

國立嘉義大學資訊工程學系  
計算機專題報告

Department of Computer Science and  
Information Engineering  
National Chiayi University  
Computer Project Report

戀 AI 物語

指導教授： 陳宗和 老師

年度： 一百一十三學年度

組別： 347-114-16

學生： 1102934 王雍心

1102936 黃建澄

中華民國 一百一十三 年 十二 月

# 國立嘉義大學資訊工程學系 計算機專題報告推薦書

國立嘉義大學資訊工程學系

\_\_\_\_\_、\_\_\_\_\_ 君

所提之計算機專題報告(題目)：

\_\_\_\_\_係由本人指導撰述，經審核同意交付本系歸檔留存。

指導教授 \_\_\_\_\_ (簽章)

系(所)主任 \_\_\_\_\_ (簽章)

\_\_\_\_\_年\_\_\_\_\_月\_\_\_\_\_日

# 戀 AI 物語

指導教授：陳宗和 老師 學生：王雍心、黃建澄

國立嘉義大學資訊工程學系

## 摘要

本專案旨在設計一個多模態互動網頁應用，讓使用者透過文字、語音和圖像與後端接入的 LLM（如 ChatGPT、Claude）進行自然的交流。前端集成了 live2D 人物模型以增強互動效果，並實現多種功能，如 AI 模型切換、背景音樂播放、表情符號反應、語音回放等。後端使用 Python 整合語音和語言處理 API，JavaScript 負責前端互動事件。技術層面上，本研究微調了 AI 模型回應，使其更具個性化和情感化，並通過開源技術訓練聲音模型，使語音輸出自然且生動。

關鍵字：live2D、LLM、微調、語音合成

# 目錄

摘要 .....	i
目錄 .....	ii
圖目錄 .....	iv
表目錄 .....	v
第一章、緒論 .....	1
1.1 研究背景 .....	1
1.2 研究動機與目的 .....	2
1.3 研究貢獻 .....	3
第二章、文獻回顧及探討 .....	5
2.1 Transformer 架構 .....	5
2.2 Fine-tuning .....	7
2.3 GPT-SoVITS(GSV) .....	8
2.4 現有 AI 聊天 app 比較 .....	10
第三章、研究方法 .....	11
3.1 系統架構設計 .....	11
3.2 技術實現 .....	12
3.3 模型微調 .....	13
3.3.1 語音模型設定 .....	13
3.3.2 語言模型微調 .....	16
3.3.3 訓練過程 .....	17
3.4 提升互動性 .....	17
3.4.1 人物模型 .....	17
3.4.2 情緒回應與背景音樂 .....	18

第四章、 實驗結果與討論 .....	20
4.1 實驗結果 .....	20
4.2 技術實作與挑戰 .....	22
4.3 系統性能觀察 .....	23
第五章、 結論與未來展望 .....	24
5.1 結論 .....	24
5.2 目前挑戰與未來展望 .....	24
參考文獻 .....	26

## 圖目錄

圖 2.1：Transformer 架構圖[3] .....	7
圖 2.2：各 app 比較圖 .....	10
圖 3.1：系統架構圖 .....	12
圖 3.2：校對畫面 .....	14
圖 3.3：模型微調介面 .....	15
圖 3.4：語音模型推理界面 .....	16
圖 3.5：資料集格式 .....	17
圖 3.6：微調語言模型[7] .....	17
圖 4.1：介面及功能說明 .....	20
圖 4.2：切換模型選單 .....	20
圖 4.3：live 2d 模型和情緒回應 .....	21
圖 4.4：對話範例一 .....	21
圖 4.5：對話範例二 .....	21
圖 4.6：可使用圖片及語音聊天 .....	22

## 表目錄

表 3.1：語音模型超參數設定 .....	14
-----------------------	----

# 第一章、緒論

## 1.1 研究背景

本研究背景探討了當前科技趨勢下，人機互動和虛擬角色應用的發展，並指出如何將人工智慧（AI）整合到使用者介面設計中以提升互動體驗。

隨著人工智慧和自然語言處理（NLP）[1]技術的進步，像 ChatGPT 和 Claude 這類大型語言模型（LLMs）[2]已經廣泛應用於對話式 AI 系統中，為用戶提供更加自然、智能的互動體驗。同時，語音合成技術也取得了顯著進展，從傳統的合成語音轉向更加自然、擬真的語音輸出，這促使許多應用領域逐漸將語音交互作為核心功能之一。

虛擬角色技術，特別是 live2D 與 3D 動畫技術的應用，已成為提升使用者介面體驗的趨勢之一。這類技術通過動畫化、情感化的虛擬形象，讓用戶與 AI 的互動更加具象化和生動化，尤其是在娛樂、教育、心理諮詢等領域逐漸成為主流。這樣的互動不僅能加強沉浸感，還能促進人機情感聯繫。

此外，現代使用者介面設計強調個性化與情境感知，能夠根據用戶的情緒或行為自動調整反應，使得整體體驗更加符合用戶需求。微調 AI 模型以模擬特定性格的回應，正是這類個性化設計的應用。透過語音模型的進一步優化，生成的語音不僅更加擬真，甚至能根據不同情境展現出可愛、友善或其他情緒特質。



整合語言、語音和虛擬角色互動的系統設計，代表了未來人機交互的新趨勢，不僅限於工具性應用，也逐步向娛樂性、陪伴性發展。本研究正是在這樣的背景下，探索如何透過前端和後端技術的結合，提升使用者與 AI 系統的互動品質和沉浸感，進一步推動人機協作和交流的未來發展。

## 1.2 研究動機與目的

隨著人工智慧技術的日益成熟，AI 在日常生活中的應用越來越廣泛，從智能語音助理到對話機器人，無不影響著人們的互動方式。然而，現有的 AI 互動系統仍然多以工具性為主，缺乏情感連結與擬真互動，使用者往往無法從中感受到溫度或個性化的回應。這使得 AI 系統在實際應用中，難以滿足現代使用者對於更真實、更具情感深度的需求。

因此，本研究的主要動機是探索如何通過微調語言模型，實現更具個性化的 AI 回應，讓 AI 不僅能理解並回應使用者的語言，還能反映出特定性格和情緒，從而營造出更加生動有趣的對話情境。同時，語音合成技術的擬真化與情感化也是本研究的重要焦點。現代使用者對於語音互動不僅要求準確，還期望語音輸出能夠傳達情感，讓人機對話更加自然流暢。因此，如何訓練語音模型使其能夠以擬真且可愛的方式輸出情感化語音，成為研究的核心議題之一。

在此基礎上，提升使用者互動體驗是輔助的研究目標。透過整合 live2D 虛擬角色、個性化回應及情感化語音輸出，研究旨在打造一個更具沉浸感的 AI 互動系統。這不僅能讓使用者在與 AI 互動時感受到更高的趣味性，還能增強人機情感連結，使 AI 在不同行業中的應用更加廣泛，如娛樂、教育、心理支持等。

總結而言，本研究的核心目的是實現個性化 AI 回應與語音的擬真化與情感化，並以此推動人機互動技術向更自然、更具情感的方向發展，最終提升使用者的互動體驗作為輔助效果。

### 1.3 研究貢獻

本研究的貢獻主要體現在個性化 AI 回應的實現以及語音合成技術的擬真化與情感化，這兩者共同推動了人機互動體驗的質變。

首先，在個性化 AI 回應方面，本研究通過對語言模型的微調，使 AI 能夠根據特定性格或情境來調整回應的語調與內容。這不僅讓 AI 不再是單純的工具，還賦予了其一定的「人格特質」，讓使用者感受到更具親和力的互動體驗。本研究展示了如何通過技術手段，讓 AI 回應更加符合特定性格或角色的預期，為未來個性化和情境感知的 AI 系統提供了範例。

其次，在語音輸出的擬真化與情感化方面，本研究利用先進的語音合成技術，訓練出能夠表達不同情緒的語音模型，進一步提升了 AI 的表現力。相較於傳統單調、機械的語音合成技術，本研究所開發的模型能夠輸出更自然且具有情感的聲音，讓人機對話更加擬真且富有生命力。這項貢獻不僅應用於對話式 AI，還可延伸至娛樂、教育等需要語音互動的領域。

此外，透過整合虛擬角色與語音合成技術，本研究展示了如何將 live2D 人物模型與 AI 系統結合，使得虛擬角色不僅能透過動畫進行互動，還能以個性化的語音進行回應，從而

加強了互動的沉浸感。這種多維度的人機互動形式突破了傳統的 UI 設計範疇，為未來虛擬角色與 AI 交互系統的設計提供了技術參考。

最後，本研究的技術應用不僅局限於人機互動領域，還為**多場景應用**提供了可能性，如在教育中，AI 能根據學生需求做出個性化的教學回應；在心理諮詢中，AI 則能以擬真且情感化的語音表達安慰和支持，為使用者帶來更加貼心的陪伴和服務。

總結而言，本研究不僅在技術層面上為個性化 AI 回應與語音合成提供了新的解決方案，也從整體上提升了人機互動體驗，為 AI 技術在多元場景中的應用提供了全新的視角與可能性。

## 第二章、文獻回顧及探討

### 2.1 Transformer 架構

Transformer 架構的誕生，源自於 2017 年 Vaswani 等人在論文《Attention is All You Need》[3]中提出的突破性概念。當時，序列模型如 RNN 和 LSTM[4]在自然語言處理(NLP)任務中表現出一定的能力，但由於它們的運算方式需要依賴逐步處理的特性，這導致它們在處理較長文本序列時遇到了效率和性能上的瓶頸，特別是在捕捉長距離依賴關係時效果不佳。為了解決這一問題，Transformer 引入了「自注意力機制」(Self-Attention Mechanism)，從根本上改變了序列數據的處理方式。

Transformer 架構的核心原理是自注意力機制，它允許模型在處理每個單詞時，不僅僅考慮相鄰的單詞，還能同時考慮序列中所有其他單詞，這使得模型能夠有效地捕捉長距離的依賴關係。具體來說，自注意力機制會通過計算「查詢(Query)、鍵(Key)和值(Value)」三個向量之間的關係，來決定每個單詞與其他單詞之間的關聯度。模型通過這種方式動態調整每個單詞的表示，從而生成更具全局性的語意理解。這與過去的序列模型形成鮮明對比，因為 RNN 和 LSTM 需要逐步處理序列中的每個單詞，難以並行化，而 Transformer 能夠同時處理整個序列中的所有單詞，極大地提高了計算效率。

Transformer 架構由多層的編碼器 (Encoder) 和解碼器 (Decoder) 組成。編碼器負責接收輸入序列，通過多層的自我注意力機制和前饋神經網絡來提取特徵，並生成內部表示。

解碼器則在生成輸出時，利用與編碼器的聯繫，逐步生成最終輸出序列。這種結構使 Transformer 能夠靈活應對機器翻譯、文本生成等任務，並能同時處理輸入與輸出的依賴關係。

與傳統模型相比，Transformer 有許多顯著優勢。首先，因為它不依賴逐步處理的方式，它可以實現並行計算，大大加快了訓練速度，尤其是在處理長序列數據時。其次，由於自注意力機制能夠捕捉到序列中任意位置的依賴關係，Transformer 在處理長距離依賴方面表現優異。再者，Transformer 中的「多頭注意力機制」允許模型在多個不同的子空間中學習不同的語意表示，從而使模型能夠更好地理解文本中的複雜結構和多層次含義。

然而，Transformer 並非沒有缺點。由於其高度依賴自注意力機制和多層結構，Transformer 對計算資源的需求非常高，尤其是當模型規模擴大時，訓練過程需要大量的 GPU 或 TPU 資源，這使得訓練和部署大型模型的成本變得極高。此外，儘管 Transformer 可以捕捉長距離依賴，但其架構本身並沒有內在的序列性，這意味著模型在處理具有強序列關聯的數據時（如時間序列）可能會表現出一定的劣勢。因此，研究者引入了位置編碼（Positional Encoding）來幫助模型學習序列信息，但這種方法在某些情況下仍然無法完全彌補缺乏序列性的問題。

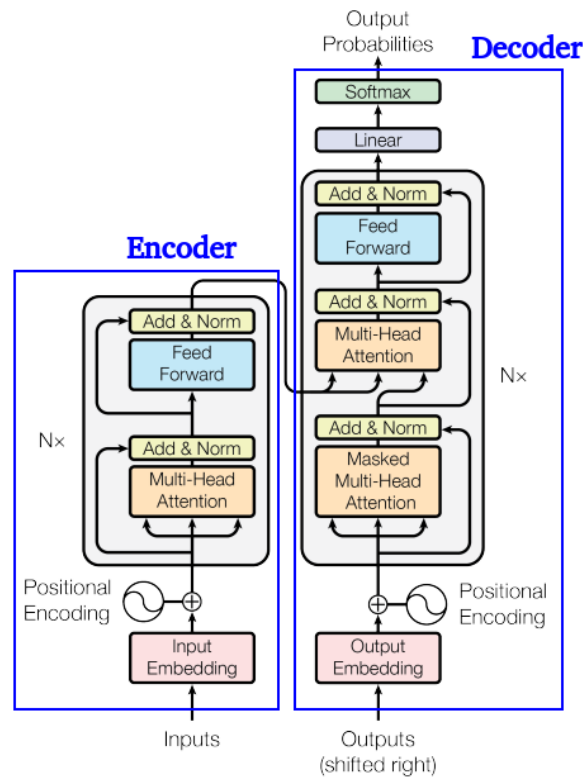


圖 2. 1：Transformer 架構圖[3]

## 2.2 Fine-tuning

微調（Fine-tuning）[7]是一種深度學習中的技術，通過對已經預訓練過的大型模型進行特定任務的調整，來達到更高的應用性能。這種方法的出現源自於預訓練模型的發展，尤其是在自然語言處理（NLP）和電腦視覺等領域。大型預訓練模型，如 BERT、GPT 等，通常會在龐大的通用數據集上訓練，獲得了廣泛的語言理解或影像識別能力。然而，這些模型最初並不是針對具體應用場景進行優化的，因此，它們在面對特定任務時可能無法充分發揮出最佳效果。

微調的核心思想是，利用這些已經訓練好的模型作為基礎，進行額外的小範圍調整，來適應具體的任務需求。具體來說，首先會在大量的通用數據上對模型進行預訓練，使模型學會捕捉廣泛的語言或圖像特徵。接著，當面對一個特定任務（如情感分析、文本分類

或特定圖像識別)時,研究者會使用與這個任務相關的數據集對預訓練模型進行微調。這個過程不需要像預訓練階段那樣耗費大量的計算資源,因為預訓練模型已經學會了基本的特徵表示,微調僅是對這些表示進行任務特定的調整。

微調的過程通常涉及到對模型的部分參數進行訓練,這樣可以在保持預訓練模型已經學到的知識的同時,讓模型適應新的任務。例如,在自然語言處理中,一個語言模型可能已經在大量文本上學會了語法結構和語義表示,但它可能不熟悉某個特定領域的專業術語或具體需求。透過微調,模型可以利用這些預先學到的知識,並根據新數據集的特定需求進行針對性學習,從而大幅提升在特定任務中的表現。

微調的優勢在於,它避免了從零開始訓練模型的高昂成本,並充分利用了預訓練模型的強大能力。同時,由於預訓練階段已經捕捉到了大量通用特徵,微調通常只需要少量的任務相關數據即可產生良好的效果。這使得微調成為許多實際應用中的理想方法,特別是在數據量有限的情況下,仍然能夠從大型模型中獲益。

這種技術在許多應用領域得到了廣泛使用,無論是文本分類、機器翻譯還是圖像識別,都可以通過微調現有的預訓練模型來快速獲得精確的結果。這使得微調成為現代深度學習模型應用中的關鍵步驟,不僅提高了模型的實用性,還加速了 AI 技術的發展與部署。

## 2.3 GPT-SoVITS(GSV)

GSV 是融合了 GPT (Generative Pre-trained Transformer)模型,和 SoVITS (Speech-to-Video Voice Transformation System)變聲器技術,在最近由中國作者開源的語音合成方法,用以實現高品質的語音複製和文字到語音轉換(TTS)[6]。

在閱讀相關論文後[5]，我們考慮到針對中文使用者，同時為了更好的整合 API，採用了由 python 開發的 GPT-SoVITS。

GPT-SoVITS 的主要功能如下：

- 少樣本 TTS 文字到語音轉換：用 1 分鐘的訓練資料，對模型進行精細調整，從而提高聲音的相似度和真實感。
- 聲音複製：透過訓練，GPT-SoVITS 能學會並複製特定說話人的聲音特徵，然後生成與特定說話人聲音非常像的合成語音。
- 跨語言支援：GPT-SoVITS 支援多種語言的語音合成，讓使用者能在各種語言環境中靈活使用這個工具。目前已經支援英語、日語和中文三種語言。
- Webui 工具：整合了包括聲音伴奏分離、自動訓練集分割、中文 ASR (自動語音識別)和文字標註等實用工具，給初學者建立訓練資料集和 GPT/SoVITS 模型帶來很大方便。

可以預見，在未來 GSV 將可應用的地方包含個性化語音助手、虛擬角色配音、有聲讀物製作，甚至是無障礙服務等相關領域。



## 2.4 現有 AI 聊天 app 比較

	Wysa	Nova	Replica	戀AI物語
功能	文字聊天	文字聊天	文字聊天	文字,語音聊天, 可調整個性,動態調 整背景音樂
語言	英文	中,英,日,韓	中,英	中,英,日,韓
可用模型	<b>1</b> (內建)	<b>1</b> (內建)	<b>1</b> (內建)	<b>2</b> (Chat-GPT,Claude)

圖 2.2：各 app 比較圖

## 第三章、研究方法

本專題旨在設計並實作一個具多模態交互功能的網頁介面，透過結合文本、語音與圖片處理能力，實現使用者與人工智慧大模型之間的自然互動。研究方法分為系統架構設計、技術實現、模型微調、以及功能測試與驗證四個部分。

### 3.1 系統架構設計

在本專題中，系統架構設計分為前端與後端：

- **前端：**使用 HTML、CSS、JavaScript 開發網頁，並引入 Live2D 角色模型，增強互動性和視覺效果。前端介面負責處理使用者輸入，包括文字輸入、語音錄製與圖片上傳，並根據使用者的需求播放回應的背景音樂及 AI 聲音。
- **後端：**後端主要使用 Python 開發，並整合語言與語音處理 API，負責接收前端傳來的數據，調用大語言模型 API，並將模型生成的回應轉換為文字與語音形式傳回前端。

## 戀AI物語

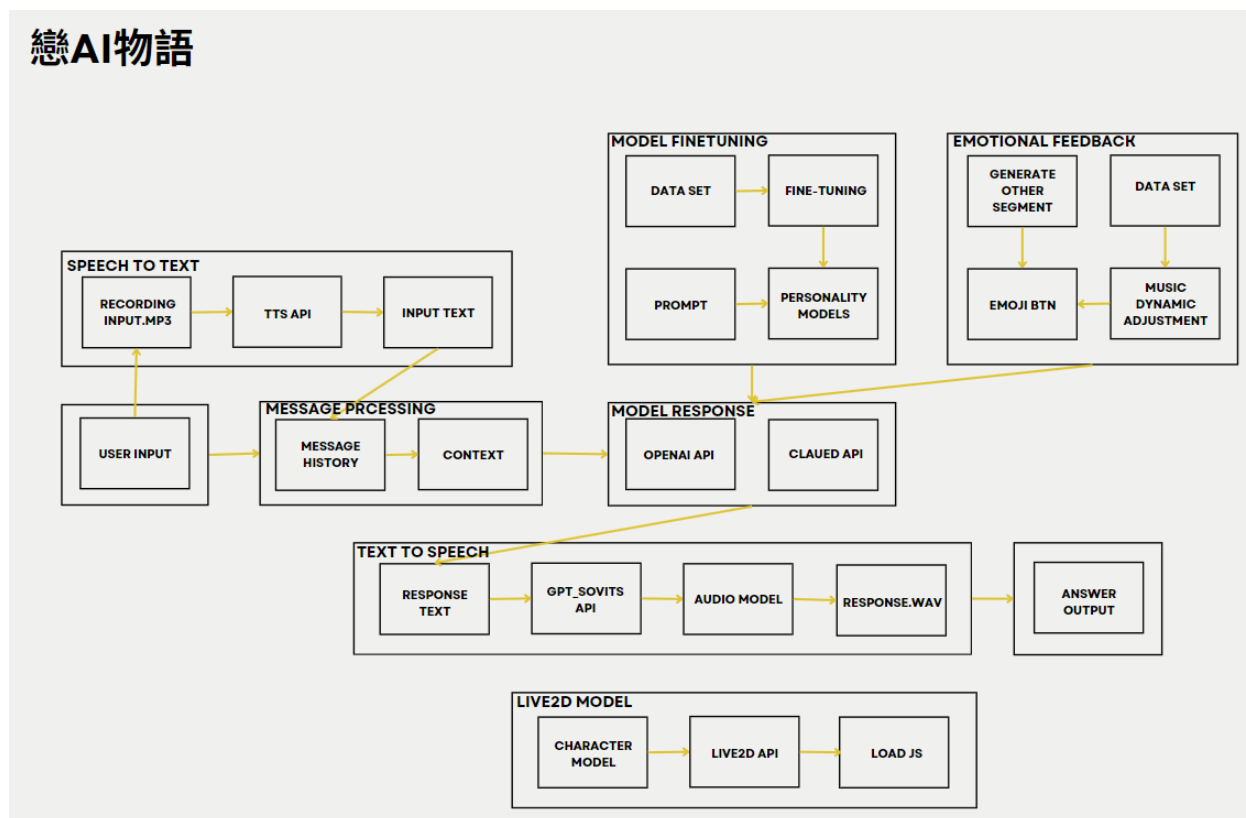


圖 3.1：系統架構圖

## 3.2 技術實現

- STT 與 TTS 處理：**本專題採用專業的語音轉文字（STT）和文字轉語音（TTS）API[9]，確保語音輸入與輸出具備高準確性與自然性。語音輸入會被即時轉換為文字，並作為大語言模型的輸入之一。模型的文字回應則經由 TTS 技術生成多樣化的語音輸出。
- 文本與圖片處理：**使用 JavaScript 與 Python 協同處理文本與圖片資料。文本輸入直接傳送至後端進行處理，圖片則會經由前端預處理後傳送至後端，調用大語言模型 API 進行分析。
- 背景音樂與情境互動：**設計了多種背景音樂，以增強聊天過程中的情境體驗。背景音樂會根據不同的互動情境自動切換或由使用者自選播放。

## 3.3 模型微調

為了使 AI 角色具備多樣化的語音和個性，我們進行了以下微調工作：

### 3.3.1 語音模型設定

使用專門的語音合成工具 GPT-SoVITS，對多個聲音模型進行微調，使不同 AI 角色能夠擁有獨特且自然的聲音。這包括調整語速、音調與情緒參數，讓聲音表現更加生動。

在參考官方說明後進行實作[10]，我們將語音模型的訓練分為兩個部分，前者是資料集的準備及處理，後者是模型的訓練和推理。

**資料集處理：**

Step 1：準備 1 至 10 分鐘的音檔，以此為想要訓練出該聲音的資料集。使用 GPT-

SoVITS 的優勢在於：只需要少量的資料集即可訓練。但仍須注重音檔的品質，避免雜音影響訓練。

Step 2：對資料集進行切割、標注以及校對。切割資料集將整個音檔的文本分割成多份，讓模型更好的針對個別語句學習；標注和校對是為了糾正語音轉文字時可能出現的錯誤，需要仔細對分割後的語句進行勘誤，因此 GPT-SoVITS 的訓練是需要人工監督的。

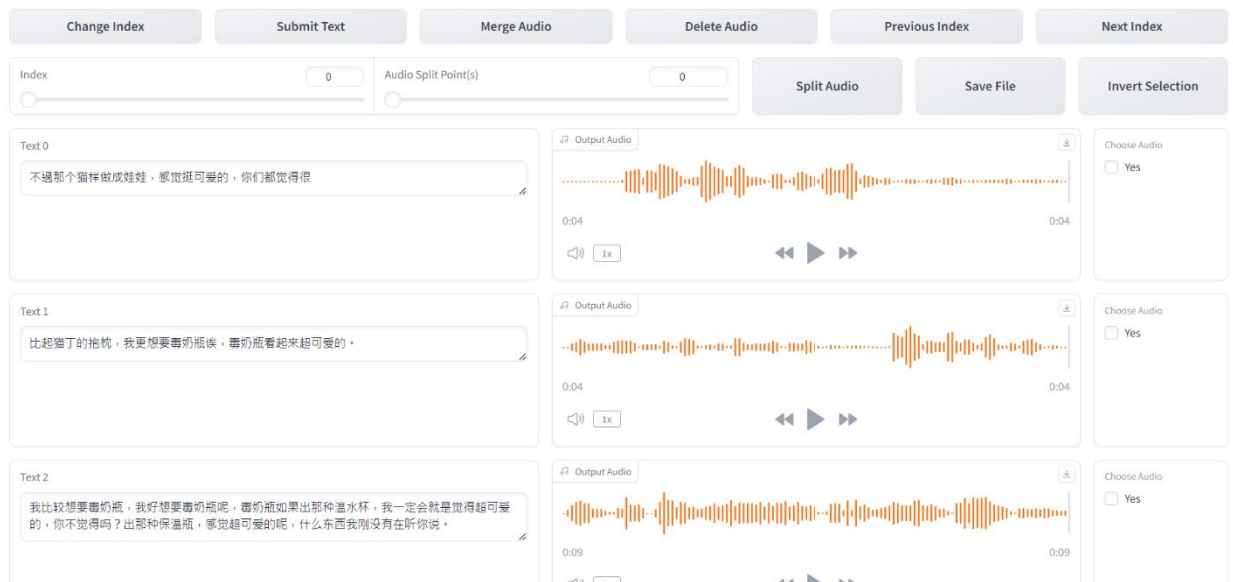


圖 3.2：校對畫面

### 模型訓練和推理：

Step 3：在資料集處理完畢後，將其餵給模型訓練。在此我們分別設定 Sovits 和 GPT

的超參數為：

表 3.1：語音模型超參數設定

	Batch size	Epoch	Learning rate
Sovits	7	8	0.4
GPT	7	15	0.4

The interface is divided into three main tabs: 0-Fetch dataset, 1-GPT-SOVITS-TTS, and 2-GPT-SOVITS-Voice Changer. The top section contains fields for Experiment/model name, GPU Information (0 Tesla T4), and three Pretrained model paths (SoVITS-G, SoVITS-D, and GPT). Below this, the 1A-Dataset formatting tab is active. The 1B-Fine-tuned training section is highlighted with a red dashed border and contains two sub-sections: 1Ba-SoVITS training and 1Bb-GPT training. Each sub-section has its own set of parameters (Batch size, Total epochs, Text model learning rate, Save frequency, etc.) and a large orange button to start training. The 1C-Inference tab is also visible at the bottom.

圖 3.3：模型微調介面

Step 4：最後，使用訓練好的模型進行推理，並把輸出的音檔放入實驗中。

參考圖 3.4，將推理分成四個部分：

1. 模型列表：選用已訓練的目標模型。
2. 參考文本：放上參考音頻以及該音頻的文本。參考音頻擔任推理中的重要腳色。

訓練好的模型可以模擬 AI 音色，參考音頻則決定了 AI 說話的情緒。

3. 合成文本：輸入要說出的文本，可以選擇切分方法，較長的句子採用不同的切分方法可以讓 AI 斷句時有更好的表現。

4. 輸出音頻：推理最後輸出的音檔。



圖 3.4：語音模型推理界面

### 3.3.2 語言模型微調

根據角色設定進行大語言模型的微調[11]，確保每個 AI 角色具備特定的說話風格與個性特徵，如幽默風趣、冷靜理智或熱情奔放等。

#### 1. 準備數據集

**數據集準備**是微調過程中的關鍵步驟。數據集的質量會直接影響模型的最終性能。

這一步主要包括：

- **數據收集**：收集相關的文本資料，這些文本應涵蓋希望模型學習的知識或語氣風格，格式如圖 3.5。例如，對話數據、技術文檔、專業領域的資料等。

#### 2. 設定模型與微調參數

- **預載模型**：從預訓練的 GPT 模型開始，如 gpt-4o-mini，該模型已經具備基本的語言理解能力，並且可以在特定應用下進行微調。
- **設置訓練超參數**：包括學習率、批次大小（batch size）、訓練輪數（epochs）等。這些參數需要根據硬件性能和數據集大小調整。

### 3. 微調過程

- **數據加載與預處理**：將文本數據轉換為模型可以理解的格式，通常是將文本轉換為標記（tokens）。

#### 3.3.3 訓練過程

1. 將訓練數據輸入模型，進行前向傳播（forward pass），生成預測結果。
2. 計算損失函數（如交叉熵損失）來衡量模型輸出的質量。
3. 執行反向傳播（backpropagation）來更新模型的參數。

```
1 [{"messages": [{"role": "system", "content": "你是一個性格文靜且靦腆的女朋友"}, {"role": "user", "content": "你今天怎麼樣?"}, {"role": "assistant", "content": "還好，想你。"}]]  
2 [{"messages": [{"role": "system", "content": "你是一個性格文靜且靦腆的女朋友"}, {"role": "user", "content": "你喜歡吃什麼?"}, {"role": "assistant", "content": "巧克力蛋糕。"}]]  
3 [{"messages": [{"role": "system", "content": "你是一個性格文靜且靦腆的女朋友"}, {"role": "user", "content": "我們今天出去玩嗎?"}, {"role": "assistant", "content": "可以啊。"}]]  
4 [{"messages": [{"role": "system", "content": "你是一個性格文靜且靦腆的女朋友"}, {"role": "user", "content": "你今天穿什麼?"}, {"role": "assistant", "content": "裙子，還有圍巾。"}]]  
5 [{"messages": [{"role": "system", "content": "你是一個性格文靜且靦腆的女朋友"}, {"role": "user", "content": "你喜歡看電影嗎?"}, {"role": "assistant", "content": "嗯，喜歡。"}]]
```

圖 3.5：資料集格式

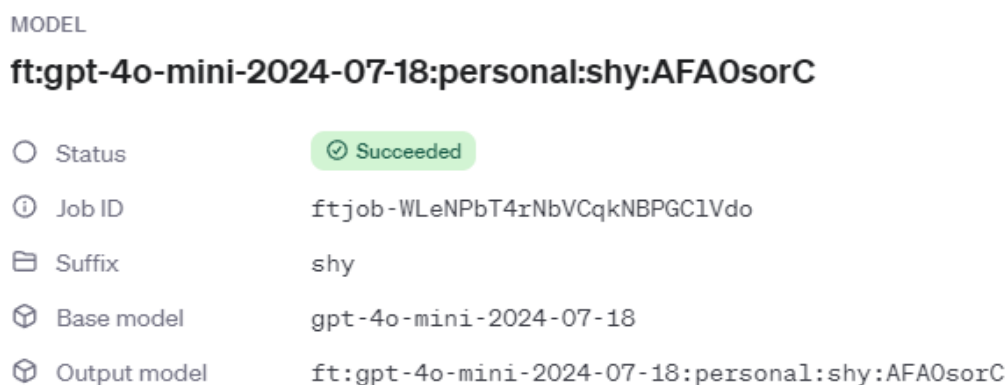


圖 3.6：微調語言模型[7]

## 3.4 提升互動性

### 3.4.1 人物模型

將 live2d 模型至於網頁上讓用戶更有與人互動的感覺，過程如下：

1. 工具準備



- **Live2D Cubism SDK**：Live2D 官網提供的開發套件[8]。這個 SDK 為 Web 提供了一個 JavaScript 庫 `live2dcubismframework.js`，用來控制 Live2D 模型。
- **模型文件**：使用 Live2D Cubism 軟體設計並導出 `.moc3` 文件、貼圖（`.png`）、表情設定（`.exp3.json`）、動畫（`.motion3.json`）等。這些文件會用來在網頁上呈現 Live2D 模型。

## 2. 網頁設計

- **HTML**：頁面結構設置，通常包括一個 `canvas` 元素來顯示 Live2D 模型。
  - **CSS**：美化頁面樣式，調整模型顯示位置等。
  - **JavaScript**：加載和控制 Live2D 模型。
3. 實現眼睛隨滑鼠移動：根據滑鼠和模型的相對位置調整 Live2D 模型的參數，用於控制眼睛的移動。
  4. 調整模型大小與位置：可以透過 Live2D SDK 提供的方法來調整模型的縮放與位置，使其在不同解析度下適配網頁。

### 3.4.2 情緒回應與背景音樂


#### 1. 功能設計理念


這個功能的核心目標是提升用戶與 chatbot 之間的互動性和情感交流體驗。每當機器人給出回覆時，用戶可以通過點擊對應的表情符號來快速表達自己的情緒或回饋。例如，當用戶對某則回覆感到滿意時，可以點擊「笑臉」或「大拇指」符號；若回覆不符合預期，則可以選擇「困惑」或「不滿」表情。這樣的設計不僅使用戶


能夠在不影響對話流程的情況下提供即時反饋，還幫助開發者收集有價值的用戶情緒數據，以優化機器人未來的回答表現。

## 2. 功能設計結構

- **表情符號選項：**每則機器人回覆下方都設置了一組常見的表情符號，例如：

（開心）：用戶對機器人的回答感到滿意或愉快。

（不滿）：用戶對回答感到不滿，期待更好的內容或反應。

（困惑）：用戶覺得回答令人困惑，可能需要更多的說明或改善。

- **設計考量：**

使用簡單而直觀的符號，讓用戶可以迅速理解並做出選擇。

每組表情符號的排列方式應合理分布，避免讓用戶感到選擇困難。

3. JavaScript 處理回饋：使用 JavaScript 編寫函數，當用戶點擊表情符號時，會切換目前的背景音樂並與機器人回覆關聯。

## 第四章、實驗結果與討論

### 4.1 實驗結果



圖 4.1：介面及功能說明

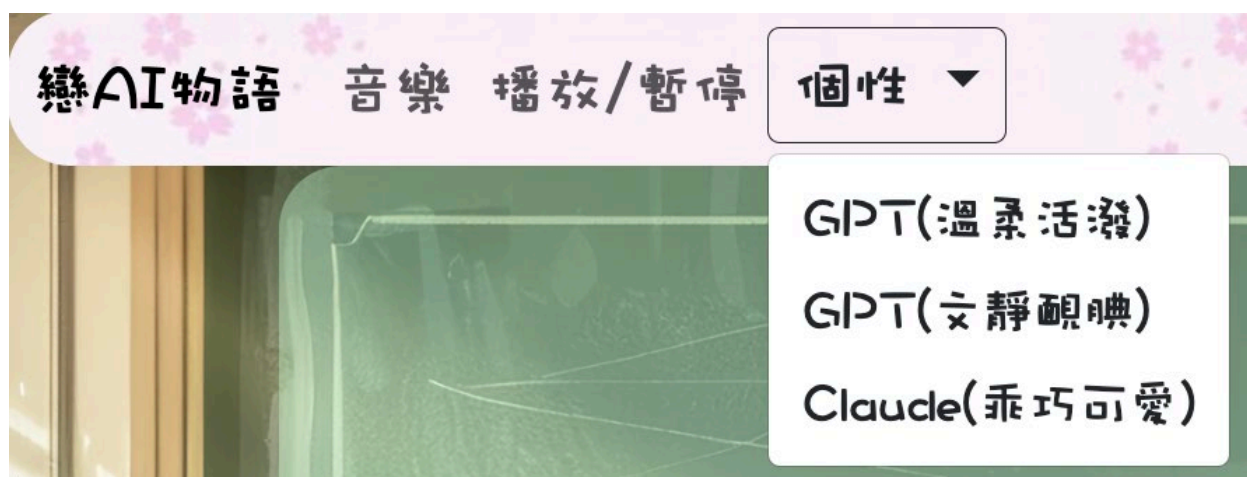


圖 4.2：切換模型選單



圖 4.3：live 2d 模型和情緒回應



圖 4.4：對話範例一

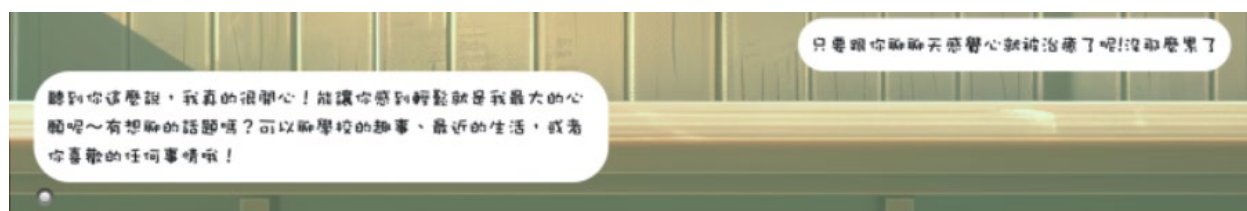


圖 4.5：對話範例二

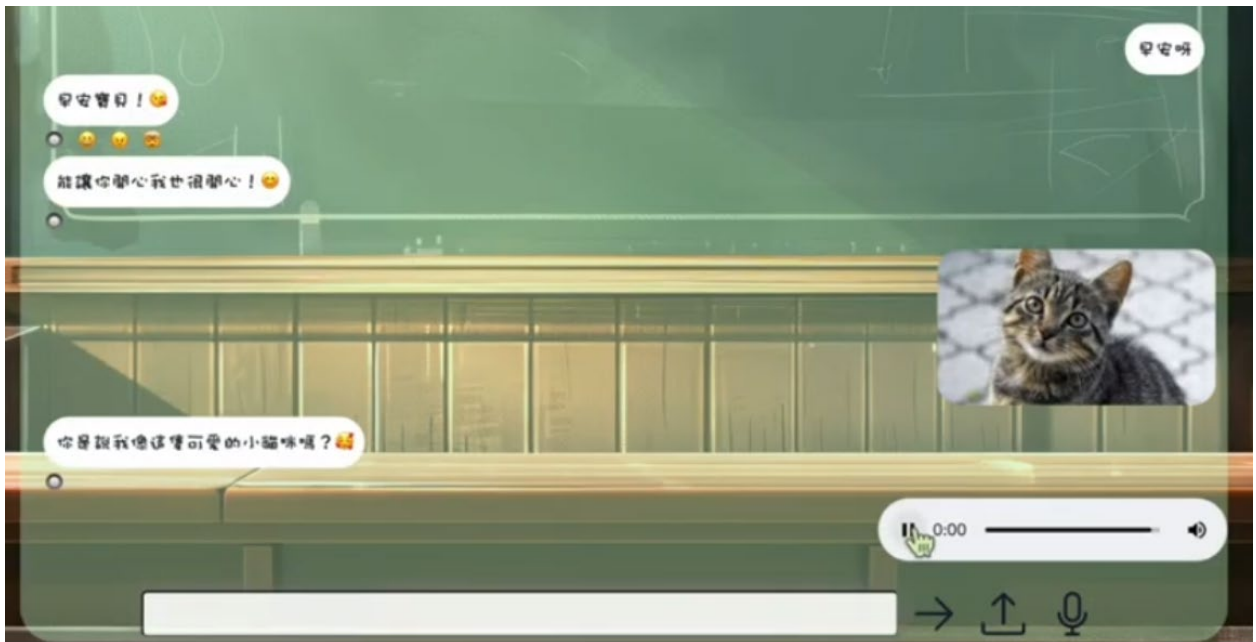


圖 4.6：可使用圖片及語音聊天

## 4.2 技術實作與挑戰

我們的實驗過程主要集中在模型的微調和 API 的集成，整個開發流程分為幾個關鍵階段，並且我們在每個階段遇到不同的挑戰和解決方法：

- **語言模型微調：**我們使用現有的語言模型，如 ChatGPT 和 Claude，對其進行微調，使其能夠輸出更符合特定角色個性的回應。微調過程中，我們針對角色設定（如幽默、冷靜或熱情）選擇合適的訓練語料，並運用數次迭代調整模型參數，以達到更自然且一致的語言風格。
- **語音合成 API 集成：**使用文字轉語音（TTS）API，我們將模型的文字回應轉換為多樣化的語音輸出。我們嘗試了不同的聲音模型，並對語速、音調和情感參數進行調整，以確保聲音更加自然和符合角色設定。
- **挑戰：**語音合成的自然度和情感表達難以平衡，尤其在高情感變化的場景下。部分 API 生成的聲音在特定語境下會顯得不夠生動，因此我們優化了參數設置並探索新的開源工具以改善表現。

## 4.3 系統性能觀察

雖然我們沒有進行正式的性能測試或用戶評估，但在開發過程中，我們注意到幾種情況：

- **語音合成性能**：模型能夠即時生成語音，延遲在可接受範圍內，但高情感變化語音的品質在部分情境下仍有待改善。
- **語言模型表現**：模型回應的自然度和一致性有所提升，尤其在符合特定性格方面，但某些複雜語境仍存在語言邏輯不夠連貫的問題。

## 第五章、結論與未來展望

### 5.1 結論

本研究成功將微調後的 LLM 模型與 live2D 動畫及語音合成技術集成，開發出一個多模態互動系統。雖然未進行正式的用戶測試，但從開發觀察來看，系統能夠提供自然的語音輸出和生動的虛擬角色互動，展示了初步的可行性和技術潛力。這些技術的集成為未來在娛樂、教育和心理支持等領域的應用提供了有力的支持。

### 5.2 目前挑戰與未來展望

目前的專題面臨著幾項挑戰：

1. **資料集的不足**：在本專題之前並未有針對個性的語言資料集，因此暫時只能透過 GPT 只需要少量資料集的優點，我們自己提供訓練資料。但若是能增加資料集的內容，想必可以讓對話更多元。
2. **模型切換**：目前我們已訓練多個個型的語言及語音模型，但由於在切換模型時，不同的 API 有所衝突，因此無法完整的同時將語言和語音模型進行切換。若能完整將切換模型加入介面，可以給使用者帶來更多的方便跟體驗。

而未來的研究和開發可以著重以下幾個方向：

- **正式性能測試與優化**：設計專門的性能測試計劃，量化系統的延遲、語音品質和語言模型回應的自然度，進一步優化模型和 API 集成的效率。
- **用戶體驗評估**：邀請用戶參與系統評估，收集反饋以改進互動設計，特別是 live2D 角色的情感表現和語音合成的生動性。

- **語氣情緒判斷與回應處理：**若是能先分析用戶語氣情緒，透過切換參考音頻的方式，以合適的情緒回復，就能讓對話更貼近真實，同時生成的音頻也能以該語氣呈現。
- **語音合成技術改進：**開發或採用更先進的語音合成技術，進一步提升情感表達的真實性和自然度。
- **改進音樂切換與情緒回復：**透過判斷語氣情緒來合理的切換音樂，達到讓用戶放鬆情緒，而不只是透過表情符號切換。而採用情緒回復時若能用更生動的方式呈現，也有助於提升聊天的趣味性。
- **擴展應用場景：**研究如何將系統應用於具體場景，如教育中的個性化教學輔助，或心理支持服務中能提供情感化安慰的虛擬角色，擴大技術影響力。

這些改進將進一步增強系統的 UX 與實用性，推動人機互動技術向更自然、更具情感的方向發展。



## 參考文獻

- [1] Zaremba, A., & Demir, E. (2023). ChatGPT : Unlocking the future of NLP in finance. *Modern Finance*, 1(1), 93-98.
- [2] Alto, V. (2023). *Modern Generative AI with ChatGPT and OpenAI Models : Leverage the capabilities of OpenAI's LLM for productivity and innovation with GPT3 and GPT4*. Packt Publishing Ltd.
- [3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [4] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D : Nonlinear Phenomena*, 404, 132306.
- [5] Danylov, V. (2024). OPEN SOURCE AND PROPRIETARY SOFTWARE FOR AUDIO DEEPFAKES AND VOICE CLONING : GROWTH AREAS, PAIN POINTS, FUTURE INFLUENCE. *Baltic Journal of Legal and Social Sciences*, (1), 105-113
- [6] Xue, L., Soong, F. K., Zhang, S., & Xie, L. (2022). Paratts : Learning linguistic and prosodic cross-sentence information in paragraph-based tts. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30, 2854-2864.
- [7] Api Reference: <https://platform.openai.com/docs/api-reference/introduction>
- [8] Live2D Cubism Tutorials: <https://docs.live2d.com/en/cubism-editor-tutorials/top/>
- [9] Text to speech: <https://platform.openai.com/docs/guides/text-to-speech>
- [10] GPT-SoVITS 指南: <https://www.yuque.com/baicaigongchang1145haoyuangong/ib3gle>
- [11] Fine-tuning GPT Assistants with OpenAI: <https://blog.weblab.technology/fine-tuning-gpt-assistants-with-openai-ec83b6d35006>