

CS101 Advanced Engineering Mathematics (I)

工程數學(一)

【Guidelines】

1. 每組限 2~3 人，組員須固定，可跨班編組，但本學期不得任意變更。
2. 組員應合作共同解題，但嚴禁跨組合作。
3. 程式設計必須使用 Python 程式語言 (原則上，請下載與安裝 Anaconda)。
4. 可參考課本、參考書籍或網站資料等進行解題，但嚴禁抄襲他組程式，組員有責任保護程式不被他組抄襲。若發現抄襲屬實，兩組均以零分計。
5. Python 程式須附上組員姓名及學號，以利評分。例如：

```
# 工程數學
# 學號: 11227XXX / 11227XXX
# 姓名: 陳○○ / 林○○
# 中原大學資訊工程系
```

【General Instructions】

To get a good grading on the homework assignments, you are advised to do the following:

- Start early, don't wait until last minutes! (請盡早開始，不然你肯定做不完)
- Do not copy other classmate's works! (請遵守學術倫理，嚴禁抄襲)
- Prepare your written reports in good quality (使用 Template 檔，編寫專業報告)
- Meet the deadline! Late homework will **NOT** be collected. (按時繳交，逾時不候)

【課程助教】

計算機視覺研究室 (電學大樓 603 室)

電話：(03) 265-4726

指導教授：張元翔

Homework Assignment 1

State-of-the-Art Face-Swap Technique in Digital Video

Deadline: 11 / 7 / 2025 (星期五)

【Objectives】

Develop a Python program that performs face swap technique on a video using the InsightFace library. The user should be able to specify the source face image, input video, output video, and implement face swap technique with pipeline processes.

【History of Face Swap Techniques】

| Era | Method | Key Features | Tools / Models |
|-----------|-----------------------------|--|--------------------------------------|
| Pre-2010 | Manual Editing | Manual alignment and blending | Photoshop |
| 2010–2014 | Landmark + Seamless Cloning | Affine warping with triangle mesh and Poisson blending | OpenCV, Dlib |
| 2014–2018 | DeepFakes (Autoencoders) | Shared encoder with separate decoders for two faces | DeepFaceLab, FaceSwap |
| 2018–2020 | FaceShifter | GAN-based swapping with occlusion awareness | FaceShifter |
| 2021–2022 | SimSwap | Identity embedding (ArcFace) + Generator | SimSwap |
| 2022–Now | InsightFace / InSwapper | Fast, high-quality swaps with ONNX models | InsightFace, inswapper_128.onnx |
| 2023–Now | Diffusion-based (FaceChain) | Style-consistent identity swaps over frames | Stable Diffusion, ControlNet, FaceID |

本次作業將採用目前最先進的換臉技術 InsightFace

【Learning Goals】

- Use Argparse to create a flexible command-line Python program.
- Apply InsightFace for face detection and face swap.
- Process video with OpenCV and generate output.
- Understand and implement the AI multimedia pipeline using Python programming.

【Software Packages】

請同學安裝下列開發環境與程式庫：

- **Python (Anaconda)** 開發環境
- **OpenCV**

```
pip install opencv-python
```

OpenCV 為開源電腦視覺程式庫，可以用來處理數位影像 (Digital Images) 或數位視訊 (Digital Video)，但無法處理音訊 (Audio)。

- **InsightFace**

```
pip install insightface --no-deps
```

InsightFace 的 dependencies 中的 albumentations 會偷安裝 opencv-headless (破壞 OpenCV 環境)

建議在 Python 目錄 Lib\site-packages\insightface\app__init__.py 註解：

```
# from .mask_renderer import *
```

- **Onnxruntime**

```
pip install onnx
pip install onnxruntime-gpu (或 onnxruntime)
```

若你的電腦配備 GPU 顯卡，則運算速度較快，但須先設置好環境。

- **超解析度 (Super-Resolution) 生成技術**

(1) Real-ESRGAN (<https://github.com/xinntao/Real-ESRGAN>)

(2) GFPGAN (<https://github.com/TencentARC/GFPGAN>)

- 使用 **yt-dlp** 自 YouTube 下載影片

```
pip install yt-dlp
```

- **FFmpeg 影音編輯應用軟體** (<https://ffmpeg.org>)

【Python 程式設計】

Design a Python program to perform face swapping pipeline
(Pipeline 是指從輸入到輸出「一條龍」完成)：

```
python face_swap.py --source source.jpg --input input.mp4 --output output.mp4
```

你可以自定義其他參數 (Arguments)，例如：--help、--reference 等。

【Requirements】

請同學以組為單位，完成下列事項：

- 使用 Argparse 處理參數。
- 載入臉部偵測與換臉模型 (務必一次性載入)。
- 以逐幀 (frame-by-frame) 方式處理數位視訊。
(原則上，建議先使用單張影像進程式設計與測試，影像範例如下圖)。
- 偵測每一幀中的臉部區域。
- 使用來源人臉影像進行換臉 (若臉部區域略超出畫面範圍，仍然須進行換臉)。
- 對置換後的臉部區域使用 Real-ESRGAN 或 GFPGAN 技術增強細節，並貼回目標影像。建議在換臉過程，顯示影像結果。
- 將換臉後的影片儲存至 --output 指定的路徑。
- 若有音訊，則使用 FFmpeg 重新合成視訊與音訊 (使用 Subprocess)，並輸出視訊檔案結果。



圖 1 InsightFace 換臉結果影像範例



圖 2 InsightFace 換臉結果影像範例 (使用 GFP-GAN 增強臉部細節)



圖 3 InsightFace 換臉結果範例 (使用數位視訊)

【Improvements】

請同學以組為單位，進一步增強下列程式功能：

- 首先，於影片中擷取主角的正面臉部影像，作為**參考臉部影像** (Reference Face Image)。
- 使用 InsightFace 的人臉 **embedding vector** (嵌入特徵向量) 比對來辨識「主角」。
- 只有當影片中偵測到的臉與 --reference 主角臉非常相似 (例如： $\text{cosine similarity} \geq 0.3$) 時，才進行換臉，以確保不會誤套其他人臉。
- 貼回臉部影像區域時，可嘗試採用 Alpha Blending 技術進行拼貼，使得臉部區域邊緣區域較為貼合。
- 使用 FFmpeg 或威力導演 (不要使用試用版，會有浮水印)，進行最後的調整。

【註】

- 請採用組員的臉作為來源臉部影像，禁止使用翔哥的臉。
- 請同學發揮創意，我可能找機會在課堂上分享同學的作品。

【作業繳交內容】

- 須根據 Template 檔編寫書面報告一份 (請繳交 Word 或 PDF 檔)。
- Python 程式一支 (Face Swap Pipeline)，請盡量提供程式註解。
- 每組須至少繳交 3 支 Demo 影片，影片內容不可重複，請根據學號(該組學號最前者)命名，例如：
11227XXX_1.mp4
11227XXX_2.mp4
11227XXX_3.mp4

【註】原則上，請不要使用 Jupyter Notebook。

【評分原則】

- 書面報告 (50%)
- Demo 影片創意與完成度 (50%)

【註】以組為單位，於繳交期限內上傳至 i-learning。若有影片檔案過大無法上傳至作業區同學，請直接攜帶 USB 繳交至計算機視覺研究室 (電學 603)。

【Python 程式範例】

使用 Python 程式與 OpenCV 播放數位視訊 (影片)：

```
import numpy as np
import cv2

cap = cv2.VideoCapture("video.mp4")

# 顯示視訊相關資訊
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
fps = cap.get(cv2.CAP_PROP_FPS)

print(f"Resolution: {width} x {height}") # Frame 解析度
print(f"FPS: {fps:.2f}")                # frame-per-second (fps)

# 播放影片
while True:
    ret, frame = cap.read()
    if not ret:
        break
    cv2.imshow("Frame", frame)
    if cv2.waitKey(int(1000 // fps)) & 0xff == 27: # 使用 ESC 提早結束
        break

# 關閉視訊檔案與釋放資源
cap.release()
cv2.destroyAllWindows()
```

InsightFace 的數學原理

InsightFace 是以 PyTorch 為基礎的人臉分析工具包，整合**臉部偵測** (Face Detection)、**關鍵點定位** (Facial Landmark Localization)、**臉部對齊** (Face Alignment)、**臉部辨識** (Face Recognition)、**換臉** (Face Swapping) 等功能。InsightFace 的數學原理主要來自於**臉部辨識模型** (Face Recognition Model) 的**特徵嵌入學習** (Face Embedding Learning)，以及 ONNX-based 模型進行臉部影像合成。以下是主要的數學原理：

I. 臉部偵測 (基於 RetinaFace)

InsightFace 使用 RetinaFace 作為人臉偵測主幹網路，其數學基礎如下：

- **Anchor-based 检测器**
- **損失函數組合：**

$$L = L_{cls} + \lambda_1 L_{bbox} + \lambda_2 L_{landmark}$$

其中：

L_{cls} ：分類 (是否為人臉)

L_{bbox} ：邊界框

$L_{landmark}$ ：臉部關鍵點位置

II. 臉部對齊 (Face Alignment)

根據偵測到的五個關鍵點，使用仿射轉換對齊到標準座標：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} + b$$

III. 換臉 (Face Swap)

InsightFace 的換臉技術使用 **ONNX 模型** (如: inswapper_128.onnx)，是經過訓練的神經網路模型，其目標是將**來源臉部特徵**映射到**目標臉部特徵**。數學上可以視為一個條件生成模型：

$$\hat{I}_{target} = G(I_{target}, f_{source})$$

其中：

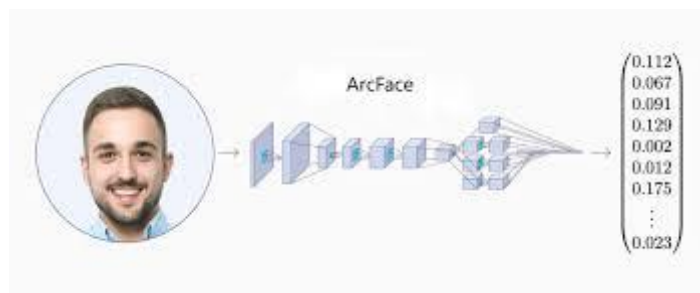
I_{target} ：目標影像

f_{source} ：來源臉部的嵌入特徵

G ：神經網路生成模型

IV. 相似度評估

首先，透過 ArcFace 模型計算的**臉部嵌入向量** (Face Embedding Vectors)。ArcFace 模型是一種基於深度學習的臉部辨識模型，如下圖。



臉部辨識與比對是透過下列「餘弦相似度」公式計算：

$$\cos_similarity(x_1, x_2) = \frac{x_1 \cdot x_2}{\|x_1\| \cdot \|x_2\|}$$

其中：

x_1, x_2 ：透過 ArcFace 模型計算的**臉部嵌入向量** (Face Embedding Vectors)

「相似度」的數值範圍介於 $-1 \sim 1$ 之間。當相似度高於某閾值 (例如： ≥ 0.3) 即可辨識為同一人。

V. 無縫複製

無縫複製 (Seamless Clone) 是 OpenCV 中基於 Poisson Image Editing 技術的一種影像融合函式，能把來源影像的一部分無縫貼入目標影像中，使融合處在視覺上看起來自然連貫，不會出現明顯的邊界、亮度不一致或色調突兀等問題。

數學原理為求解一個區域 Ω 中的影像 $f(x, y)$ ，其梯度場近似來源影像 $g(x, y)$ ，目的是使得下列的**目標函數** (Objective Function) 最小化：

$$\min_f \iint_{\Omega} \|\nabla f - \nabla g\|^2 dx dy$$

其中， ∇ 稱為**梯度運算子** (Gradient Operators)，定義為：

$$\nabla f(x, y) = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$