

# 中華民國第 65 屆中小學科學展覽會

## 作品說明書

---

高級中等學校組 電腦與資訊學科

052506

跨語言交流與環境感知的智能眼鏡—EchoLens

學校名稱： 臺北市立內湖高級工業職業學校

作者：	指導老師：
職二 顏吉良	陳昭安
職二 蔡言紘	王宛琦
職二 林晉緯	

關鍵詞： 人工智能、即時翻譯、天氣預報

## 摘要

本研究提出一款結合生成式 AI 能進行場景分析，進而綜合情緒、語氣、文化禮節給予最佳回覆的人工智慧眼鏡。目的希望使用者不需要透過手機即可實現即時語音翻譯、情境分析與天氣資訊抓取功能。系統以 ESP32 為主要控制核心，搭配 Android Studio 開發之應用程式，透過 WiFi 採用 UDP 通訊協定進行無線通訊，提供精簡的翻譯與應答建議。每次翻譯處理後，GPT 根據語境與提示詞再將翻譯結果與應答建議，傳送回並顯示於智慧眼鏡的 OLED 螢幕，將即時資訊呈現在使用者眼前，實現流暢的跨語言互動體驗。

## 壹、前言

### 一、研究動機

在一次國外旅行中透過 Google 翻譯順利完成與外國人的溝通，雖然過程最終達成目的，但反覆使用手機進行語音輸入及操作按鈕，導致對話中出現多次停頓，雙方需禮貌地等待彼此回應，造成一定程度的不便與溝通障礙。此經驗引發筆者思考：若能將翻譯內容直接顯示於視線前方，是否能有效提升跨語言對話的流暢度與便利性？

為實踐此構想在高一的行動裝置應用實習課程中，利用 Android Studio 開發一款可連接 Google Translate API 的中英文翻譯應用程式；高二的專題實作課程則進一步結合 Google Cardboard、ESP32 與 OLED 顯示器，製作出第一代智慧型眼鏡。系統架構上，語音訊息經手機翻譯後，透過 UDP 協定傳送至 ESP32，再由 OLED 顯示結果，讓使用者得以即時觀看翻譯內容。

在實際測試中發現，部分翻譯內容出現語意不清或難以理解的情形。進一步比較後發現，相較於傳統翻譯工具，ChatGPT 等大型語言模型（Large Language Model, LLM）不僅可提供翻譯結果，亦能針對語境進行說明與解釋。這顯示語言的理解不僅止於字面翻譯，更涉及語言結構、文化差異、溝通情境與情緒氛圍等層面。「語言的社會文化規則對於母語使用者而言為直覺反應，然而對外語學習者則需長時間觀察與學習」（陳麗娟, 2015）。此一嘗試讓我們意識到語言的理解不僅止於字面翻譯，更涉及語言結構、文化差異、溝通情境與情緒氛圍等層面。



圖 1 英文 break a leg 在字面上的意義，會誤會成斷一條腿，但實際上英文是祝您好運。(Chat GPT 生成)

因此，本研究旨在結合智慧型眼鏡與 OpenAI 所提供的語言模型技術，透過語音與影像資訊判讀當前環境與語境，產生更貼近實際情況的翻譯與應對建議，並即時於眼鏡前端 OLED 顯示器呈現。此裝置期望協助使用者完成一段自然流暢、符合語境的跨語言溝通過程，進而提升其外語應用信心與學習動機，達到輕鬆跨語言互動的目標。

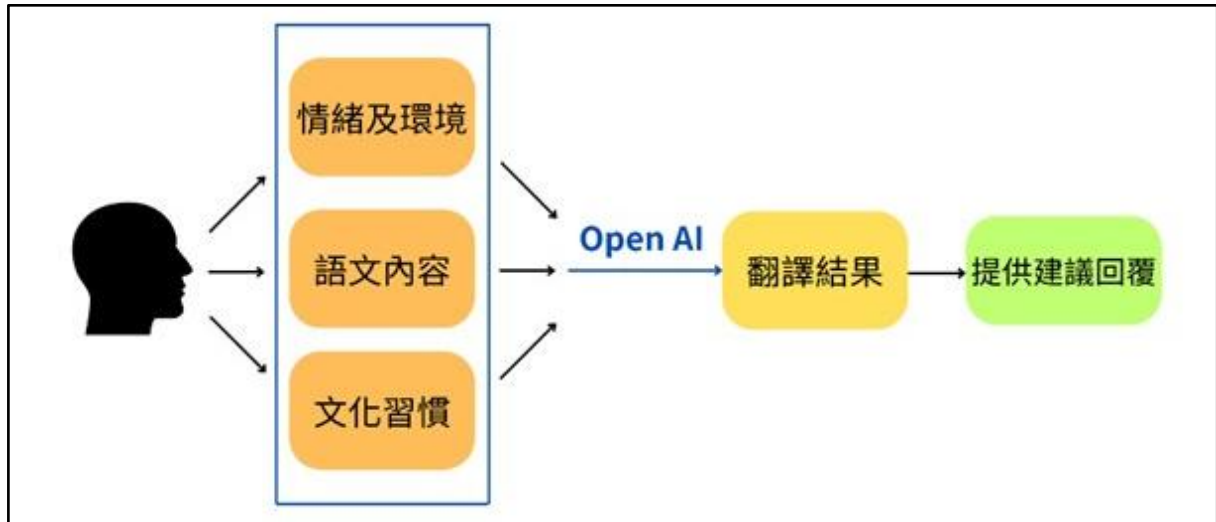


圖 2:Open AI 綜合情緒環境、語文內容及文化習慣產生翻譯結果及回覆建議  
(研究者自繪)

## 二、研究目的

本研究旨在透過整合多項人工智慧與物聯網技術，開發一款具備語音翻譯、情境分析及即時顯示功能的智慧型眼鏡系統，以提升跨語言溝通效率與互動品質。具體研究目的如下：

- (一) 使用 Android Studio 建構測試 Google Translate API 及 GPT API 的語言翻譯 App
- (二) 應用提示工程 (Prompt Engineering) 強化 AI 語言理解與回應品質。
- (三) 結合 ESP32 與 OLED 顯示模組實現穿戴裝置應用
- (四) 整合氣象 API 提供情境感知功能。
- (五) 測試整翻譯正確性及系統穩定性。

### 三、文獻回顧

#### (一) 翻譯學

在進行語言翻譯時，有幾個基本的要點像是：情緒因素、文化背景、語句結構、專業術語等等。必須需要多項影響準確度的因素才能做到正確翻譯。(eTranslation Services,2023)

1. 文化敏感性是翻譯工作的核心之一。語言背後往往隱含著特定文化脈絡與情感，如果沒有經過了解，容易導致誤解。例如成語與俚語在不同語言中未必有對等表達，譯者需靈活採用意譯或文化對等轉換，以傳達原文真正的意圖與語氣。
2. 語言結構的差異也是翻譯過程中常見的問題。不同語言之間的語法規則和語序安排可能會有差異，若採用逐字翻譯，往往會產生生硬、不自然的錯誤句子。
3. 對於技術文件與專業資料而言，專業術語的準確性格外重要。不當的術語使用可能造成誤解，甚至帶來法律或操作風險。因此，譯者不僅需具備該領域基本知識，也應參考官方詞彙表（如 Google Glossary 功能）或行業慣例，確保用詞正確且一致。(萊克伍德,2022)

#### (二) Google Translation API

經由 Google Cloud 以神經機器翻譯（Neural Machine Translation, NMT）為基礎的商業翻譯服務。允許開發者將文字翻譯為超過 100 種語言的語言，適用於網站、應用程式或其他軟體系統。

主要 Google Translation API 的功能如下表所示:

表 1:Google Translation API 功能表(Google Cloud. 2025)

功能名稱	功能說明	適用場景
<b>translateText</b>	翻譯輸入文字為目標語言， 可自動偵測來源語言	智慧眼鏡、翻譯 App
<b>detectLanguage</b>	自動辨識文字內容所使用的	語音翻譯前語言辨識

	語言	
<b>getSupportedLanguages</b>	回傳所有支援的語言列表	語言選單、國際化介面
<b>translateDocument</b>	同步翻譯 PDF、Word 等文件	文件閱讀、翻譯工具
<b>batchTranslateText</b>	非同步翻譯大量文字資料 (支援雲端儲存輸入/輸出)	大型客服資料、自動化處理
<b>batchTranslateDocument</b>	非同步翻譯大量文件	檔案管理系統
<b>romanizeText</b>	將非拉丁文字轉換為拼音 (如拼音、羅馬字)	教學系統、語音朗讀
<b>Glossaries</b>	自訂術語表，提升領域翻譯 準確性(如醫療、工程)	專業領域系統翻譯
<b>Adaptive MT</b>	訓練專屬翻譯模型，可根據 自備語料提升準確率	企業品牌專屬翻譯風格
<b>Operations</b>	查詢、等待或取消長時間操 作(如批次翻譯)	任務管理、非同步任務追蹤

### (三) ChatGPT OpenAI API

ChatGPT OpenAI API：OpenAI 提供的 Chat Completion API 支援自然語言對話，透過上下文記憶生成情境化回應，適合本研究的跨語言翻譯需求。Whisper API 將語音轉為文字並支援多語言辨識，確保語音輸入的準確性。Vision API 則透過圖像分析提供環境情境理解，輔助生成符合語境的翻譯與建議（OpenAI, 2023）。

1. Whisper API（語音辨識）：Whisper API 是一套語音轉文字的服務，可將語音檔案（如:mp3、m4a、wav）轉換為對應的文字內容，該系統內建多語言自動辨識功能，亦能直接將輸入語音翻譯成英文，Whisper 對長錄音與環境雜音具良好容錯能力。

2. **Vision API（圖像理解）**：Vision API 為 GPT-4o 模型新增的功能，支援圖文並陳的輸入格式，可對圖片進行辨識與分析。該服務可識別圖像內容、物件、場景，並解析圖中的文字、表格與視覺結構。此能力使其適合應用於智慧助理、視障輔具、圖像問答系統及多模態互動平台，強化人機互動的感知能力。
3. **Embedding API（語意向量化）**：Embedding API 可將文字資料轉換為高維數值向量，保留語意特徵並可用於語意比較與匹配。此技術常應用於搜尋引擎、推薦系統、問句相似性比對與資料分群分析。透過語意嵌入的方式，機器能以向量方式「理解」語言內容間的語意距離，進而提供更精準的語意搜尋與內容配對。

#### (四) 提示工程 (Prompt)

1. 透過 API（Application Programming Interface）與大型語言模型互動已成為開發應用與系統整合的主流方式。特別是在 OpenAI 所推出的 ChatGPT API 或其他 LLM API 服務中，開發者透過精心設計的提示詞（Prompt）撰寫，可引導模型產出預期回應或特定格式資料，這樣的過程即為「提示詞工程（Prompt Engineering）」。
2. 相較於一般使用者透過圖形介面輸入文字指令，開發者更常在應用程式中撰寫 JSON 格式的 API 請求，並內嵌精煉的提示詞，引導模型做出目標導向的回應。因此 API 的提示詞設計需同時考慮三大要素：語意精準、格式明確與結果穩定。
3. 為了使大型語言模型能在不同情境下發揮最大效益，開發者需進行提示詞的微調與測試，包含指令語氣、角色扮演、情境說明、輸出格式等變項的搭配與調整。尤其在自動化應用、機器控制或資料處理上，更需明確標示結構化的輸出格式，例如要求回應以 JSON、XML 或特定欄位對應的格式呈現，以利後端程式自動解析與執行。

提示工程：透過設計提示詞引導大型語言模型產出預期回應，需確保語意精準、格式明確與結果穩定。本研究採用零樣本與單樣本提示策略，結合角色扮演與格式引導，實現智能眼鏡的即時翻譯與回應生成（Alibaba Cloud, n.d.）。

##### (1) 提示功能導向 (Prompt Functional Purposes)

根據常見的應用需求，提示功能可分為以下類型：

- A. **生成（Generate）**：產生文字、回覆、對話、文章，用於內容創作、自動回應。

- B. 摘要 (Summarize)：擷取內容重點，濃縮文本，應用於新聞摘要、會議記錄等。
- C. 改寫 (Rewrite)：轉換語氣或風格，如正式化、口語化、翻譯。
- D. 資訊抽取 (Extract)：從文本中擷取特定資料，如數據、關鍵字、事件、角色等。
- E. 分類 (Classify)：依據語意將文本歸類至既定標籤，例如情感分類、主題分類。
- F. 群聚 (Cluster)：將多筆資料依語意相似性進行分群，常見於評論分析與主題探索。
- G. 相似度查詢 (Search / Similarity)：比對語句與語料的語意相似度，用於搜尋與推薦。(Microsoft, 2023)

## (2) 提示策略導向 (Prompting Strategies)

在上述功能應用中，開發者可依任務特性選擇不同的提示策略來提升效能，常見策略包含：


- A. 零樣本提示 (Zero-shot Prompting)：僅提供任務說明，不附範例，適用於模型已具備任務知識的情境。
- B. 單樣本提示 (One-shot Prompting)：提供範例以利模型模仿任務格式與語氣。
- C. 多樣本提示 (Few-shot Prompting)：提供兩個以上範例，幫助模型學習更穩定的輸入與輸出邏輯。
- D. 鏈式思考提示 (Chain-of-Thought, CoT)：引導模型逐步推理、逐層說明中間思考過程，適用於邏輯與計算任務。
- E. 角色扮演提示 (Role Prompting)：要求模型扮演特定角色（如客服、老師、律師），以提供風格化、情境化回應。
- F. 格式引導提示 (Schema Prompting)：明確規範輸出格式（如欄位名稱、JSON 結構），常用於系統整合任務。

## 貳、研究設備及器材

研究使用的設備與器材分類說明如下：

### 一、硬體

表 2:智慧眼鏡總元件(研究者自繪)

微控制器	電源供應	顯示螢幕	眼鏡框架
			
ESP32 C3	鋰電池 18650	0.91 吋 OLED SSD1306	Google CardBaord

### 二、軟體

#### (一) 系統平台

1. Chat GPT 建立 API 將接收到的資訊在這邊進行處理在將結果回傳回去。
2. Google Cloud 用來儲存運算前跟運算後的資料。
3. Blender 是一個功能全面、靈活且易於入門的 3D 創作工具，我們將來會使用他來設計眼鏡外觀。

#### (二) 程式開發環境與語言

1. ESP32 的網路連線及 OLED 的顯示都是使用 Arduino IDE 做為開發環境，使用 C++ 進行撰寫。
2. 手機應用程式是透過安裝 Android studio 的程式來執行的，可使用內部按鈕來執行。



## 參、研究方法和過程

規劃本作品主要分為跨語言翻譯 App，關於可綜合外國語文語境、語用的 GPT 的提示工程以生成合理三大部分進行設計：

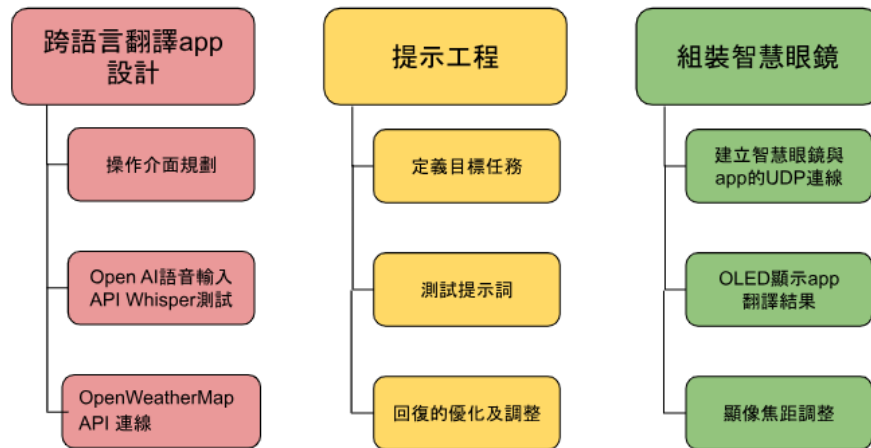


圖 3: 研究流程圖(研究者自繪)

### 一、GPT 智慧翻譯眼鏡規劃及設計

本研究將設計一套以 ChatGPT 的大型語言為基礎的智慧翻譯眼鏡。智慧眼鏡透過無線網路連接一套專用的 App，App 可接收語音後，再進行語音轉成文字，文字內容透過 GPT API 發送到 ChatGPT 進行翻譯，App 在接收到翻譯結果後，隨即傳回眼鏡上一個小型的 OLED 顯示幕，呈現翻譯的結果，使用者佩戴眼鏡可以再如下圖的位置上直接看到翻譯結果。

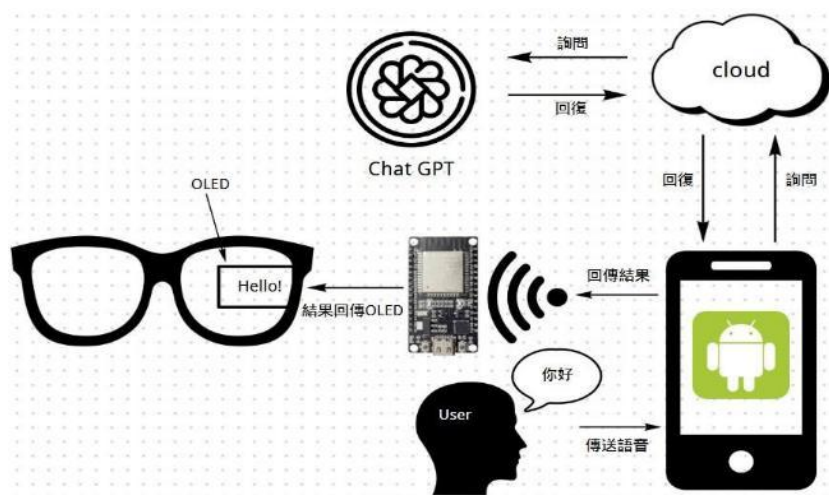


圖 4: 使用 GPT 翻譯的智慧翻譯眼鏡的運作架構(研究者自繪)

## (一) 跨語言翻譯 App 設計

### 1. 介面設計

為順利使用 Android Studio 設計翻譯 App，將不同語言翻譯成指定的語言，設計 App 操作介面如下：

- (1) 網路連線：設置 IP 及 Port 的文字輸入方塊，讓 App 執行 UDP 協定與智慧眼鏡的 ESP32 控制器連線。

- (2) 記憶語言：按下記憶語言按鈕後，會記憶最近一次按下「翻譯」按鈕所收錄的語言。記憶後接下來按下「翻譯」所收錄的任何國家語言，都會翻成該語言。如：按「翻譯」說中文：「翻譯成中文」，第一次翻譯會翻譯成「Translate to Chinese」，當按下「記憶按鈕」後，再按下「翻譯」收錄對方對方的語音講：「How are you?」，則會依照記憶語言自動翻譯成中文「你好嗎？」。

- (3) 清除語言：將記憶的語言清除，恢復成翻譯成英文。

- (4) 獲取天氣：取得所在地點當地的天氣資訊。

- (5) 翻譯：長按這個按鈕收錄語音，將任何語言翻譯成英文。搭配「記憶語言」按鈕，則翻譯成被記憶的語言。

- (6) 中文翻譯：長按這個按鈕收錄語音，將任何語言翻譯成中文。

- (7) GPT 圖形分析：分析接收的照片情境。

### 2. 網路連接

網路使用 UDP 通訊協定進行連接，主要是溝通手機 App 與 ESP32 微控制器之間的連線，連線電路及關係如下圖所示。



圖 5:手機 APP 介面(研究者拍攝)

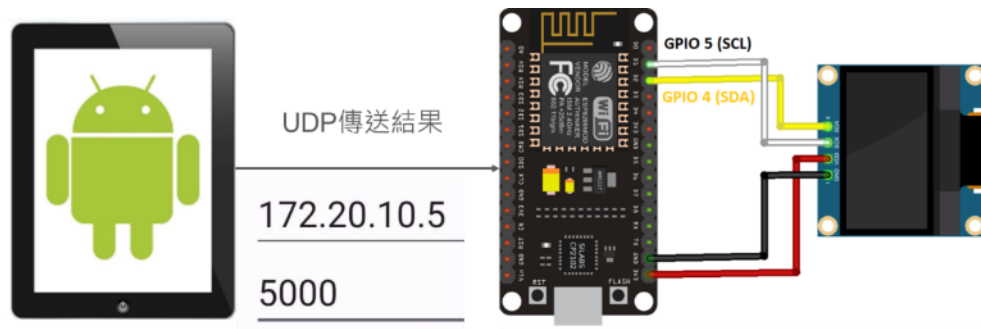


圖 6:元件接線圖(研究者自繪)

(1) App 端連線設計：

```
private void sendUdpMessage(final String message) {
    new Thread() -> {
        try {
            if (esp32Ip == null || esp32Ip.isEmpty()) return;

            DatagramSocket socket = new DatagramSocket();
            socket.setBroadcast(true); // 可選
            InetAddress address = InetAddress.getByName(esp32Ip);
            byte[] buffer = message.getBytes();
            DatagramPacket packet = new DatagramPacket(buffer, buffer.length, address, esp32Port);
            socket.send(packet);
            socket.close();
        }
    };
}
```

圖 7:App 資料透過 UDP 傳送到 ESP32 (研究者拍攝)

設計說明如下：

(1) ESP32 端連線設計：

程式引入了幾個必要的函式庫，包括控制 ESP32 Wi-Fi 功能、處理 UDP 通訊、顯示圖形以及音訊處理，設定 Wi-Fi 的連線參數，其中 ssid 和 password 分別指定了要連接的無線網路名稱與密碼。ESP32 啟動後會嘗試連上這個網路，以便進行網路通訊。

(2) UDP 通訊的設定部分：

udpTargetIP 是目標裝置（例如 Android 手機）的 IP 位址，ESP32 將資料傳送到這個位址上。udpPort 是對方裝置監聽的端口號，而 localPort 則是 ESP32 自己用來接收 UDP 資料的端口這些參數允許雙向資料的傳輸，例如將翻譯或語音辨識的結果從 ESP32 傳送到手機，或由手機傳回命令給 ESP32。

### 3. 語音接收、翻譯及回覆設計

使用 Android Studio 設計於 App，主要是在按下翻譯按鈕，可以接收外界的音訊，程式使用 OpenAI 提供的 Whisper 深度學習模型模型，將收錄的音訊轉換成文字，重點是這個模型可以同時判斷語言。

- (1) 收錄音訊辨識及轉換文字：自動語言辨識和語音轉文字，使用 File 物件開啟以 UUID 命名的音訊檔，使用 MediaRecorder 物件進行錄製，程式如下。

```
private void startRecording() {
    try {
        File outputDir = getCacheDir();
        audioFile = new File(outputDir, child: UUID.randomUUID().toString() + ".m4a");
        recorder = new MediaRecorder();
        recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
        recorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
        recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AAC);
        recorder.setOutputFile(audioFile.getAbsolutePath());
        currentAudioPath = audioFile.getAbsolutePath();
        recorder.prepare();
        recorder.start();
    } catch (Exception e) {
        e.printStackTrace();
        runOnUiThread() -> outputText.setText("！ 錄音啟動失敗");
    }
}
```

圖 8:收音訊辨識及轉換文字(研究者拍攝)

- (2) 自動辨識語音和記憶語言：將 Whisper 語音轉文字結果傳送至 GPT 模型，判斷語句所屬語言代碼（如：zh、en、ja），並回傳給主程式。用於記憶非中文語言，程式如下：

```
private String detectLanguageByGPT(String text) {
    try {
        OkHttpClient client = new OkHttpClient();

        JSONObject body = new JSONObject();
        body.put("model", "gpt-3.5-turbo");

        JSONArray messages = new JSONArray();
        messages.put(new JSONObject().put("role", "system")
            .put("content", "請判斷使用者輸入的文字是哪一種語言，只回答語言代碼（例如：en、ja、ko、fr、de）；不要解釋。"));
        messages.put(new JSONObject().put("role", "user").put("content", text));
        body.put("messages", messages);

        Request request = new Request.Builder()
            .url("https://api.openai.com/v1/chat/completions")
            .post(RequestBody.create(body.toString(), MediaType.get("application/json")))
            .addHeader("Authorization", "Bearer sk-proj--JdohoCybu0Yv6INtj_DIdbCPa0QooLnSSiNRiRN6x1lv7LPBUSK")
            .addHeader("Content-Type", "application/json")
            .build();
    }
```

圖 9:語言辨識及記憶(研究者拍攝)

#### 4. UDP 連線及測試

(1) ESP32 端 UDP 接收程式設計: 實現 ESP32 透過 Wi-Fi 連線並接收 UDP 封包的功能，與手機 App 或其他裝置進行資料交換，像是接收翻譯結果、語音命令等。程式引入 WiFi.h 和 WiFiUdp.h 兩個函式庫，指定本地的 UDP 接收 Port 定義為 5000，這個 Port 會在 setup() 階段啟動並持續監聽外部訊息。

(2)App 端 UDP 發送程式設計: App 的 Wi-Fi 連接和發送數據，採用 Android Studio 設計的工作流程

A. getUserInput() 讀取用戶輸入

通過 ipEditText 和 portEditText 獲取 ESP32 的 IP 地址和端口號。

B. 創建 DatagramSocket (UDP Socket)

```
DatagramSocket socket = new DatagramSocket();
```

C. 允許進行 UDP 廣播 (可選)

```
socket.setBroadcast(true);
```

D. 發送 UDP 數據包

```
DatagramPacket packet =  
new DatagramPacket(buffer, buffer.length, address, esp32Port);
```

E. 將數據發送到 ESP32

```
socket.send(packet);
```

#### 5. 天氣感測實作

使用 OkHttpClient 建立並發送請求以取得天氣資料，將 API 回傳的 JSON 資料進行解析，從而提取溫度、濕度等數據，並組合成一個字串。將取得的天氣資料透過 UDP 傳送到 ESP32，並更新 UI 顯示天氣資訊。也就是當使用者按下 weatherButton 時會先取得輸入資訊，再呼叫這個方法來調用 API 抓取天氣資料。

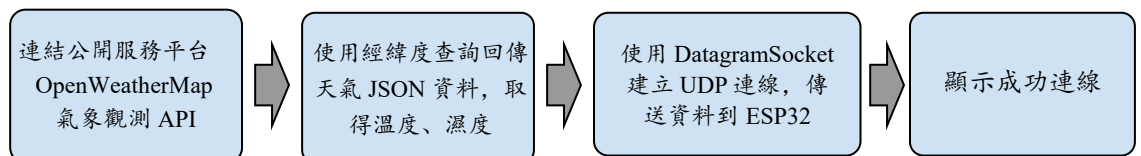


圖 10:天氣資訊抓取流程圖(研究者自繪)

(1) 氣象 API 連線：

```
String url = "API 連接的網址，使用者輸入緯度、經度等參數";  
OkHttpClient client = new OkHttpClient();  
Request request = new Request.Builder().url(url).build();
```

(2) 天氣的 JSON 資料：連線氣象 API 後，回傳該緯度、經度地天氣資訊。

```
"current": {  
  "feelsLike": {  
    "unit": "°C",  
    "value": "27"  
  },  
  "humidity": {  
    "unit": "%",  
    "value": "76"  
  },  
  "pressure": {  
    "unit": "hPa",  
    "value": "1013"  
  },  
  "pubTime": "2025-04-07T14:36:00+08:00",  
  "temperature": {  
    "unit": "°C",  
    "value": "30"  
  },  
  "windSpeed": {  
    "unit": "km/h",  
    "value": "12"  
  }  
},  
"location": {  
  "lat": "25.08",  
  "lon": "121.58"  
}
```

经121.58 纬25.08|温度30°C|湿度76%|  
★ 情境建議：  
早上好！氣溫已經達到 30 度，濕度也比較高，建議注意補充水分，適時避免暴曬，注意保濕喔！祝您有個愉快的一天！

圖 11:json 格式天氣資訊(研究者拍攝) 圖 12:app 顯示天氣資訊及建議(研究者拍攝)

(3) 傳送到 ESP32：於 App 端將接受的 JSON 資料傳送到 ESP32。

```
DatagramSocket socket = new DatagramSocket();  
socket.setBroadcast(true); // 允許廣播  
InetAddress address = InetAddress.getByName(esp32Ip);  
byte[] buffer = message.getBytes();  
DatagramPacket packet =  
    new DatagramPacket(buffer, buffer.length,  
        address, esp32Port);  
socket.send(packet);
```

(4) OLED 顯示氣象資訊：



圖 13:螢幕顯示天氣資訊(研究者拍攝)



## 6. 智慧眼鏡硬體設計

本專題以使用者說出語音開始，經由手機收音後將音訊內容傳至雲端，在藉由 Open AI 的 API 傳給 Chat Gpt 進行翻譯，將結果回傳至手機，透過 WI-FI 傳給 ESP32，並將翻譯結果傳至 OLED 顯示螢幕上，讓使用者可以透過光學鏡片看到近距離的螢幕顯示內容。

本作品使用了 Google cardboard 做為外觀，但因為元件大小關係，我們決定對他做擴充，內部包含了 ESP32、OLED ssd1306、充電式鋰電池 18650、電池座和光學透鏡，實際分布圖如下：



圖 14:內部元件分布圖(研究者自繪)

## 7. 肉眼觀看焦距測試

人的眼睛在極小的距離下觀看是無法對焦的，人眼的最短對焦距離是 10~15cm，這個距離不可能放置在眼鏡內部，所以我們使用了 45mm 焦距透鏡，讓使用者可以透過這個鏡片來看到極小距離的螢幕顯示內容。

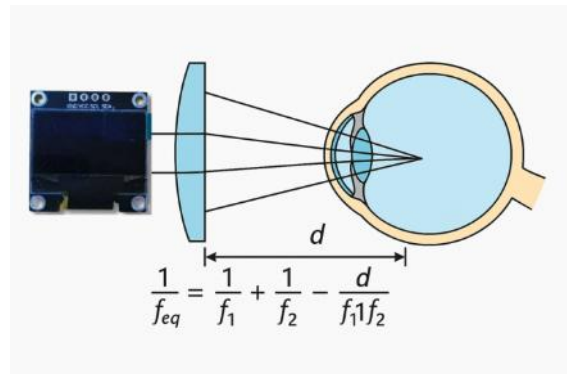


圖 15: 透過透鏡觀看示意圖(AI 生成)

表 3: 焦距計算及測試數據(研究者自繪)

鏡片焦距  $f_1(\text{mm})$ : 45mm 眼鏡焦距  $f_2(\text{mm})$ : 22mm

鏡片與眼睛距離 $d(\text{mm})$	等效焦距 $f_{eq}(\text{mm})$	觀看清晰度
5	15.97	最低
10	17.37	低
15	19.04	中等
20	21.06	次高
25	23.57	高

#### 8. 圖像環境情境分析

本研究的另外一個構想，是希望可以讓眼鏡擷取以前的影像，再交給 GPT 解析影像的環境場域、情境氛圍，已得到最合適的翻譯及建議回覆的內容。使用 Android Studio 設計流程如下：

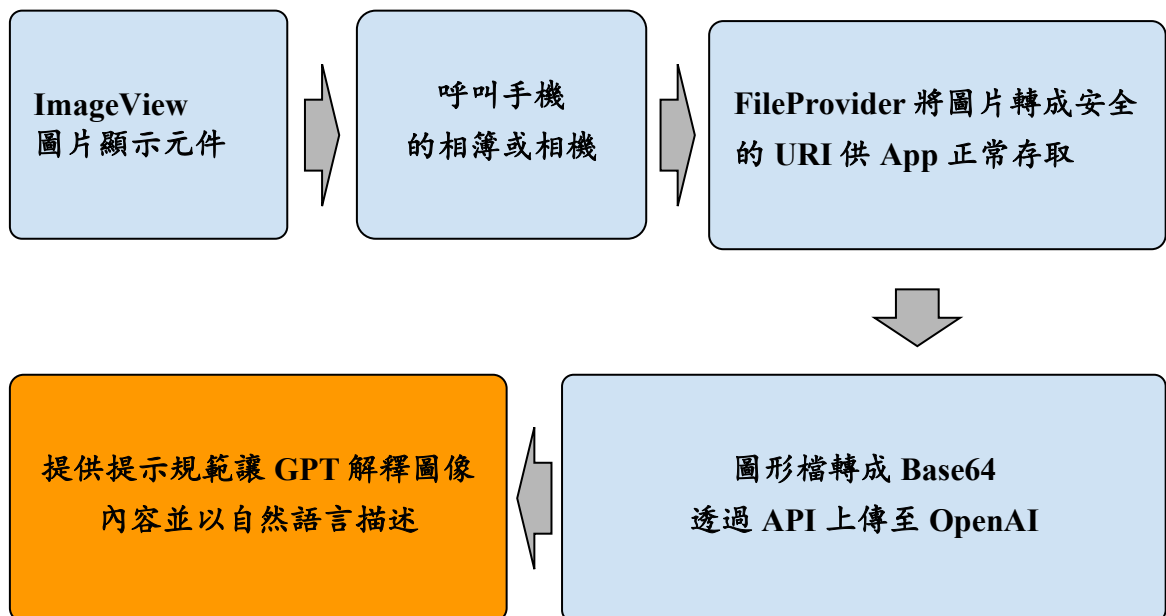


圖 16: 圖像環境情境分析流程(研究者自繪)

(1)圖片取得與顯示：使用者透過手機的相機或相簿選擇圖片，並將其載入 App 中的 ImageView 預覽元件中，提供直觀的使用體驗。



- (2)檔案安全管理：為符合 Android 安全規範，系統用 FileProvider 將圖片轉換成安全的 URI，使 App 在不同模組或權限範圍內皆可安全讀取圖片檔案。
- (3)圖像轉碼與上傳：圖片檔案會轉換為 Base64 編碼格式，便於透過 HTTP 或 API 傳送至伺服器或雲端模型，避免檔案直接傳輸帶來的格式不兼容或存取限制。
- (4)AI 圖像分析：伺服器端透過 GPT-4o 等支援 Vision 的模型，依據設定的提示語（prompt）來辨識圖片內容，並以自然語言生成對圖片的說明、敘述或判讀結果。

## 二、提示工程

研究設計跨語文交流、環境認知，也可以當日氣象輔佐產生環境情境，設計了 9 組 GPT 提示詞 (Prompt)，讓 ChatGPT 根據口語的語音敘述，自動執行對應任務。在硬體規格、語言翻譯、情緒分析、建議生成，運用提示語導引導 AI 生成符合問題的自然回應，讓 AI 更理解你的語氣、意圖與情境，以貼近人類思維的對話體驗。

表 4:提示詞設計原則(研究者自繪)

中文 → 記憶語言（或英文）	情緒判斷（表層）	透過文句直接猜測情緒（英文）
1. 明確指定語言。 2. 用「只輸出」簡潔輸出。 3. 適合用小螢幕 OLED 展示。	1. 定義模型為心理專家。 2. 以「簡短回答一種情緒」避免輸出冗長分析。 3. 定義適合快速語音播報。	1. 加入同理心避免命令語氣。 2. 「You might」「It could be」溫和的語句。 3. 搭配翻譯功能譯成對方語言。
外語 → 中文翻譯	情緒判斷（心理層面）	情緒安慰（心理建議）
1.指定「繁體中文」台灣用語，避免簡體或港式翻法。 2.加上「不要加任何說明」避免「翻譯如下」的冗字。	1.適合用於需要更嚴謹心理觀點的模式下。 2.與前一個相比，語意更完整，有學術感。 3.避免 GPT 回覆「我認為使用者可能是...」這類格式，使判斷內容更加深層。	1.適合用於日常語句的情緒撫慰與支持場景。 2.提示語強調「溫暖」「正向」「不帶命令語氣」，讓使用者感到被理解與安慰。 3.回覆為一句輕柔建議，提升使用者心理舒適感。

ChatGPT 問答（原語言回答）	情境式天氣建議	圖片分析
1.簡潔自然，讓 ChatGPT 回應時自動 <b>使用偵測語言</b> （Whisper 偵測出來的）。 2. <b>適合多語情境</b> ：講日文回日文、講韓文回韓文等。 3.搭配 Whisper 語音偵測效果最佳。	1.每日提醒、早安問候或穿搭建議等生活情境下。 2.透過時間、天氣與地理資訊動態生成建議，讓內容具即時性與個人化。 3.語氣自然親切，強調「 <b>繁體中文口語化</b> 」，提升使用者親切度與信賴感。	1.適合用於物品辨識、場景描述、視覺輔助等場景。 2.提示語採「 <b>繁體中文簡要說明</b> 」風格，避免回覆過於技術或制式。 3.可應用於拍照即分析功能，作為多模態（語音+圖像）應用的一環。

## 六、Google 及 OpenAI 的翻譯 API 結果比較

考驗 Google 及 OpenAI 的翻譯能力，實作如下：

1. 在 ChatGPT 以「**英文特殊用語或俚語，容易翻譯錯誤的詞句，請列出並提供例句**」詢問。
2. 將條列的例句，提供給 Google 及 OpenAI API 翻譯結果比較。

採用 Google API 進行翻譯會直譯而產生語意的偏差，OPEN AI 翻譯的結果則正確。

表 5:OPEN AI API 和 GOOGLE API 翻譯對比(研究者自繪)

語句(粗體為片語)	OPEN AI API 翻譯	GOOGLE API 翻譯
<i>You have a big performance tonight—<b>break a leg!</b></i>	你今晚有重要的演出—祝你好運！	今晚你有精彩的表演—打斷腿！
That designer bag <b>cost me an arm and a leg!</b>	那個名牌包超貴的！	那個名牌包花了我一條手臂和一條腿！
I've done my part. Now <b>the ball's in your court.</b>	我該做的做完了，接下來看你了。	我已經盡了我的職責。現在球在你的球場上。
Are you serious, or are you just <b>pulling my leg?</b>	你是認真的，還是在跟我開玩笑？	你是認真的還是只是在開我的玩笑？



## 肆、研究結果

### 一、功能介紹

本專題主要的功能有天氣資訊抓取，英文翻譯，多語文翻譯，圖像分析及記憶語言功能，以下是成果展示及說明：

(一) 天氣資訊抓取:使用 API 連線 OpenWeatherMap 網站並透過經緯度回傳 json 格式的溫溼度資訊，將資訊傳給 Open AI 分析再將結果回傳並由螢幕進行顯示。

表 6:天氣資訊 App 及螢幕顯示(研究者自繪)

App 顯示	OLED 顯示
	

(二) 英文翻譯:使用者說出英文後，手機進行收音並透過大型語言模型中的 Whisper 自動辨識語言並將音訊內容轉成文字，再透過 API 傳給 Open AI 進行翻譯並生成回覆建議，最後再將結果回傳至螢幕顯示。

(三) 多語文翻譯:使用者說出除了英文外的語言，透過 Whisper 自動辨識語言後傳給 Open AI 進行分析並產生回覆建議。

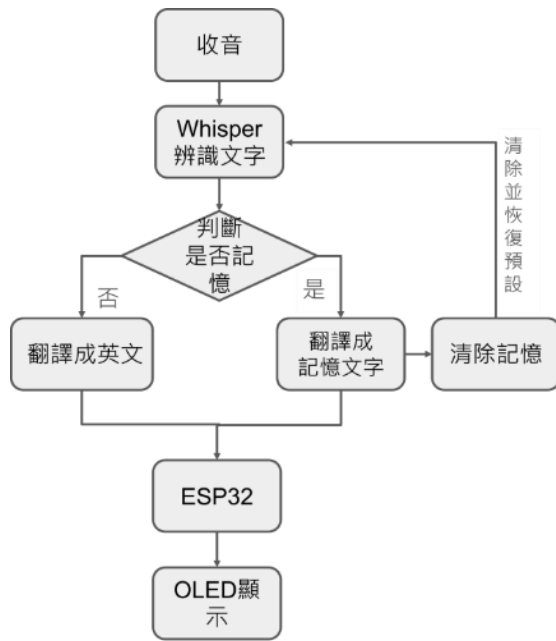


圖 17:記憶功能流程圖 (研究者自繪)



圖 18:記憶功能 app 介面(研究者拍攝)

(四) 記憶語言功能:透過 button 記憶使用者當前說的語言，將之後收錄的音訊直接翻譯成記憶中的語言。例如:使用者說出日文並按下記憶，使用者之後不管說了甚麼都會翻譯成記憶中的日文直到清除記憶。

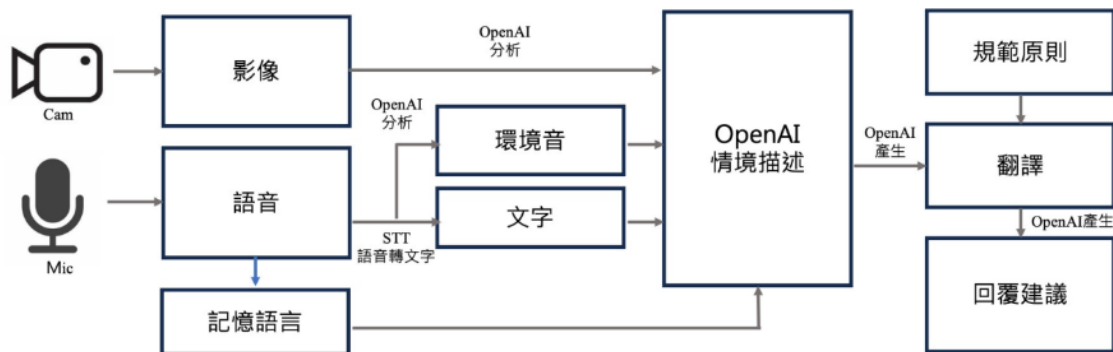


圖 19:語音和影像分析流程(研究者自繪)

(五) 圖像分析:使用者透過手機拍照或從相簿選擇照片，FileProvider 會將圖片轉換成安全的 URI 以提供 App 存取，存取後的圖片檔案會轉換為 Base64 編碼格式，便於透過 HTTP 或 API 傳送至伺服器或雲端模型，避免檔案直接傳輸帶來的格式不兼容或存取限制，伺服器端透過 Open AI 依據設定的提示語來辨識圖片內容，並以自然語言生成對圖片的說明、敘述或判讀結果。



圖 20:圖像分析 app 顯示(研究者拍攝)

## 二、雙人對話流程

### (一) 單人配戴智能眼鏡溝通模式

外國人說出英文，眼鏡配戴者收音後螢幕顯示中文並生成英文回復，再由配戴者回覆英文

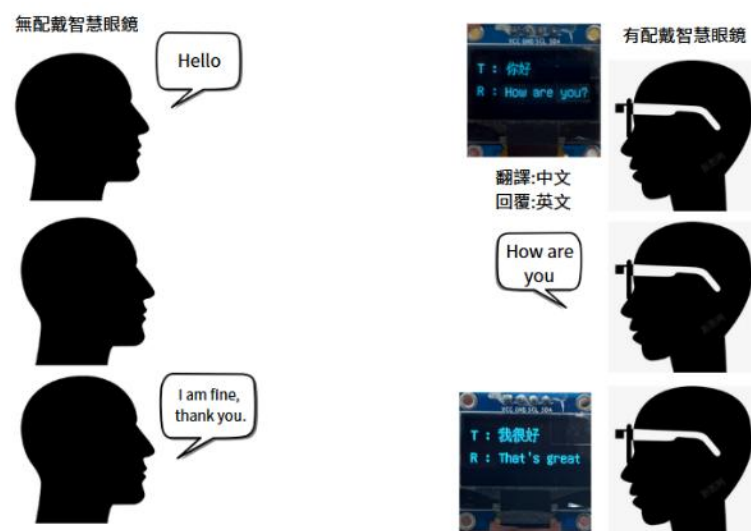


圖 21: 單人配戴智能眼鏡溝通示意(研究者自繪)

## (二) 雙人配戴智能眼鏡溝通模式

雙方皆配戴眼鏡由一方說出自己的母語後。另一方接收後產生自己母語的翻譯結果和回覆。

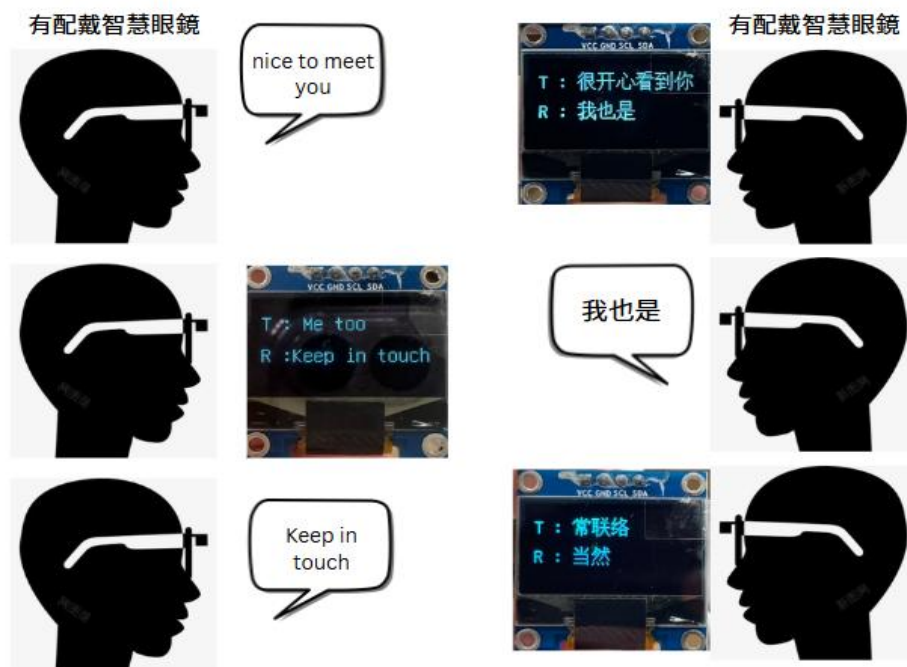


圖 22:雙人配戴智能眼鏡溝通(研究者自繪)

## 三、EdgeCortex SAKURA-II 晶片科技展現場模擬

地點：晶片科技展現場

角色一：對方是外國參觀者（說英文）

角色二：你是展區主持人

以下訓練內容為使用這在科展時遇到專業用詞的場景模擬訓練

表 7: 晶片科技展專業術語模擬對話(研究者自繪)

外國參觀者	Does the SAKURA-II PCIe card include onboard DRAM, or does it rely on host memory?
GPT 翻譯	SAKURA-II PCIe 卡上有內建 DRAM 還是依賴主機記憶體？
GPT 回覆	It includes onboard LPDDR4 DRAM—16GB on the single-chip module and 32GB on the dual-chip variant.
外國參觀者	What's the peak AI compute performance in INT8 and BF16?



GPT 翻譯	它在 INT8 和 BF16 模式下的 AI 運算效能分別是多少？
GPT 回覆	You'll get up to 60 TOPS in INT8 and 30 TFLOPS in BF16 mode.
外國參觀者	Which AI frameworks are supported by the SAKURA-II SDK?
GPT 翻譯	SAKURA-II 的 SDK 支援哪些 AI 框架？
GPT 回覆	It supports PyTorch, TensorFlow, ONNX, and includes a compiler for automatic graph optimization.
外國參觀者	What's the typical inference latency for a model like YOLOv5?
GPT 翻譯	像 YOLOv5 這類模型，在這顆晶片上的推論延遲大概是多久？
GPT 回覆	Around 3 to 5 milliseconds per frame, depending on the model size and input resolution.
外國參觀者	What kind of power consumption should we expect during sustained AI inference?
GPT 翻譯	在持續執行 AI 推論時，大約的功耗表現如何？
GPT 回覆	It consumes about 10W in the single-chip config, and 20W in the dual-chip config under standard workloads.
外國參觀者	Is PCIe Gen 3.0 x8 sufficient for full bandwidth utilization?
GPT 翻譯	PCIe Gen 3.0 x8 的頻寬是否足夠發揮完整效能？
GPT 回覆	Yes, it's been tested to deliver optimal throughput for real-time multi-stream inference applications.
外國參觀者	Which industries are currently integrating the SAKURA-II into their edge platforms?
GPT 翻譯	目前有哪些產業已經將 SAKURA-II 整合到他們的邊緣平台中？
GPT 回覆	Industries like autonomous systems, defense, smart cities, and high-performance robotics are adopting it for real-time AI at the edge.

#### 四、美式派對笑話模擬

本實驗為了模擬到了外國可能聽不懂當地的流行語或笑話，導致場面尷尬或冷場。

表 8:美式派對對話模擬(研究者自繪)

朋友	Don't you think this skeleton costume looks insanely real? I'm totally killing it tonight!
GPT	You're definitely killing something.

朋友	Hey! I'm the coolest skeleton at this party. Wanna see me throw down some bone-crunching moves?
GPT	Yeah? But do you know why skeletons don't fight each other?
朋友	Uh... why?
GPT	Because they don't have the guts.
朋友	Dude! That was hilariously bone-dry!
GPT	I'm the backbone of bad puns, what can I say.

笑點說明："guts" 同時指「膽量」與「內臟」

這種情境既符合笑話的主題（骷髏、膽量），也能自然地帶出冷笑話風格的雙關，適合在輕鬆派對、萬聖節中使用。

## 伍、討論

### 一、SSD1306 無法正常顯示

一開始使用 ssd1306 128x64 的螢幕時如圖 23 我們發現了很多的問題像是有雜訊、供電不穩.....等等後來我們也用了電容濾波但效果還是不好，所以我們改用 ssd1306 128x32 如圖 24 解決了這些問題螢幕可以正常顯示，不僅沒有任何雜訊，內容也能清晰地顯示。



圖 23: ssd1306 128x64 雜訊畫面

(研究者拍攝)



圖 24: ssd1306 128x32 正確顯示

(研究者拍攝)



## 二、智慧眼鏡問題討論

表 9:作品問題及解決(研究者自繪)

功能	問題	解決方案
硬體	1.硬體整體大小不變長時間攜帶 2.戴上後視野範圍受限	利用 3D 建模製作一個眼鏡框，達到輕量化設計使使用者配戴舒服且可以提升視野範圍
	硬體內部空間受限導致無法匹配任何使用者的眼距	將螢幕構置在鏡片之外，並透過光學鏡片讓使用者能夠看到近距離的螢幕顯示內容
收音	只能用手機收音 溝通耗時長	改用耳機收音，並將其加裝在鏡框側面，讓耳機收音後把資料透過網路傳給 ESP32，省去掉操作設備的時間。
OLED 顯示	字體無法完全顯示或顯示空間不足	更改螢幕，雖然體積稍微增加，但螢幕清晰度和效能都會提升

## 三、智慧眼鏡硬體改善

現版本眼鏡外殼不僅龐大，戴上後使用者視野受限，影響到使用者的安全，基於以上問題我們決定使用 3D 建模製作一個輕便且視野更廣闊的佩戴型智慧眼鏡



圖 25:使用者配戴視野受限(研究者拍攝)



圖 26:3D 建模裝配式智慧眼鏡(研究者拍攝)

## 陸、結論

### (一) 智慧眼鏡的研究結果

本專題以跨語言溝通與環境感知為核心，成功開發了 EchoLens 智能眼鏡，結合了多語言翻譯、天氣資訊與圖片分析。透過 Android studio 實現與 Open AI 的連結並建立 UDP 連線將 Open AI 分析的結果直接顯示在使用者眼前，讓使用者享受到人工智慧帶來的便利性。

1. 藉由手機收音並透過 Whisper 自動辨識語言後傳給 Open AI 進行翻譯跟回覆。
2. 使用 API 連結天氣資訊網站，並透過 API 回傳 json 格式的溫溼度資訊。
3. 經過實驗調整眼睛與鏡片的最佳距離，使讓使用者清晰地看到螢幕顯示結果。
4. 使用提示工程將 Open AI 的回覆內容符合預期
5. 即時語音翻譯、情緒理解、天氣資訊抓取與圖像情境分析的功能。

### (二) 未來展望

1. 利用耳機來收音，取代本來使用者必須要對手機講話的方式，結合麥克風將翻譯結果講出來實現無障礙溝通
2. 使用 3D 建模製作輕便的外觀
3. 加入鏡頭使眼鏡可以進行物件辨識並顯示於螢幕上
4. 透過使用者提供的內容，回覆一些認知之外的東西，例如:圖中細微的線索或存在安全隱患、發生緊急情況該如何應對
5. 透過 node-red 編寫出一個可以一對多的 UDP 傳輸，實際可以應用於演講、會議等等。演講者在台上說話的內容會透過 UDP 傳輸給底下所有觀眾並顯示在觀眾的眼鏡裡，無論演講者說什麼語言或聆聽者的母語是甚麼，都能夠透過事先設定後在眼鏡中呈現聆聽者看得懂的語言，這樣就算台上的講師跟你語言不通你也可以透過智能眼鏡清楚的理解演講內容。

## 柒、參考文獻

- [1]陳麗娟(2007)。外語教學中的跨文化溝通問題。外國語文研究期刊 第 6 期。
- [2] ESP32 介紹：<https://github.com/espressif/arduino-esp32>
- [3] OLED 雜訊排除：<https://makerpro.cc/2020/10/oled-principles-and-applications-part3/>
- [4] ESP32 連線：[https://crazymaker.com.tw/esp32-connect-to-wifi-network/#google\\_vignette](https://crazymaker.com.tw/esp32-connect-to-wifi-network/#google_vignette)
- [5] 肉眼最短觀看距離和對焦：<https://nga.178.com/read.php?tid=35418669&rand=138>
- [6] Arduino 基本語法：<https://pse.is/7ju7nk>
- [7]神經機器翻譯介紹：<https://reurl.cc/NYL7Np>
- [8]翻譯學介紹：<https://slpparty.blogspot.com/2019/08/blog-post.html>
- [9]提示工程介紹：<https://aws.amazon.com/tw/what-is/prompt-engineering/>
- [10]文化敏感的翻譯：<https://pse.is/7ju7h4>
- [11]Cloud Translation API：<https://cloud.google.com/translate/docs/reference/rest>
- [12]Prompt engineering techniques：<https://pse.is/7k9cun>
- [13]Language Models are Few-Shot Learners：<https://arxiv.org/abs/2005.14165>
- [14]Covariate-adaptive randomization inference in matched designs：  
<https://arxiv.org/abs/2207.05019>
- [15]Optimistic Curiosity Exploration and Conservative Exploitation with Linear Reward Shaping：  
<https://arxiv.org/abs/2209.07288>
- [16] 語音辨識及翻譯：[https://nlp.ict.ac.cn/xsdt/202410/t20241009\\_247113.html](https://nlp.ict.ac.cn/xsdt/202410/t20241009_247113.html)

## 【評語】 052506

1. 本研究提出一款結合生成式 AI 的智慧眼鏡，具備場景分析能力，能綜合判斷使用者所處情境中的情緒、語氣與文化禮節，提供最合適的回應建議。
2. 此作品以實作為主，整合多項技術，相當具難度，但報告中不易看到完整作品的結果，較缺乏系統的效能評估。
3. 本作品雖有製作「EchoLens」智能眼鏡，但研究核心為與 ChatGPT 的互動以取得較佳的互動翻譯品質，在技術創新性上較為不足，完整的實驗設計與實驗結果討論亦可再強化。另亦展現結合「情緒、語氣、文化」等的翻譯與回應差異。

作品海報

# 跨語言交流與環境感知的智能眼鏡- EchoLens

*EchoLens: Intelligent Glasses for Real-Time Translation and Environmental Sensing*

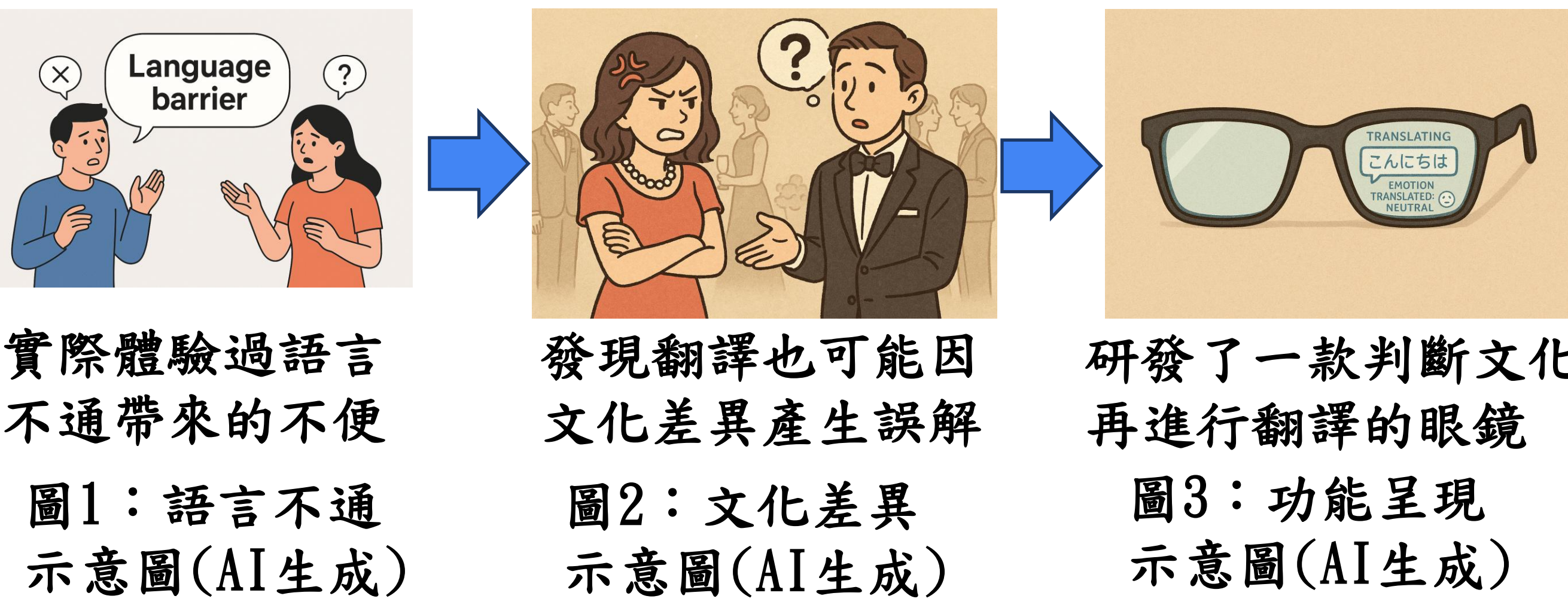


摘要

本研究提出一款結合生成式AI能進行場景分析，進而綜合情緒、語氣、文化禮節給予最佳回覆的人工智慧眼鏡。目的希望使用者不需要透過手機即可實現即時語音翻譯、情境分析與天氣資訊抓取功能。系統以ESP32為主要控制核心，搭配Android Studio開發之應用程式，透過WiFi採用UDP通訊協定進行無線通訊，提供精簡的翻譯與應答建議。每次翻譯處理後，GPT根據語境與提示詞再將翻譯結果與應答建議，傳送回並顯示於智慧眼鏡的OLED螢幕，將即時資訊呈現在使用者眼前，實現流暢的跨語言互動體驗。

研究動機

本研究的開發初衷，來自於日常生活中對於語音互動、即時翻譯與資訊顯示的需求。雖然現有的翻譯App或設備能夠提供基本的語音翻譯功能，但多數需要手動操作，導致溝通仍然不夠順暢，又或者因翻譯結果缺乏語境、情緒理解導致文化或禮節上面的誤會。因此，本研究希望透過 OPEN AI API 語言處理技術，打造更加自然、並能夠判讀情緒的智能翻譯裝置，讓語言或文化不再成為溝通的障礙。開發一套智能多語言翻譯的智慧眼鏡。



研究目的

- 一、研究使用氣象公開資料的API取得溫濕度的方法
- 二、探討使用Open AI API 進行語文翻譯的方法
- 三、建構即時多語文翻譯及回覆建議的手機App
- 四、建立ESP32連線App將資訊即時在OLED
- 五、使用Open AI分析場合及文化來避免翻譯造成誤解

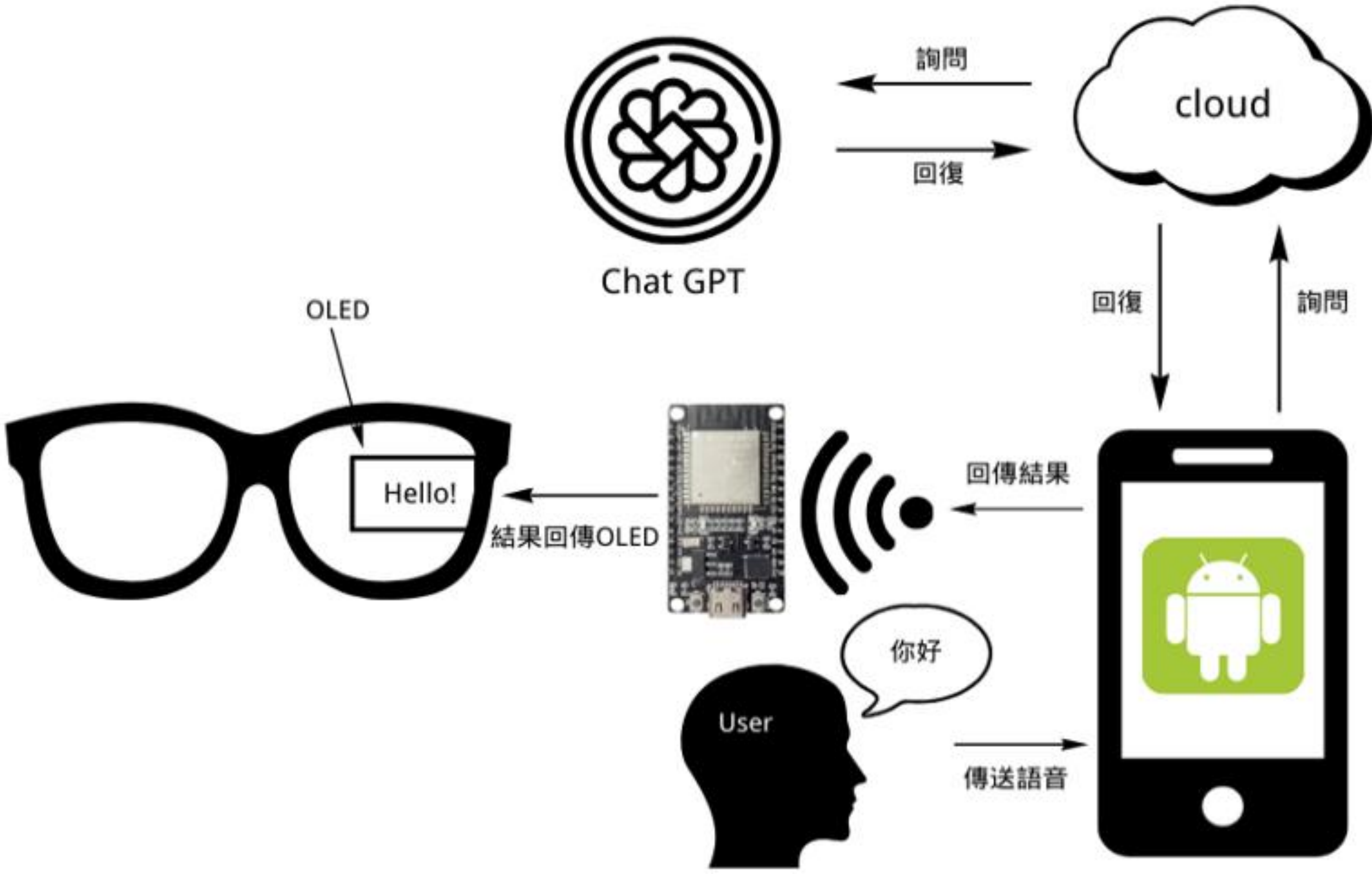


圖4：功能流程圖(研究者自繪)

研究過程及方法

作品製作流程



圖5：製作流程(研究者自繪)

智慧眼鏡硬體設計

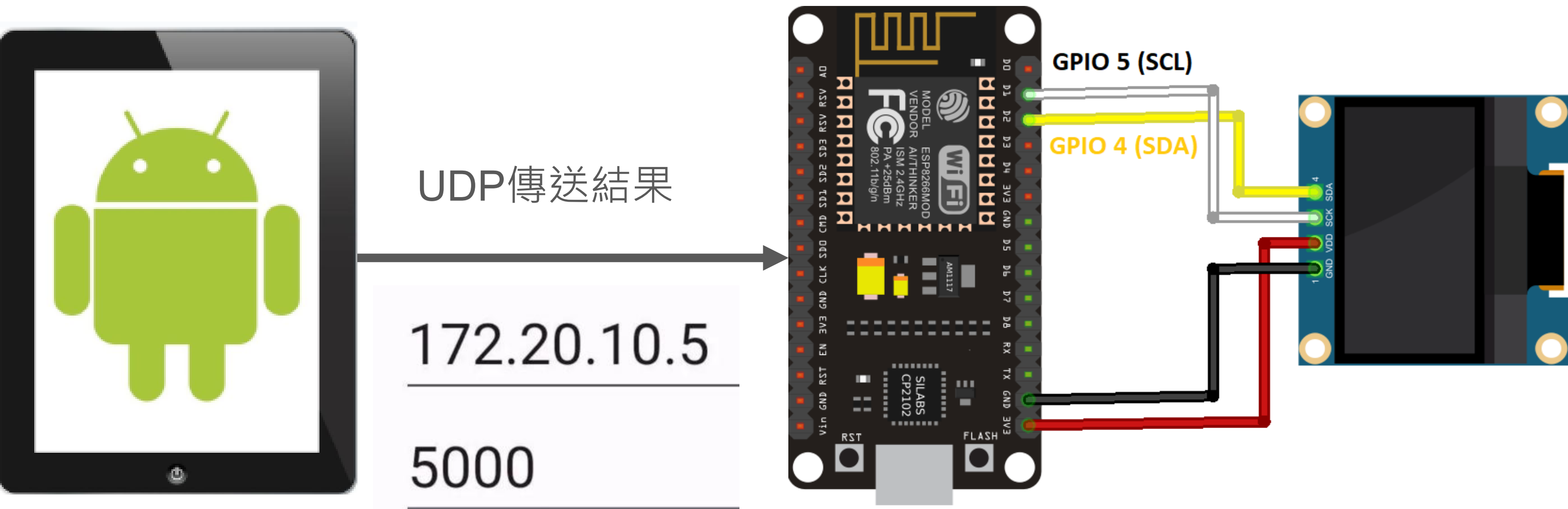


圖6：資料傳輸及接線方式(研究者自繪)

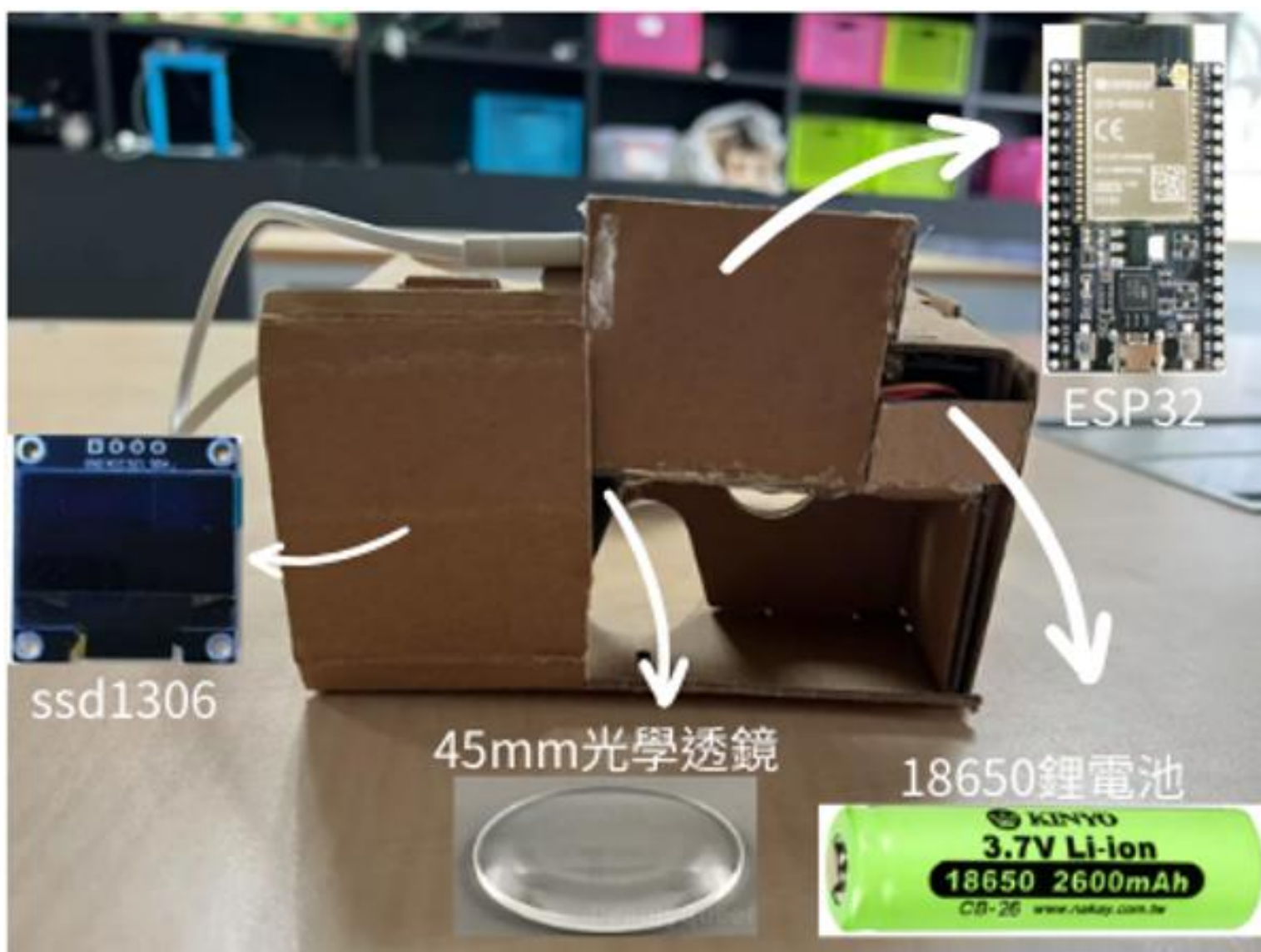


圖7：元件分布圖(研究者拍攝)

翻譯功能

使用者長按 Button 來進行語音輸入並透過 Whisper 自動辨識語言並轉文字在使用UDP傳給Open AI透過語言內容判斷環境再根據環境推斷文化並做出一個最適當的翻譯及回覆

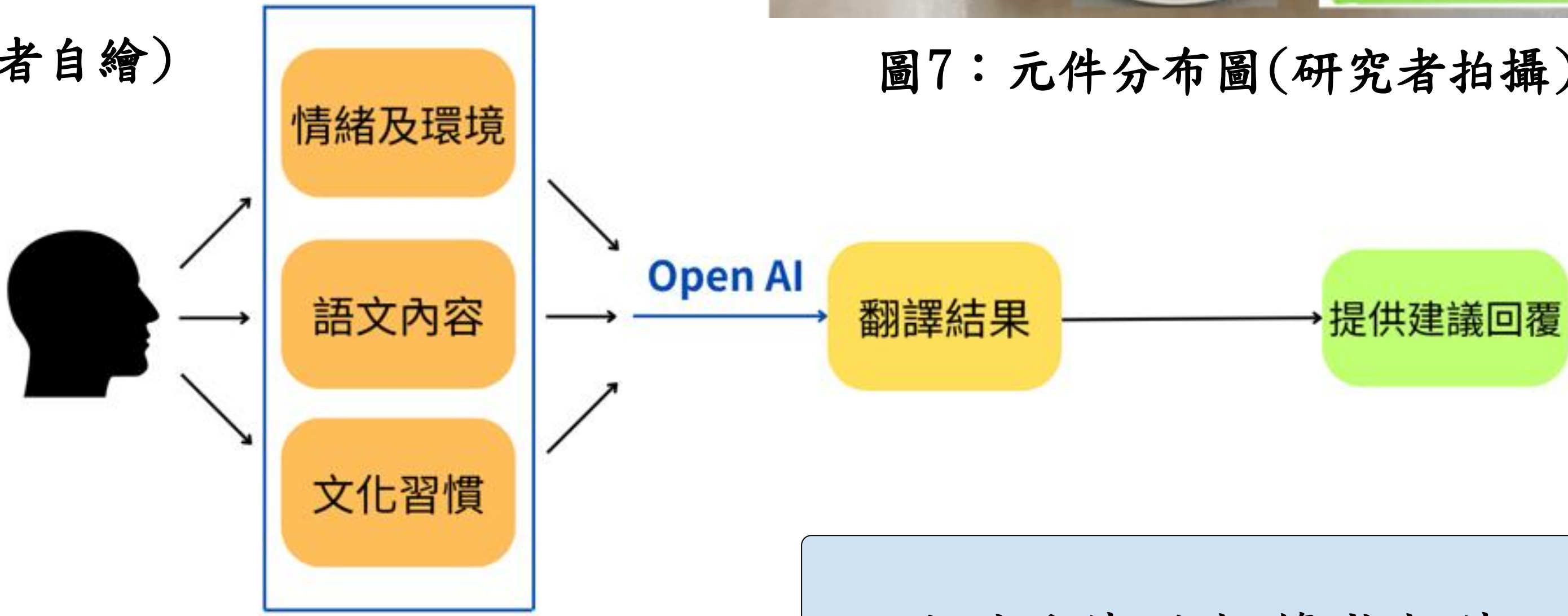


圖8：翻譯憑據圖(研究者自繪)

記憶語言

透過button記憶使用者當前說的語言，將之後收錄的音訊直接翻譯成記憶中的語言。例如:使用者說出日文並按下記憶，使用者之後不管說了甚麼都會翻譯成記憶中的日文直到清除記憶。

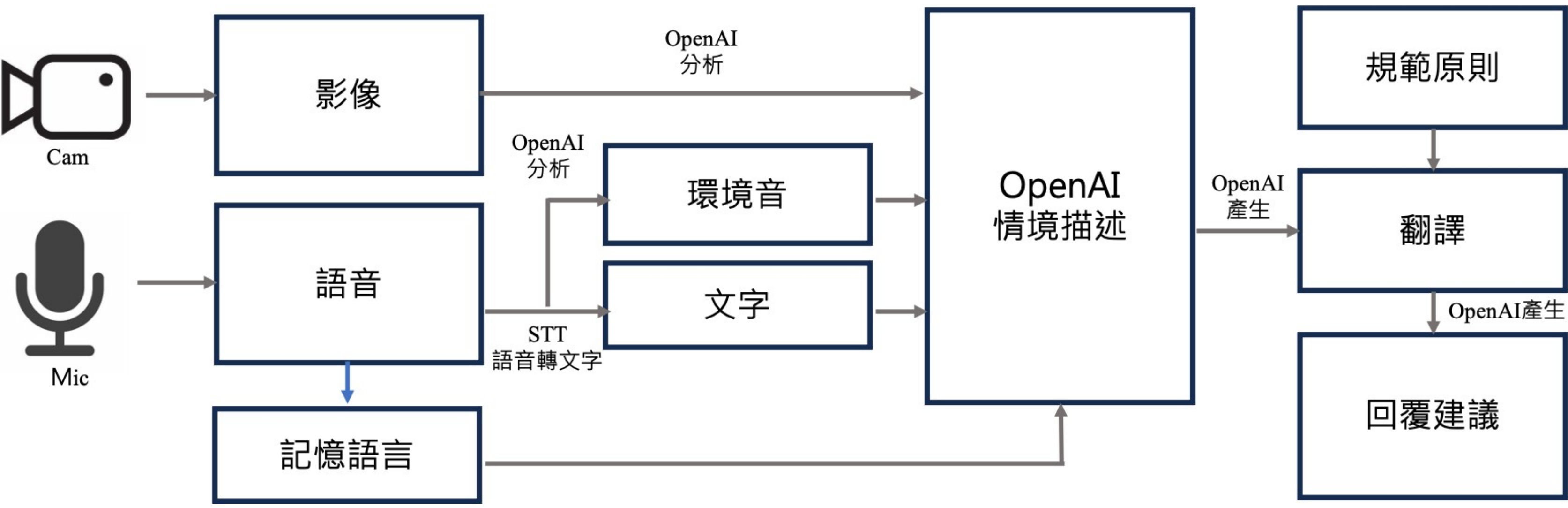


圖9：Open AI分析圖像音訊流程圖(研究者自繪)

圖像分析

呼叫手機的相簿或相機來獲取照片後透過FileProvider將圖片轉成安全的 URI供App 正常存取再將圖形檔轉成Base64透過API上傳至OpenAI 進行分析再回傳Open AI結果。

環境音

環境音分析是指針對非語音內容的音訊進行特徵提取與分類，目的在於了解音訊中的環境背景或事件，本專題使用環境音來精進對場合的判斷。

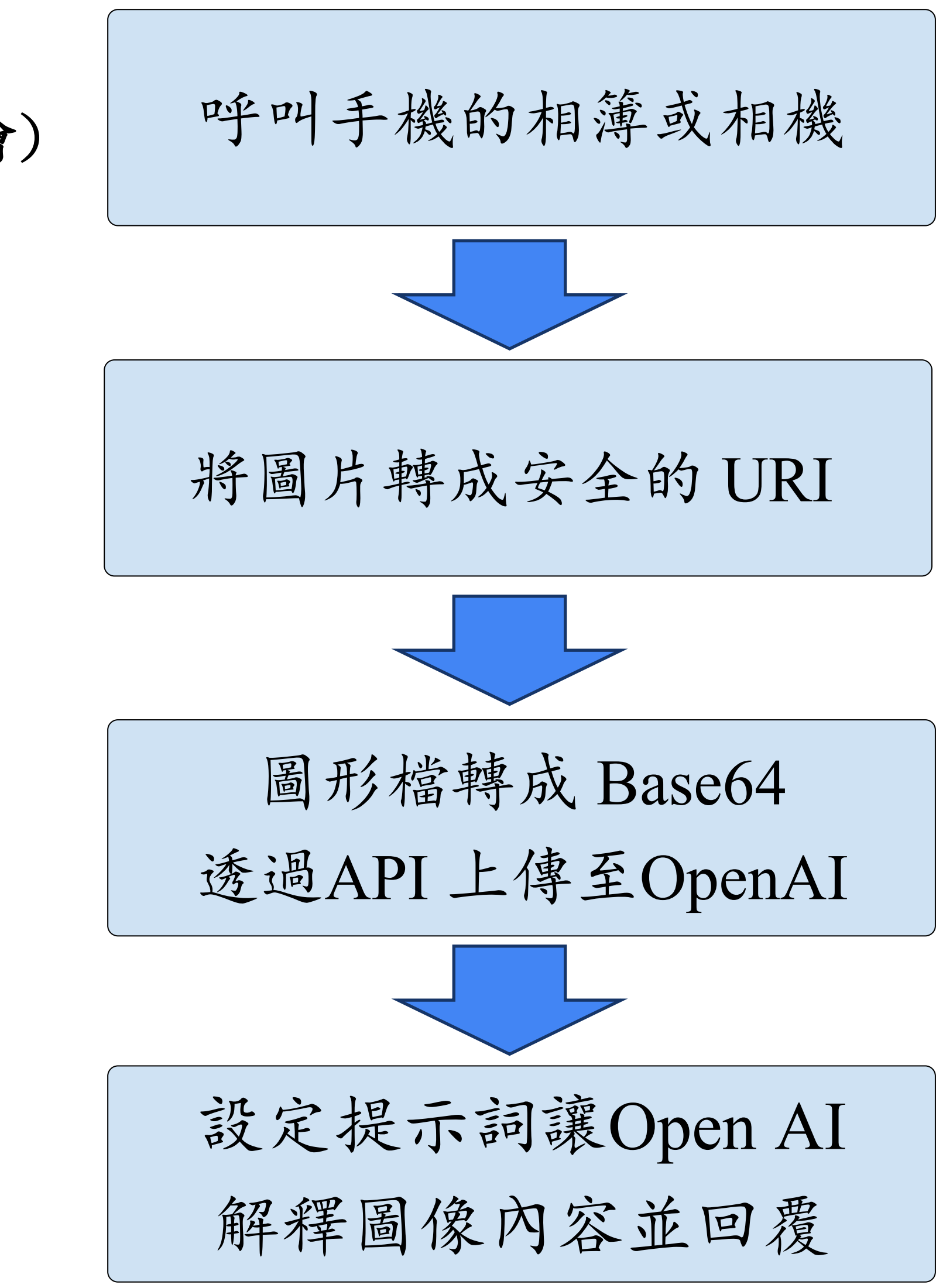


圖10：圖像上傳Open AI流程圖(研究者自繪)



圖像搭配環境音和文字產生口語翻譯

我現在在一場社區派對，活動使用英文，我是中文母語者並且配戴智慧眼鏡，能略懂與看懂英文。朋友與我之間的對話圍繞著派對開始至結束。請你假設是我朋友，與我對話，眼鏡會顯示中文翻譯內容與建議回覆

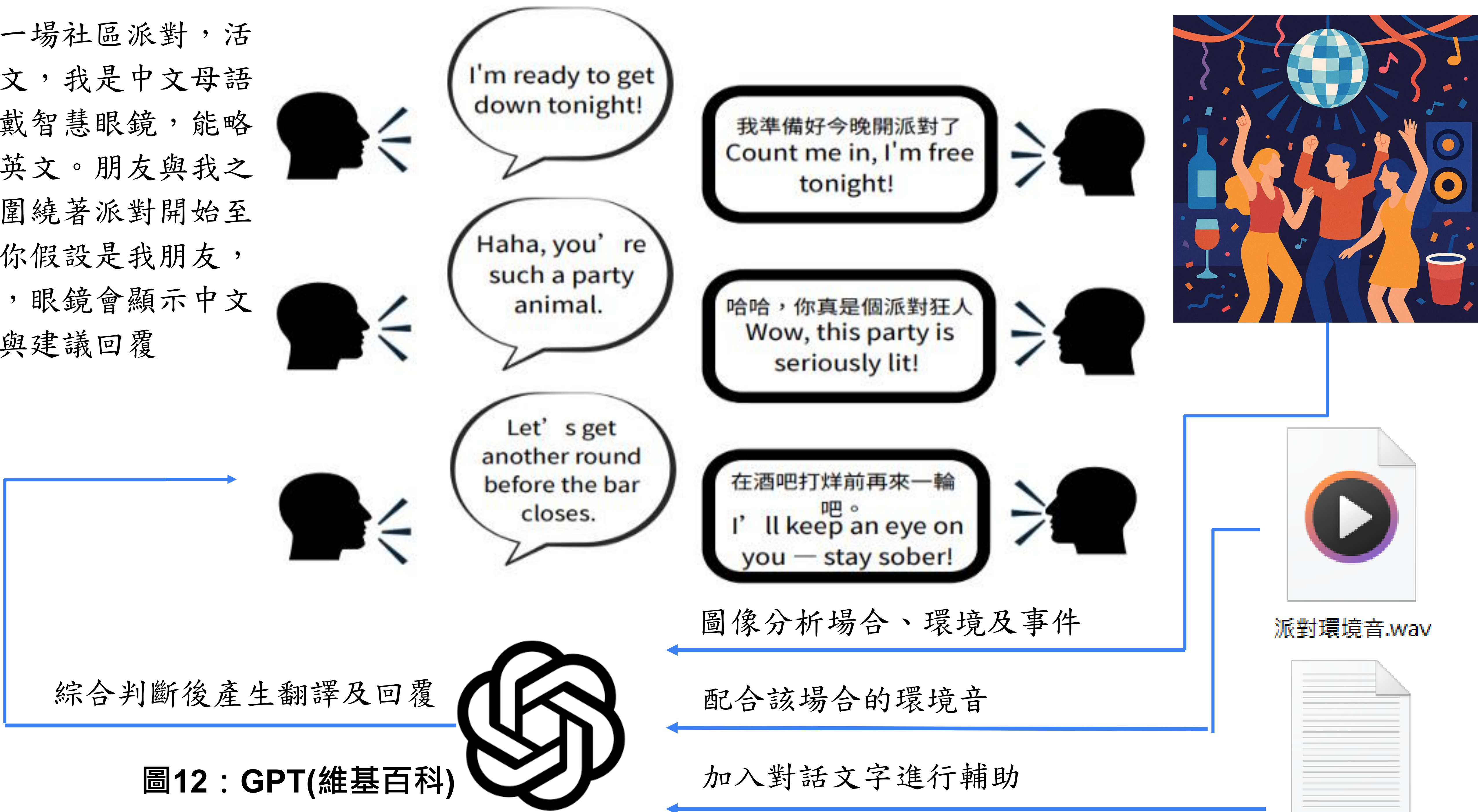


圖12：GPT(維基百科)

圖13：三重判斷進行翻譯及回覆(維基百科)

實驗一：Open AI透過音檔與文字分析場合

透過純音檔跟音檔加文字兩種不同分析情況發現純音檔分析連基礎的場合都無法分辨，只能給出一個攏統的範圍，後來我們加入文字檔來輔助判斷如表2可知，加入文字後可以具體判斷出場合，也能透過文字內容來達到更精確的判斷結果。

表1:純音檔分析

場所類別	機率 (%)
室內安靜空間	28.5
戶外自然環境	10
街道或公共空間	35
辦公室或教室	26.5

表2:音檔+文字分析

場所類別	機率 (%)
服飾店	94.5
百貨公司	4
一般零售店	1
商場	0.5

實驗二：Google及OpenAI的翻譯API結果比較

表3 對比兩種翻譯圖(作者繪製)

考驗Google及OpenAI 的翻譯能力，實作如下：

1. 在ChatGPT以「英文特殊用語或俚語，容易翻譯錯誤的詞句，請列出並提供例句」詢問。
2. 將條列的例句，提供給Google 及OpenAI API翻譯結果比較。

如表3，採用 Google API進行翻譯會直譯而產生語意的偏差，OPEN AI翻譯的結果則正確。

語句(粗體為片語)	GOOGLE API翻譯	OPEN AI API翻譯
You have a big performance tonight— <b>break a leg!</b>	今晚你有精彩的表演 - 打斷腿！	你今晚有重要的演出—祝你好運！
That designer bag <b>cost me an arm and a leg!</b>	那個名牌包花了我一條手臂和一條腿！	那個名牌包超貴的！
I've done my part. Now <b>the ball's in your court.</b>	我已經盡了我的職責。現在球在你的球場上。	我該做的做完了，接下來看你了。
Are you serious, or are you just <b>pulling my leg?</b>	你是認真的還是只是在開我的玩笑？	你是認真的，還是在跟我開玩笑？

提示工程

本研究設計當日氣象、跨語文交流及環境認知三種環境情境，設計了6組GPT提示詞 (Prompt)，讓ChatGPT根據口語的語音敘述，自動執行對應任務。在硬體規格、語言翻譯、情緒分析、建議生成，運用提示語導引導 AI 生成符合問題的自然回應，讓 AI 更理解你的語氣、意圖與情境，以貼近人類思維的對話體驗。

表4 本研究使用的提示語整理（作者繪製）

中文 → 記憶語言（或英文）	情緒判斷（表層）	情緒安慰（心理建議）
1. 明確指定語言，避免 GPT 主動翻成多語或原文。 2. 使用「 <b>只輸出</b> 」限制輸出說明，使輸出簡潔。 3. 加上「 <b>不要加任何說明</b> 」避免「 <b>翻譯如下</b> 」的冗字適合用於展示於螢幕或 OLED。	1. 定義模型角色為心理專家更貼近任務。 2. 指定「 <b>簡短回答一種情緒</b> 」避免輸出冗長分析。 3. 定義適合快速語音播報或在OLED 顯示。	1. 適合用於日常語句的情緒撫慰與支持場景。 2. 提示語強調「 <b>溫暖</b> 」「 <b>正向</b> 」「 <b>不帶命令語氣</b> 」，使使用者感到被理解與安慰。 3. 內容設計簡潔，回覆通常為一句「 <b>輕柔建議</b> 」，提升使用者心理舒適感。
ChatGPT問答（原語言回答）	情境式天氣建議	圖片分析
1. 簡潔自然，讓ChatGPT 回應時自動「 <b>使用偵測語言</b> 」（Whisper 偵測出來的）。 2. 「 <b>適合多語情境</b> 」：講日文回日文、講韓文回韓文等。 3. 搭配 Whisper 語音偵測效果最佳。	1. 每日提醒、早安問候或穿搭建議等生活情境下。 2. 透過時間、天氣與地理資訊動態生成建議，讓內容具即時性與個人化。 3. 語氣自然親切，使用「 <b>建議的口吻</b> 」，提升使用者親切度與信賴感。	1. 適合用於物品辨識、場景描述、視覺輔助等場景。 2. 提示語採「 <b>簡要說明</b> 」風格，避免回覆過於攏長或複雜。 3. 可應用於拍照即分析功能，作為多模態（語音+圖像）應用的一環。

聆聽系統

開始後持續收音，目的是為了能夠收錄從頭到尾的對話內容並分析使用者的情緒，並經由分析後總結出當前情緒，並生成出符合使用者的建議，讓我們的智慧眼鏡不僅是語言上的翻譯，也能夠翻譯內心的情緒

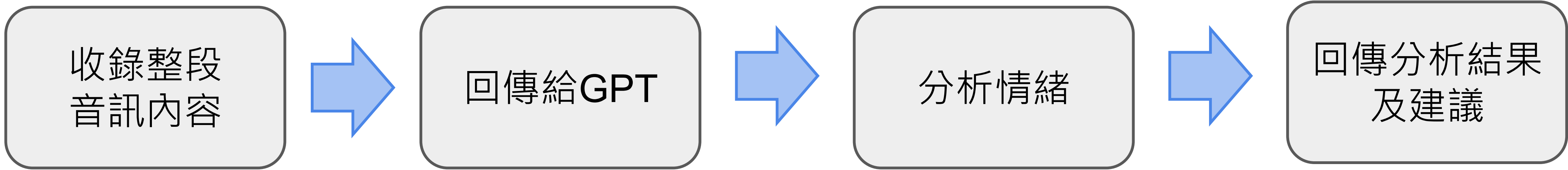


圖14：聆聽模式流程(研究者自繪)



# 研究結果

## App 執行結果

1. 天氣資訊抓取：Button輸入後上傳經緯度回傳json格式的溫溼度資訊
2. 多語言翻譯&記憶語言功能：本作品可分析多種語言，長按進行收音後上傳至Open AI進行翻譯及回覆，例如：使用者說出日語後按下記憶語言按鈕，App就會將日語記憶，之後使用者說出的語言會直接翻譯成日語直到清除記憶
3. 情緒分析：透過收錄音訊來判斷當時的情緒，再根據情緒給予回覆及建議
4. 圖像分析：使用者選擇圖片後轉檔上傳Open AI分析圖片內容和場景並回傳至App



圖15：各功能App展示(研究者拍攝)



圖16：天氣及翻譯OLED顯示(研究者拍攝)

## 討論

表5 作品功能檢討和解決(作者繪製)

功能	問題	解決方案
軟體	• App介面化簡使其更直接好理解使用方法	隨機測試路人並讓他們提供使用感受及建議
眼距	• 硬體內部空間受限導致無法匹配任何使用者的眼距	將螢幕構置在鏡片之外，並透過光學鏡片讓使用者能夠看到近距離的螢幕顯示內容
收音	• 只能用手机收音	改用耳機收音，並將其加裝在鏡框側面，省去掉操作設備的時間，實現解放雙手的無障礙溝通

## 結論

本次專題研究中，設計了一個以GPT為核心可進行天氣感知、多語言交流及環境認知的智能眼鏡，透過Android Studio實現與OpenAI的連結，並將結果使用UDP通訊協定傳送到ESP32驅動OLED顯示訊息，主要的成果如下：

一、天氣資訊抓取:系統可即時獲取天氣、溫度與濕度，根據環境狀況提供建議，提升日常生活的便利性。

二、多語言交流:透過語音辨識，自動語言偵測，記憶語言功能實現多語言之間的即時溝通。

三、環境認知:運用 GPT的图片辨識能力，讓系統可解讀拍攝畫面、辨識物品與場景內容，並做出回覆

本智慧眼鏡透過語音輸入，結合 Whisper 語音辨識、GPT 翻譯、情緒分析、圖片辨識，實現多語言即時溝通與溫暖建議並同步顯示於 OLED 螢幕，實現即時資訊的呈現。

## 未來展望

- 一、利用耳機來收音，取代本來使用者必須要對手機講話的方式，結合麥克風將翻譯結果講出來實現無障礙溝通
- 二、使用3D建模製作輕便的外觀
- 三、加入鏡頭使眼鏡可以進行物件辨識並顯示於螢幕上
- 四、透過使用者提供的內容，回覆一些認知之外的東西，例如:圖中細微的線索或存在安全隱患、發生緊急情況該如何應對。

## 參考資料與文獻

Ithelp(2023) ChatGPT 大解密：帶您探討機器學習背後的秘密以及利用與發展，[https：//ithelp.ithome.com.tw/articles/10311343](https://ithelp.ithome.com.tw/articles/10311343)

尤濬哲（2023）。IoT 物聯網應用：使用 ESP32 開發板與 Arduino C 程式語言。台科大出版。

Aeneas (2015)。I2C通訊協定 [PDF]，<https://ppt.cc/ffRUMx>

FMUUSER(2021)如何將 OLED 顯示器與 ESP32 接口？ESP32 OLED 顯示教程，[https：//zh-tw.fmuser.net/content/?14594.html](https://zh-tw.fmuser.net/content/?14594.html)

ibest(2023)什麼是生成式AI？5分鐘帶你了解生成式人工智慧，[https：//www.ibest.com.tw/news-detail/aigc/](https://www.ibest.com.tw/news-detail/aigc/)

Amazon Web Services ( n.d. )。什麼是提示詞工程（ Prompt Engineering ），<https://aws.amazon.com/tw/what-is/prompt-engineering/>