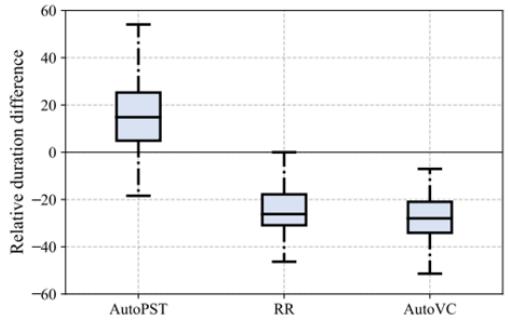
Global rhythm style transfer without text transcriptions

Student : Sian-Yi Chen

Advisor : Tay-Jyi Lin and Chingwei Yeh

Outline

- Action item
  - 使用 [1] 的 open source，並將執行程式時遇到的錯誤排除



圖一

- Status report
  - 論文內容：引用了對於 Voice Style 已證明有效的 AutoVC [2] 作為編解碼器架構，改動其演算法並達到有效轉換正確的 Prosody Style Transfer (PST)
    - 數據：
      1. 使用箱形圖比較不同神經網路架構的 Relative Duration Difference(RDD)，若為正則表示語音轉換中成功轉換了 rhythm information (圖一)，以及透過主觀聽覺來評估轉換果 (圖二)
      2. Paper Speech Demo Link
    - 架構：Autoencoder-based (圖三) 具有三個 model，Z(t) 代表 frame，顏色深淺相似的則表示 frame 的相似度高 (下一頁做進一步介紹)
  - 實作進度
    - 已將論文提供的 open source 環境架設完成並執行
    - Open source 有兩個主程式，預設迭代次數皆為 100 萬次，目前都把迭代次數下修為 1 萬次，並且正在執行 main_2，執行 1 萬次時間約為 10 小時

圖三

Table 1. Subjective evaluation results.

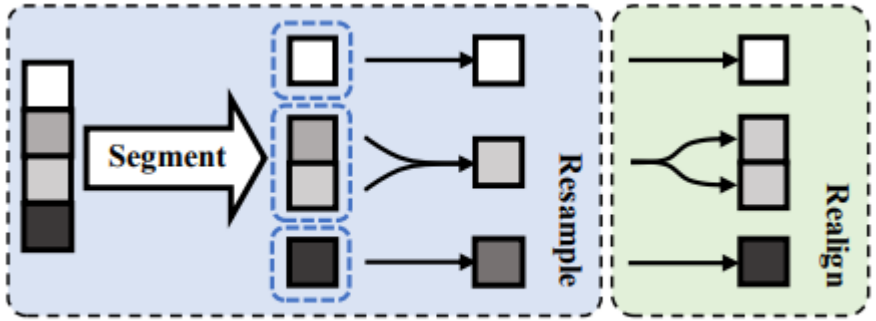
	AUTOPST	RR	AUTOVC
Timbre	**4.29 ± 0.032**	4.07 ± 0.037	4.26 ± 0.034
Prosody	**3.61 ± 0.053**	2.97 ± 0.063	2.64 ± 0.066
Overall	**3.99 ± 0.036**	3.63 ± 0.045	3.49 ± 0.052

圖二

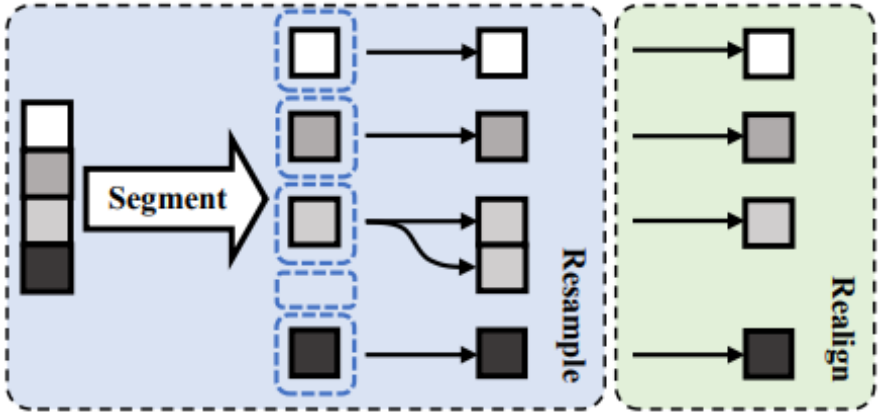
[1] K. Qian et al., "Global rhythm style transfer without text transcriptions," *arXiv:2106.08519 [eess.AS]*, 2021

[2] K. Qian et al., "AUTOVC: zero-shot voice style transfer with only autoencoder loss, " *arXiv:1905.05879 [eess.AS]*, 2019

Resampling



(a) The downsampling case ($\tau \leq 1$)



(b) The upsampling case ($\tau > 1$)

Goal：縮短具有較高相似性的 segment

τ ≦ 1

Step 1：藉由 [1] 找出鄰近相似的 frame

Step 2：將兩兩小於臨界值 τ 的 frame 作分割

Step 3：使用 mean-pooling 做下採樣

Goal：延長具有較高相似性的 segment

τ > 1

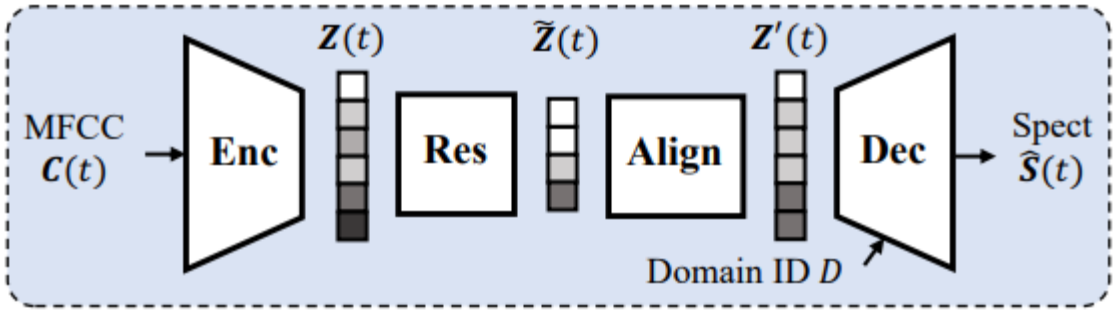
每一段都只包含一個 frame，並在相似度較高的地方插入空段，並複製之前的 frame 填入

[1] S. Bhati et al., "Self-expressing autoencoders for unsupervised spoken term discovery," *arXiv:2007.13033 [eess.AS]*, 2020

Two-Stage Training

雖然有 resampling model，但編碼器與解碼器仍可能找到方法互相傳遞一些 rhythm information，因此又使用了兩階段的訓練來防止任何可能。

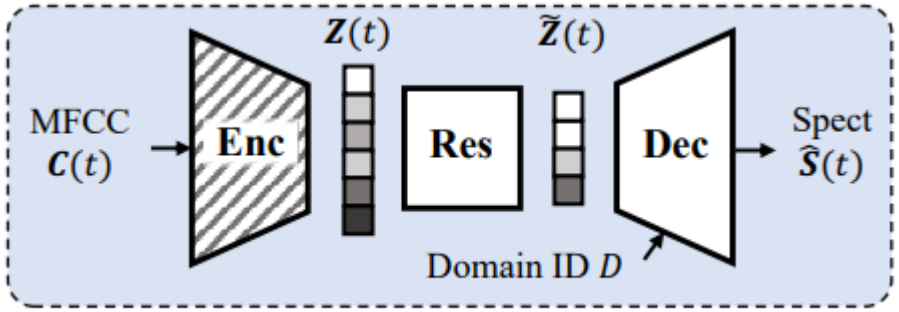
Stage 1：Synchronous training



(a) Synchronous training.

將 $\tilde{Z}$ 與 Z 對齊
下採樣 segment：複製 $\tilde{Z}$ 並對齊原始長度
上採樣 segment：刪除新插入的 $\tilde{Z}$
解碼器將會學到 content information 而非 rhythm information

Stage 2：Asynchronous training



(b) Asynchronous training.

移除 align model，並將 encoder 的參數固定，只更新 decoder，藉以讓 decoder 更難學習

附錄

論文摘要

在大多數非平行語音或是演算法中並不會轉換 prosody 這個重要資訊，作者提出了 AutoPST (Autoencoder-based Prosody Style Transfer)，其中架構由編解碼器夾著一個 resampling 所組成 (圖一)，在 resampling 這個 model 會藉由對時間重採樣來取得一個模糊的節奏 (obscure rhythm)，這樣解碼器就難以猜到原始語音的節奏資訊，增加解碼器的難度已達到效果更好的去學習韻律特徵。

那什麼是節奏資訊呢？

假設我們講了一個單詞為 CAT，它可以拆分成 3 個音素，說話慢的人，它的這個單詞就會由較多的音素所構成，而說話快的人，所需的音素就較少。

圖一

作者在論文中表示，在觀察中發現人類說話速度改變時，音素縮放的比率是不均勻的，其中母音的變化率大於子音、輔音，因此 resampling 需要去模仿這樣的機制。

因此利用上、下採樣來模仿人類的行為，並訂定一些規則來設計 resampling model。

GitHub 專案所需環境、套件以及檔案

Dependencies

- Python 3.6 — 查看 python 版本 python –version **OR** python –V
- Numpy — conda install numpy
- Scipy — conda install scipy
- PyTorch == v1.6.0 — conda install pytorch=1.6 (python 3.6 版本只支援到1.4)
- librosa — conda install -c conda-forge librosa
- pysptk — pip install pysptk，需要安裝 VS C++
- soundfile — pip install SoundFile
- wavenet_vocoder `pip install wavenet_vocoder==0.1.1` for more information, please refer to https://github.com/r9y9/wavenet_vocoder — pip install wavenet_vocoder==0.1.1

To Run Demo

Download pre-trained models to `assets` — pretrained_models.zip - Google 雲端硬碟

To Train

Download training data to `assets`. The provided training data is very — vctk16-train-wav.zip - Google 雲端硬碟