

(一)研究題目

美式手語拼字辨識 (American Sign Language Fingerspelling Recognition)

(二)指導教授姓名與組員姓名

指導教授：葉梅珍

組員：許哲安

(三)摘要

在現今科技進步的時代，語音助理的發展大幅增加，像是Siri、Google Nest 與 Amazon Alexa 等裝置，都提供人類更進步與便利的生活，但是世界上超過七千萬的聾啞人士與十五億以上的聽力受損者，由於聽力受損與先天差異，無法使用這些語音助理技術，為此，本計劃希望可以通過Kaggle辦的ASL recognition的競賽中，挑選一個訓練好的模型，用此模型為基礎，嘗試應用。

(四)研究問題與背景

在語音助理裝置的大幅發展下，對於人類的生活都產生了便利性，結合物聯網的應用，現在可以利用語音下命令，一句話就可以控制電燈、窗簾等，然而，對於全球超過七千萬依賴手語作為主要溝通方式的聾啞人士來說，無法以語音來溝通，使得這些語音裝置，難以使用。

拼字辨識，是利用人們比出的手勢對應到各個英語字母與數字，進而達到辨識的效果。在應用方面，對於聾啞人士拼名字、地址、電話號碼都能提供幫助，但大家可能會思考到一個問題，為何不能以手機替代成一種溝通媒介，利用手機鍵盤來拼字，傳達資訊？那是因為對於聾啞人士來說，利用手勢拼字會比手機打字快上許多，有些甚至達到兩倍的差距，根據統計，以美式拼字為例子，他們於手機鍵盤上每分鐘可以打三十六個字，但是以手勢拼字，每分鐘可以打五十七個字，這顯示了聾啞人士還是較熟悉以手勢來達到拼字的目標。

在上半學期，我們已經嘗試實際運行模型並實作了一個即時辨識系統，並且實作了一個即時辨識計算機系統（[demo影片](#)）。本學期的研究將重點放在以下兩個方面：

一、提升模型的辨識正確率：在現有模型的基礎上，我們將研究如何提升模型的準確性，使其在實際應用中能夠更可靠地辨識手勢。這涉及到優化算法、以及調整模型參數等多方面的工作。

二、訓練出一個新的手勢：我們將訓練新的手勢，以擴展現有手語拼字系統的功能，使其能夠識別更多的手勢，進一步提升其應用價值。訓練新手勢不僅能夠增加手語拼字系統的手勢量，我們希望也能夠提供其他有需求想增加新手勢的人，參考我們的方法與流程，為聾啞人士提供更加靈活和實用的溝通工具。

通過這兩個研究方向，我們希望能為聾啞人士提供一種更有效的溝通工具，讓他們也能享受到現代科技帶來的便利。我們相信，隨著技術的不斷進步，手語辨識系統將會在未來的智能設備中扮演重要角色，讓更多人能夠平等地享受科技的成果。

(五)相關文獻探討

本次專題所選擇的模型是利用連結時序分類(Connectionist temporal classification, CTC)與多層 Transformer 與一維卷積神經網路(1D Convolution Neural Network, CNN)訓練而成，CTC主要是針對連續型的資料，如語音處理、手寫辨識等，在訓練資料與類別資料尚未對齊的情況下，使用CTC能省去對齊的預處理要花費大量的人力和時間；Transformer，主要針對Sequence to Sequence 的模型，多用於自然語言處理，其中主要技術，Multi-Head Attention 提升了計算速度，能平行計算相似度，與循環神經網路(RNN)產生了效率上的區別；1D CNN多使用於處理時間序列數據，如股票價格預測，以及文本數據，如情感分析。以下將分別簡述CTC, Transformer 以及1D CNN 的概念。

一、連結時序分類(Connectionist temporal classification)

在手語拼字的模型訓練中，由於訓練資料是一段影片，而標籤資料是一段文字，在訓練之前，訓練資料與標籤資料是尚未對應的，換句話說，對於每一幀的資料，我們需要知道對應的標籤(label)才能進行有效的訓練，但困難的點是，對齊的預處理要花費大量的人力和時間，而且對齊之後，模型預測出的標籤(label)又只是區域性分類的結果，而無法給出整個序列的輸出結果，往往要對預測出的label做一些後處理才可以得到我們最終想要的結果。

CTC是利用將空白字元加入分類所有可能的字元，讓空字元也可能是 frame 預測的結果，接著在刪除連續相同字元成單一字元(例如：hh-el-loo => hello)，將所有可能預測成hello的機率相加，也就是以下方的公式(圖一)來說X為input sequence的集合，Y是output sequence的集合，以這個例子，Y是我們預期的output，也就是hello，X就是所有可能成為hello 的input，像是hh-el-loo, h-ee-l-llo...等

$$p(Y | X) = \sum_{A \in \mathcal{A}_{X,Y}} \prod_{t=1}^T p_t(a_t | X)$$

The CTC conditional probability	marginalizes over the set of valid alignments	computing the probability for a single alignment step-by-step.
---	--	--

圖一：CTC條件機率公式，本圖為Sequence Modeling With CTC[1]中的Figure

二、Transformer

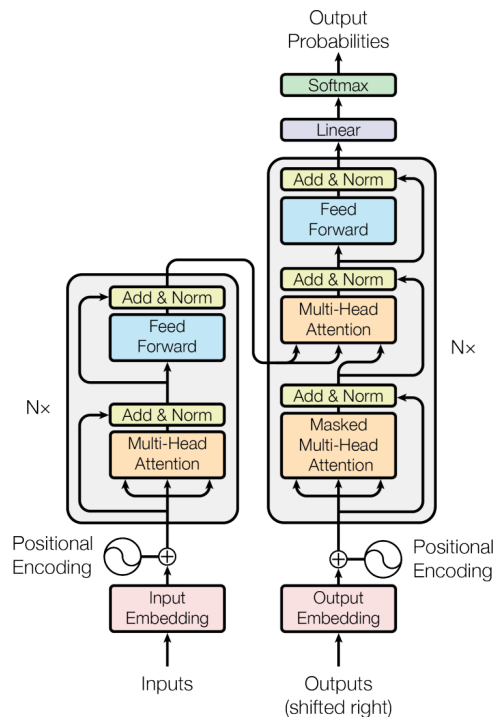
Transformer 主要是一個encoder-decoder的結構（圖三），encoder 主要是計算input中的相似性，會先利用Input Embedding 將input轉換成vector，再加上位置訊息(Positional Encoding)，因為Transformer主要是針對連續型的資料，所以順序會被記下來，之後再利用transformer核心技術Multi-Head Attention來一次計算所有input組

合的相似性，此技術解決了RNN計算大量資料的效率問題，因為RNN hidden state 為連續型計算，，假設要計算position t 的 hidden state h_t ，需要position t-1 的 hidden state h_{t-1} 與 position t 的input x_t ，造成RNN面對較大的資料，計算效率會逐漸變差。Transformer 做完Multi-Head Attention後會將結果再送進Feed Forward Network 中，此網路主要是將複雜的非線性關係計算出來，比較像是多個字的關係。

$$h_t = f_W(h_{t-1}, x_t)$$

圖二：RNN計算hidden state 的公式，圖片來源：[Recurrent Neural Network-RNN](#)

Decoder 的架構與Encoder相似，唯一不同的點是使用了Masked Multi-Head Attention，之所以需要masked是因為在預測position t 的 output時，我們只需要考慮position t 與position t 以前的字的相似性。

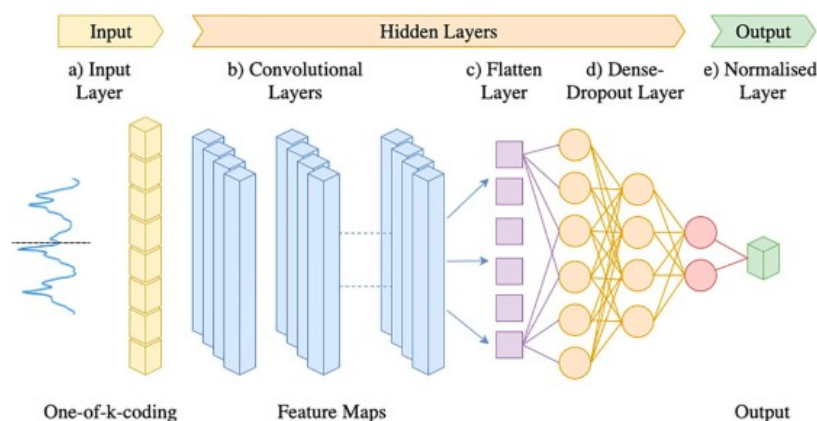


圖三：Transformer整體架構圖，本圖為Attention is all you need[2]中的Figure 1

三、一維卷積神經網路(1D Convolution Neural Network)

1D CNN是深度學習領域中的一種神經網路結構（圖四），主要應用在處理連續型數據，如時間序列或文本。這種網路是利用filter進行局部特徵提取(feature extraction)，通常使用1D filter在序列上進行滑動，抓取數據中的模式。另外，池化層(Pooling layer)也被使用，以降低輸入數據的維度並保留重要特徵。激活函數(Activation function)如ReLU引入非線性關係，而全連接層(Fully connected layer)則

用於做分類或迴歸預測。1D CNN與其他連續型處理模型相比，具有參數較少和計算效率較高的優點，適用於處理中等長度的序列數據。但對於長距離相似性的抓取表現可能較差。



圖四: 1D CNN架構圖，圖片來源: [One-dimensional convolutional neural networks for low/high arousal classification from electrodermal activity - ScienceDirect](#)

(六)研究方法與步驟

一、提升模型正確率：

在提升正確率的部分，主要嘗試以下方法

(一) 調整參數：

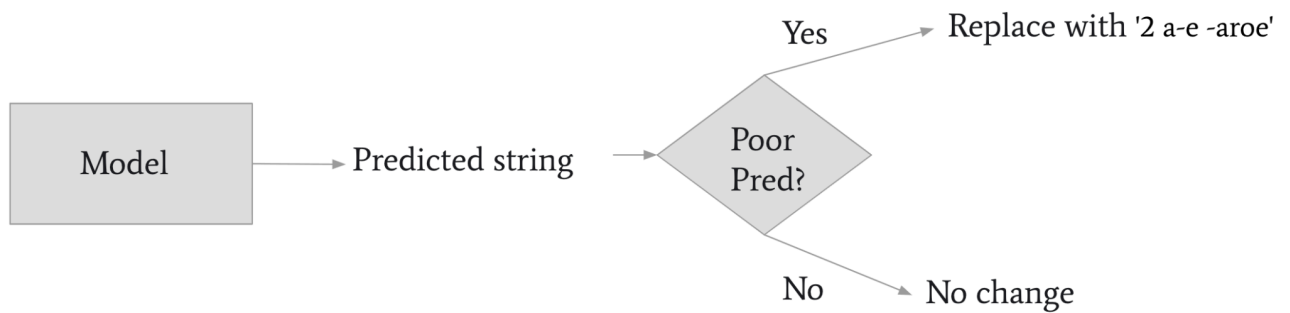
我們調整了訓練參數，包括epoch、max learning rate和warm up epoch的數值。通常來說，在沒有overfitting的情況下，增加訓練組數（epoch）能有效提升模型的最終正確率。然而，本次Kaggle競賽規定模型訓練時間的上限為9小時，這意味著我們不能僅依靠增加訓練組數來提升正確率，因為這會超過時間限制。為此，我們通過反覆試驗不同參數組合，尋找最優的模型辨識正確率。

(二) 資料前處理：

在原始模型中，部分完全沒有手部的幀數被保留。然而，由於訓練時間的限制，我們認為刪除這些幀數可以節省訓練時間。此外，手勢辨識的主要目標是手部動作，因此完全沒有手部的幀數可能對訓練幫助不大。我們嘗試將這些幀數刪除，以觀察模型正確率的變化。

(三) 預測後處理調整：

我們針對模型預測中的較短輸出進行後處理（圖五）。本次競賽使用Levenshtein distance來評估模型預測輸出與target label的loss。[此篇作者](#)利用貪婪法找出當字串為"2 a-e -aroe"時，模型可以在訓練資料中達到最高正確率0.1602。我們的調整是，當輸出長度小於某個數值時，我們會視為不好的預測(Poor Prediction)，將輸出替換為"2 a-e -aroe"，防止正確率下降太多。



圖五: 預測後處理流程圖

(四) 損失函數調整：

原始的損失函數為CTC Loss，我們嘗試加入KL Divergence Loss來防止模型overfitting。KL Divergence Loss則進一步增加模型的正則化效果，通過衡量兩個概率分佈之間的差異來防止模型對特定標籤的過度依賴。

二、訓練新手勢：

訓練新手勢的部分，主要想訓練手比愛心的手勢（圖六）。



圖六：手比愛心手勢 圖片來源：[finger heart korean style isolated on white Stock Photo - Alamy](#)

流程為以下四個步驟：

(一) 搜集資料集：

在網路上尋找真人手比愛心的影片，好的資料會是影片包含人體的臉部、上半身與比手勢的手。

(二) 提取資料集身體節點：

將資料經過影片剪輯後，利用Google Mediapipe套件，提取影片中每一幀的身體節點，包含臉部468個節點、上半身33個節點與左右手各21個節點，每一個節點會偵測x,y,z三個維度的數值，所以每一幀影片總共會偵測1629個數值。

(三) 資料前處理：

提取資料身體節點後，需將每段影片的資料合併成一個資料，並且1629個column命名必須與訓練資料的column相同（x_face_0, y_face_0, z_face_0 …..），另外在紀錄target label的資料中也需要新增我們新手勢的資料，也必須在紀錄手勢的資料中加上手比愛心的符號，” <3 ”。

除此之外，需要將現在的parquet file，轉成與訓練資料相同的TFRecord file。

（四）與原始資料一併訓練：


在訓練程式碼中與原始資料一併訓練模型。

訓練新手勢主要最大的問題是，在資料搜集取得不易，總共在網路上搜尋到大約44部影片，提取資料集身體節點後，資料大小約50MB，其他原始資料大小約85GB，所以預期在訓練之後，新手勢的辨識狀況會非常差，為了解決此問題，我先在訓練新手勢資料集中加入我的影片（4部影片），目的是先確保模型訓練後，針對訓練資料的學習是有效的，換句話說，先確認模型訓練是往正確的道路前進。


（七）研究結果

一、提升模型正確率：

原始的模型正確率（圖七）為0.748，而Fine Tune過後的模型正確率（圖八）提升到了0.753，提升了+0.005。

	Competition Notebook	Run	Private Score	Public Score
	<u>Google - American Sign Language Finge...</u>	29413.1s - GPU P100	0.719	0.748

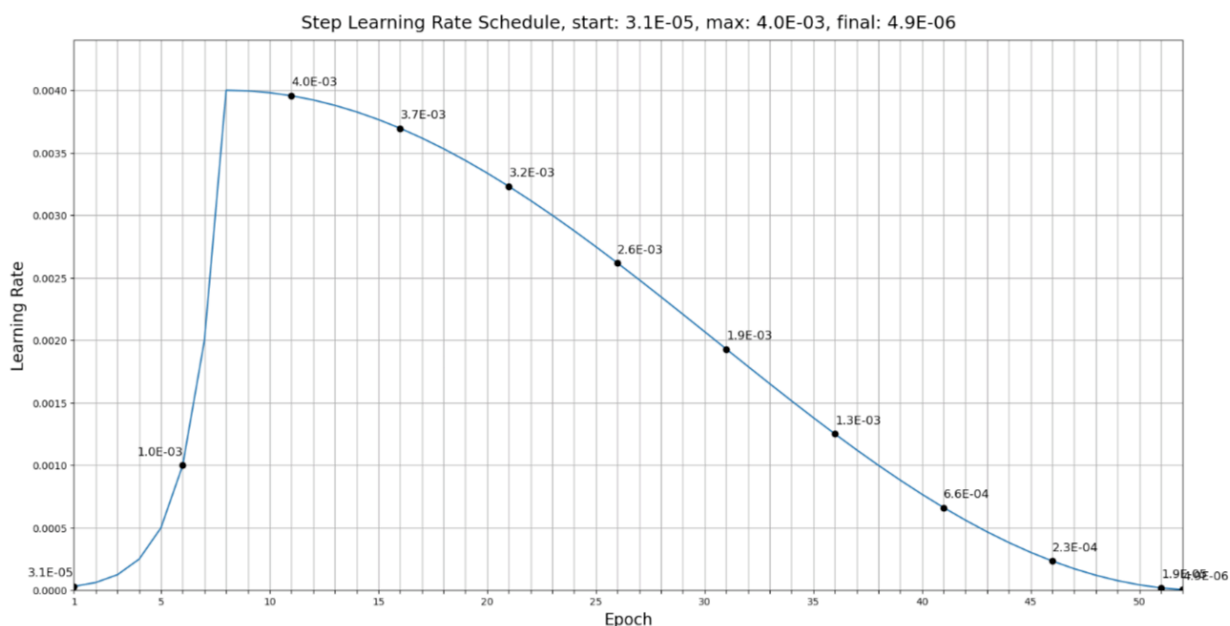
圖七: 原始模型正確率

	Competition Notebook	Run	Private Score	Public Score
	<u>Google - American Sign Language Finge...</u>	30680.7s - GPU P100	0.725	0.753

圖八: Fine Tune過後的模型正確率

（一）調整參數：正確率 +0.001

最後的參數調整為Max Learning Rate: 4.00E-03, N_Epoch: 52, N_warm_up_Epoch: 7 (圖九)



圖九: Learning Rate變化圖

(二) 資料前處理：正確率 -0.03

在實驗過程將無手部資料的偵數完全刪除會降低模型的辨識正確率，推測的原因為，雖然在辨識手勢的過程中，手勢是個重要的指標，然而，在訓練資料中，我們也會將臉部、上半身等身體資料納入一起做訓練，所以刪除無手部資料的偵數，可能會造成一併刪除臉部、上半身等重要訊息，造成模型沒有學習到臉部、上半身等資訊。

(三) 預測後處理調整：正確率 無明顯變化

(四) 損失函數調整：正確率 +0.004

在加入KL Divergence Loss之後，觀察到正確率有顯著的提升，推測可能的原因為原本的損失函數只包含CTC Loss，但CTC Loss僅關注模型輸出與目標字串(Target string)對齊的準確性，它可能會導致模型Overfit。加上KL Divergence Loss後，我們會計算模型輸出與均勻分布(Uniform Distribution)的相似程度，能降低模型Overfit的可能性，鼓勵模型在各類別之間不要有極端值的概率分佈。CTC Loss + KL Divergence Loss能讓模型不但保持正確性且也防止Overfit的情況產生。

$$\mathcal{L}(\theta_{\text{online}}) \triangleq (1 - \alpha)\mathcal{L}_{\text{CTC}} + \alpha \sum_{t=1}^T D_{KL}(P_t || U)$$

圖十: 損失函數公式，本圖為Improved training for online end-to-end speech recognition systems[5]中的第11條公式

二、訓練新手勢：

(一) 延伸問題與解決辦法

在訓練完成新手勢後，有嘗試測試新手勢的辨識，發現到模型不但能辨識我本人，還可以辨識不在訓練資料中的其他人，然而，在測試的途中，”<3”（愛心）的辨識狀況都還不錯，但假設我嘗試比出”<3a”（愛心加上字母a）的情況下，模型無法辨識出a，因為在訓練資料中沒有這一類型的資料（愛心加上其他字母），以至於模型沒有學習到。

解決辦法為產生”<3[a-z]”與”<3[1-9]”的訓練資料，共70部影片（左右手各一次），然而，未來可能的問題會是辨識效率不佳，因為這70部影片都是我個人產生的。

(二) 結果分析

結果分析是利用訓練資料去測試，目的是確保模型訓練是往正確的方向前進(接近正確率100%)，為何不使用測試資料？因為必須自己人工生成，不但需要不同人的測試資料，還必須產生一定的數量，對於現階段要產生大量且不同人的測試資料，我認為是無法達成的，所以最後決定使用訓練資料來測試，至少確保模型是在正確的方向。

結果分析 (圖十一)可以參考附圖，也可以參考[demo影片](#)。

分析圖中X軸分別為Original Model（原始模型）、Model 1（新手勢模型：訓練新手勢資料只有包含”<3”）與Model 2（最終新手勢模型：訓練新手勢資料包含”<3”，”<3[a-z]”與”<3[1-9]”）。

Y軸為正確率，是用[Levenshtein Distance](#)來評估，Levenshtein Distance是計算兩個字串的差別，計算要利用幾個(replace, insert, delete)將prediction string變成target string。

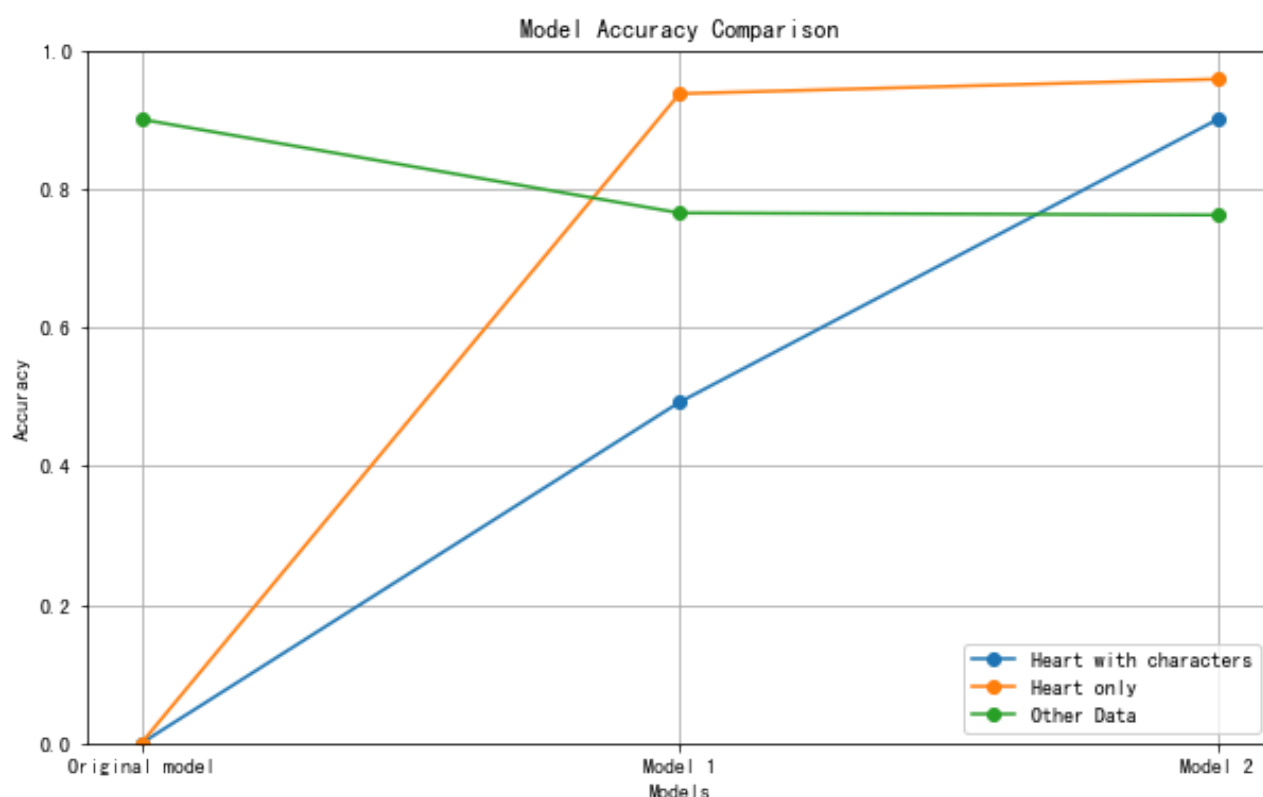
圖中包含三個資料集，綠色為少部分的原始資料集(Other Data)，共有999 samples，橘色為只有”<3”的資料集(Heart Only)，共有48 samples，藍色為”<3[a-z]”與”<3[1-9]”的資料集(Heart with characters)，共有70 samples。

Original Model，Other Data 正確率為0.9，其餘新手勢的正確率為0，因為Original Model的訓練資料中，沒有新手勢的訓練資料。

Model 1，Other Data 正確率為0.765，heart only的正確率為0.937，heart with characters的正確率為0.492。觀察到Other Data的正確率有些微的下降，原因是Model 1多了新手勢的”<3”的辨識，所以可能在某些字元的辨識上有錯誤，不過這個下降是可接受的，另外heart only的正確率說明模型學習是往正確的方向，而heart with characters的正確率接近50%，因為雖然Model 1只能辨識”<3”，但在輸入資料”<3a”，Model 1至少能辨識出”<3”，所以在每一筆資料中預期的loss為0.5。

Model 2，Other Data 正確率為0.762，heart only的正確率為0.958，heart with characters的正確率為0.9。觀察到Other Data的正確率無明顯變化，另外heart only的正確率提升了一點，可能的原因是加上了”<3[a-z]”與”<3[1-9]”

的資料，訓練資料變多，學習效果提升，而heart with characters的正確率可以說明模型學習是往正確的方向。



圖十一: 模型表現比較圖

(八)後續研究/實作方向

一、提升模型辨識正確率：

(一) 嘗試使用排名第一模型的方法：

在Kaggle ASL競賽中，排名第一的模型正確率為0.836，使用Squeezeformer Encoder & Transformer Decoder模型架構，而我們使用的模型架構是Transformer Encode & Transformer Decoder，未來會嘗試將Squeezeformer Encoder使用在本次模型，比較模型正確率的差異。

(二) 現有模型加入輸出長度參數：

於此篇Fingerspelling PoseNet: Enhancing Fingerspelling Translation with Pose-Based Transformer Models [6]論文中，作者在模型中加入輸出長度的參數，並且發現當模型能預測輸出的長度時，對於辨識正確率有提升的效果。

二、訓練新手勢：

在本次研究中，訓練新手勢的最主要的挑戰是無法獲取足夠且多樣的訓練資料。因此，未來的研究將著重於以下幾個方面以擴展訓練資料集：

(一) 資料來源多樣化：

與聾啞學校、手語培訓機構等專業機構合作，獲取更真實和多樣的手語拼字影片資料。

（二）自行錄製資料：

在現有資料不足的情況下，自行錄製手語拼字視頻是一個有效的方法。為此，但需要以下幾個步驟：

標準化流程：制定統一的錄製標準，包括手語的動作範圍、速度、角度等，以確保資料的一致性和質量。

多樣化參與者：邀請不同年齡、性別、種族的手語使用者參與錄製，以增加數據集的多樣性和代表性。

（三）資料擴充技術：

利用數據擴充技術來增加訓練資料的數量 and 多樣性，例如：

數據增強：對現有的手語視頻進行旋轉、平移、縮放、調整亮度和對比度等操作，以生成更多的訓練樣本。

生成對抗網絡（GANs）：利用GANs技術生成新的手語拼字視頻，擴充數據集。

（九）參考文獻與資料

- [1] Hannun, A. (2017). Sequence Modeling With CTC. Distill, <https://distill.pub/2017/ctc/>
- [2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, I. (2017). Attention is All You Need. In Advances in Neural Information Processing Systems (pp. 5998-6008)
- [3] “Week 4: CTC”, 12 Aug. 2021, <https://hackmd.io/@computerVision/Hk3chA6Qd>
- [4] “Static Greedy Baseline [0.157 LB]”, 2023, [Static Greedy Baseline \[0.157 LB\] \(kaggle.com\)](https://kaggle.com/Static-Greedy-Baseline-0.157-LB)
- [5] Suyoun Kim, Michael L. Seltzer, Jinyu Li, Rui Zhao (2018). Improved training for online end-to-end speech recognition systems
- [6] Pooya Fayyazsanavi, Negar Nejatishahidin , Jana Košecká (2023). Fingerspelling PoseNet: Enhancing Fingerspelling Translation with Pose-Based Transformer Models