

## 具深度學習結合互動式資訊服務站之旅遊景點推薦系統

### A Tourist Attraction Recommendation System Using Deep Learning On Kiosk

專題組員：施柏亘、陳勇豪、邱偉城、陳彥廷

專題編號：PRJ-NTPUCSIE-108-002

執行期間：2019 年 9 月至 2020 年 6 月

#### 一、簡介

互動式資訊服務站(又名 Kiosk)，外觀很像電話亭，是一種電腦終端機，提供人性化且互動式的使用者介面。最早追溯於 1977 年，主要放在校園中提供學校資訊。如今，在百貨公司、便利商店、機場都隨處可見，這樣大型笨重的機器就這樣悄悄地進入我們的生活之中，現今有人拿來當成廣告看板，也有像最常見的 YOUBIKE 站台，提供租借腳踏車的服務。同樣的一台機器依照匯入軟體的不同，使得機台可以有多種不同的功能。這次的專題，就是想將 Kiosk 結合網頁、深度學習以強化現有的 Kiosk 功能。用攝像頭拍攝即時的人臉圖片資訊，透過深度學習完成人臉辨識，判斷使用者的性別、年齡區間和情緒，分析出使用者適合哪些去處，經過計算且排序後，給予對應的景點推薦列表。使用者能查詢推薦結果與當地天氣資訊，選擇想前往的景點遊玩。此系統能應用在捷運站或火車站等交通便利之處，使得背包客或是來到此地的外地人在不知哪處可去時，有個可以推薦景點的機台，實現

便利的智慧城市。

#### 二、系統架構



[圖一 system scenario]

本系統大致可以分為三個部份來做簡單的概述，第一個是使用者介面，第二是所使用的 AI model，最後一個則是存取資訊的資料庫：

##### 一、景點資料收集

系統會依照當時機器所在的地點，利用 Google Maps API 取得該地點周邊之景點資訊。並利用簡單的 json 檔分析程式取得所需之景點資訊，並放入由 MySQL 建置的資料庫中。

##### 二、即時特徵截取

系統一開始會先拍一張使用者的臉部圖片，隨後使用三種 AI 模型，分別得出性別、年齡及情緒三個使用者臉部圖像的分析結果。及

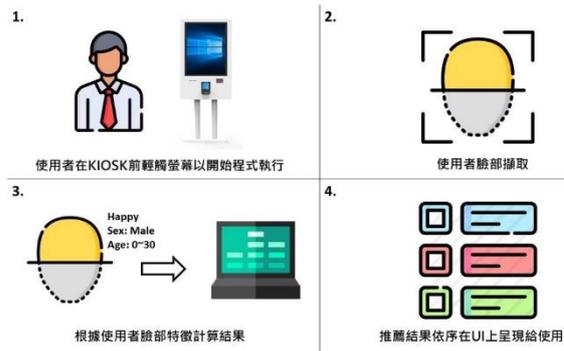
使用 OpenWeather API 取得當地的天氣資訊。

### 三、推薦與結果呈現介面

1. 推薦機制：根據使用者當時的性別、年齡、情緒以及當地當時的天氣，運用統計分析的公式計算，得出當時適合的結果列表，供使用者做選擇。
2. 結果呈現介面：將經由推薦機制計算後的結果呈現在使用者介面上，並透過 Google API 取得各景點的導航、街景等資訊。

### 三、系統實作

#### I. 系統三部份實作說明



[圖二 系統流程圖]

本節將依照上述系統架構的三個部份做詳細的說明：

#### 一、景點資料收集

透過 Google API 及機器所在地之經緯度，即可得到當地的景點資訊，如景點名稱、圖片、經緯度等。利用 Place API 取得最近的特定類型地標的資訊，再利用 Street View Static API 取得該坐標的街景服務。

#### 二、即時特徵擷取

此部分包括兩個細項，一個是運用 AI 模型，由使用者的臉部擷取出

其年齡、性別和情緒。另一項是即時天氣資訊的取得。以下將會依照下面兩個部分做詳細說明：

#### I. AI 模型

依照一個人的臉部特徵及表情，要完整分辨他的年齡、性別及情緒，其實並不是一件容易的事情。因此，為了降低問題的複雜度以及提升訓練結果的準確度，年齡與性別將使用 OpenCV 套件來完成 [3]，而情緒則是利用 Google MobileNet 模型 [4]，以避免分類的組合數過多，而造成系統的判斷結果不準確，進而影響到推薦之內容。

在情緒模型的部分，我們是利用上網搜尋的方式來收集各種表情的人臉圖片並將其分類為普通 (Neutral)、快樂 (Happy)、生氣 (Angry)、難過 (Sad) 四類。每一個類別約 500 張圖片，總共約 2000 張做模型的訓練，訓練後的準確率約為 70%。

#### II. 即時天氣資訊取得

在選擇旅遊行程與景點時，天氣狀態一直都扮演一個非常重要的角色。因此，能不能得到即時的天氣資訊就成為系統完整與否的重要評斷關鍵。

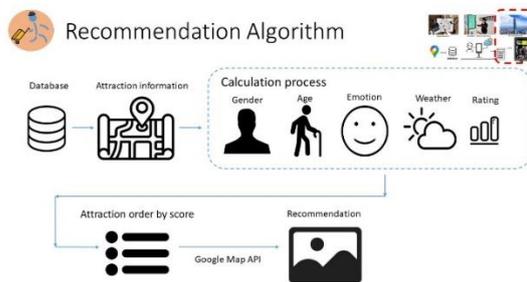
起初以為只要利用爬蟲程式，存取中央氣象局網站上的公開資訊，即可得到系統所需要的天氣資訊。但在研究後發現，中央氣象局的網頁原始碼中，各部分(包含氣溫、濕度與天氣狀態等)的資料都不是原始數值的呈現，而是後端資料

庫的存取連結。因此，爬蟲程式還是無法取得天氣資訊。

後又經過探索，發現有一網站—open weather map，只要在官網申請一個免費的金鑰，並於 python 程式中引入函式庫 pyowm 後，即可利用其 API 取得其各測站之即時天氣資訊。

### 三、推薦演算法

#### 1. 推薦機制：



[圖三 推薦演算法]

首先，我們可以先將由 Google API 得來的景點分為 Museum, Park, Shopping Mall, Department Store, Movie Theater 和 Tourist Attraction 等六個不同的類別。並依照參考論文[2]中得出八個不同的旅遊動機，如下：

1. To visit history / culture sites
2. To increase knowledge of new place
3. To meet local people
4. To mix with fellow people
5. To have fun
6. To relax
7. To get close to nature
8. To engage in sports

並依照論文中的各年齡層對不同旅遊動機的影響程度之統計數據，在標準化後等比縮放成 0 到 1 的區間，以供

計算使用。之後，依照各個不同的類別，選出其對應的旅遊動機。

其後，依照上面得到的各類別景點之旅遊動機的數據，與臉部特徵擷取的結果做相乘，以得到該特徵對旅遊動機的影響值，公式如下：

$$F_j = \sum_{i=1}^8 M_{ij} * R_{ij}$$

(其中，i 為八項旅遊動機，j 為特徵擷取的類別(分為年齡、性別與情緒等三種)，M 為旅遊動機之影響值，R 為一景點類別中，各個旅遊動機對其的影響比例)

在分別計算完三項特徵的 F 值後，便會將其三個值加總，並乘上各景點在 Google 上的評分，得到該景點類別的分數。

$$H = \left( \sum_{j=1}^3 F_j \right) * G$$

(其中，j 為特徵擷取的類別，G 為由 Google API 得到之景點的 rating)

接下來，依照景點類別，將天氣狀態也放入分數計算。這裡是依照景點類別做天氣狀態下合適與否的判別，並除以整體景點的 Google rating，以防止天氣影響佔整體評分的比例過高，公式如下：

$$C = \frac{w}{G_{avg}}$$

(其中，w 為現在天氣狀態之影響值， $G_{avg}$  為所有景點的 google rating 平均值)

最後，將各景點的 H 和 C 加總，即可得到最終的分數。系統會依照分數之

高低排序，並將推薦列表呈現給使用者。

$$S = H + C$$

## II. 軟/硬體配置



[圖四 軟硬體配置圖]

### 1. Kiosk

本次專題使用的平台為 Kiosk，其作業系統為 Ubuntu 系統。負責使用者照片的擷取和介面的呈現。當使用者按下開始時，Kiosk 上的拍照程式會一直不斷的擷取使用者的照片，並利用 socket 回傳給 Server 進行臉部特徵擷取及推薦結果計算。

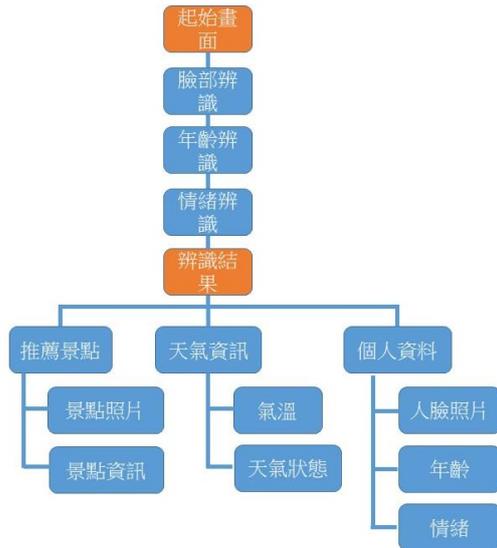
### 2. Server

我們將所有會用到的程式以及資料庫等等都存放於 Server 中，程式的部分由 python 來撰寫，主要會運用到 TensorFlow、Keras 還有 OpenCV 等函式庫來進行判斷的動作，並且所有的程式都將在背景執行。當使用者開始使用時，會由 Kiosk 端的拍照程式透過 socket 將使用者的照片傳回 Server。接下來，會依照從 Kiosk 端收到的照片做即時特徵擷取，以得到其年齡、性別和情緒。同時，透過 OpenWeather API 取得即時的天氣資訊。有了使用者特徵和天氣資訊後，系統將會從資料庫中取出所需要用到

的標籤，再依各項權重進行運算，最後將得出的結果回傳至使用者介面中呈現給使用者。

### 3. 使用者介面(UI)

說到使用者介面(UI)，最常讓人聯想到的就是網頁前端設計。我們這次專題的使用者介面是用全螢幕的網頁來呈現，說到網頁想當然爾就離不開 HTML、CSS、JavaScript 這三大元素。HTML 主要負責產生網頁的整體架構，其中包括所有會呈現出來的元素，例如：按鈕、文字等，這時網頁上的所有元素由上到下毫無排列的擠在一起，雜亂無章並不美觀。CSS 負責管理網頁的外貌，CSS 之於網頁有如化妝品之於人類，所以 CSS 相當考驗設計者的美感，它將網頁中所有的元素放到其正確的位置，讓網頁的外貌美觀一些。最後，JavaScript 管理控制網頁的內容與使用者之間的操作行為。JavaScript 最重要的概念就是控制 HTML 裡的元素，像是監聽按鈕，又或是計畫中要使用的景點滑動，也是用到 JavaScript 裡的 JQuery 函式庫[1]，這些都是常見的 JavaScript 用法。



[圖五 使用者介面流程圖]

#### 四、實作結果



[圖六 辨識結果介面]

經過人臉辨識與後台計算後跑出的頁面，上方是辨識出的資訊，下方是為您推薦的景點，左右滑動可以挑選不同的景點。



[圖七 景點資訊介面]

點下去挑選的景點可以看到這個景點的資訊，包括景點圖片、景點街景，與景點導航的 QRcode，下方可以在挑選不同的景點。

#### 五、結語與展望

本系統希望結合當地店家，還有與當地政府以及觀光局合作，藉由判斷的結果，使得對於當地不熟悉的觀光客，可以根據推薦結果前往最適合當時的天氣以及觀光客的心情、年齡以及性別之景點，我們認為這個系統對於像是背包客這類型沒有跟團的旅客是很友善的，不需要做太多的作業，而到了當地就可以透過我們的推薦系統，找到適合的景點，體驗當地風情，除了增加了互動性之外，也可以有效率的推廣各地的觀光產業。

#### 六、參考文獻

[1] jQuery UI: slick

(<https://github.com/kenwheeler/sli>)

ck/)

- [2] Cristina Jonsson, Dwayne Devonish. "Does Nationality, Gender, and Age Affect Travel Motivation? A Case of Visitors To the Caribbean Island of Barbados", p6~8.  
(<https://www.onecaribbean.org/content/files/TRAVEL%20MOTIVATIONS%20OF%20TOURISTS.pdf>)
- [3] DataFlair Team, Jan. 2020. Interesting Python Project of Gender and Age Detection with OpenCV(<https://data-flair.training/blogs/python-project-gender-age-detection/>)
- [4] Ritesh, Jun. 2018. Facial-Expression-Detection(<https://github.com/MauryaRitesh/Facial-Expression-Detection>)