Fastapi串接前端

Rich
4 min read · 3 days ago

#fastapi # react

先在後端資料夾的main.py設置路由(這裡後端用內插法示範)

前端會傳送2張照片(一張舊照片、一張新照片)到後端，後端處理完後會傳送一張內插過程圖到前端

main.py的路由(這裡只有路由的設定)

```
from interpolated import getImgInterpolated

@app.post("/interpolated/")
async def interpolated(image1: UploadFile = File(...), image2: UploadFile = Fil
    try:
        image_data1 = await image1.read()
        image_data2 = await image2.read()
        oldImage = Image.open(io.BytesIO(image_data1)).convert("RGB")
        newImage = Image.open(io.BytesIO(image_data2)).convert("RGB")
        response = getImgInterpolated(oldImage, newImage)
        return StreamingResponse(response, media_type="image/jpeg")  # 直接回傳圖

    except Exception as e:
        return JSONResponse(content={"error": str(e)}, status_code=400)
```

interpolated.py

```
from PIL import Image, ImageOps
import torch
import cv2
import numpy as np
from diffusers import StableDiffusionPipeline
from io import BytesIO

def getImgInterpolated(oldImage, newImage):
    # 內插法過程

    # 轉換資料再return
    img_bytes = BytesIO()  # 建立一個 BytesIO 物件
    merged_image.save(img_bytes, format="JPEG")  # 把圖片存進 BytesIO
    img_bytes.seek(0)  # 將指標移到開頭
    return img_bytes  # 回傳 BytesIO 物件
    # 顯示合併後的圖片
    # merged_image.show()
```

前端發送請求

```
    try {
      const response = await fetch("http://localhost:8000/interpolated/", {
        method: "POST",
        body: formData,
      });

      if (!response.ok) {
        throw new Error("圖片處理失敗");
      }

      const blob = await response.blob(); // 解析後端返回的 JSON 數據
      const imageUrl = URL.createObjectURL(blob); // 創建圖片 URL
      setResult({
        image: imageUrl, // 使用後端返回的圖片 URL
        text: "完成處理",      // 使用後端返回的文字
      });
    } catch (error) {
      console.error("上傳錯誤", error);
      setError("圖片處理失敗，請重試"); // 設置錯誤訊息
    } finally {
      setUploading([false, false]); // 重置上傳狀態
    }
  };
```

要注意資料型態的傳送否則很容易出問題，也可以先透過fastapi的docs檢查自己的後端路由

結果