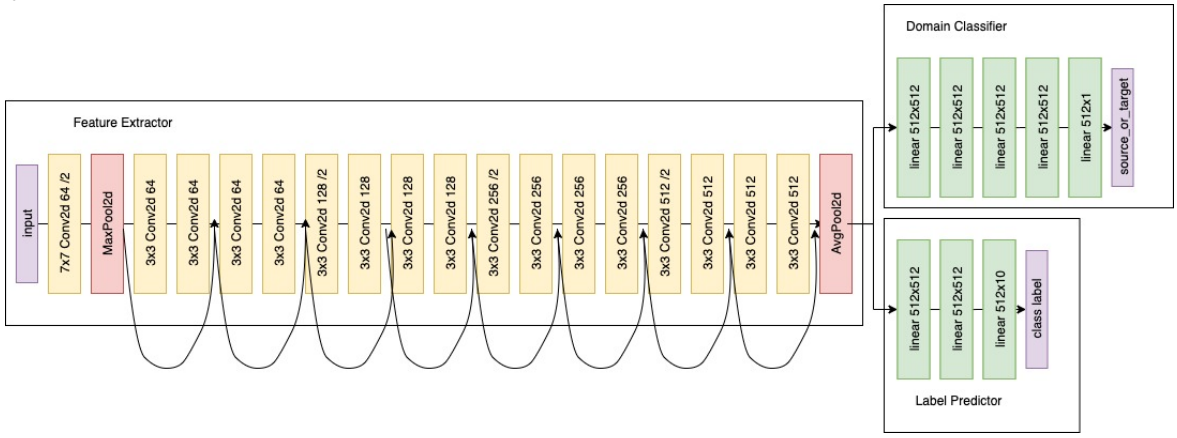
gg 學號：b07902047　　系級：資工二　　姓名：羅啟帆

1. 請描述你實作的模型架構、方法以及 accuracy 為何。其中你的方法必須為 domain adversarial training 系列 (就是你的方法必須要讓輸入 training data & testing data 後的某一層輸出 domain 要相近)。(2%)

   由於我最好的結果是 blending，這裡只描述其中一個 model（其他的有類似的架構只是參數不同）。我使用類似 resnet 的架構。下面的圖省略 Batch Normalization 和 ReLU 等等輔助的 layer。彎曲的箭頭代表的是 residual connection。



   Data augmentation 的部分則是分別對 source 和 target 採取不同的方式，主要的差別在於 Source 有使用 Canny。細節如下：

   - Source:
     - transforms.Grayscale()
     - transforms.Lambda(lambda x: cv2.Canny(np.array(x), 170, 300))
     - transforms.ToPILImage()
     - transforms.RandomHorizontalFlip()
     - transforms.RandomRotation(15, fill=(0,))
     - transforms.ToTensor()
     - transforms.Normalize([0.5], [0.5])
   - Target:
     - transforms.Grayscale()
     - transforms.Resize((32, 32))
     - transforms.RandomHorizontalFlip()
     - transforms.RandomRotation(15, fill=(0,))
     - transforms.ToTensor()
     - transforms.Normalize([0.5], [0.5])

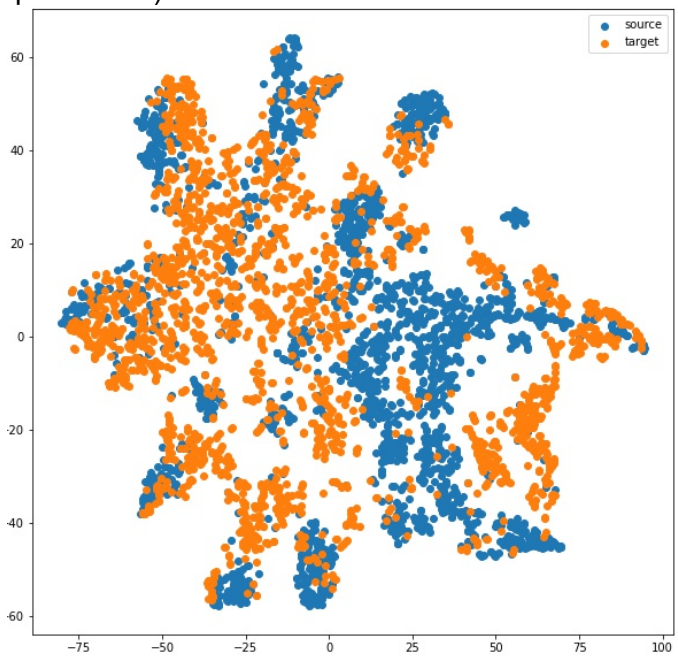
   最後，training 的部分使用 Adam 作為三個 model 的 optimizer，而 DaNN loss 裡面使用了 CrossEntropyLoss() 和 BCEWithLogitLoss()，並且採用 adaptive lambda，從 0 慢慢遞增到 1 ，公式為$\frac{2}{1+e^{-\gamma * p}} - 1.0$其中 p 為 training progress（也就是 current training step / total training step）。Training epoch 為 2500。

   以上的設定可以在 public test set 拿到 75.972%的準確率。

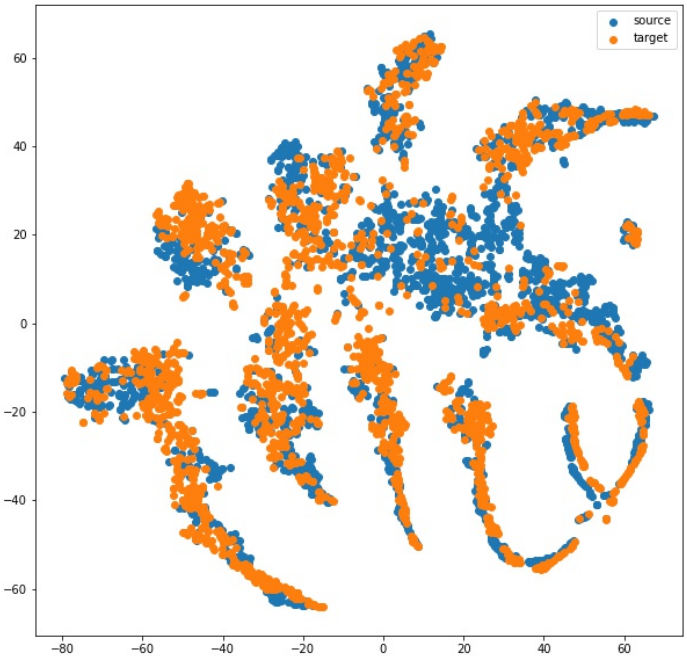
2. 請視覺化真實圖片以及手繪圖片通過沒有使用 domain adversarial training 的 feature extractor 的 domain 分布圖。(2%)

   原本的 feature 維度是 512，我使用 kernelPCA 和 t-SNE 降維以方便繪圖。細節如下：

- KernelPCA(n_components=100, kernel='rbf', n_jobs=-1)
- TSNE(n_components=2)



3. 請視覺化真實圖片以及手繪圖片通過有使用 domain adversarial training 的 feature extractor 的 domain 分布圖。(2%)
   降維方法和參數同上一題。



比較 2&3 可以發現有 domain adversarial training 的話，feature 疊合的情況比較好，也就是說他們分布比較相近。