: When I run streamlit in the local, the message keeps prompting "Please wait", and then I found that I need to run the following code

```
streamlit run face_mesh_app.py --server.enableWebsocketCompression=false
```

1.Basic setting :

```
import streamlit as st
import mediapipe as mp
import cv2
import numpy as np
import tempfile
import time
from PIL import Image
import style
import saved_models

mp_drawing = mp.solutions.drawing_utils
mp_face_mesh = mp.solutions.face_mesh

DEMO_IMAGE = 'women.jpg'
DEMO_VIDEO = 'video.mp4'
```

2.streamlit setting

```
st.title(' ***THE FACE WEB*** ')

st.markdown (
    """
    <style>
    [data-testid = "stSidebar"][aria-expanded = "true"] > div:first-child{
        width:350px
    }
    [data-testid = "stSidebar"][aria-expanded = "false"] > div:first-child{
        width:350px
        margin-left : -350px
    }
    </style>
    """,
    unsafe_allow_html=True,
)

#sidebar title
st.sidebar.title("***THE FACE*** Sidebar")
st.sidebar.subheader('Mode selection')
```

```
#select bar
app_mode = st.sidebar.selectbox('Choose the App mode',
                                ['Introduction','Image Recognition','Video Recognition','Style Transfer']
                                )
```

3.opencv maintain the aspect ratio of the original image

```
@st.cache()
def image_resize(image,width=None,height=None,inter=cv2.INTER_AREA):
    dim = None
    (h,w) = image.shape[:2]

    if width is None and height is None:
        return image
    if width is None:
        r = width/float(w)
        dim = (int(w*r),height)
    else:
        r = width/float(w)
        dim = (width,int(h*r))

    resized = cv2.resize(image,dim,interpolation=inter)

    return resized
```

4. [Introduction] part

```
if app_mode == 'Introduction':
    st.subheader('Introductory video')
    st.markdown('The video below describes how to use this website 👇 ')


    st.markdown (
        """
        <style>
        [data-testid = "stSidebar"][aria-expanded = "true"] > div:first-child{
            width:350px
        }
        [data-testid = "stSidebar"][aria-expanded = "false"] > div:first-child{
            width:350px
            margin-left : -350px
        }
        </style>
        """,
        unsafe_allow_html=True,
    )

    #insert video
    st.video('https://youtu.be/2u-2UKVB-ws')

    st.subheader('Introduction:')
    st.markdown("""

                Welcome to the website of THE FACE WEB. 😊 😊 😊 🎉 🎉 🎉 \n
                This Website is created using tools such as opencv, python, PyTorch, streamlit, numpy, and more. \n
                On this Website, there are three different modes.✌ ✌ ✌ \n
                📷 In Image Recognition mode: This is a picture recognition system.You can upload your own images to see the number of faces
                👀 In Video Recognition Mode: This is a video recognition system where you can upload videos, and the website will recogniz
                🌹 In Style Transfer mode: You can select different pictures about "faces" provided by the system, and then select differen

                """)
```

4. [Image Recognition] part

(1) The line of recognition on the face

```
elif app_mode == 'Image Recognition':
  drawing_spec = mp_drawing.DrawingSpec(thickness=2,circle_radius=1)
```

(2) Streamlit setting

```
#sidebar
    st.sidebar.markdown('-------')

    st.markdown (
        """
        <style>
        [data-testid = "stSidebar"][aria-expanded = "true"] > div:first-child{
            width:350px
        }
        [data-testid = "stSidebar"][aria-expanded = "false"] > div:first-child{
            width:350px
            margin-left : -350px
        }
        </style>
        """,
        unsafe_allow_html=True,
    )

#detect the number of face
    st.markdown("***Detected faces***")
    kpil_text = st.markdown("0")

    # sidebar
    st.sidebar.subheader('Face Detection Value Adjustments')
    max_faces = st.sidebar.number_input('The Number Of Detected Faces',value= 2,min_value=1)

    detection_confidence = st.sidebar.slider('Min Detection Confidence',min_value=0.0,max_value=1.0,value=0.5)
    st.sidebar.markdown('----')

    face_count = 0
```

(3) Browse local image

```
img_file_buffer = st.sidebar.file_uploader("Upload an Image",type=["jpg","jpeg","png"])
    if img_file_buffer is not None:
        image = np.array(Image.open(img_file_buffer))
    else:
        demo_image = DEMO_IMAGE
        image = np.array(Image.open(demo_image))

    st.sidebar.text('Original Image')
    st.sidebar.image(image)
```

(4) Dashboard

```
with mp_face_mesh.FaceMesh(
    static_image_mode= True,
    max_num_faces=max_faces,
    min_detection_confidence= detection_confidence) as face_mesh:
        results = face_mesh.process(image)
        out_image = image.copy()

        #Face landmark drawing
        for face_landmarks in results.multi_face_landmarks:
            face_count +=1

            mp_drawing.draw_landmarks(
            image = out_image,
            landmark_list= face_landmarks,
            connections = mp_face_mesh.FACEMESH_CONTOURS,
            landmark_drawing_spec = drawing_spec)

           # write the number of face
            kpil_text.write(f"<h1 style='text-align:center;color:red;'>{face_count}</h1>",unsafe_allow_html=True)

        #show image
        st.subheader('Output Image')
        st.image(out_image,use_column_width=True)
```

5. [Video Recognition] part

(1) Camera setting

```
elif app_mode == 'Video Recognition':

  st.set_option('deprecation.showfileUploaderEncoding', False)

    st.sidebar.subheader('Webcam Mode')
    use_webcam = st.sidebar.button('Use Webcam')
    st.sidebar.markdown('---')
```

(2) Record video

```
st.sidebar.subheader('Record Video Mode')
    record = st.sidebar.checkbox("Record Video")
    if record:
        st.checkbox("Recording",value=True)
    st.sidebar.markdown('---')
```

(3) Streamlit setting

```
st.markdown (
        """
        <style>
        [data-testid = "stSidebar"][aria-expanded = "true"] > div:first-child{
            width:350px
        }
        [data-testid = "stSidebar"][aria-expanded = "false"] > div:first-child{
            width:350px
            margin-left : -350px
        }
        </style>
        """,
        unsafe_allow_html=True,
    )


    st.sidebar.subheader('Face Detection Value Adjustments')
    max_faces = st.sidebar.number_input('The Number Of Detected Faces',value= 5, min_value=1)

    detection_confidence = st.sidebar.slider('Min Detection Confidence',min_value=0.0,max_value=1.0,value=0.5)
    tracking_confidence = st.sidebar.slider('Min Tracking Confidence', min_value=0.0, max_value=1.0, value=0.5)
    st.sidebar.markdown('----')

    st.markdown("## Output")
```

(4) Upload video

```
    st.sidebar.subheader('Upload Mode')
    stframe = st.empty()
    video_file_buffer = st.sidebar.file_uploader("Upload a Video",type = ['mp4','mov','avi','asf','m4v'])
    tffile = tempfile.NamedTemporaryFile(delete=False)

    if not video_file_buffer:
        if use_webcam:
            vid = cv2.VideoCapture(0)
        else:
            vid = cv2.VideoCapture(DEMO_VIDEO)
            tffile.name = DEMO_VIDEO
    else:
        tffile.write(video_file_buffer.read())
        vid = cv2.VideoCapture(tffile.name)

    width = int(vid.get(cv2.CAP_PROP_FRAME_WIDTH))
    height = int(vid.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fps_input = int(vid.get(cv2.CAP_PROP_FPS))

    #Recording
    codec = cv2.VideoWriter_fourcc('N','J','P','6')
    out = cv2.VideoWriter('output1.mp4',codec,fps_input,(width,height))
    st.sidebar.text('Input Video')
    st.sidebar.video(tffile.name)
```

(5) Facemesh predictor

```
    fps = 0
    i = 0

    #the line
    drawing_spec = mp_drawing.DrawingSpec(thickness=2, circle_radius=1)

    #show number
    kpi1,kpi2,kpi3 = st.columns(3)

    with kpi1:
        st.markdown("**Frame Rate**")
        kpi1_text = st.markdown("0")

    with kpi2:
        st.markdown("**Detected Faces**")
        kpi2_text = st.markdown("0")

    with kpi3:
        st.markdown("**Image Width**")
        kpi3_text = st.markdown("0")

    st.markdown("<hr/>",unsafe_allow_html=True)

    # facemesh predictor
    with mp_face_mesh.FaceMesh(
    max_num_faces=max_faces,
    min_detection_confidence=detection_confidence,
    min_tracking_confidence=tracking_confidence
    ) as face_mesh:
        prevTime = 0

        while vid.isOpened():
            i += 1
            ret,frame = vid.read()
            if not ret:
                continue


            results = face_mesh.process(frame)
            frame.flags.writeable = True


            face_count = 0
            if results.multi_face_landmarks:
                for face_landmarks in results.multi_face_landmarks:
                    face_count += 1

                    mp_drawing.draw_landmarks(
                        image= frame,
                        landmark_list=face_landmarks,
                        connections=mp_face_mesh.FACEMESH_CONTOURS,
                        landmark_drawing_spec=drawing_spec,
                        connection_drawing_spec=drawing_spec)

            currTime = time.time()
            fps = 1/(currTime - prevTime)
            prevTime = currTime

            if record:
                out.write(frame)
```

(6) Dashboard

```
kpi1_text.write(f"<h1 style='text-align:center;color:red;'>{int(fps)}</h1>", unsafe_allow_html=True)
kpi2_text.write(f"<h1 style='text-align:center;color:red;'>{face_count}</h1>", unsafe_allow_html=True)
kpi3_text.write(f"<h1 style='text-align:center;color:red;'>{width}</h1>", unsafe_allow_html=True)

frame = cv2.resize(frame,(0,0),fx = 0.8,fy=0.8)
frame = image_resize(image = frame,width = 640)
stframe.image(frame,channels = "BGR",use_column_width = True)
```

6. [Style Transfer] part

(1) Streamlit setting

```
elif app_mode == 'Style Transfer':
    st.subheader('The Face Style Transfer')
    st.sidebar.markdown('___')
    st.sidebar.subheader('Style Transfer Mode')
    img = st.sidebar.selectbox(
        'Select Image',
        ('women.jpg','women_group.jpg','man.jpg','man_group.jpg','team.jpg')
    )

    style_name = st.sidebar.selectbox(
        'Select Style',
        ('candy', 'mosaic', 'rain_princess', 'udnie')
    )
```

(2) Image path

```
    model = "saved_models/" + style_name + ".pth"
    input_image = "images/content-images/" + img
    output_image = "images/output-images/" + style_name + "-" + img

    st.write('Source Image:')
    image = Image.open(input_image)
    st.image(image, width=400)  # image: numpy array

    st.markdown('Press the button below to make a picture style transfer 👇 ')
```

(3) Click button

```
    clicked = st.button('Stylize')

    if clicked:
        model = style.load_model(model)
        style.stylize(model, input_image, output_image)

        st.write('### Output image:')
        image = Image.open(output_image)
        st.image(image, width=400)
```

Reference

https://numpy.org/

https://python.plainenglish.io/face-mesh-detection-with-python-and-opencv-complete-project-359d81d6a712

https://streamlit.io/

https://www.python.org/

https://pytorch.org/

https://mediapipe.dev/

https://docs.python.org/3/library/tempfile.html

https://www.youtube.com/watch?v=-IM3531b1XU&t=565s

https://docs.streamlit.io/knowledge-base/deploy/remote-start