

即時影像分割+風格轉換

第七組

組員:N26120870林耕澤 N26120820羅文璟 N26120799吳炯霖 N26120309謝亞城

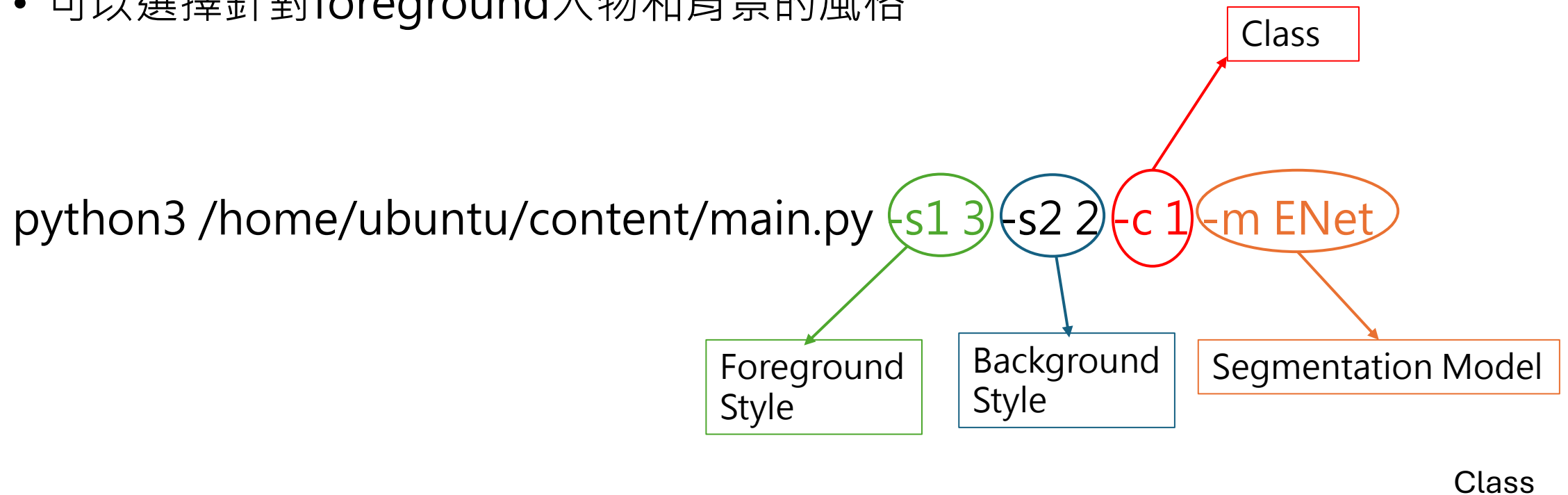
目錄

- 系統介紹
- 語意分割模型
- 風格轉換模型
- CB Styling
- 部署

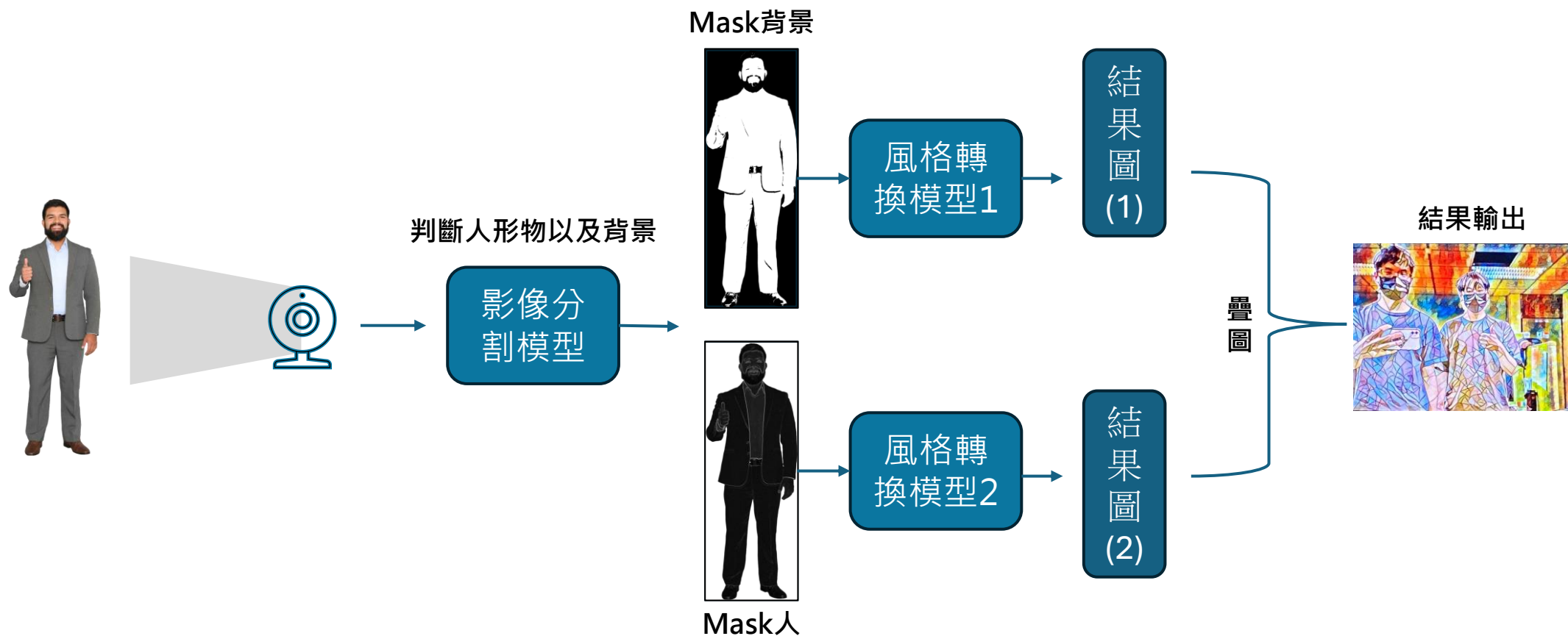
系統介紹

系統功能介紹

- 透過Webcam接收串流，實現即時多區塊風格轉換
- 可以選擇針對foreground人物和背景的风格

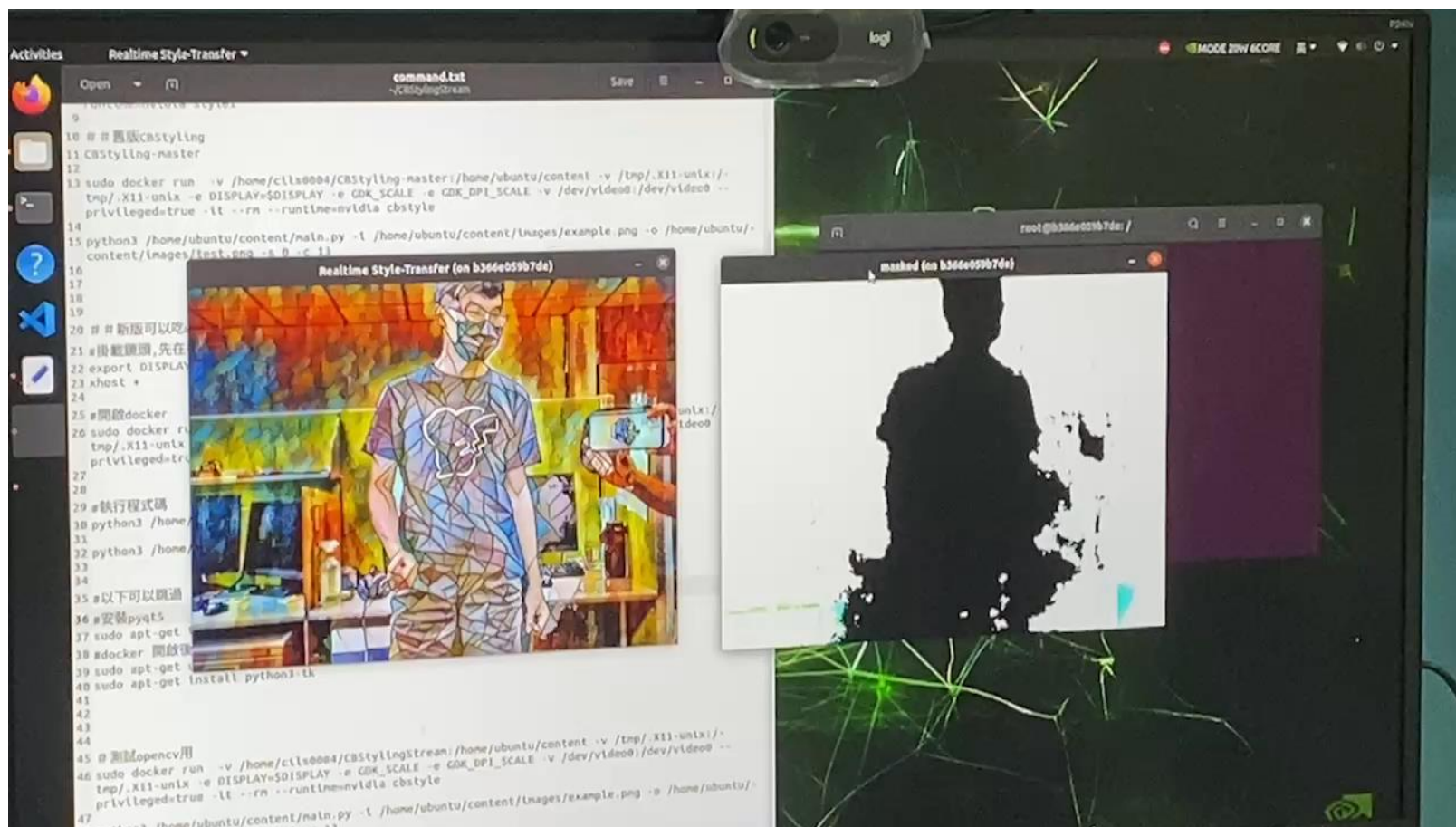


系統流程



忘記留原圖，用讚讚人代替

系統展示



<https://youtu.be/QXZZo53d1IE>

語意分割模型

資料集







- Human Segmentation Dataset

- 2667張原始照片以及Mask

- **Content**

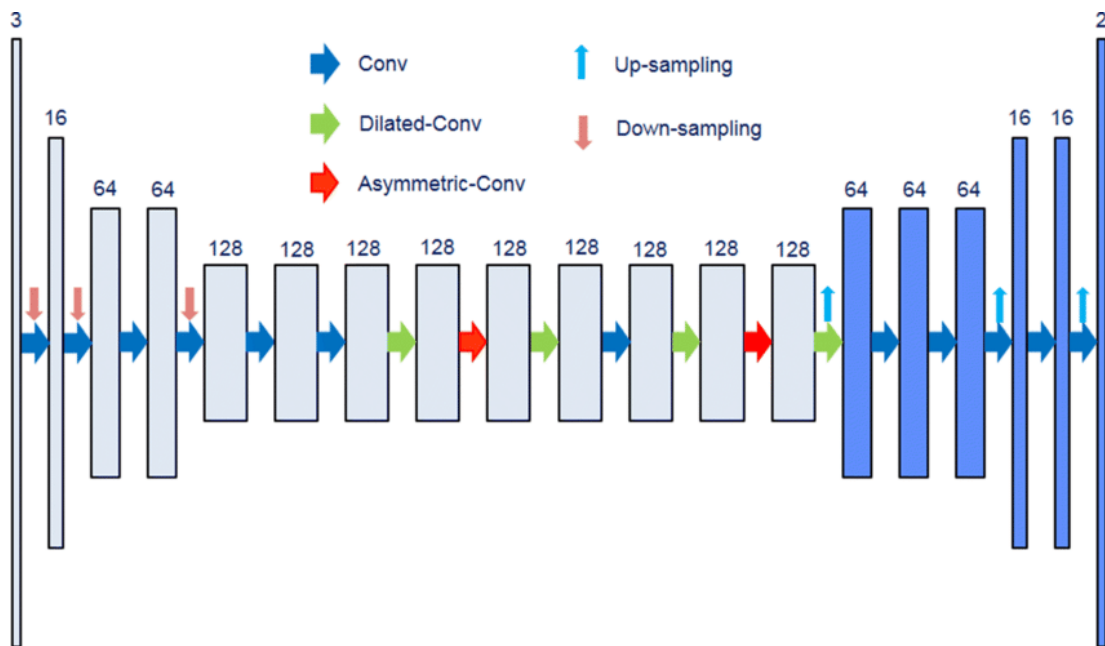
- The dataset consists of 3 folders:
- collages - collages of original photo, human figure flooded with color and segmentation mask
- images - original images
- masks - segmentation masks for images



- ▼  supervisely_person_clean.
- ▼  supervisely_person_cle
 - ▶  collage
 - ▶  images
 - ▶  masks
-  df.csv

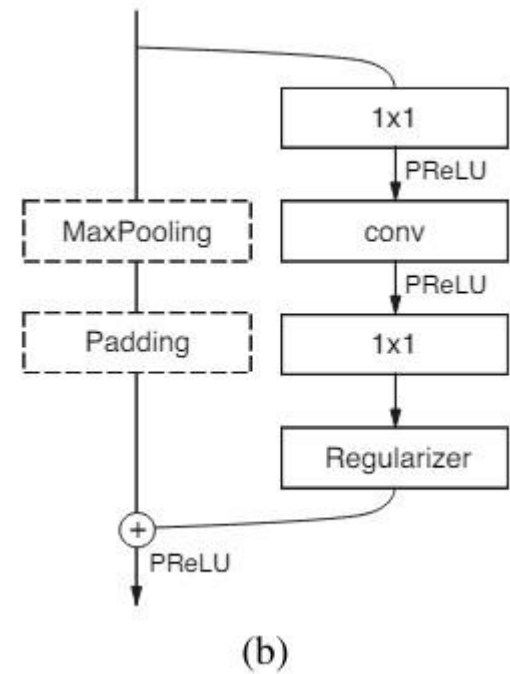
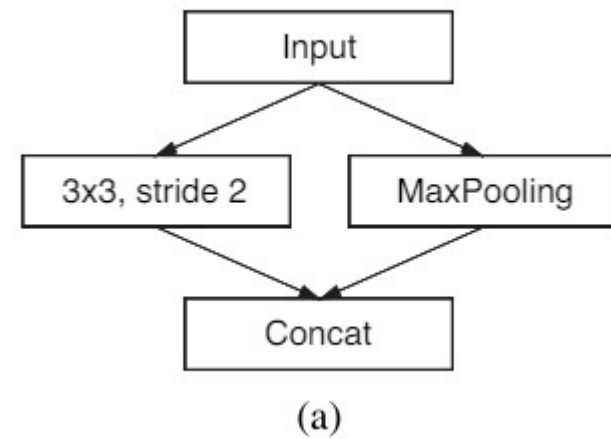


模型架構(ENet)

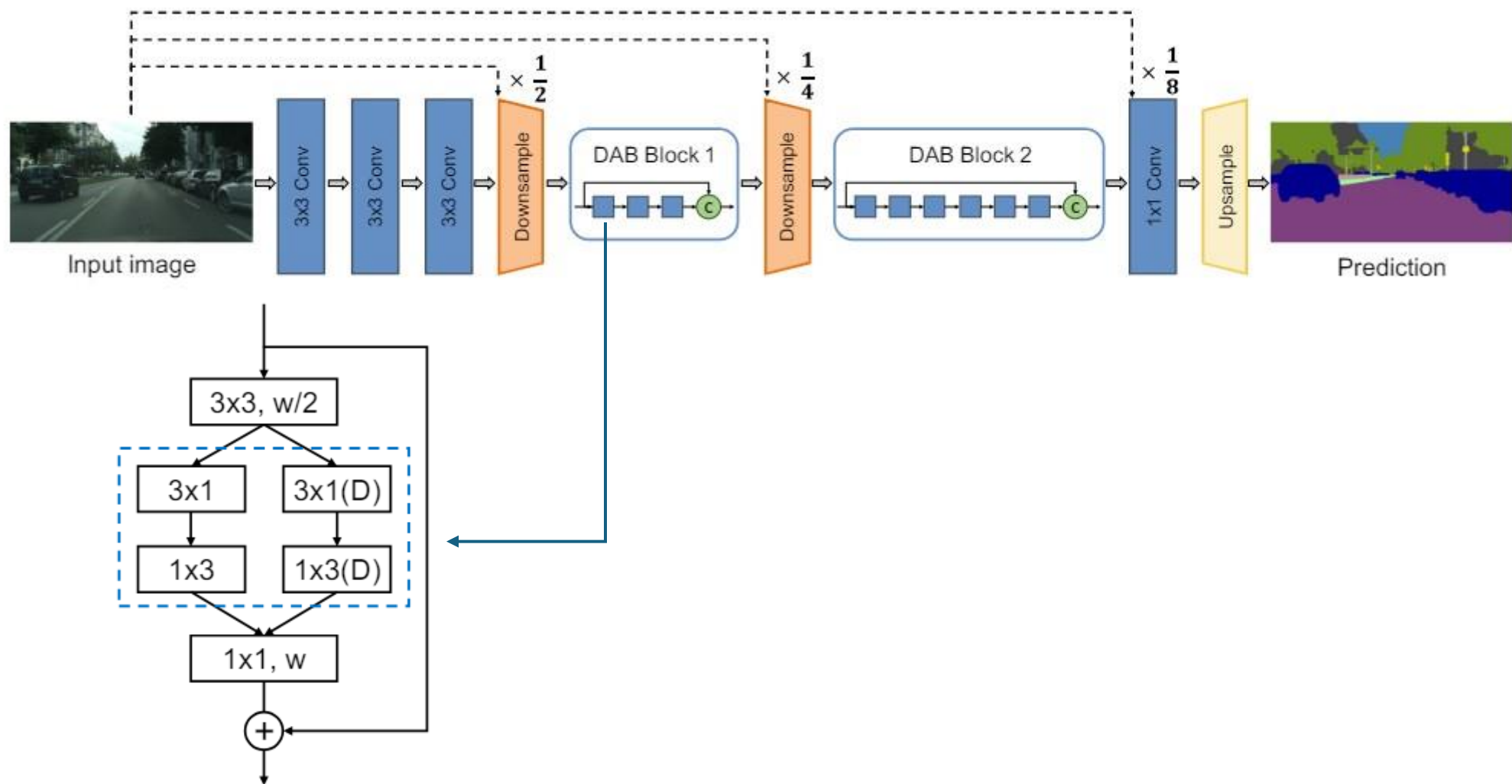


https://www.researchgate.net/figure/The-complete-architecture-of-the-ENet-model-is-shown-The-model-includes-both-encoder_fig5_326883590

<https://arxiv.org/pdf/1606.02147>



模型架構(DABNet)

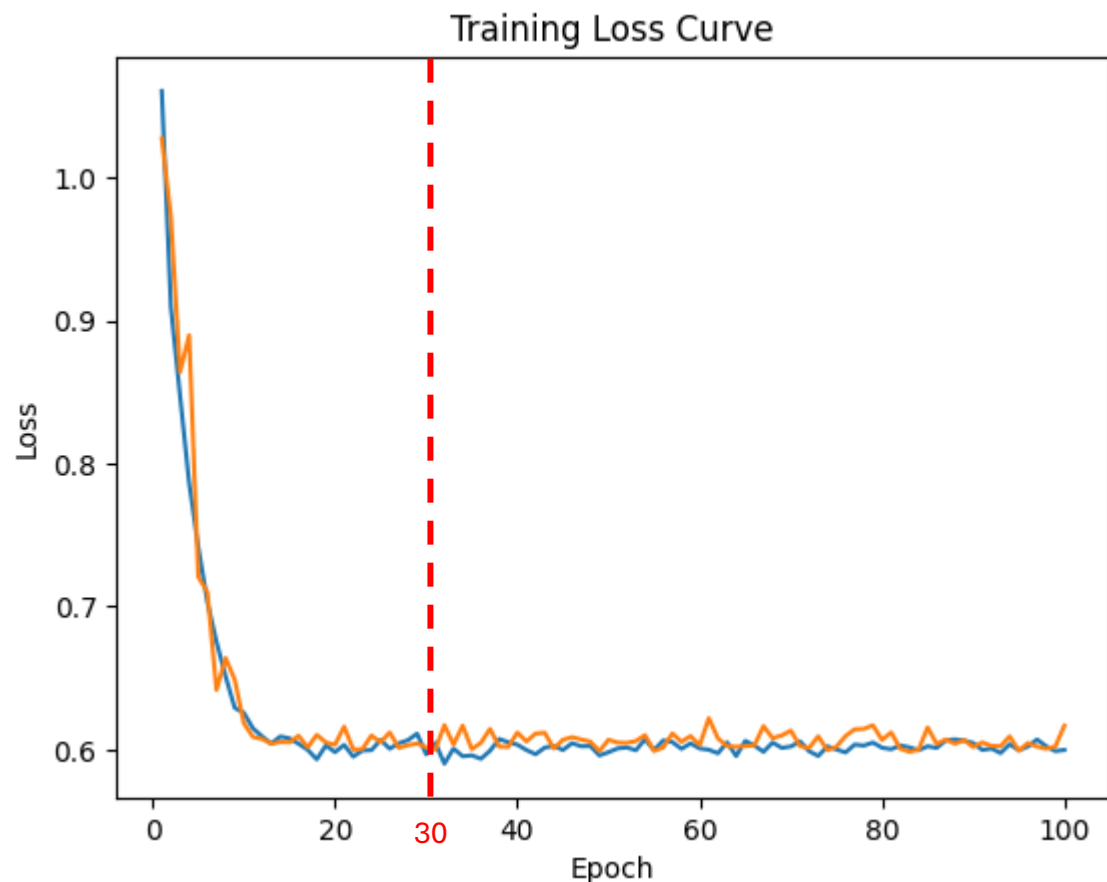


模型訓練設置

- Dataset: [Human Segmentation Dataset](#)
- 輸入大小: 480*640
- 資料增強: 水平翻轉0.5、垂直翻轉0.5
- 損失函數: binary dice loss , binary cross entropy
- 優化器: Adam
- 初始學習率: 0.1
- 學習率衰減: epoch *0.65
- Batch Size: ENet:16 DABNet: 8 (超過8會OOM)
- Epoch: 30
- 類別數量: 1

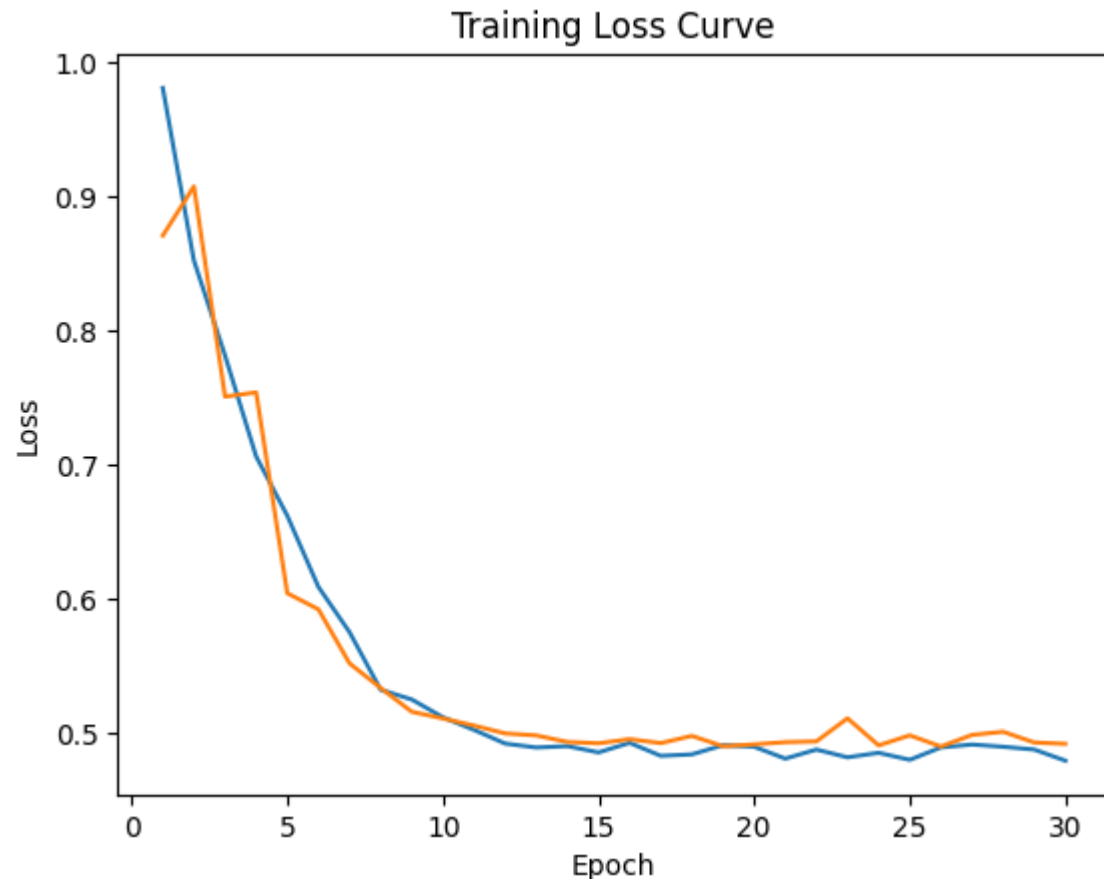
結果: Loss Curve

ENet



train loss: 0.5967304544217551
valid loss: 0.6016958916888517

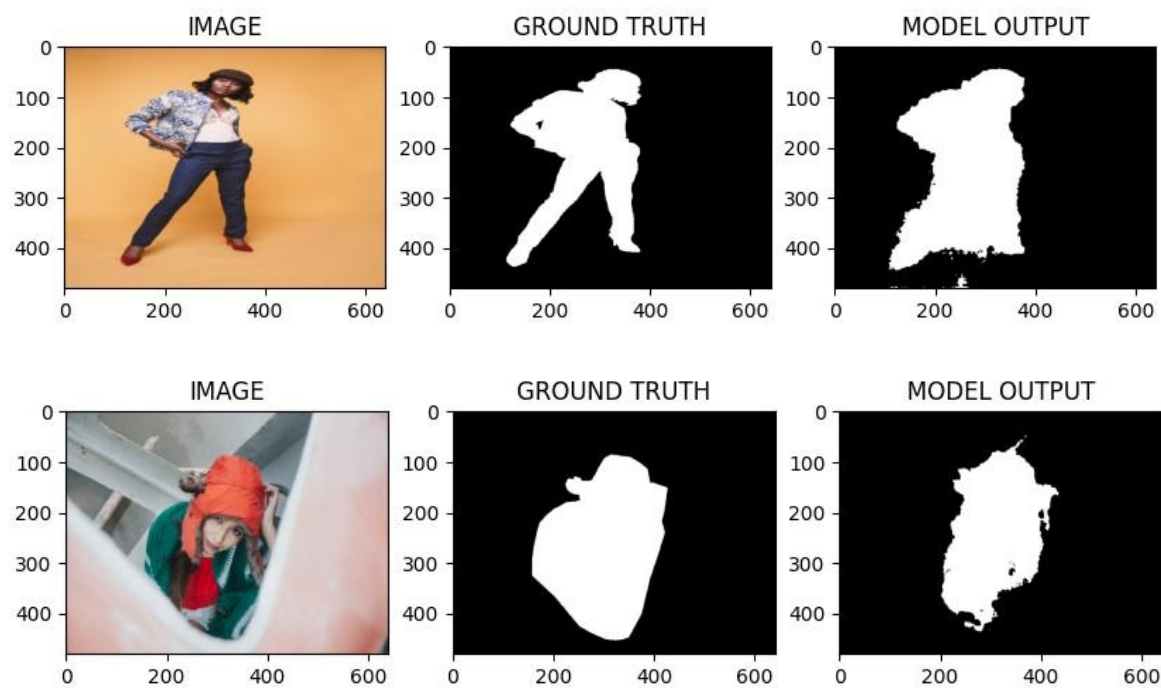
DABNet



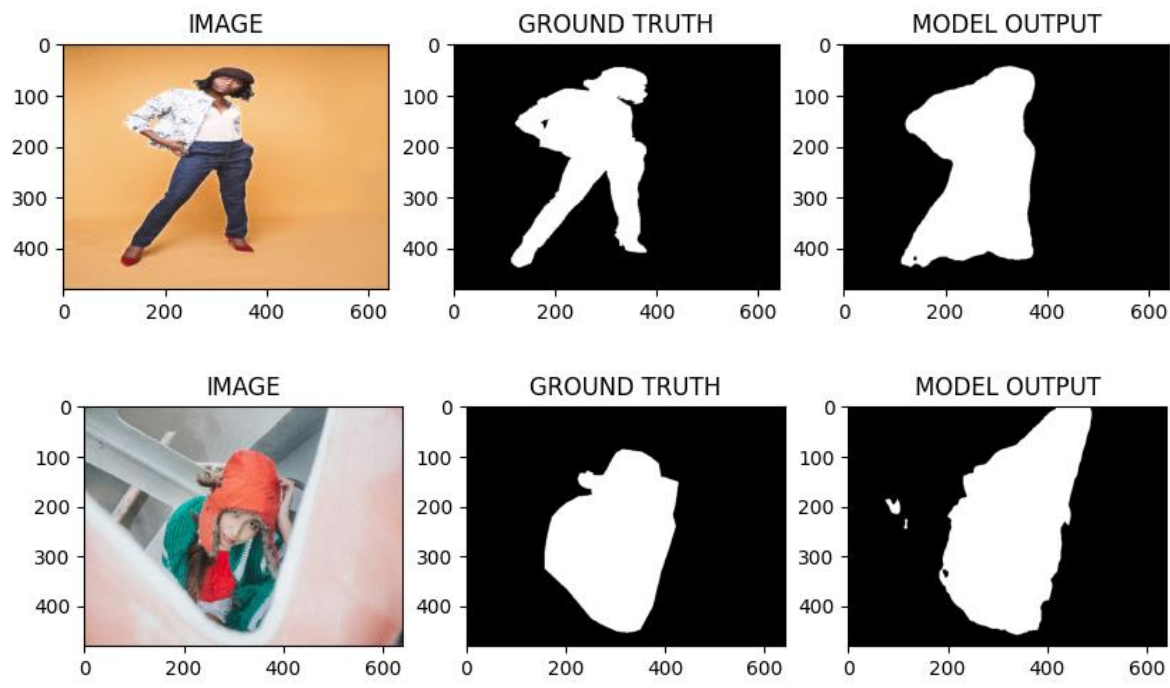
train loss: 0.47956186075335583
valid loss: 0.49215039577501923

資料集輸出比較

ENet

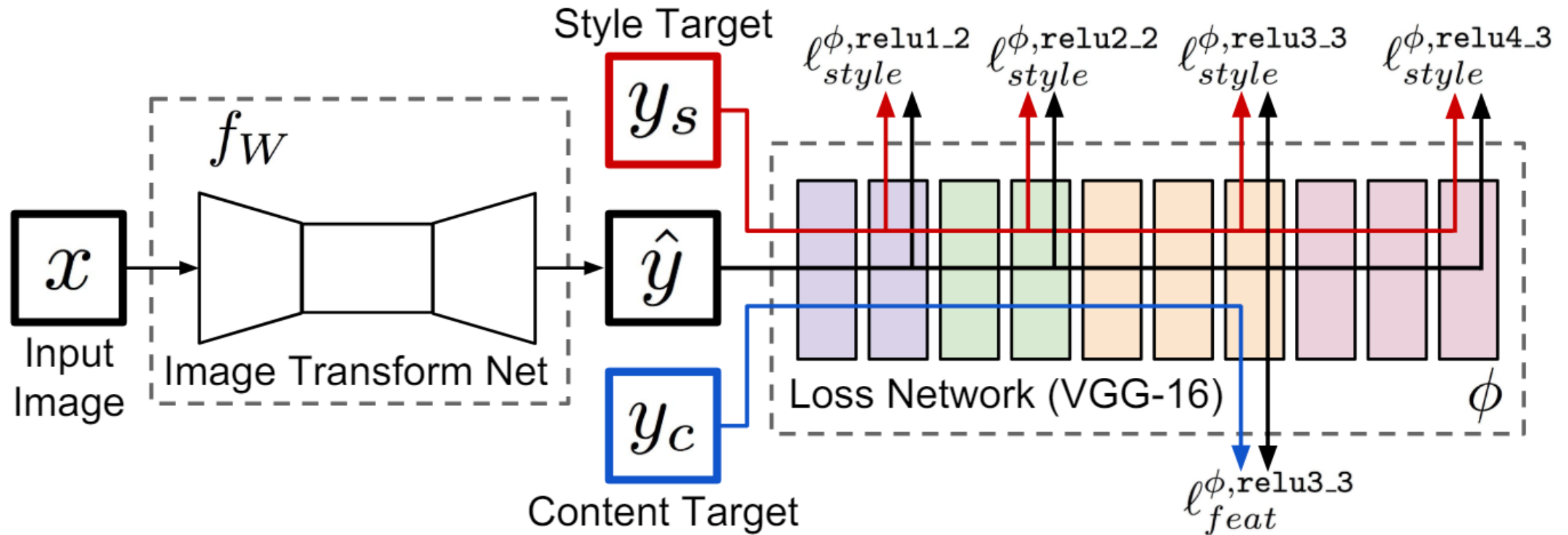


DABNet



風格轉換模型

Fast Style Transfer [fast-neural-style](#) (Pytorch)



Perceptual Losses for Real-Time Style Transfer and Super-Resolution
<https://arxiv.org/pdf/1603.08155>



CB Styling

16

CB Styling

- [IssamLaradji/CBStyling: Styling individual objects in an image \(github.com\)](https://github.com/IssamLaradji/CBStyling)
- Class-Based Styling: Real-time Localized Style Transfer with Semantic Segmentation
- 以DABNet實現Segmentation + Perceptual Losses for Real-Time Style Transfer and Super-Resolution (風格轉換)

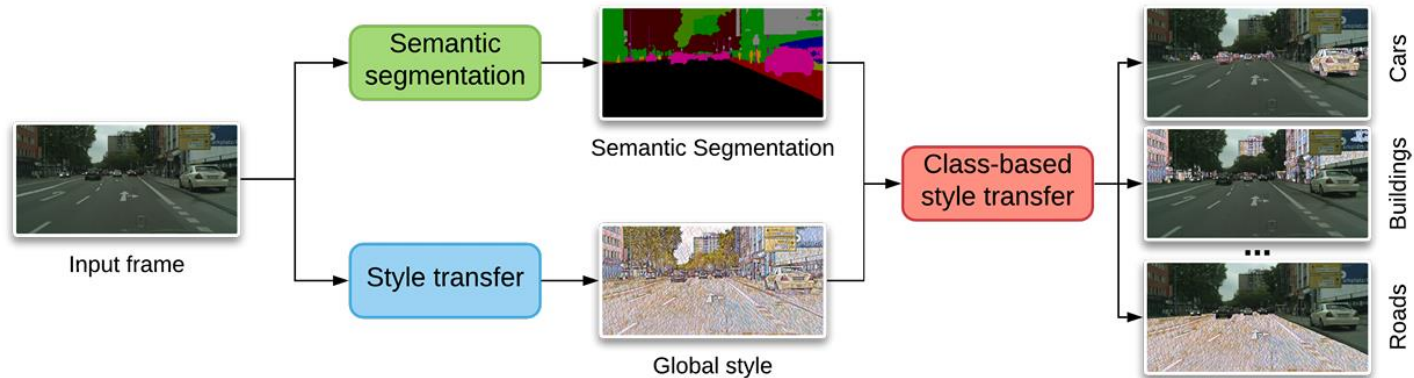


Figure 2: CBS pipeline. First, CBS takes a video frame as an input image and performs two operations in parallel: semantic segmentation and global style transfer. Then, the segmentation mask is combined with the styled image to style only the object classes of interest.

系統效能(Style Transfer處理速率)

```
Stylizing class 1...  
0.21991300582885742 sec  
Video capture FPS: 7.5  
Stylizing class 1...  
0.21946263313293457 sec  
Video capture FPS: 7.5  
Stylizing class 1...  
0.21941542625427246 sec  
Video capture FPS: 7.5  
Stylizing class 1...  
0.21657538414001465 sec  
Video capture FPS: 7.5  
Stylizing class 1...  
0.21741962432861328 sec  
Video capture FPS: 7.5  
Stylizing class 1...  
0.22135519981384277 sec  
Video capture FPS: 7.5  
Stylizing class 1...  
0.2186279296875 sec  
Video capture FPS: 7.5  
Stylizing class 1...  
0.21947097778320312 sec  
Video capture FPS: 7.5  
Stylizing class 1...
```

Image Size: 640 * 480 (Webcam)情況下
單純style transfer速度: 210 ms/frame

系統效能(語意分割模型處理速率)

- 參考論文: DABNet: Depth-wise Asymmetric Bottleneck for Real-time Semantic Segmentation
- 效能比較
 - ENet: 大概使用 63ms/frame
 - DABNet: 使用 34ms/frame

```
def get_masked_image(model, image, category, bg=0, model_name = "ENet"):  
    if model_name == "ENet":  
        mask=mask.to(device)  
        start_time = time.time()  
        logits_mask = model(mask.unsqueeze(0))  
        print("ENet:"+str(time.time()-start_time))  
        #print('logits_mask shape:', logits_mask.shape)  
    else:  
        #when coreml_model_loaded...  
        start_time = time.time()  
        output = model(input_var)  
        print("DABNet:"+str(time.time()-start_time))  
        torch.cuda.synchronize()
```

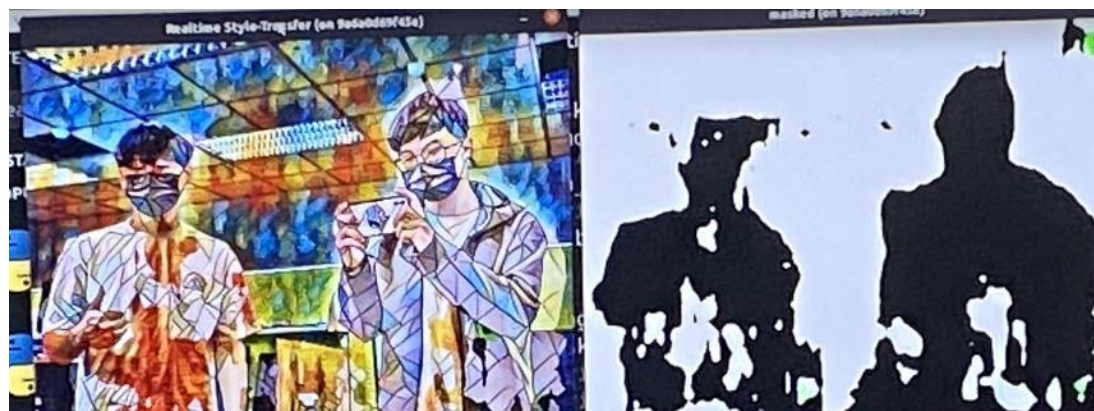
```
0.5753076076507568 sec  
Video capture FPS: 7.5  
Stylizing class 1...  
ENet:0.06161069869995117  
ENet:0.06517481803894043  
0.5758774280548096 sec  
Video capture FPS: 7.5  
Stylizing class 1...  
ENet:0.06375432014465332  
ENet:0.06374669075012207  
0.5761730670928955 sec  
Video capture FPS: 7.5  
Stylizing class 1...  
ENet:0.06170821189880371  
ENet:0.062262773513793945  
0.5752956867218018 sec  
Video capture FPS: 7.5  
Stylizing class 1...  
ENet:0.06379413604736328  
ENet:0.06179237365722656  
0.5739941596984863 sec  
Video capture FPS: 7.5  
Stylizing class 1...  
ENet:0.07126402854919434  
ENet:0.0658421516418457
```

```
DABNet:0.037450313568115234  
DABNet:0.03442788124084473  
0.8671872615814209 sec  
Video capture FPS: 7.5  
Stylizing class 11...  
DABNet:0.035620689392089844  
DABNet:0.034632205963134766  
0.8628950119018555 sec  
Video capture FPS: 7.5  
Stylizing class 11...  
DABNet:0.03575921058654785  
DABNet:0.03473472595214844  
0.8659448623657227 sec  
Video capture FPS: 7.5  
Stylizing class 11...  
DABNet:0.03518247604370117  
DABNet:0.03440713882446289  
0.8632023334503174 sec  
Video capture FPS: 7.5  
Stylizing class 11...  
DABNet:0.03648114204406738  
DABNet:0.034439802169799805  
0.8655211925506592 sec  
Video capture FPS: 7.5  
Stylizing class 11...  
DABNet:0.034439802169799805
```


效果



ENet



DABNet



DABNet (CBStyling)

模型部署

Docker 部署

- 先在terminal 打jetson_release可以查看jetpack版本

```
Software part of jetson-stats 4.2.7 - (c) 2024, Raffaello Bonghi  
Model: NVIDIA Jetson Xavier NX Developer Kit - Jetpack 5.1.1 [L4T 35.3.1]
```

- 搜尋NVIDIA L4T ML docker就可以找到官網或是github(dustynv)
- 在指令中加入--runtime=nvidia就可以用到GPU

Catalog > Containers > NVIDIA L4T ML

NVIDIA L4T ML



[jetson-containers](#) / [packages](#) / [l4t](#) / [l4t-ml](#) /



dusty-nv updated docs

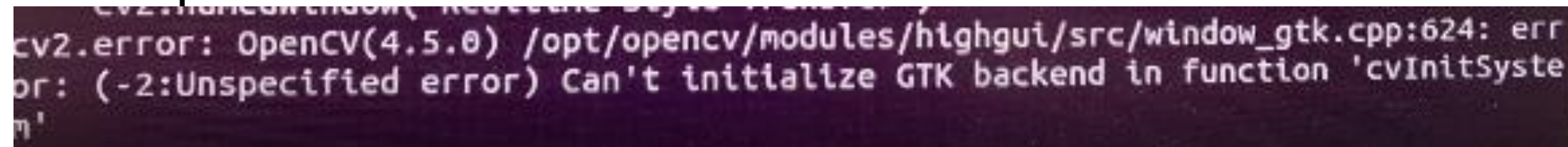
Docker
載點

套
件

- TensorFlow 2.11.0
- PyTorch v2.0.0
- torchvision v0.15.1
- torchaudio v2.0.1
- onnx 1.13.1
- onnxruntime 1.16.0
- optimum 1.8.8
- CuPy 13.0.0
- numpy 1.23.5
- numba 0.56.4
- PyCUDA 2022.2
- OpenCV 4.5.0 (with CUDA)
- pandas 2.0.1
- scipy 1.10.0
- scikit-learn 1.2.2
- diffusers 0.17.1
- transformers 4.30.2
- xformers 0.0.20
- JupyterLab 3.6.3

Docker 問題

問題1. OpenCV視窗開不起來



```
cv2.error: OpenCV(4.5.0) /opt/opencv/modules/highgui/src/window_gtk.cpp:624: error: (-2:Unspecified error) Can't initialize GTK backend in function 'cvInitSystem'
```

解決方法:

後來發現應該是沒有給予container XServer權限，參考之前學長yolov8教學再開啟docker前先在本地端打上xhost + 開啟所有權限就可以顯示opencv視窗。

問題2. 安裝PyQt5可能pip會裝不起來

解決方法:

要改從docker內的linux安裝:

```
sudo apt-get update
```

```
sudo apt-get install python3-pyqt5
```

分工

- Docker環境: 吳炯霖，羅文璟
- 模型訓練: 林耕澤，謝亞城
- PPT: 吳炯霖，羅文璟，林耕澤

Q&A