

Module 01

認識機器學習

學習目標：

- 1-1: 機器學習介紹
- 1-2: 機器學習學習種類
- 1-3: 迴歸與分類

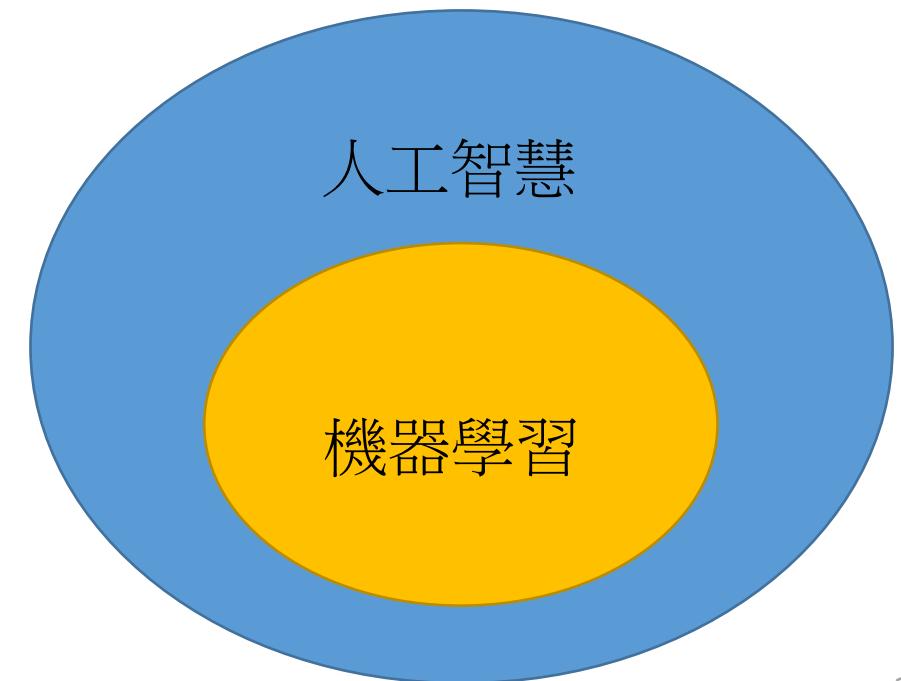
學習目標：

1-1:機器學習介紹

- 人工智能與機器學習
- 學習機器學習所需要的知识
- 電腦如何學習

人工智慧與機器學習

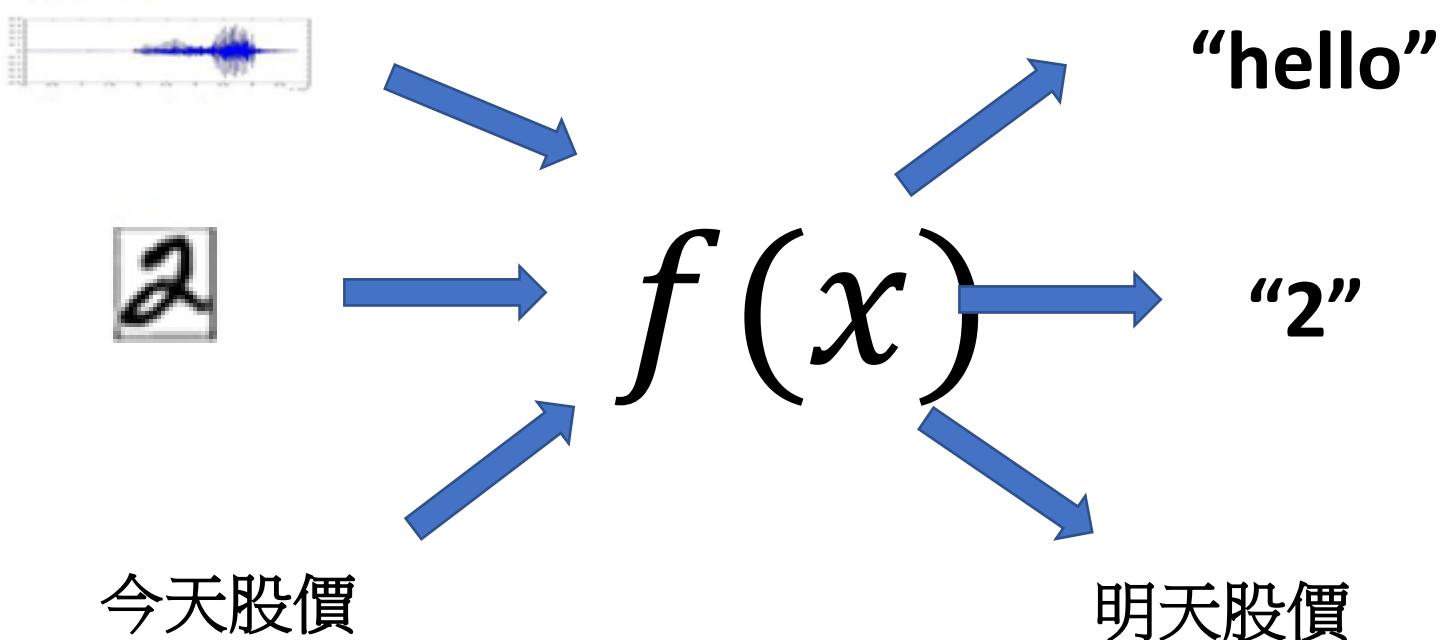
- 人工智能是計算機科學領域裡一門學問
 - 其宗旨在讓電腦跟人腦一樣聰明
- 機器學習是人工智能領域裡一門分支
 - 使用大量統計、機率來讓電腦變聰明



- 我們可以將大量的資料輸入給機器學習演算法去學習
 - 機器將會自己找到資料之間的規律
 - 機器可以挖掘出直覺無法發現的資料價值
 - 機器也可以運用資料提供更客製化的服務

- 學習機器學習所需要的知識非常多，包含
 - 統計學、機率論、最佳化理論、線性代數、微積分等
- 如果對於影像相關的問題想要更深入了解還需要多學習
 - 影像處理
- 如果對於自然語言處理相關的問題想要更深入了解還需要多學習
 - 訊號處理、自然語言處理
- 這門課程我們會使用Azure API來去呼叫已經寫好的機器學習演算法
 - 這些核心的技術Azure的平台都已經寫好供我們使用

- 電腦在學習就是在找到一個很厲害的函數
 - 此函數能幫我們把特定的事情做得很好
 - 例如語音辨識就是要找到一個函數，其輸入是音檔，輸出可以很好的辨識此音檔在說什麼



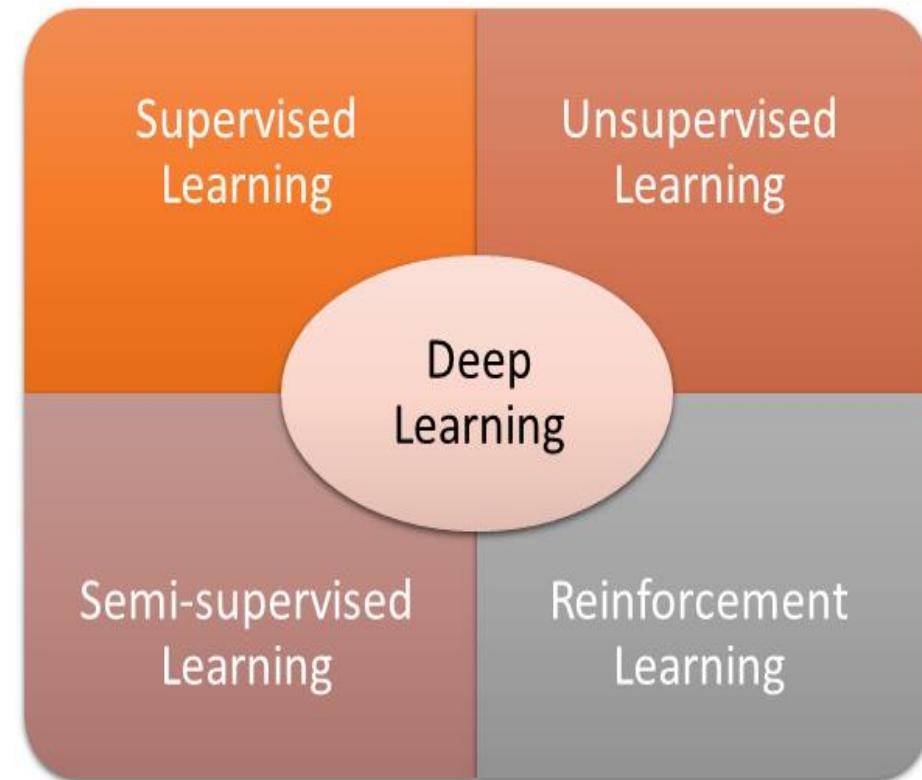
學習目標：

1-2:機器學習學習種類

- 常見之學習方法
- 監督式學習
- 非監督式學習
- 半監督式學習
- 強化學習

常見之學習方法

- 一般來說，我們可以根據資料輸入的型態來將學習分成四種
 - 監督式學習、非監督式學習、半監督式學習、強化學習



- 監督式學習表示給予機器資料以及每筆資料所對應的標籤
 - 這些標籤就好像要用來教電腦辨別東西的答案
 - 目前比較成熟的AI應用大部分都是此類別



這是0



這是1



這是1



這是0

- 非監督式學習表示給予機器資料但沒有對應的標籤
 - 類聚就是一種常見的非監督式學習，其會把相似度高的資料放在一起，相似度相對低的放遠一點



半監督式學習

- 半監督式學習顧名思義就是部分是監督式學習，部分是非監督式學習
 - 部分資料有標籤，部分資料沒有標籤



這是0



這是1

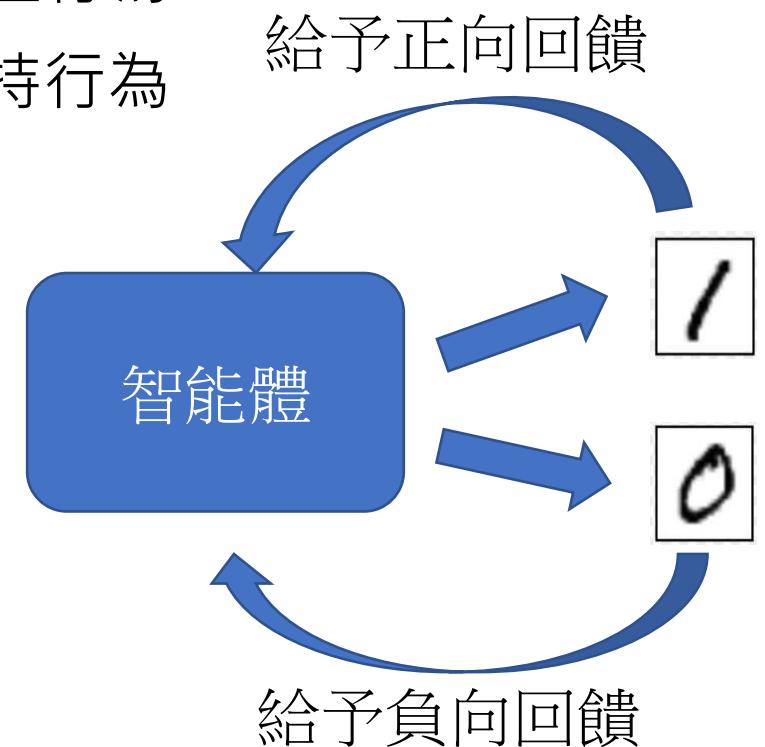


???



???

- 強化學習是藉由發號命令，並讓智能體做動作、給予正/負向賞酬、修正智能體的一連串學習行為
 - 當智能體做錯事的時候，給予負向回饋請它修正行為
 - 當智能體做對事的時候，給予正向回饋請它保持行為



1-3:迴歸與分類

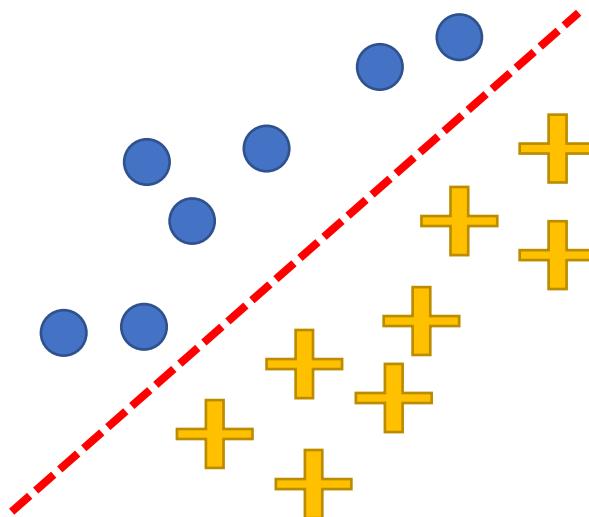
- 迴歸與分類
- 迴歸問題介紹
- 分類問題介紹

迴歸與分類

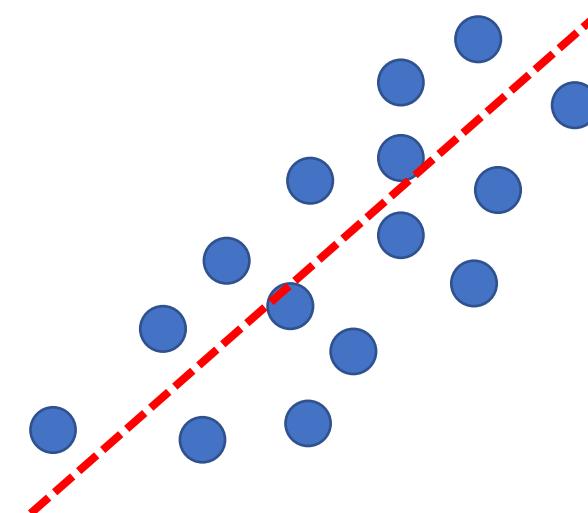
- 我們可以根據想要預測的目標來將機器學習分成兩類
 - 如果預測的目標是一個連續的數值，這樣的問題就是迴歸問題
 - 如果預測的目標是一個類別，這樣的問題就是分類問題

- 在幾何上，我們可以看到
 - 迴歸問題就是要找到一個超平面來逼近資料的趨勢
 - 分類問題就是要找一個超平面來把不同類型的資料分開來

分類問題



迴歸問題



迴歸問題介紹

- 迴歸問題最常見的案例就是房價預測
 - 紿予房屋大小、屋齡、幾間房等資訊，來機器學習來預測房價

| Size (feet ²) | Number of bedrooms | Number of floors | Age of home (years) | Price (\$1000) |
|---------------------------|--------------------|------------------|---------------------|----------------|
| 2104 | 5 | 1 | 45 | 460 |
| 1416 | 3 | 2 | 40 | 232 |
| 1534 | 3 | 2 | 30 | 315 |
| 852 | 2 | 1 | 36 | 178 |
| ... | ... | ... | ... | ... |

分類問題介紹

- 分類問題最常見的案例就是貓狗照片分類
 - 紿予一堆貓或狗的照片，用機器學習來預測那些照片是貓，那些照片是狗



- 安裝Anaconda
- 安裝jupyter notebook
- 撰寫Hello world並執行

本章重點精華回顧

- 機器學習介紹
- 四種學習方法
- 迴歸與分類問題



- Lab01: 安裝Anaconda
- Lab02: 安裝Jupyter Notebook
- Lab03: 撰寫Hello world並執行

Estimated time:
20 minutes



學習目標：

Module 02 機器學習的流程

- 2-1: 機器學習的流程
- 2-2: 衡量模型的方法
- 2-3: 機器學習的特性

Estimated time:
45 min.

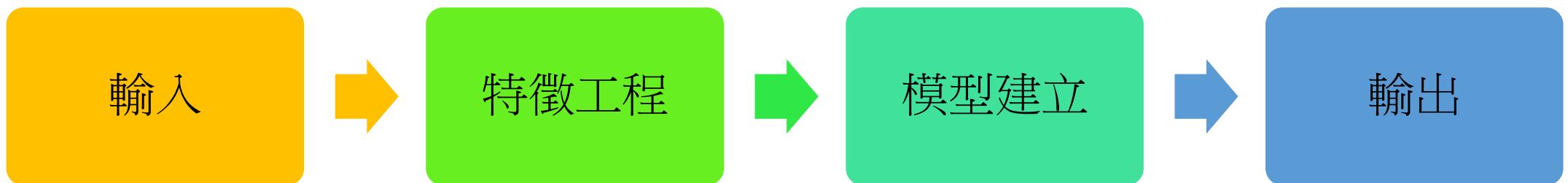
學習目標：

2-1:機器學習的流程

- 機器學習的流程
- 資料分割
- 特徵工程
- 建立模型

機器學習的流程

- 機器學習的流程如下
 - 從輸入→特徵工程→模型建立→輸出



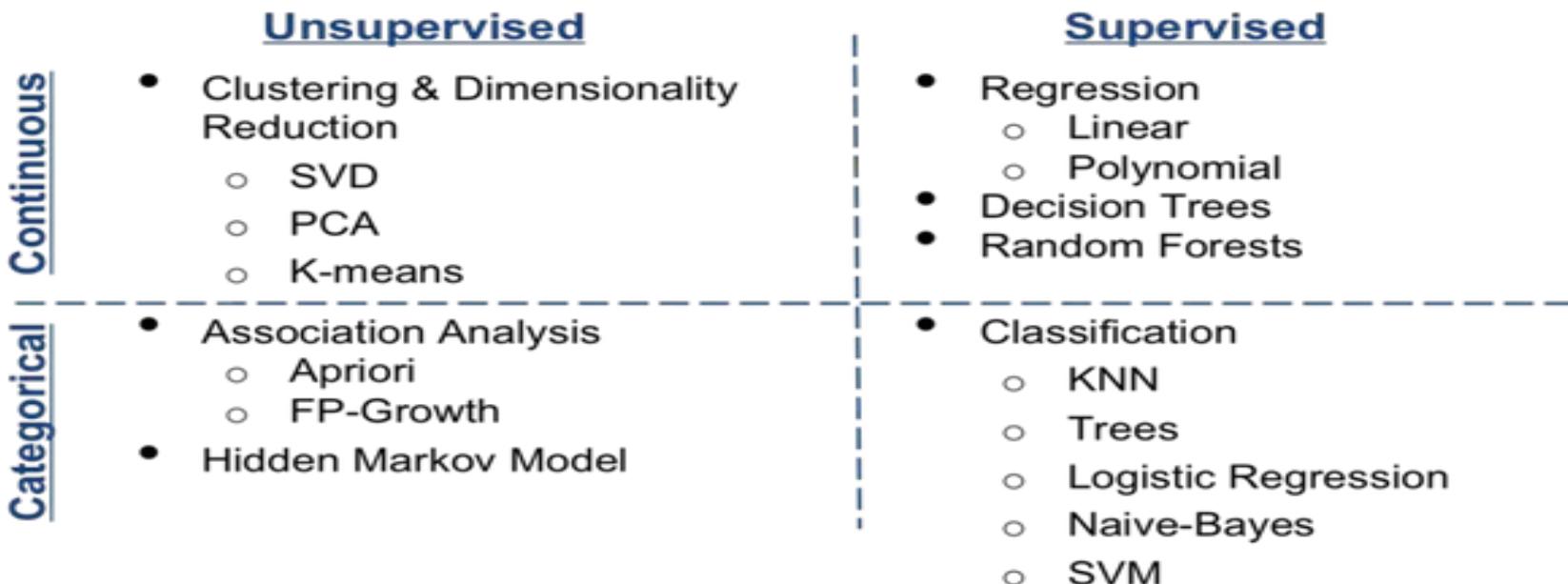
- 在輸入前，我們通常會把所有的資料分成訓練資料以及測試資料
 - 訓練資料的目的是讓電腦去學習的資料
 - 測試資料的目的是當電腦學習完以後，想要驗證它是否真的學得那麼好
- 一般來說，訓練資料與測試資料最常見的分割比例為8:2



- 所謂的特徵工程指的是我們需要將輸入的原始資料做整理，包含
 - 缺失值的處理、極端值的移除、資料分布的轉換等
- 在機器學習裡面，特徵工程是最花時間的一個步驟
 - 根據實務上的統計，特徵工程佔了整體專案的60%的時間

- 在將資料輸入到模型內前，也需要將資料做標準化
 - 標準化的目的通常都是希望將資料壓所到某的特定的區間，例如[0, 1]
- 常見的標準化方法有
 - $\frac{x_i - \mu}{\sigma}$ (Z-score normalization)
 - $\frac{x_i - x_{min}}{x_{max} - x_{min}}$ (Min-max normalization)

- 在機器學習裡，有非常多不同種類的模型可以做選擇
 - 包含監督式、非監督式等
 - 包含迴歸、分類演算法



2-2:衡量模型的方法

- 衡量迴歸模型
- 衡量分類模型

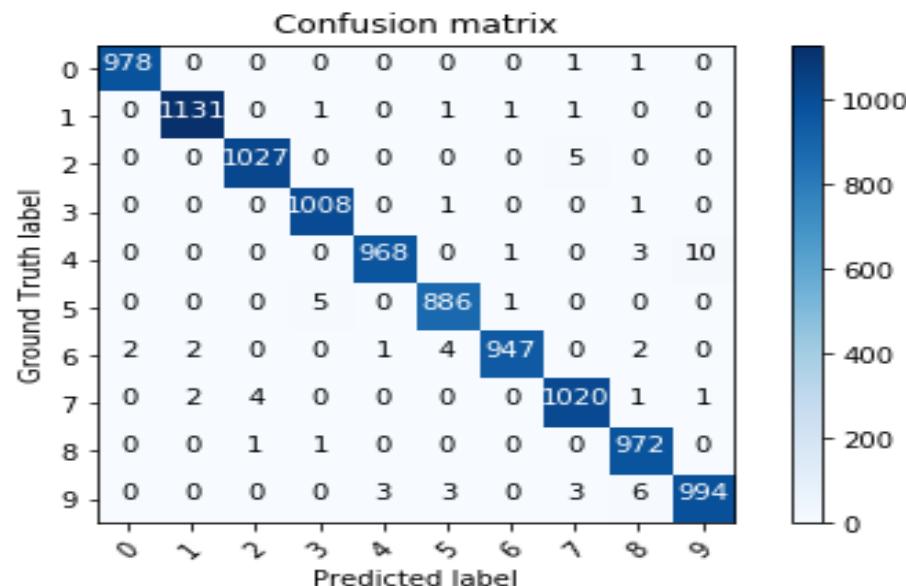
- 衡量迴歸模型的指標有非常多，常見的有
 - MSE以及 R^2 等
- MSE表示預測值f以及實際值差異平方的總和
 - 越小表示模型與實際資料越貼近
- R^2 表示預測值f與實際資料離散程度及實際資料之間離散程度的比值
 - 越接近1代表模型越與實際資料貼近

- 衡量分類模型的指標有非常多，常見的有
 - 準確度、混淆矩陣
- 準確度是最簡單衡量分類模型的指標之一
 - 將測試資料輸入並計算猜對資料的筆數除以總資料筆數

- 準確度也有人將其分得更細作探討
 - 預測正樣本且真實正樣本的True Positive、預測正樣本且真實負樣本的True Negative
 - 預測負樣本且真實正樣本的False Positive、預測負樣本且真實負樣本的False Negative



- 混淆矩陣
 - 是一種衡量多分類問題模型的指標
 - 橫軸代表測猜測的類別、縱軸代表真實的類別
 - 對角線代表猜對的部分、非對角線代表猜錯的部分



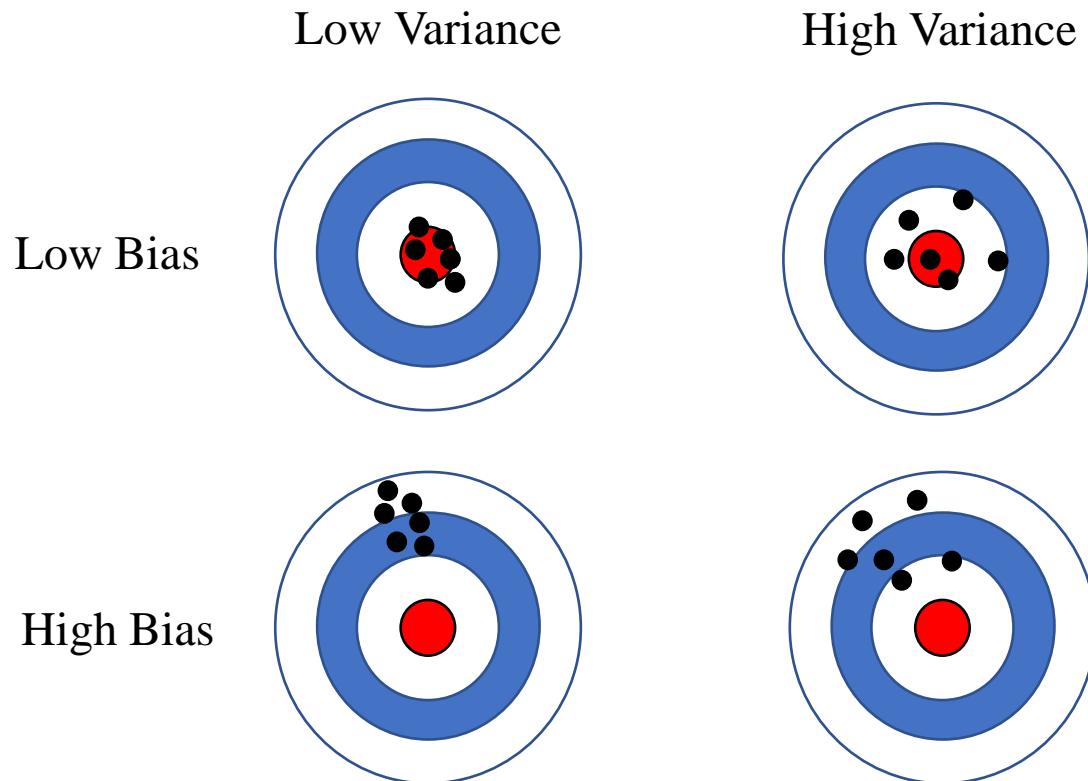
學習目標：

2-3:機器學習的特性

- Bias-Variance兩難
- 天下沒有白吃午餐
定裡

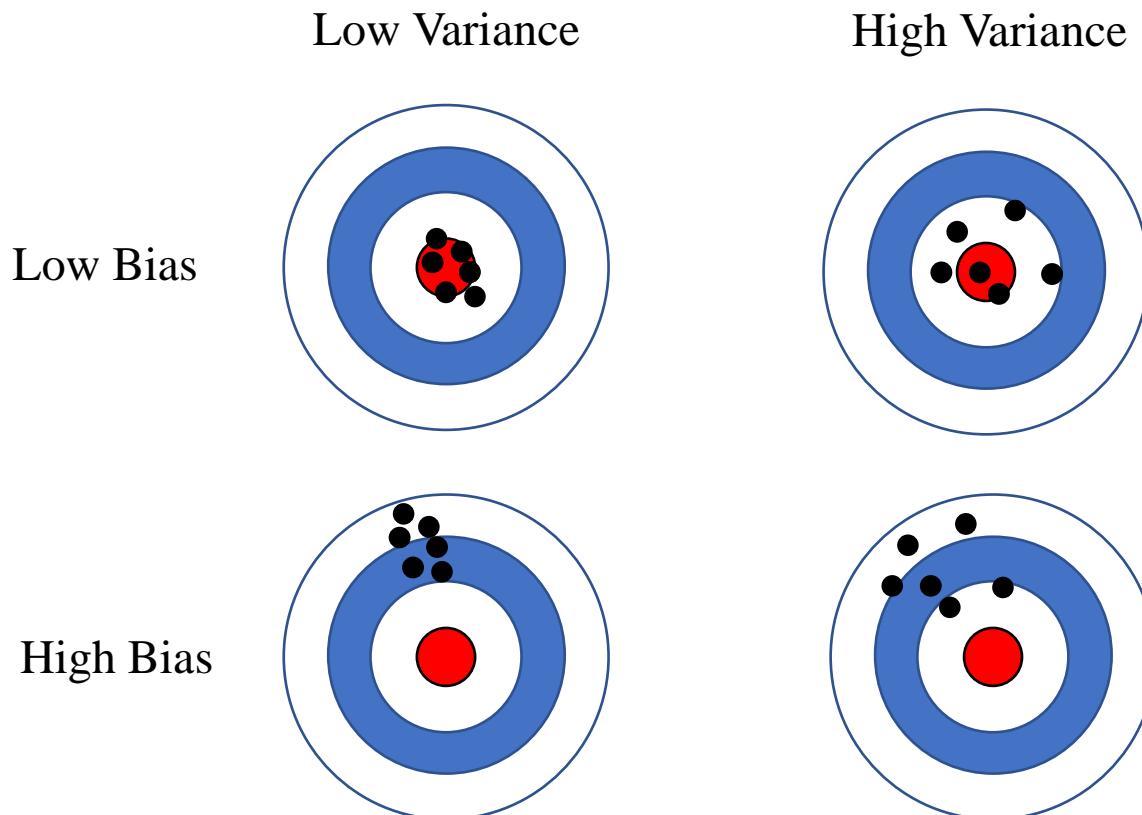
Bias-Variance兩難

- 在機器學習裡面，常常會面對Bias-Variance兩難的問題
 - Bias指的是整體平均準確度，可以看到low bias平均來說都有打到紅心
 - Variance指的是資料離散程度，可以看到low variance整理資料離散程度低

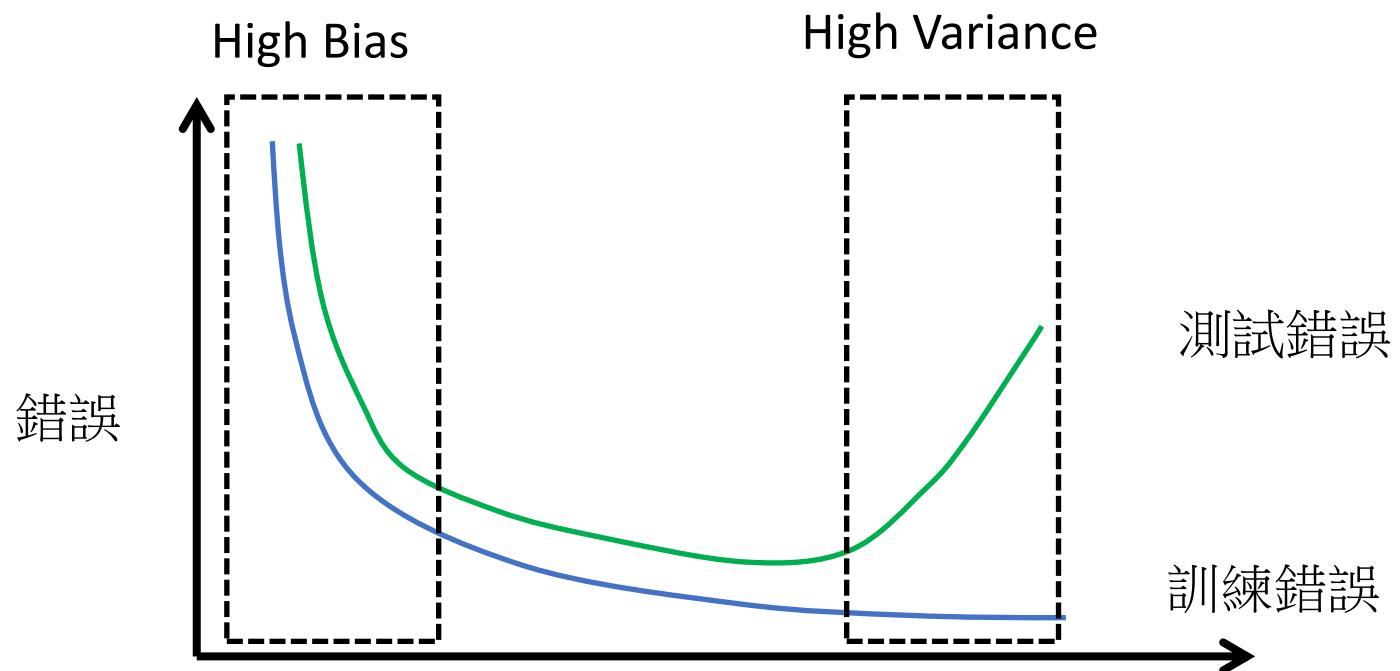


Bias-Variance兩難

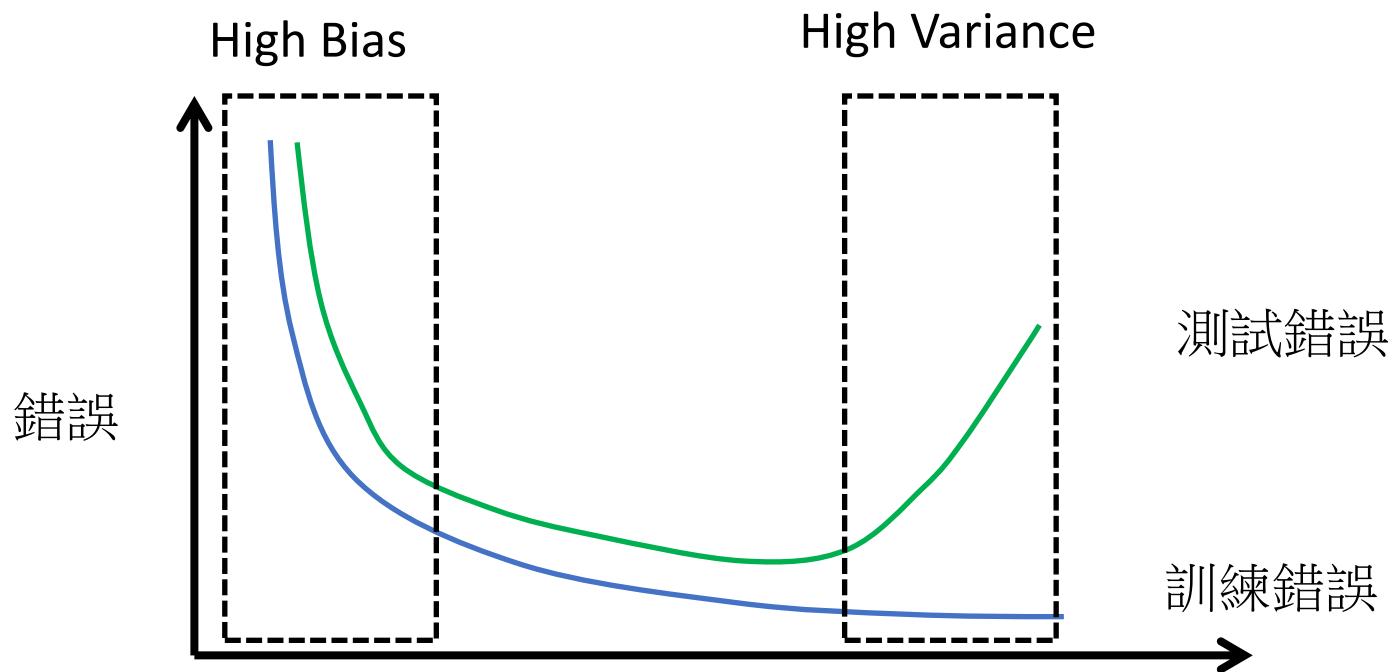
- 最理想的情況當然是low bias且low variance



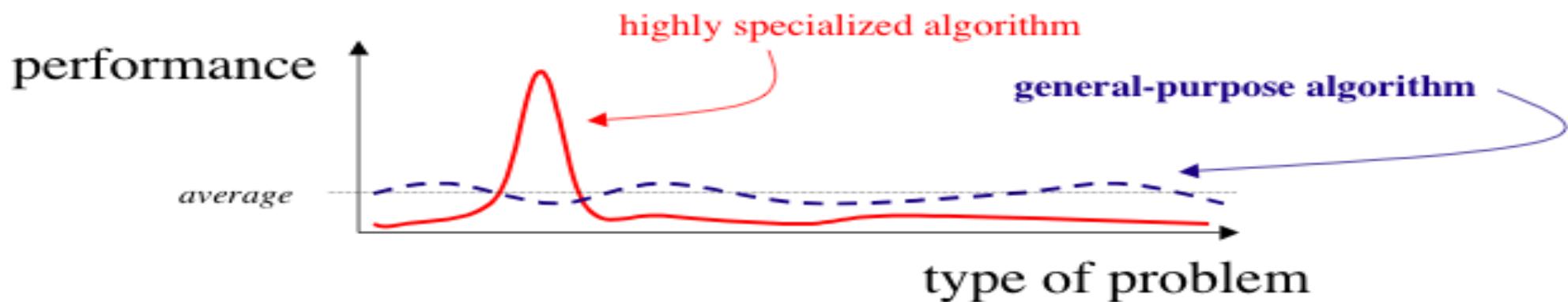
- 一般來說，在機器學習模型剛訓練的時候，模型會在high bias的區域，但當持續優化模型後，會漸漸跑到low bias區域



- 正常情況下，訓練錯誤會跟測試錯誤隨著訓練過程中持續下降，但當訓練到一段時間後，測試錯誤會突然又開始往上升
 - 這個現象叫做overfitting，也是做機器學習時非常棘手的問題
 - 模型只要overfitting，實際部屬時常常會表現不好



- 天下沒有白吃的午餐定理指的是機器學習的演算法有那麼多，但不可能找到一個機器學習演算法在所有情況下都勝過其他演算法
 - 這個已經在數學上被嚴格證明出來了
 - 越是針對某個方向優化的演算法，其在其他方向必會比較弱
 - 如果要求演算法在每個方向都很強，則會學到每個方向都很平庸的演算法



- 安裝Pandas
- 開啟Demo_2-3.ipynb
- 執行Pandas基礎操作
- 執行scikit-learn基礎操作

- 機器學習的流程
- 衡量模型的方法
- Bias-Variance兩難
- 天下沒有白吃午餐定裡



- Lab01: Pandas基礎使用
- Lab02: 安裝scikit-learn
- Lab03: scikit-learn基礎使用

Estimated time:
20 minutes



機器學習的套件

學習目標：

- 3-1: 機器學習常見套件
- 3-2: 機器學習的應用
- 3-3: 資料科學競賽平台

Estimated time:
45 min.

學習目標：

3-1:機器學習常見套件

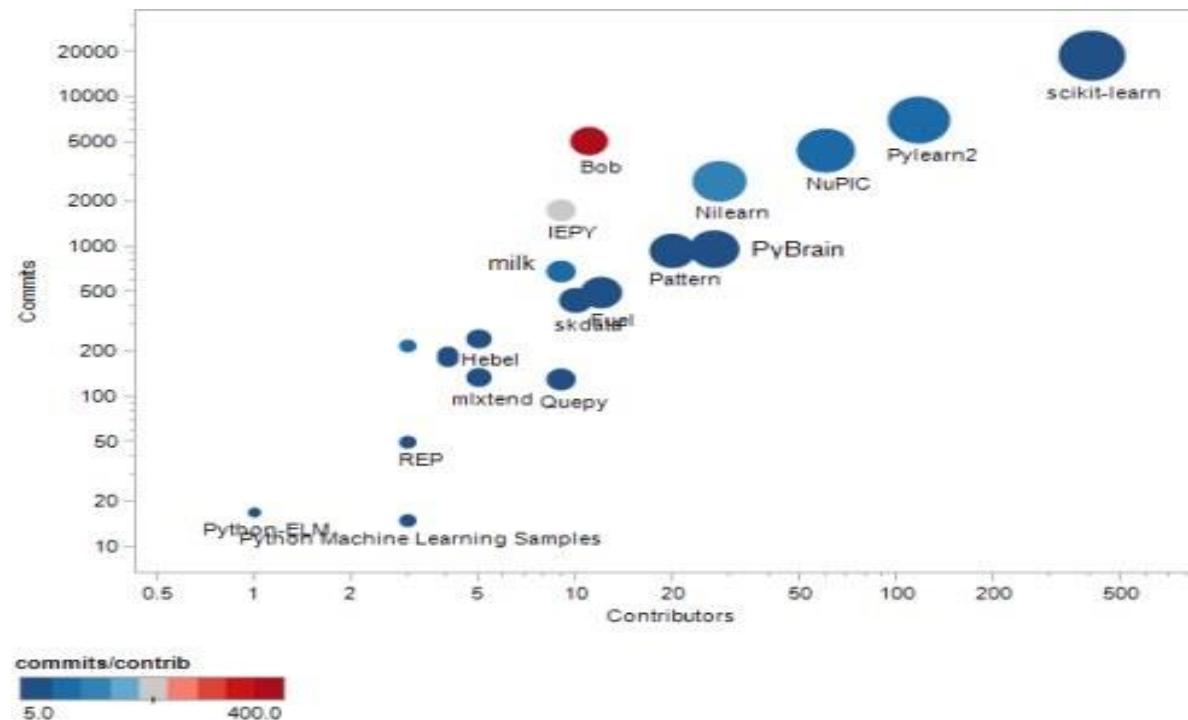
- scikit-learn介紹
- 雲端付費API介紹

- scikit-learn是一個Python程式語言的機器學習套件
 - 支援非常多不同種類的機器學習演算法，包含迴歸、分類、類聚等
 - <http://scikit-learn.org/stable/>



scikit-learn介紹

- 雖然機器學習的開源軟體非常多，但是scikit-learn目前穩居冠軍的寶座
 - 最多程式碼commit次數、最多開發者在使用



- 使用scikit-learn時，常都會搭配像是Numpy或是Pandas這類數據處理的套件一起使用
 - Numpy及Pandas負責讀寫資料、特徵工程
 - scikit-learn負責機器學習模型的部分

- 除了使用開源程式碼外，許多公司也有提供雲端付費機器學習的API讓使用者快速上手
 - 像是Microsoft Azure、Amazon AWS、Google GCP等
 - 我們這門課之後會使用Azure來讓同學操作看看

3-2:機器學習的應用

- 醫療業的應用
- 廣告行銷業的應用
- 金融業的應用
- 交通業的應用

- 在醫療業當中，我們可以用機器學習來做醫療影像辨識
 - 例如偵測病人X光照片，並用機器學離來判斷此病患是否有問題



廣告行銷業的應用

- 在廣告行銷業中，我們可以用收集來的使用者數據去預測每位使用者的偏好
 - 例如電商平台分析過使用者的偏好後，可以做更精準的行銷



- 在金融領域裡，每天有成千上萬的資料，可以用機器學習去輔助投資的決策
 - 例如某避險基金可以利用機器學習去買賣股票



交通業的應用

- 在交通領域裡，可以用機器學習影像辨識的方法，讓交通工具更聰明
 - 例如許多汽車廠商紛紛投入的自駕車研究



學習目標：

3-3:資料科學競賽平 台

- Kaggle平台介紹
- 天池平台介紹

Kaggle平台介紹

- Kaggle是一個資料科學競賽平台
 - 平時會有很多與資料科學相關的競賽在上面
 - 上面有非常多不同領域的資料集以及各公司想要解決的問題
- Google於2017年買下Kaggle這個平台

The logo for Kaggle, featuring the word "kaggle" in a lowercase, bold, sans-serif font. The letters are a vibrant blue color.

Kaggle平台介紹

- Kaggle能看到許多過往比賽的資料集以及觀看許多參賽者如何解決問題
 - 非常適合想要精進自己能力的同學善加利用
 - 同時也能觀察不同領域的企業有哪些問題想要解決

- Kaggle針對每個資料集都會有基本描述、資料集下載區域、討論區、即時排名、比賽規則等資訊

The screenshot shows the Kaggle competition page for "House Prices: Advanced Regression Techniques". At the top, there's a navigation bar with the Kaggle logo, a search bar, and a notification bell icon. Below the header, the competition title "House Prices: Advanced Regression Techniques" is displayed in bold, along with a small house icon featuring an "F SOLD E" sign. A brief description follows: "Predict sales prices and practice feature engineering, RFs, and gradient boosting". It also indicates "4,761 teams · Ongoing". Below this, a horizontal menu bar includes "Overview" (which is underlined), "Data", "Notebooks", "Discussion", "Leaderboard", and "Rules". To the right of the menu is a blue "Join Competition" button. The main content area starts with an "Overview" section. On the left, a sidebar lists "Description", "Evaluation", "Tutorials", and "Frequently Asked Questions". On the right, a large text block titled "Start here if..." explains the competition's target audience: "You have some experience with R or Python and machine learning basics. This is a perfect competition for data science students who have completed an online course in machine learning and are looking to expand their skill set before trying a featured competition." At the bottom, another section titled "Competition Description" is partially visible.

天池平台介紹

- 大陸版本的Kaggle，上面一樣有非常多的資料集以及比賽可以供大家參考
- 天池主要是由阿里巴巴所主持的



3-3 Demo

- 開啟Demo_3-3.ipynb
- 使用scikit-learn跑線性迴歸
- Kaggle平台註冊
- Kaggle加入比賽

本章重點精華回顧

- scikit-learn介紹
- 機器學習的應用
- Kaggle資料科學競賽平台



- Lab01: scikit-learn模型建立
- Lab02: Kaggle平台註冊
- Lab03: Kaggle加入比賽

Estimated time:
20 minutes



Azure認知服務

學習目標：

- 4-1: Azure認知服務介紹
- 4-2: Azure帳號設定
- 4-3: Azure工作環境建置

4-1: Azure 認知服務 介紹

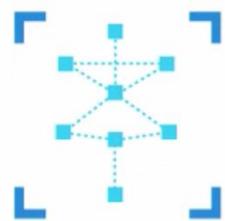
- Microsoft Azure AI 介紹
- Azure 認知服務內容
- Azure 認知服務應用

Microsoft Azure AI介紹

- Azure AI是Microsoft提供的AI雲端運算平台
 - 平台內有機器學習服務（認知服務、機器人服務、認知搜尋服務...）
 - 透過API讓開發者減少開發（訓練模型）時間
 - <https://azure.microsoft.com/zh-tw/overview/ai-platform/>



- Azure的認知服務讓所有開發人員都可以不須具備機器學習專業知識就可以使用API
- 只要呼叫API，就可以將機器學習中的視訊、音訊、語音、搜尋等等功能內嵌於應用程式中



Vision



Speech



Language



Decision



Web Search

- Azure提供的認知服務種類分五類
 - 辨識：識別及分析影像、影片和數位筆跡中的內容
 - 語音：將語音處理整合到應用程式與服務
 - 語言：從非結構化文字擷取意義
 - 決策：更快做出更聰明的決策
 - 網路內容搜尋：從全球資訊網尋找您想要的內容

- KPMG
 - 使用Azure認知服務中的語音、語意、文本分析、文本翻譯服務來分析數以萬小時的電話錄音，減少人力負擔
- Volkswagen
 - 當新文件發佈時使用文本翻譯、語意服務將所有發布平台同步翻譯成40國語言，進而保持全球文件發表的零時差
- Uber
 - 於其應用程式中應用Azure人臉辨識服務，進而確保司機與乘員的人身安全

學習目標：

4-2:Azure帳號設定

- 新增Azure帳號
- Azure帳號設定

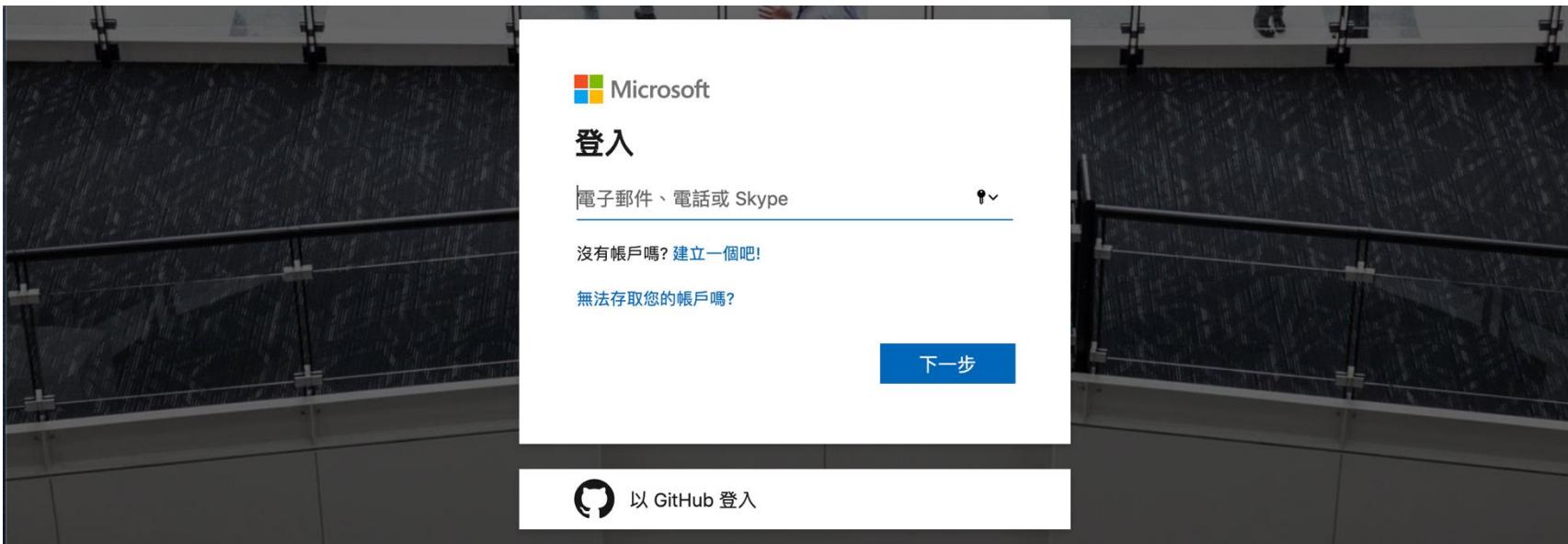
- 帳號登入

- 於瀏覽器中輸入以下網址
- <https://azure.microsoft.com/zh-tw/services/cognitive-services/>
- 並於右上角點選登入



新增Azure帳號

- 有沒有Microsoft mail、skype、Github帳號？
 - 有→直接下一步
 - 沒有→新增帳號



- 帳戶資訊填寫
 - 手機驗證
 - 信用卡資訊驗證
 - 點選同意與註冊
 - 再點選前往入口網站



免費試用 Azure
請遵循下列步驟開始使用。我們會要求提供這些詳細資料，以保護帳戶與資訊。沒有任何預付款或費用。

包含項目

- 12 個月的免費產品**
前 30 天免費存取虛擬機器、儲存體與資料庫等熱門產品。升級為隨用隨付訂閱後，則可繼續免費存取 12 個月。
- \$6,300 的點數**
前 30 天超過免費產品用量可使用您的 \$6,300 點數試用任何 Azure 服務。
- 25 項以上永遠免費的產品**
利用 25 個以上永遠免費的產品，包括無伺服器，容器及人工智慧。前 30 天可免費使用，選擇升級後，即可永遠免費使用。
- 不會自動收費**
除非選擇升級，否則無須付費。我們會在在前 30 天屆滿之前通知您，而您也能選擇升級為隨用隨付訂閱，在超過免費的額度之後，依據資源的使用量付費。

1 關於您

2 依據手機進行身分識別驗證

3 依據卡片進行身分識別驗證

4 協議

我同意訂閱合約、供應項目詳細資料和隱私權聲明。

我希望收到 Microsoft 或所選夥伴提供與 Azure 相關的資訊、提示和供應項目，包含 Azure 電子報、定價更新及其他 Microsoft 產品與服務。

註冊 正在確認您的資訊...

- 完成 Azure 帳號申請後，進入 Azure 服務的入口網站

The screenshot shows the Microsoft Azure portal homepage. At the top, there's a navigation bar with the Microsoft Azure logo, a search bar, and various icons for account management and help. Below the header, a banner displays account information: "尚餘點數 NT\$6,100.00 點" and "訂用帳戶 '免費試用版' 尚餘價值 NT\$6,100.00 元的點數。" A button says "按一下這裡以升級為隨用隨付訂用帳戶。". The main content area is titled "Azure 服務" and features several service icons: 建立資源 (New), 虛擬機器 (Virtual Machines), 應用程式服務 (App Services), 儲存體帳戶 (Storage Accounts), SQL 資料庫 (SQL Database), 適用於 PostgreSQL ... (PostgreSQL), Azure Cosmos DB, Kubernetes 服務 (Kubernetes Service), 函數應用程式 (Functions), and 更多服務 (More Services). Below this, there are sections for "瀏覽" (Browse) with links to 訂用帳戶 (Subscription), 資源群組 (Resource Groups), 所有資源 (All Resources), and 儀表板 (Dashboard); "工具" (Tools) with Microsoft Learn and Cost Management; and "實用的連結" (Useful Links) for Technical Documentation, Azure Services, and Azure Migrate Tools. In the center, a prominent white box welcomes users with "歡迎使用 Microsoft Azure" and "在您開始使用之前，展示一下內容。" It includes "開始導覽" and "稍後再說" buttons. At the bottom right, there's an "Azure 行動裝置應用程式" section with download links for the App Store and Google Play.

學習目標：

4-3: Azure工作環境建置

- 地區位置選擇
- 工作環境建置

地區位置選擇

- 雖然Azure在全球眾多地區都有服務，但服務內容略有差異



地區位置選擇

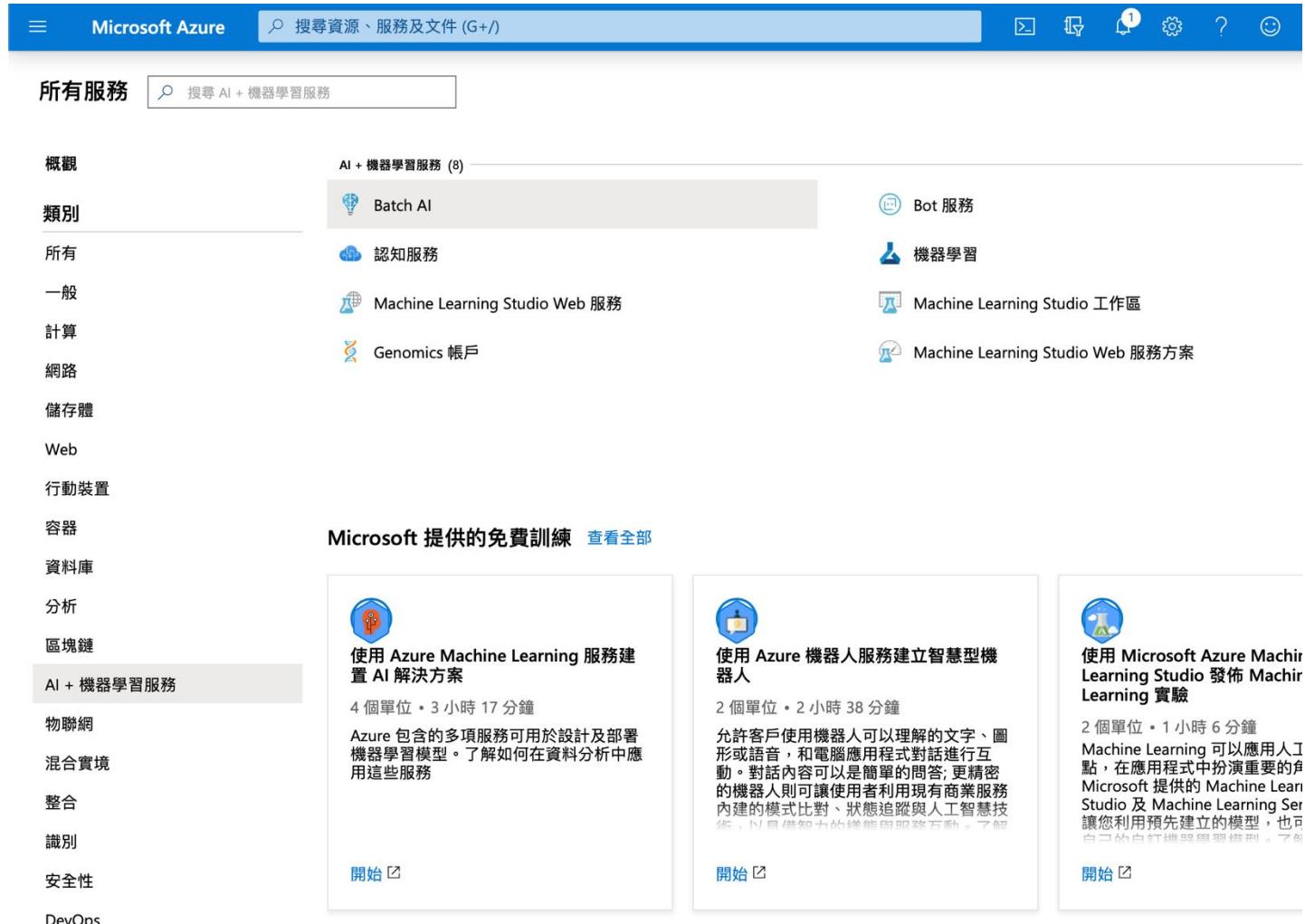
- 點選以下連結查詢所選取地區是否有所需要的服務
 - <https://azure.microsoft.com/zh-tw/global-infrastructure/services/?products=cognitive-services®ions=all>

The screenshot shows the Azure Cognitive Services page with the 'Regions' tab selected. In the sidebar, the 'AI + 機器學習服務' category is highlighted with a red box and a red arrow pointing to it. The main content area displays various Azure services under the '認知服務' category, including Azure Machine Learning, Azure Databricks, Azure Bot Service, Azure 認知搜尋, and Microsoft Genomics. On the right side, there is a '地區' (Regions) section where users can select specific regions for their Azure services. The '全選/全不選' (Select All/Unselect All) checkbox is checked. Several regions are selected, including South Korea, United States, Japan, Canada, and Asia Pacific.

| 地區 |
|----------------------|
| 非區域 |
| Azure Government |
| 中國 |
| 巴西 |
| 日本 |
| 加拿大 |
| 印度 |
| 亞太地區 |
| 法國 |
| 與 Azure Stack Hub 比較 |
| 阿拉伯聯合大公國 |
| 非洲 |
| 南韓 |
| 美國 |
| 英國 |
| 挪威 |
| 瑞士 |
| 德國 |
| 歐洲 |
| 澳洲 |

工作環境建置

- 點選建立資源
- 點選AI+機器學習服務
- 點選認知服務



The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and various icons for account management and notifications. Below the navigation bar, the main content area has a blue header bar with the text '所有服務' (All Services) and a search bar for '搜尋 AI + 機器學習服務' (Search for AI + Machine Learning Services). On the left, there is a sidebar with a tree view of service categories: 概觀 (Overview), 類別 (Category), 所有 (All), 一般 (General), 計算 (Compute), 網路 (Network), 儲存體 (Storage), Web, 行動裝置 (Mobile), 容器 (Container), 資料庫 (Database), 分析 (Analytics), 区塊鏈 (Blockchain), AI + 機器學習服務 (AI + Machine Learning Services), 物聯網 (IoT), 混合實境 (Mixed Reality), 整合 (Integration), 識別 (Identity), 安全性 (Security), and DevOps. The 'AI + 機器學習服務' category is currently selected and highlighted in grey. To the right of the sidebar, there is a list of services under the heading 'AI + 機器學習服務 (8)': Batch AI, 認知服務 (Cognitive Services), Machine Learning Studio Web 服務, Genomics 帳戶, Bot 服務, 機器學習 (Machine Learning), Machine Learning Studio 工作區, and Machine Learning Studio Web 服務方案. Below this list, there is a section titled 'Microsoft 提供的免費訓練' (Free Microsoft-provided training) with three cards: '使用 Azure Machine Learning 服務建置 AI 解決方案' (Build an AI solution using the Azure Machine Learning service), '使用 Azure 機器人服務建立智慧型機器人' (Build a intelligent robot using the Azure Bot service), and '使用 Microsoft Azure Machine Learning Studio 發佈 Machine Learning 實驗' (Publish a Machine Learning experiment using Microsoft Azure Machine Learning Studio).

- 在認知服務主控台新增、刪除已設定好的服務
 - 下一章節將會說明如何新增一個Azure認知服務

The screenshot shows the Microsoft Azure Cognitive Services management portal. The top navigation bar includes the Microsoft Azure logo, a search bar, and various navigation icons. The main title is "所有服務 > 認知服務". Below the title, there's a sub-header "認知服務" and a "預設目錄". The toolbar contains buttons for "新增" (Create), "管理檢視" (Manage Overview), "重新整理" (Refresh), "匯出至 CSV" (Export to CSV), "指派標籤" (Assign Labels), and "意見反應" (Feedback). There are also filter buttons for "依名稱篩選..." (Filter by Name), "訂用帳戶" (Subscription), "資源群組" (Resource Group), "位置" (Location), and "加入篩選" (Add Filter). The main content area displays a message: "正在顯示 0 至 0 筆記錄 (共 0 筆)" (Showing 0 to 0 records (Total 0)). Below this, there are sorting and filtering options for columns: "名稱" (Name), "種類" (Type), "位置" (Location), "自訂網域" (Custom Domain), "定價層" (Pricing Tier), "狀態" (Status), and "Created ...". A large, stylized cloud icon is centered in the middle of the page. Below the icon, the text "沒有 認知服務 可顯示" (No Cognitive Services available to display) is shown. At the bottom, there's a descriptive text: "Embed the ability to see, hear, speak, search, understand and accelerate decision-making with Azure Cognitive Services.", followed by "Learn More" and "Documentation" links, and a prominent blue "建立 認知服務" (Create Cognitive Service) button.

- 建立Azure帳戶
- 設定Azure工作環境
- 建立一個Azure服務

- Azure認知服務介紹
- Azure認知服務五種分類
- Azure認知服務計價方式與地區相關



- Lab01: 建立Azure帳戶
- Lab02: 設定Azure工作環境
- Lab03: 建立一個Azure服務

Estimated time:
20 minutes



Module 05

電腦視覺

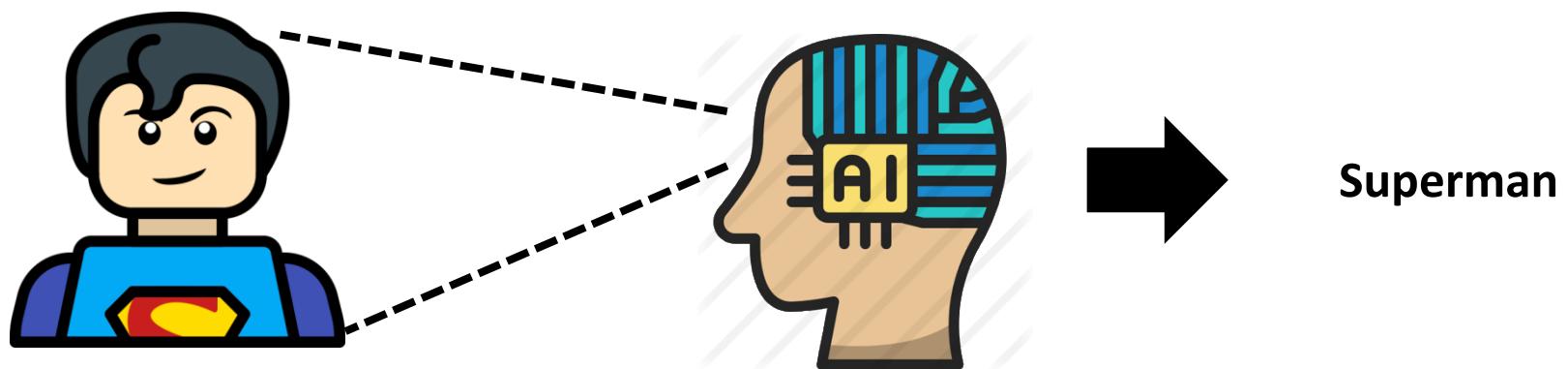
學習目標：

- 5-1: Azure 電腦視覺類別介紹
- 5-2: Azure 電腦視覺類別設定
- 5-3: 內容標記服務介紹

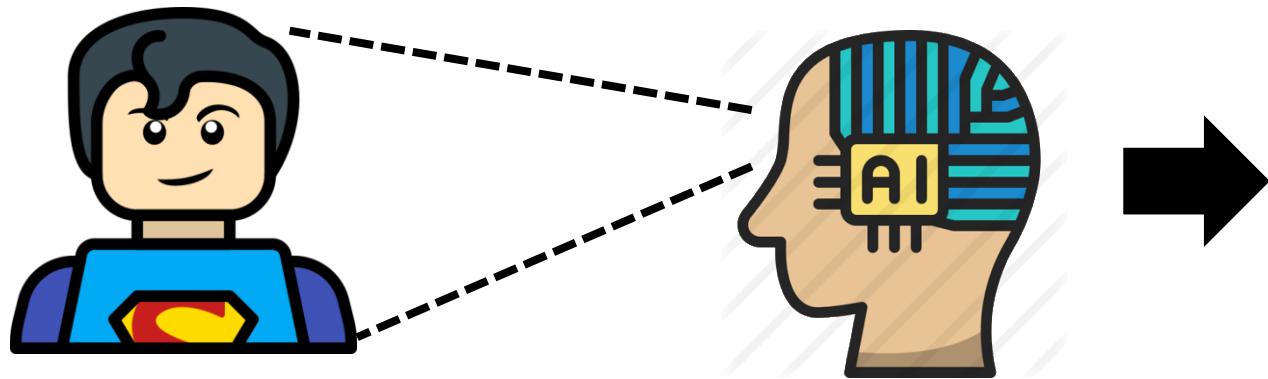
5-1:電腦視覺類別介紹

- 認識電腦視覺
- 深入分析影像
- 電腦視覺概念分類

- 電腦視覺也稱為機器視覺
 - 利用代替人眼的圖像感測器將物體的圖像轉換成數字圖像
 - 藉由機器學習方式理解和識別圖像達到分析圖像與作出結論等目的
 - 常應用於航空照片或是太空衛星照片、視頻解釋與分析



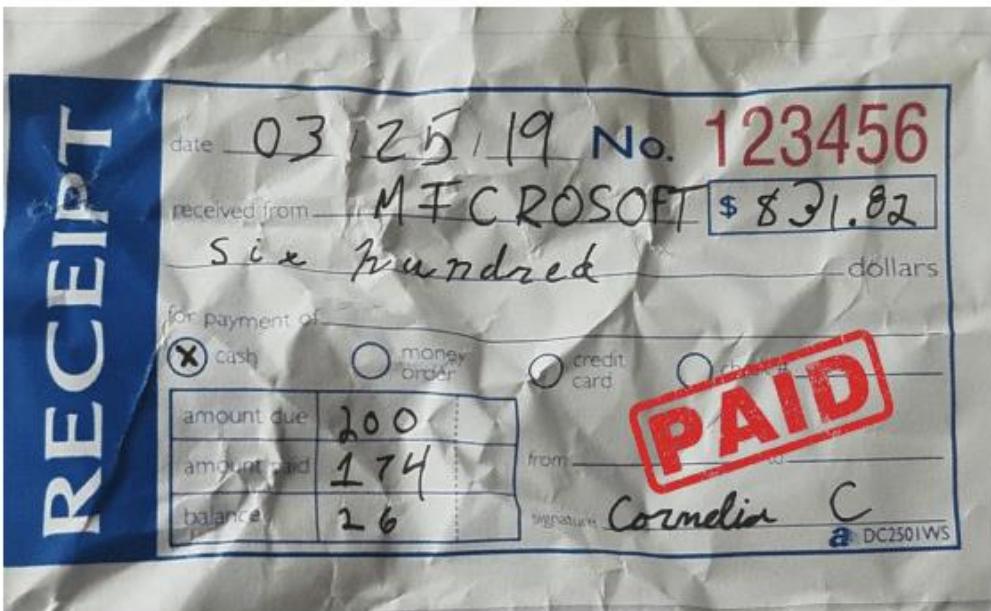
- Azure的電腦視覺服務
 - 藉由SDK或是REST API方式供開發人員存取相關演算法來分析影像
 - 電腦視覺演算法可根據您感興趣的視覺化功能，以不同的方式分析影像的內容。



Superman
Man with black hair
Blue shirt
Clip art picture
Don't have adult content
Not black & white picture

- 範例

- 以Azure網站上的範例說明，右下表格為Azure電腦視覺分析左下圖像後傳回的資料內容



| | |
|------|--|
| 物件數 | [{"rectangle": {"x": 86, "y": 56, "w": 467, "h": 343}, "object": "Office supplies", "confidence": 0.577}] |
| 標籤 | [{"name": "text", "confidence": 0.9999335}, {"name": "handwriting", "confidence": 0.9937743}] |
| 描述 | {"tags": ["text"], "captions": [{"text": "a close up of text on a white background", "confidence": 0.8143099}]}] |
| 影像格式 | "Jpeg" |
| 影像尺寸 | 430 x 558 |
| 美工圖案 | 0 |

- Azure分析影像API提供的功能有
 - 偵測並提供與其視覺特徵和特性有關的深入解析、擷取影像中的文字、調節影像中的內容
 - 標記視覺特徵、偵測物件、偵測品牌、將影像分類、說明影像、偵測臉部、偵測影像類型、偵測特定領域內容、偵測色彩配置、產生縮圖、取得關注區域等等會在後面章節一一介紹與如何使用。

學習目標：

5-2:電腦視覺類別設定

- 初探電腦視覺服務
- 開始電腦視覺服務

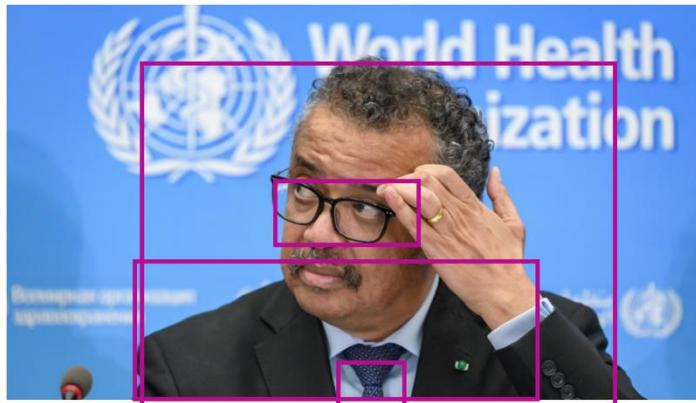
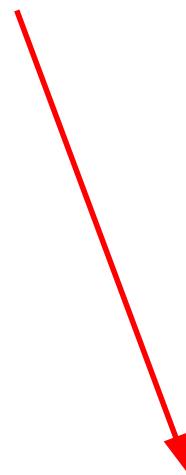
初探電腦視覺服務

- 點選以下Azure電腦視覺Demo網址
 - <https://azure.microsoft.com/zh-tw/services/cognitive-services/computer-vision/#features>



| 功能名稱 : | 值 |
|--------|---|
| 物件數 | [{ "rectangle": { "x": 93, "y": 178, "w": 115, "h": 237 }, "object": "person", "confidence": 0.764 }, { "rectangle": { "x": 0, "y": 229, "w": 101, "h": 206 }, "object": "person", "confidence": 0.624 }, { "rectangle": { "x": 161, "y": 31, "w": 439, "h": 423 }, "object": "subway train", "parent": { "object": "train", "parent": { "object": "Land vehicle", "parent": { "object": "Vehicle", "confidence": 0.926 }, "confidence": 0.923 }, "confidence": 0.917 }, "confidence": 0.801 }] |
| 標籤 | [{ "name": "train", "confidence": 0.9975446 }, { "name": "platform", "confidence": 0.9955431 }, { "name": "station", "confidence": 0.979800761 }, { "name": " |

- 試著於影像URL輸入影像的網址
 - <https://www.statnews.com/wp-content/uploads/2020/02/Coronavirus-CDC-1600x900.jpg>



影像 URL 提交

"suit", "confidence": 0.985000849 }, { "name": "tie", "confidence": 0.9640306 }, { "name": "outdoor", "confidence": 0.962736845 }, { "name": "clothing", "confidence": 0.892711163 }, { "name": "wearing", "confidence": 0.881226242 }, { "name": "glasses", "confidence": 0.652098656 }, { "name": "text", "confidence": 0.6495545 }, { "name": "sunglasses", "confidence": 0.614187241 }, { "name": "white-collar worker", "confidence": 0.5253712 }, { "name": "older", "confidence": 0.491324157 }]

描述 { "tags": ["person", "man", "suit", "outdoor", "wearing", "glasses", "looking", "sunglasses", "front", "holding", "older", "talking", "standing", "old", "phone", "gray", "giving", "shirt", "sign", "hat", "blue"], "captions": [{ "text": "a man wearing a suit and tie", "confidence": 0.98805 }] }

- 試著於影像URL輸入影像的網址
 - <https://www.statnews.com/wp-content/uploads/2020/02/Coronavirus-CDC-1600x900.jpg>
 - 但也不是每次都很精準



影像 URL 提交

| | |
|-------|---|
| 功能名稱： | 值 |
| 物件數 | 0 |
| 標籤 | [{ "name": "cake", "confidence": 0.976046 }, { "name": "fruit", "confidence": 0.728532732 }, { "name": "reef", "confidence": 0.5530258 }, { "name": "pattern", "confidence": 0.5472472 }, { "name": "plant", "confidence": 0.3064567 }, { "name": "colored", "confidence": 0.212851048 }] |
| 描述 | { "tags": ["cake", "fruit", "chocolate"], "captions": [{ "text": "a close up of a cake", "confidence": 0.654741049 }] } |

開始電腦視覺服務

- 於認知服務主控台點選新增
 - 進到Marketplace後選擇AI + 機器學習服務
 - 右側Cognitive Services區域選擇查看更多
 - 點選電腦視覺圖像



A screenshot of the Azure Marketplace search results for 'Cognitive Services'. The results are categorized under 'Bot Service' and 'Cognitive Services'. Under 'Bot Service', there are two items: 'Web App Bot' and 'Bot Channels Registration'. Under 'Cognitive Services', there are four items: 'Bing 自動建議', 'Bing 自訂搜尋', 'Bing 實體搜尋', and 'Bing 搜尋'. Each item has a brief description and a 'View details' button.

• 建立電腦視覺服務

- 點選建立
- 填寫資料：名稱、位置、定價層、新建資源群組

電腦視覺
Microsoft

實用的連結
[深入了解電腦視覺 API](#)
[文件](#)
[API 參考](#)
[定價](#)
[區域可用性](#)

所有服務 > 認知服務 > Marketplace > 電腦視覺 > 建立

建立

電腦視覺

名稱 *

ComputerVision

訂用帳戶 *

免費試用版

位置 *

(US) 美國中南部

定價層 ([檢視完整定價詳細資料](#)) *

資源群組 *

新建

開始電腦視覺服務

- 完成電腦視覺服務
 - 完成部署後可以在認知服務中控台看到建立的服務內容

The screenshot displays two main sections of the Azure portal:

Top Section (Deployment Status):

- Page Title: Microsoft.CognitiveServicesComputerVision | 概觀
- Section: 部署
- Message: 部署正在進行中... (Deployment is in progress...) - 下午12:52
正在部署至資源群組 'CogAI'。
- Search Bar: 搜尋 (Cmd +/)
- Buttons: 刪除, 取消, 重新部署, 重新整理
- Section: 概觀
- Information: 部署名稱: Microsoft.CognitiveServicesComputerVision 開始時間: 22/3/2020 下午12:52:34
訂用帳戶: 免費試用版
資源群組: CogAI
- Section: 部署詳細資料 (下載)
- Table Headers: 資源, 類型, 狀態, 作業詳細資料
- Text: 沒有結果。
- Links: 資訊安全中心, 免費 Microsoft 教學課程

Bottom Section (Cognitive Services Blade):

- Page Title: 首頁 > 認知服務
- Section: 認知服務
- Buttons: 新增, 管理檢視, 重新整理, 匯出至 CSV, 指派標籤, 意見反應
- Filters: 依名稱篩選..., 訂用帳戶 == 全部, 資源群組 == 全部, 位置 == 全部, 加入篩選
- Text: 正在顯示 1 至 1 筆記錄 (共 1 筆)。
- Table Headers: 種類 ↑↓, 位置 ↑↓, 自訂網域名稱 ↑↓, 定價層 ↑↓, 狀態 ↑↓, Created ... ↑↓
- Data Row: moduleComputerVision, ComputerVision, 東南亞, modulecomputervision, F0, Succeeded, 2020-03-22T0...

學習目標：

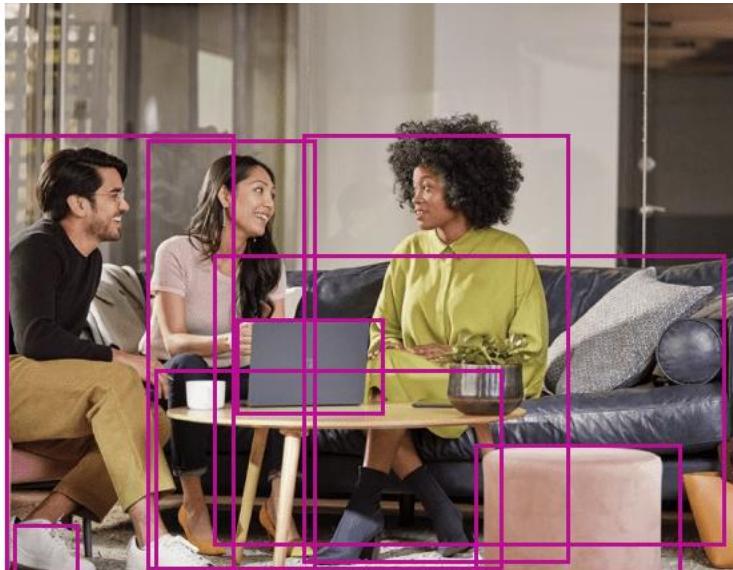
5-3:內容標記服務介紹

- 內容標記服務介紹
- 使用內容標記服務

內容標記服務介紹

- 定義：

- 在上傳影像或指定影像 URL 之後，電腦視覺的演算法會根據在影像中識別出的物件、生物和動作來輸出標記。
- 標記內容包含周遭環境 (室內或室外)、家具、工具、植物、動物、配件和小工具等。
- 目前僅支援英文。



| | |
|----|--|
| 標籤 | <pre>[{"name": "person", "confidence": 0.994588}, {"name": "laptop", "confidence": 0.9619576}, {"name": "computer", "confidence": 0.9575019}, {"name": "clothing", "confidence": 0.9374012}, {"name": "furniture", "confidence": 0.9298235}, {"name": "indoor", "confidence": 0.9139405}, {"name": "table", "confidence": 0.7971237}, {"name": "chair", "confidence": 0.722082436}, {"name": "people", "confidence": 0.7205929}, {"name": "woman", "confidence": 0.543776631}]</pre> |
| 描述 | <pre>{"tags": ["person", "laptop", "indoor", "computer", "sitting", "table", "man", "people", "group", "woman", "front", "using", "room", "living", "talking", "desk", "young", "food", "restaurant", "train", "standing"], "captions": [{"text": "a group of people sitting at a table using a laptop", "confidence": 0.9693954}]}]</pre> |
| 影像 | "Jpeg" |

使用內容標記服務

- 影像標記範例
- 將內容標記套用到影像
 - 下列 JSON 回應說明了在範例影像中偵測到標記視覺功能時，電腦視覺傳回的內容。



```
{  
  "tags": [  
    {  
      "name": "grass",  
      "confidence": 0.999995231628418  
    },  
    {  
      "name": "outdoor",  
      "confidence": 0.99992108345031738  
    },  
    {  
      "name": "house",  
      "confidence": 0.99685388803482056  
    },  
    {  
      "name": "sky",  
      "confidence": 0.99532157182693481  
    },  
    {  
      "name": "building",  
      "confidence": 0.99436837434768677  
    },  
    {  
      "name": "tree",  
      "confidence": 0.98880356550216675  
    },  
    {  
      "name": "lawn",  
      "confidence": 0.788884699344635  
    },  
    {  
      "name": "green",  
      "confidence": 0.71250593662261963  
    },  
    {  
      "name": "residential",  
      "confidence": 0.70859086513519287  
    },  
    {  
      "name": "grassy",  
      "confidence": 0.46624681353569031  
    }  
  "requestId": "06f39352-e445-42dc-96fb-0a1288ad9cf1",  
  "metadata": {  
    "height": 200,  
    "width": 300,  
    "format": "Jpeg"  
  }  
}
```

- 安裝用戶端程式庫
 - pip install --upgrade azure-cognitiveservices-vision-computervision
- 建立Python應用程式
 - 載入相關套件

```
from azure.cognitiveservices.vision.computervision import ComputerVisionClient
from azure.cognitiveservices.vision.computervision.models import TextOperationStatusCodes
from azure.cognitiveservices.vision.computervision.models import TextRecognitionMode
from azure.cognitiveservices.vision.computervision.models import VisualFeatureTypes
from msrest.authentication import CognitiveServicesCredentials

from array import array
import os
from PIL import Image
import sys
import time
```

使用內容標記服務

- 取得Azure金鑰與端點

- 到認知服務主控台
複製API key與端點



- 完成環境變數設定

- Windows :

```
setx COGNITIVE_SERVICE_KEY "your-key"    key取代 "your-key" 後輸入
```

- 重新開啟所有需要環境變數的程式或是重新開機

使用內容標記服務

- 設定API key

```
# Add your Computer Vision subscription key to your environment variables.
if 'COMPUTER_VISION_SUBSCRIPTION_KEY' in os.environ:
    subscription_key = os.environ['COMPUTER_VISION_SUBSCRIPTION_KEY']
else:
    print("\nSet the COMPUTER_VISION_SUBSCRIPTION_KEY environment variable.\n**Restart your shell or IDE for changes")
    sys.exit()
# Add your Computer Vision endpoint to your environment variables.
if 'COMPUTER_VISION_ENDPOINT' in os.environ:
    endpoint = os.environ['COMPUTER_VISION_ENDPOINT']
else:
    print("\nSet the COMPUTER_VISION_ENDPOINT environment variable.\n**Restart your shell or IDE for changes")
    sys.exit()
```

- 驗證用戶端

```
computervision_client = ComputerVisionClient(endpoint, CognitiveServicesCredentials(subscription_key))
```

- 欲分析影像 U R L

```
remote_image_url = "https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/koala.jpg"
```

- 取得影像標籤

```
...
Tag an Image - remote
This example returns a tag (key word) for each thing in the image.
...
print("===== Tag an image - remote =====")
# Call API with remote image
tags_result_remote = computervision_client.tag_image(remote_image_url )

# Print results with confidence score
print("Tags in the remote image: ")
if (len(tags_result_remote.tags) == 0):
    print("No tags detected.")
else:
    for tag in tags_result_remote.tags:
        print("{}'{}' with confidence {:.2f}%".format(tag.name, tag.confidence * 100))
```

5-3 Demo

- 開啟Demo_5-3.ipynb
- 執行Azure電腦視覺Web服務操作
- 使用網路URL執行內容標記服務操作
- 上傳本機圖檔執行內容標記服務操作

- Azure電腦視覺介紹
- Azure電腦視覺分析影像方法
- 學習不同來源圖檔的內容標記服務操作



- Lab01: 操作電腦視覺Web服務
- Lab02: 建立Azure電腦視覺服務
- Lab03: 使用內容標記服務

Estimated time:
20 minutes



Module 06

電腦視覺服務應用1

學習目標：

- 6-1:物件偵測服務介紹
- 6-2:品牌偵測服務介紹
- 6-3:影像分類服務介紹

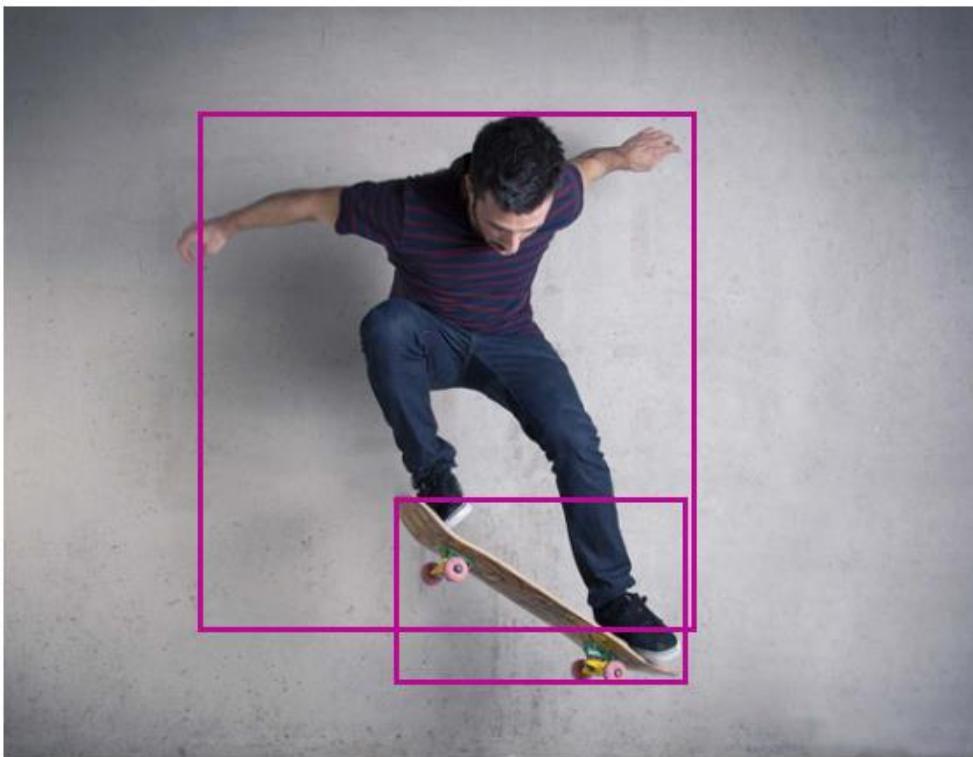
學習目標：

6-1:物件偵測服務介紹

- 物件偵測服務介紹
- 使用物件偵測服務

物件偵測服務介紹

- 物件偵測會傳回每個物件（狗、貓或人物）的週框方塊座標（以像素為單位）。



| 功能名稱： | 值 |
|-------|---|
| 物件數 | [{ "rectangle": { "x": 238, "y": 299, "w": 177, "h": 117 }, "object": "Skateboard", "confidence": 0.903 }, { "rectangle": { "x": 118, "y": 63, "w": 305, "h": 321 }, "object": "person", "confidence": 0.955 }] |
| 標籤 | [{ "name": "skating", "confidence": 0.999951541 }, { "name": "snowboarding", "confidence": 0.990067363 }, { "name": "sports equipment", "confidence": 0.9774853 }, { "name": "person", "confidence": 0.9605776 }, { "name": "roller skating", "confidence": 0.945730746 }, { "name": "boardsport", "confidence": 0.9242261 }, { "name": "man", "confidence": 0.9188208 }, { "name": "outdoor", "confidence": 0.9107821 }, { "name": "riding", "confidence": 0.900007248 }, { "name": "skiing", "confidence": 0.894337356 }, { "name": "footwear", "confidence": 0.8788208 }, { "name": "sport", "confidence": 0.86974 }, { "name": "skateboard", "confidence": 0.86974 }] |

物件偵測服務介紹

- 物件偵測範例
 - 偵測影像中的一般物件並回傳 JSON內容



```
{  
  "objects": [  
    {  
      "rectangle": {  
        "x": 730,  
        "y": 66,  
        "w": 135,  
        "h": 85  
      },  
      "object": "kitchen appliance",  
      "confidence": 0.501  
    },  
    {  
      "rectangle": {  
        "x": 523,  
        "y": 377,  
        "w": 185,  
        "h": 46  
      },  
      "object": "computer keyboard",  
      "confidence": 0.51  
    }  
  ],  
  {  
    "rectangle": {  
      "x": 471,  
      "y": 218,  
      "w": 289,  
      "h": 226  
    },  
    "object": "Laptop",  
    "confidence": 0.85,  
    "parent": {  
      "object": "computer",  
      "confidence": 0.851  
    }  
  },  
  {  
    "rectangle": {  
      "x": 654,  
      "y": 0,  
      "w": 584,  
      "h": 473  
    },  
    "object": "person",  
    "confidence": 0.855  
  }  
],  
  "requestId": "a7fde8fd-cc18-4f5f-99d3-897dcd07b308",  
  "metadata": {  
    "width": 1260,  
    "height": 473,  
    "format": "Jpeg"  
  }  
}
```

- 限制
 - 物件占全影像大小低於 5%
 - 若物件位置十分相近則通常偵測不到（例如，一疊盤子）
 - 無法以品牌或產品名稱區分物件（例如貨架上有不同種類的汽水）。
- 下一章會介紹使用品牌偵測來從影像中取得品牌資訊。

- 如內容標記服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 完成環境變數設定
 - 於Python應用程式中設定API key
 - 於Python應用程式驗證用戶端
 - 於Python應用程式加入欲分析影像URL

- 偵測物件
 - 下列程式碼會偵測影像中的一般物件，並將其輸出到主控台。

```
...
Detect Brands - remote
This example detects common brands like logos and puts a bounding box around them.
...
print("===== Detect Brands - remote =====")
# Get a URL with a brand logo
remote_image_url = "https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/images/gray-sh Image url
# Select the visual feature(s) you want
remote_image_features = ["brands"]
# Call API with URL and features
detect_brands_results_remote = computervision_client.analyze_image(remote_image_url, remote_image_features)

print("Detecting brands in remote image: ")
if len(detect_brands_results_remote.brands) == 0:
    print("No brands detected.")
else:
    for brand in detect_brands_results_remote.brands:
        print("{}'{}' brand detected with confidence {:.1f}% at location {}, {}, {}, {}".format( \
            brand.name, brand.confidence * 100, brand.rectangle.x, brand.rectangle.x + brand.rectangle.w, \
            brand.rectangle.y, brand.rectangle.y + brand.rectangle.h))
```

學習目標：

6-2: 品牌偵測服務介紹

- 品牌偵測服務介紹
- 使用品牌偵測服務

- 品牌偵測範例1

- 偵測影像中的熱門品牌
- 下列 JSON 回應說明了在範例影像中偵測到品牌時，電腦視覺傳回的內容。



```
"brands": [  
  {  
    "name": "Microsoft",  
    "rectangle": {  
      "x": 20,  
      "y": 97,  
      "w": 62,  
      "h": 52  
    }  
  }  
]
```

- 品牌偵測範例2
 - 品牌圖案與品牌名稱視為兩個不同的標誌。



```
"brands": [  
  {  
    "name": "Microsoft",  
    "rectangle": {  
      "x": 58,  
      "y": 106,  
      "w": 55,  
      "h": 46  
    }  
  },  
  {  
    "name": "Microsoft",  
    "rectangle": {  
      "x": 58,  
      "y": 86,  
      "w": 202,  
      "h": 63  
    }  
  }  
]
```

- 如內容標記服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 完成環境變數設定
 - 於Python應用程式中設定API key
 - 於Python應用程式驗證用戶端
 - 於Python應用程式加入欲分析影像URL

- 偵測品牌
 - 下列程式碼會偵測影像中的公司品牌和標誌，並將其輸出到主控台。

```
...
Detect Brands - remote
This example detects common brands like logos and puts a bounding box around them.
...
print("===== Detect Brands - remote =====")
# Get a URL with a brand logo
remote_image_url = "https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/images/gray-st
# Select the visual feature(s) you want
remote_image_features = ["brands"]
# Call API with URL and features
detect_brands_results_remote = computervision_client.analyze_image(remote_image_url, remote_image_features)

print("Detecting brands in remote image: ")
if len(detect_brands_results_remote.brands) == 0:
    print("No brands detected.")
else:
    for brand in detect_brands_results_remote.brands:
        print("{}'{}' brand detected with confidence {:.1f}% at location {}, {}, {}, {}".format( \
            brand.name, brand.confidence * 100, brand.rectangle.x, brand.rectangle.x + brand.rectangle.w, \
            brand.rectangle.y, brand.rectangle.y + brand.rectangle.h))
```

Image url

學習目標：

6-3:影像分類服務介紹

- 影像分類服務介紹
- 使用影像分類服務

影像分類服務介紹

- 依主題分類影像
 - 回傳影像中偵測到的類型（parent與child）與類別（86個英文類別）。
 - 類別部分如右圖所示



- 影像分類範例
 - 下列 JSON 回應說明根據視覺功能進行範例影像分類時，電腦視覺傳回的內容。



```
{  
  "categories": [  
    {  
      "name": "people_",  
      "score": 0.81640625  
    }  
  ],  
  "requestId": "bae7f76a-1cc7-4479-8d29-48a694974705",  
  "metadata": {  
    "height": 200,  
    "width": 300,  
    "format": "Jpeg"  
  }  
}
```

映像分類服務介紹

- 下列是一般影像傳回的類別。

類別: **people_group**



類別: **animal_dog**



影像分類服務介紹

- 下列是一般影像傳回的類別。

類別: **outdoor_mountain**



類別: **food_bread**



- 如內容標記服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 完成環境變數設定
 - 於Python應用程式中設定API key
 - 於Python應用程式驗證用戶端
 - 於Python應用程式加入欲分析影像URL

- 取得影像類別
 - 下列程式碼會取得已偵測的影像類別。

```
...
Categorize an Image - remote
This example extracts (general) categories from a remote image with a confidence score.
...
print("===== Categorize an image - remote =====")
# Select the visual feature(s) you want.
remote_image_features = ["categories"]
# Call API with URL and features
categorize_results_remote = computervision_client.analyze_image(remote_image_url , remote_image_features)

# Print results with confidence score
print("Categories from remote image: ")
if (len(categorize_results_remote.categories) == 0):
    print("No categories detected.")
else:
    for category in categorize_results_remote.categories:
        print('{{}} with confidence {:.2f}%'.format(category.name, category.score * 100))
```

6-3 Demo

- 開啟Demo_6-3.ipynb
- 執行物件偵測服務操作
- 執行品牌偵測服務操作
- 執行影像分類服務操作

- 了解如何使用物件偵測服務
- 了解如何使用品牌偵測服務
- 了解如何使用影像分類服務



- Lab01: 執行物件偵測服務操作
- Lab02: 執行品牌偵測服務操作
- Lab03: 執行影像分類服務操作

Estimated time:
20 minutes



Module 07

電腦視覺服務應用2

學習目標：

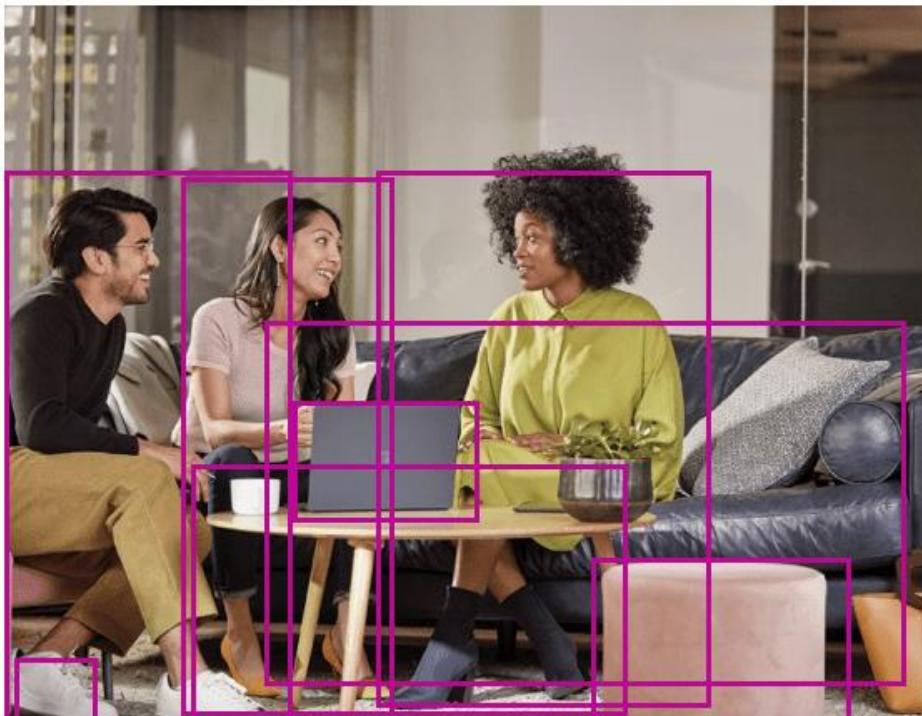
- 7-1:影像說明服務介紹
- 7-2:臉部偵測服務介紹
- 7-3:影像類別偵測服務介紹

學習目標：

7-1:影像說明服務介紹

- 影像說明服務介紹
- 使用影像說明服務

- 分析影像後產生易讀的句子來描述影像內容
 - 根據不同的視覺特徵傳回數個描述, 每個描述都有信賴分數。
 - 描述清單排列順序為信賴分數最高到最低。



| | |
|------|---|
| | { "name": "clothing", "confidence": 0.9374012 }, { "name": "furniture", "confidence": 0.9298235 }, { "name": "indoor", "confidence": 0.9139405 }, { "name": "table", "confidence": 0.7971237 }, { "name": "chair", "confidence": 0.722082436 }, { "name": "people", "confidence": 0.7205929 }, { "name": "woman", "confidence": 0.543776631 }] |
| 描述 | { "tags": ["person", "laptop", "indoor", "computer", "sitting", "table", "man", "people", "group", "woman", "front", "using", "room", "living", "talking", "desk", "young", "food", "restaurant", "train", "standing"], "captions": [{ "text": "a group of people sitting at a table using a laptop", "confidence": 0.9693954 }] } |
| 影像格式 | Jpeg |
| 影像尺寸 | 430 x 558 |

- 影像描述範例

- 將影像描述成人類可讀語言
- 下列 JSON 回應說明了在根據視覺功能描述範例影像時，電腦視覺傳回的內容。



```
{  
  "description": {  
    "tags": ["outdoor", "building", "photo", "city", "white", "black", \  
            "large", "sitting", "old", "water", "skyscraper", "many", \  
            "boat", "river", "group", "street", "people", "field", \  
            "tall", "bird", "standing"],  
    "captions": [  
      {  
        "text": "a black and white photo of a city",  
        "confidence": 0.95301952483304808  
      },  
      {  
        "text": "a black and white photo of a large city",  
        "confidence": 0.94085190563213816  
      },  
      {  
        "text": "a large white building in a city",  
        "confidence": 0.93108362931954824  
      }  
    ],  
    "requestId": "b20bfc83-fb25-4b8d-a3f8-b2a1f084b159",  
    "metadata": {  
      "height": 300,  
      "width": 239,  

```

- 如內容標記服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 完成環境變數設定
 - 於Python應用程式中設定API key
 - 於Python應用程式驗證用戶端
 - 於Python應用程式加入欲分析影像URL

使用影像說明服務

- 下列程式碼會取得為影像產生的tags清單。

```
'''  
Describe an Image - remote  
This example describes the contents of an image with the confidence score.print("===== Describe an image - remote =====")  
# Call API  
description_results = computervision_client.describe_image(remote_image_url )  
  
# Get the captions (descriptions) from the response, with confidence level  
print("Description of remote image: ")  
if (len(description_results.captions) == 0):  
    print("No description detected.")  
else:  
    for caption in description_results.captions:  
        print('{} with confidence {:.2f}%'.format(caption.text, caption.confidence * 100))|
```

學習目標：

7-2:臉部偵測服務介紹

- 臉部偵測服務介紹
- 使用臉部偵測服務

- 電腦視覺會偵測影像中的人臉，並針對偵測到的臉部產生年齡、性別和矩形。
- 臉部偵測範例1
 - 下列範例使用包含單一人臉的影像，示範電腦視覺傳回的 JSON 回應。



```
{  
  "faces": [  
    {  
      "age": 23,  
      "gender": "Female",  
      "faceRectangle": {  
        "top": 45,  
        "left": 194,  
        "width": 44,  
        "height": 44  
      }  
    }  
  ],  
  "requestId": "8439ba87-de65-441b-a0f1-c85913157ecd",  
  "metadata": {  
    "height": 200,  
    "width": 300,  
    "format": "Png"  
  }  
}
```

- 臉部偵測範例2
 - 接下來的範例使用包含多張人臉的影像來示範傳回的 JSON 回應。



```
{  
  "faces": [  
    {  
      "age": 11,  
      "gender": "Male",  
      "faceRectangle": {  
        "top": 62,  
        "left": 22,  
        "width": 45,  
        "height": 45  
      }  
    },  
    {  
      "age": 11,  
      "gender": "Female",  
      "faceRectangle": {  
        "top": 127,  
        "left": 240,  
        "width": 42,  
        "height": 42  
      }  
    }  
  ].
```

```
{  
  "age": 37,  
  "gender": "Female",  
  "faceRectangle": {  
    "top": 55,  
    "left": 200,  
    "width": 41,  
    "height": 41  
  }  
},  
{  
  "age": 41,  
  "gender": "Male",  
  "faceRectangle": {  
    "top": 45,  
    "left": 103,  
    "width": 39,  
    "height": 39  
  }  
}  
],  
"requestId": "3a383cbe-1a05-4104-9ce7-1b5cf352b239",  
"metadata": {  
  "height": 230,  
  "width": 300,  
  "format": "Png"  
}
```

- 如內容標記服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 完成環境變數設定
 - 於Python應用程式中設定API key
 - 於Python應用程式驗證用戶端
 - 於Python應用程式加入欲分析影像URL

- 偵測臉部

- 下列程式碼會傳回影像中偵測到的臉部及其矩形座標，然後選取臉部特性。

```
...
Detect Faces - remote
This example detects faces in a remote image, gets their gender and age,
and marks them with a bounding box.
...
print("===== Detect Faces - remote =====")
# Get an image with faces
remote_image_url_faces = "https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/faces.jpg"
# Select the visual feature(s) you want.
remote_image_features = ["faces"]
# Call the API with remote URL and features
detect_faces_results_remote = computervision_client.analyze_image(remote_image_url_faces, remote_image_features)

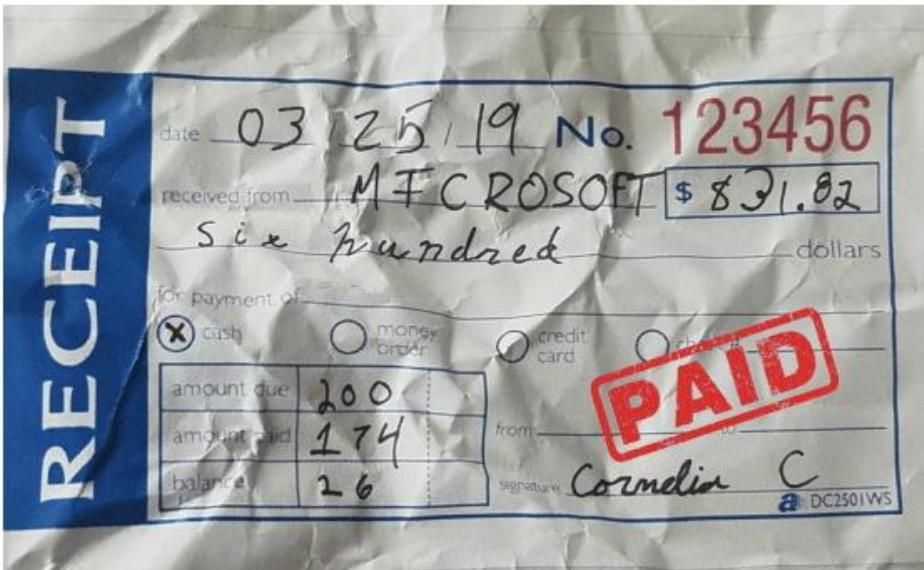
# Print the results with gender, age, and bounding box
print("Faces in the remote image: ")
if (len(detect_faces_results_remote.faces) == 0):
    print("No faces detected.")
else:
    for face in detect_faces_results_remote.faces:
        print("{}'{}' of age {} at location {}, {}, {}, {}".format(face.gender, face.age, \
face.face_rectangle.left, face.face_rectangle.top, \
face.face_rectangle.left + face.face_rectangle.width, \
face.face_rectangle.top + face.face_rectangle.height))
```

Image url

7-3:影像類別偵測服務介紹

- 影像類別偵測服務介紹
- 使用影像類別偵測服務

- 分析影像的內容, 指出影像多大機率為美工圖案 (clip art) 以及同時是否為線條繪圖 (line drawing) 。
- 偵測美工圖案
 - 依 0 到 3 的等級評估該影像是美工圖案的可能性



| 格式 | 值 | 意義 |
|----------------|-----------|--------|
| 影像 | 430 x 558 | |
| 尺寸 | 0 | 非美工圖案 |
| 美工 圖案 類型 | 1 | 不明確 |
| 線條 繪圖 類型 | 2 | 一般美工圖案 |
| 黑與 白 | 3 | 良好美工圖案 |

- 美工圖案偵測範例1

- 下列 JSON 回應說明了當評估該範例影像是否為美工圖案的可能性時，電腦視覺傳回的內容。



```
{  
    "imageType": {  
        "clipArtType": 3,  
        "lineDrawingType": 0  
    },  
    "requestId": "88c48d8c-80f3-449f-878f-6947f3b35a27",  
    "metadata": {  
        "height": 225,  
        "width": 300,  
        "format": "Jpeg"  
    }  
}
```

- 美工圖案偵測範例2

- 下列 JSON 回應說明了當評估該範例影像是否為美工圖案的可能性時，電腦視覺傳回的內容。



```
{  
    "imageType": {  
        "clipArtType": 0,  
        "lineDrawingType": 0  
    },  
    "requestId": "a9c8490a-2740-4e04-923b-e8f4830d0e47",  
    "metadata": {  
        "height": 200,  
        "width": 300,  
        "format": "Jpeg"  
    }  
}
```

- 偵測線條繪畫
 - 電腦視覺分析影像並傳回布林值，指出影像是否為線條繪畫。
- 線條繪畫偵測範例1
 - 下列 JSON 回應說明了在指出該範例影像是否為線條繪圖時，電腦視覺傳回的內容。



```
{  
    "imageType": {  
        "clipArtType": 2,  
        "lineDrawingType": 1  
    },  
    "requestId": "6442dc22-476a-41c4-aa3d-9ceb15172f01",  
    "metadata": {  
        "height": 268,  
        "width": 300,  
        "format": "Jpeg"  
    }  
}
```

- 線條繪畫偵測範例2

- 下列 JSON 回應說明了在指出該範例影像是否為線條繪圖時，電腦視覺傳回的內容。



```
{  
  "imageType": {  
    "clipArtType": 0,  
    "lineDrawingType": 0  
  },  
  "requestId": "98437d65-1b05-4ab7-b439-7098b5fdfcbf",  
  "metadata": {  
    "height": 200,  
    "width": 300,  
    "format": "Jpeg"  
  }  
}
```

使用影像類別偵測服務

- 如內容標記服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 完成環境變數設定
 - 於Python應用程式中設定API key
 - 於Python應用程式驗證用戶端
 - 於Python應用程式加入欲分析影像URL

使用影像類別偵測服務

- 取得影像類型
 - 下列程式碼可列印影像類型相關資訊—不論是美工圖案或線條繪圖。

```
...
Detect Image Types - remote
This example detects an image's type (clip art/line drawing).
...
print("===== Detect Image Types - remote =====")
# Get URL of an image with a type
remote_image_url_type = "https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/clipart/line_drawing_cat.jpg"
# Select visual feature(s) you want
remote_image_features = VisualFeatureTypes.image_type
# Call API with URL and features
detect_type_results_remote = computervision_client.analyze_image(remote_image_url_type, remote_image_features)

# Prints type results with degree of accuracy
print("Type of remote image:")
if detect_type_results_remote.image_type.clip_art_type == 0:
    print("Image is not clip art.")
elif detect_type_results_remote.image_type.line_drawing_type == 1:
    print("Image is ambiguously clip art.")
elif detect_type_results_remote.image_type.line_drawing_type == 2:
    print("Image is normal clip art.")
else:
    print("Image is good clip art.")

if detect_type_results_remote.image_type.line_drawing_type == 0:
    print("Image is not a line drawing.")
else:
    print("Image is a line drawing")
```

Image url

- 開啟Demo_7-3.ipynb
- 執行影像說明服務操作
- 執行臉部偵測服務操作
- 執行影像類別偵測服務操作

- 了解如何使用影像說明服務
- 了解如何使用臉部偵測服務
- 了解如何使用影像類別偵測服務



- Lab01: 執行影像說明服務操作
- Lab02: 執行臉部偵測服務操作
- Lab03: 執行影像類別偵測服務操作

Estimated time:
20 minutes



Module 08

電腦視覺服務應用3

學習目標：

- 8-1:特定領域偵測服務介紹
- 8-2:色彩配置偵測服務介紹
- 8-3:智慧裁切縮圖服務介紹

學習目標：

8-1:特定領域偵測服務介紹

- 特定領域偵測服務介紹
- 使用特定領域偵測服務

- 偵測特定領域內容
 - 使用影像分類86類別以外的模型來分析。例：landmarks
 - https://upload.wikimedia.org/wikipedia/commons/c/c2/01_khafre_north.jpg



```
{  
  "result": {  
    "landmarks": [{  
      "name": "Pyramid of Khafre",  
      "confidence": 0.970619797706604  
    }]  
  },  
  "requestId": "7daa04b7-3d99-4fb6-a106-99f02a2be530",  
  "metadata": {  
    "width": 667,  
    "height": 500,  
    "format": "Jpeg"  
  }  
}
```

- 如內容標記服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 完成環境變數設定
 - 於Python應用程式中設定API key
 - 於Python應用程式驗證用戶端
 - 於Python應用程式加入欲分析影像URL

- 取得特定領域內容
 - 下列程式碼以會以celebrities類別來分析影像。

```
...
Detect Domain-specific Content - remote
This example detects celebrites and landmarks in remote images.
...
print("===== Detect Domain-specific Content - remote =====")
# URL of one or more celebrities
remote_image_url_celebs = "https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/
# Call API with content type (celebrities) and URL
detect_domain_results_celebs_remote = computervision_client.analyze_image_by_domain("celebrities", \
                                         remote_image_url_celebs)

# Print detection results with name
print("Celebrities in the remote image:")
if len(detect_domain_results_celebs_remote.result["celebrities"]) == 0:
    print("No celebrities detected.")
else:
    for celeb in detect_domain_results_celebs_remote.result["celebrities"]:
        print(celeb["name"])

```

Image url

使用特定領域偵測服務

- 取得特定領域內容
 - 下列程式碼會以landmarks類別來分析影像。

```
# Call API with content type (landmarks) and URL
detect_domain_results_landmarks = computervision_client.analyze_image_by_domain("landmarks", remote_image_url)
print()

print("Landmarks in the remote image:")
if len(detect_domain_results_landmarks.result["landmarks"]) == 0:
    print("No landmarks detected.")
else:
    for landmark in detect_domain_results_landmarks.result["landmarks"]:
        print(landmark["name"])
```

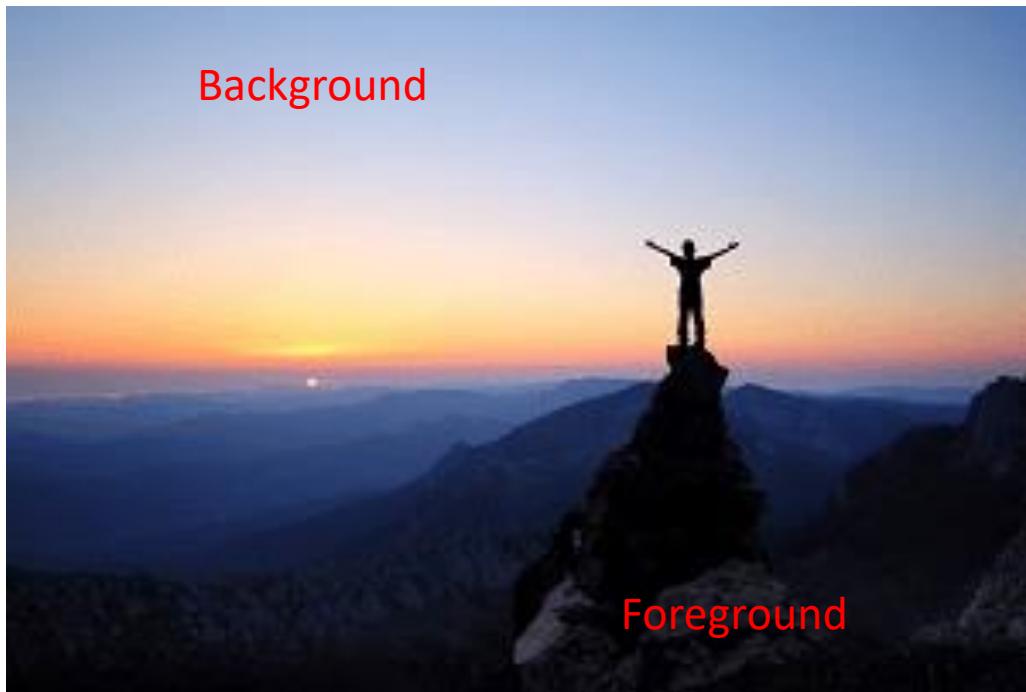
學習目標：

8-2:色彩配置偵測服務介紹

- 色彩配置偵測服務介紹
- 使用色彩配置偵測服務

- 色彩配置偵測範例

- 傳回的JSON如下。雖然不是黑白影像，但主要前景和背景色彩為黑色，而且整體影像的主要色彩為黑色和白色。



```
{  
  "color": {  
    "dominantColorForeground": "Black",  
    "dominantColorBackground": "Black",  
    "dominantColors": ["Black", "White"],  
    "accentColor": "BB6D10",  
    "isBwImg": false  
  },  
  "requestId": "0dc394bf-db50-4871-bdcc-13707d9405ea",  
  "metadata": {  
    "height": 202,  
    "width": 300,  
    "format": "Jpeg"  
  }  
}
```

色彩配置偵測服務介紹

- 主要色彩範例
 - 傳回每個範例影像的前景、背景及影像色彩。



前景：黑色
後景：白皮書
色彩：黑色、白色、綠色



前景：黑色
後景：黑色
色彩：黑色

色彩配置偵測服務介紹

- 輔色（互補色）範例

- 互補色指的是成對的顏色，組合後互相抵消產生白色或是黑色。
- 傳回每個範例影像的輔色 (十六進位 HTML 色彩值)。



"accentColor": "BB6D10"



"accentColor": "C6A205"



"accentColor": "474A84"

色彩配置偵測服務介紹

- 黑白偵測範例
 - 下表顯示電腦視覺對範例影像的黑白評估。



“isBwImg”: True



“isBwImg”: False

使用色彩配置偵測服務

- 如內容標記服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 完成環境變數設定
 - 於Python應用程式中設定API key
 - 於Python應用程式驗證用戶端
 - 於Python應用程式加入欲分析影像URL

使用色彩配置偵測服務

- 取得影像色彩配置
 - 下列程式碼會列印影像中偵測到的色彩特性，例如主色和輔色。

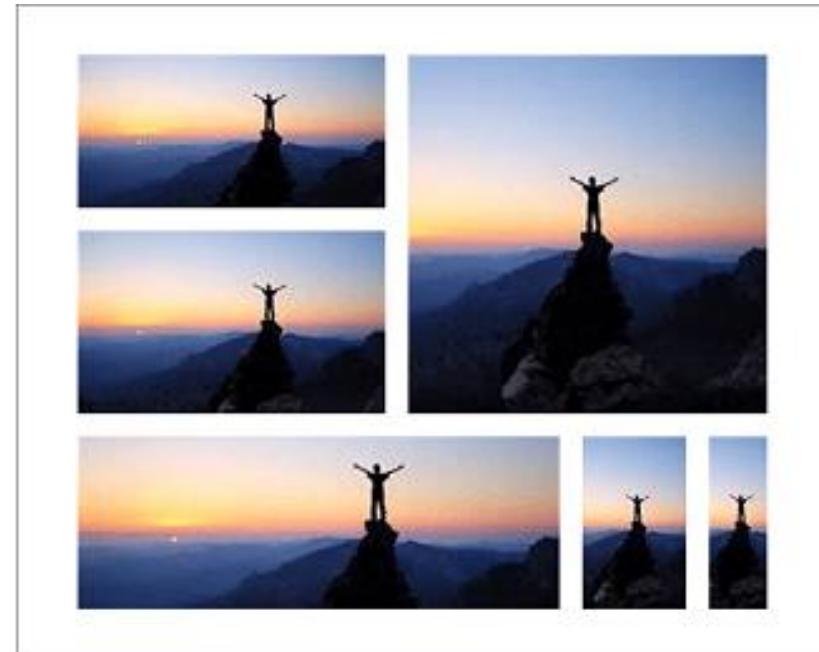
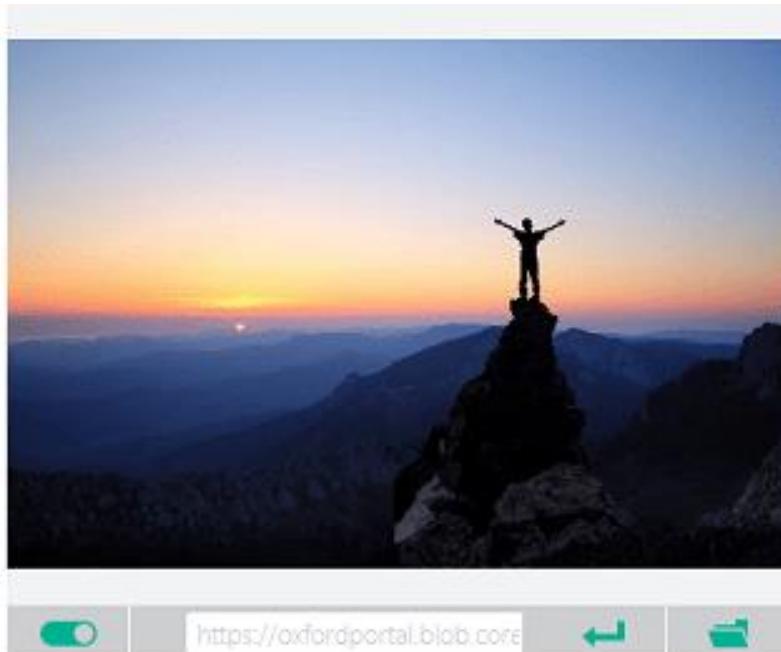
```
'''  
Detect Color - remote  
This example detects the different aspects of its color scheme in a remote image.print("===== Detect Color - remote =====")  
# Select the feature(s) you want  
remote_image_features = ["color"]  
# Call API with URL and features  
detect_color_results_remote = computervision_client.analyze_image(remote_image_url, remote_image_features)  
  
# Print results of color scheme  
print("Getting color scheme of the remote image: ")  
print("Is black and white: {}".format(detect_color_results_remote.color.is_bw_img))  
print("Accent color: {}".format(detect_color_results_remote.color.accent_color))  
print("Dominant background color: {}".format(detect_color_results_remote.color.dominant_color_background))  
print("Dominant foreground color: {}".format(detect_color_results_remote.color.dominant_color_foreground))  
print("Dominant colors: {}".format(detect_color_results_remote.color.dominant_colors))
```

學習目標：

8-3:智慧裁切縮圖服務介紹

- 智慧裁切縮圖服務介紹
- 使用智慧裁切縮圖服務

- 使用電腦視覺產生智慧裁剪的縮圖
 - 縮圖是大小縮小的影像表示。
 - 電腦視覺 API 會使用智慧裁剪，以及搭配影像大小調整，來建立特定影像的直覺式縮圖。



智慧裁切縮圖服務介紹

- 針對範例影像所產生的縮圖。
 - 目標高度和寬度指定為 50 像素，並啟用智慧裁剪功能所產生的縮圖。

Image



縮圖



- 如內容標記服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件

```
import requests
# If you are using a Jupyter notebook, uncomment the following line.
# %matplotlib inline
import matplotlib.pyplot as plt
from PIL import Image
from io import BytesIO
```

- 完成環境變數設定
- 於Python應用程式中設定API key
- 於Python應用程式驗證用戶端

- 使用電腦視覺 REST API 從影像產生縮圖。
 - 指定所需的高度和寬度，開啟智慧型裁剪。

```
thumbnail_url = endpoint + "vision/v2.1/generateThumbnail"

# Set image_url to the URL of an image that you want to analyze.
image_url = "https://upload.wikimedia.org/wikipedia/commons/9/94/Bloodhound_Puppy.jpg"

headers = {'Ocp-Apim-Subscription-Key': subscription_key}
params = {'width': '50', 'height': '50', 'smartCropping': 'true'}
data = {'url': image_url}
response = requests.post(thumbnail_url, headers=headers,
                         params=params, json=data)
response.raise_for_status()

thumbnail = Image.open(BytesIO(response.content))

# Display the thumbnail.
plt.imshow(thumbnail)
plt.axis("off")

# Verify the thumbnail size.
print("Thumbnail is {0}-by-{1}".format(*thumbnail.size))
```

- 開啟Demo_8-3.ipynb
- 執行特定領域偵測服務操作
- 執行色彩配置偵測服務操作
- 執行智慧裁切縮圖服務操作

- 了解如何使用特定領域偵測服務
- 了解如何使用色彩配置偵測服務
- 了解如何使用智慧裁切縮圖服務



- Lab01: 特定領域偵測服務操作
- Lab02: 色彩配置偵測服務操作
- Lab03: 智慧裁切縮圖服務操作

Estimated time:
20 minutes



Module 09

電腦視覺服務應用4

學習目標：

- 9-1:印刷和手寫文字辨識服務介紹
- 9-2:成人內容偵測服務介紹
- 9-3:電腦視覺總結

學習目標：

9-1:印刷和手寫文字 辨識服務介紹

- 印刷和手寫文字辨識服務介紹
- 使用印刷和手寫文字辨識服務

- 辨識印刷和手寫文字
 - 偵測影像中所顯示的列印、手寫文字。例如手寫筆記、醫療記錄。



- 下列三節說明三種不同的文字辨識 API

```

Nutrition Facts Amount Per Serving
[144.0, 0.0, 1238.0, 211.0, 1224.0, 280.0, 130.0, 57.0]
Serving size: 1 bar (40g)
[110.0, 58.0, 598.0, 157.0, 587.0, 206.0, 100.0, 108.0]
Serving Per Package: 4
[83.0, 108.0, 548.0, 206.0, 538.0, 256.0, 72.0, 157.0]
Total Fat 13g
[683.0, 213.0, 1000.0, 286.0, 989.0, 332.0, 672.0, 260.0]
Saturated Fat 1.5g
[695.0, 295.0, 1120.0, 394.0, 1108.0, 447.0, 683.0, 347.0]
Amount Per Serving
[29.0, 207.0, 491.0, 309.0, 478.0, 367.0, 16.0, 265.0]
Trans Fat 0g
[668.0, 363.0, 954.0, 435.0, 940.0, 488.0, 655.0, 416.0]
calories 190
[8.0, 293.0, 265.0, 350.0, 254.0, 396.0, 0.0, 339.0]
Cholesterol 0mg
[593.0, 424.0, 1007.0, 526.0, 993.0, 580.0, 579.0, 479.0]
ories from Fat 110
[9.0, 377.0, 398.0, 464.0, 388.0, 509.0, 0.0, 421.0]
Sodium 20mg
[561.0, 497.0, 913.0, 588.0, 899.0, 643.0, 547.0, 552.0]
nt Daily Values are based on
[7.0, 476.0, 521.0, 598.0, 511.0, 640.0, 0.0, 518.0]
Vitamin A 50%
[525.0, 597.0, 776.0, 657.0, 766.0, 699.0, 514.0, 640.0]
calorie diet.
[12.0, 534.0, 196.0, 576.0, 187.0, 615.0, 4.0, 574.0]

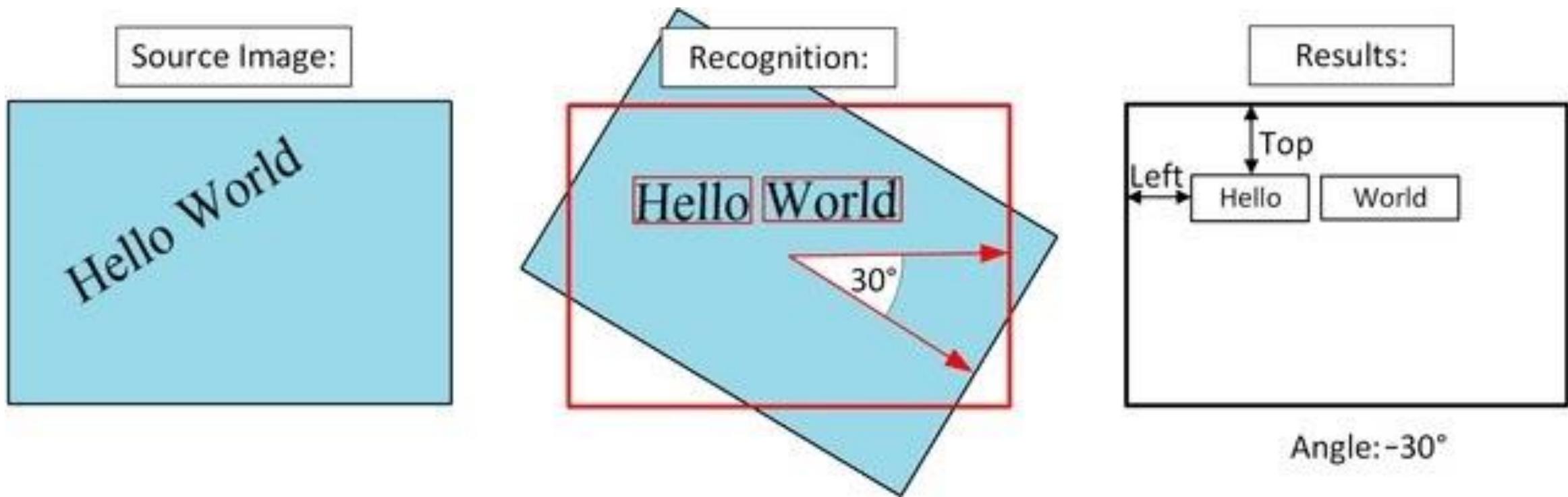
```

- 讀取API
 - 偵測影像中的文字內容，將識別的文字轉換成電腦可讀取的字元資料流。
 - 適合用於文字繁重的影像（例如已進行數位掃描的檔），以及適用於具有許多視覺雜訊的影像。
 - 每一行都有周框方塊座標，而且線條內的每個單字也有自己的座標。
 - 僅適用僅適用於英文和西班牙文（預覽）文字。

- 影像需求
 - JPEG、PNG、BMP、PDF 或 TIFF 格式。
 - 影像維度介於 50×50 與 10000×10000 圖元之間。
 - PDF 頁面小於或等於 17×17 英寸。
 - 檔案大小必須小於 20 mb
- 限制
 - 免費層的訂用帳戶只處理 PDF 或 TIFF 檔的前兩個頁面。
 - 付費訂用帳戶時，最多可處理200個頁面。
 - 每個頁面最多300行。

- OCR (光學字元辨識) API

- OCR 會以角度為單位傳回以水平影像座標軸為依據的旋轉位移，來修正偏移選轉的文字。OCR 也會提供每個字的框架座標，如下圖所示。



- 影像需求
 - 僅支援JPEG、PNG、GIF 或 BMP 格式。
 - 影像大小介於 50 x 50 與 4200 x 4200 圖元之間。
 - 影像中的文字能以 90 度的任何倍數進行旋轉，並可輔以不超過 40 度的小角度旋轉。
- 限制
 - 在主要由文字構成的相片上，部分辨識的文字可能會造成誤判。
 - 在某些相片（特別是沒有任何文字的相片）上，精確度可能會隨著影像類型而有所不同。

- 辨識文字 API
 - 辨識文字 API 與 OCR 類似，但它會以非同步方式執行。
- 影像需求
 - 影像以 JPEG、PNG 或 BMP 格式呈現。
 - 影像的維度介於 50×50 與 4200×4200 圖元之間。
 - 影像檔案小於 4 mb。

- 限制（文字辨識作業的精確度取決於影像的品質）
 - 下列因素可能會導致不正確的讀取
 - 影像模糊、複雜的背景、陰影、文字反光或透視失真
 - 草寫的文字、藝術字型樣式
 - 文字太小、文字開頭的大寫字母過大或遺失
 - 文字加上了下標、上標或刪除線

使用印刷和手寫文字辨識服務 介紹

- 如內容標記服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 完成環境變數設定
 - 於Python應用程式中設定API key
 - 於Python應用程式驗證用戶端
 - 於Python應用程式加入欲分析影像URL

使用印刷和手寫文字辨識服務 介紹

- 讀取印刷和手寫文字
 - 讀取影像中的可見文字，並將它轉換成字元資料流。
- 呼叫讀取API
 - 使用下列程式碼來呼叫指定影像的 `batch_read_file` 方法來讀取影像的內容。

```
...
Batch Read File, recognize printed text - remote
This example will extract printed text in an image, then print results, line by line.
This API call can also recognize handwriting (not shown).
...
print("===== Batch Read File - remote =====")
# Get an image with printed text
remote_image_printed_text_url = "https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/print_text.png" # Image url
# Call API with URL and raw response (allows you to get the operation location)
recognize_printed_results = computervision_client.batch_read_file(remote_image_printed_text_url, raw=True)
```

使用印刷和手寫文字辨識服務 介紹

- 取得讀取結果
 - 取得從 batch_read_file 傳回的作業識別碼，並使用它來查詢服務中的作業結果。下列程式碼會以一秒的間隔檢查作業，直到傳回結果為止。然後，它會將已解壓縮的文字資料輸出到主控台。

```
# Get the operation location (URL with an ID at the end) from the response
operation_location_remote = recognize_printed_results.headers["Operation-Location"]
# Grab the ID from the URL
operation_id = operation_location_remote.split("/")[-1]

# Call the "GET" API and wait for it to retrieve the results
while True:
    get_printed_text_results = computervision_client.get_read_operation_result(operation_id)
    if get_printed_text_results.status not in ['NotStarted', 'Running']:
        break
    time.sleep(1)

# Print the detected text, line by line
if get_printed_text_results.status == TextOperationStatusCodes.succeeded:
    for text_result in get_printed_text_results.recognition_results:
        for line in text_result.lines:
            print(line.text)
            print(line.bounding_box)
print()
```

學習目標：

9-2:成人內容偵測服務介紹

- 成人內容偵測服務
介紹
- 使用成人內容偵測
服務

成人內容偵測服務介紹

- 偵測成人內容

- 電腦視覺可以偵測影像中的成人內容，讓開發人員可以限制在其軟體中顯示這些影像。



| | |
|----------|-------------|
| 成人 內容 | true |
| 成人 分數 | 0.855978131 |
| 辛辣 內容 | true |
| 辛辣 分數 | 0.984987 |

- 這項功能大部分是由Azure 內容仲裁服務所提供之。未來章節會再說明。

- 內容旗標定義

- 「成人」分類中有幾個不同的類別：
- 成人影像會定義為本質上明確的性行為，而且通常會描繪裸體和性行為的行為。
- 猥亵影像的定義是本質上具有性暗示，而且相較於標記為成人的影像內容，通常包含內容較不露骨的影像。
- 血腥暴力影像的定義為影像中有傷口與血。

使用成人內容偵測服務

- 如內容標記服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 完成環境變數設定
 - 於Python應用程式中設定API key
 - 於Python應用程式驗證用戶端
 - 於Python應用程式加入欲分析影像URL

- 取得是否有成人內容的布林值

```
# Set image_url to the URL of an image that you want to analyze.
image_url = "https://upload.wikimedia.org/wikipedia/commons/thumb/1/12/" + \
    "Broadway_and_Times_Square_by_night.jpg/450px-Broadway_and_Times_Square_by_night.jpg"

headers = {'Ocp-Apim-Subscription-Key': subscription_key}
params = {'visualFeatures': 'Categories,Description,Color'}
data = {'url': image_url}
response = requests.post(analyze_url, headers=headers,
                         params=params, json=data)
response.raise_for_status()

# The 'analysis' object contains various fields that describe the image. The most
# relevant caption for the image is obtained from the 'description' property.
analysis = response.json()
print(json.dumps(response.json()))
image_caption = analysis["description"]["captions"][0]["text"].capitalize()

# Display the image and overlay it with the caption.
image = Image.open(BytesIO(requests.get(image_url).content))
plt.imshow(image)
plt.axis("off")
_ = plt.title(image_caption, size="x-large", y=-0.1)
plt.show()
```

9-3:電腦視覺總結

- 電腦視覺服務總結
- 電腦視覺服務定價

電腦視覺服務總結

• 電腦視覺服務

物件偵測

物件數

```
[{"rectangle": {"x": 93, "y": 178, "w": 115, "h": 237}, "object": "person", "confidence": 0.764}, {"rectangle": {"x": 0, "y": 229, "w": 101, "h": 206}, "object": "person", "confidence": 0.624}, {"rectangle": {"x": 161, "y": 31, "w": 439, "h": 423}, "object": "subway train", "parent": {"object": "train", "parent": {"object": "Land vehicle", "parent": {"object": "Vehicle", "confidence": 0.926}, "confidence": 0.923}, "confidence": 0.917}, "confidence": 0.801}]
```

內容標記

標籤

```
[{"name": "train", "confidence": 0.9975446}, {"name": "platform", "confidence": 0.9955431}, {"name": "station", "confidence": 0.979800761}, {"name": "indoor", "confidence": 0.9277198}, {"name": "subway", "confidence": 0.8389395}, {"name": "clothing", "confidence": 0.5043765}, {"name": "pulling", "confidence": 0.4317162}]
```

臉部偵測

臉部 []

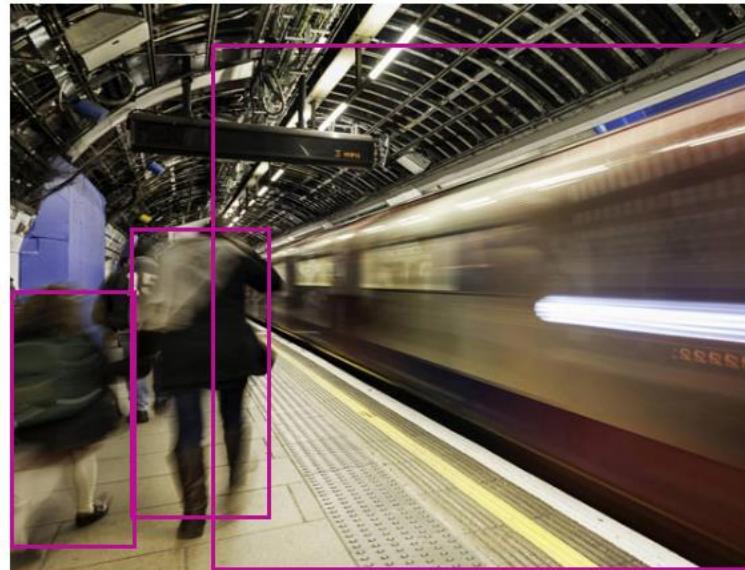
影像分類

類別 [{"name": "trans_trainstation", "score": 0.98828125}]

影像說明

描述

```
{ "tags": [ "train", "platform", "station", "building", "indoor", "subway", "track", "walking", "waiting", "pulling", "board", "people", "man", "luggage", "standing", "holding", "large", "woman", "yellow", "suitcase" ], "captions": [ { "text": "people waiting at a train station", "confidence": 0.8330992 } ] }
```



成人內容偵測

影像類別偵測

成人
內容

成人
分數

辛辣
內容

辛辣
分數

美工 0

圖案

類型

線條 0

繪圖

類型

主要
色彩
背景

主要
色彩
前景

輔色 #484C83

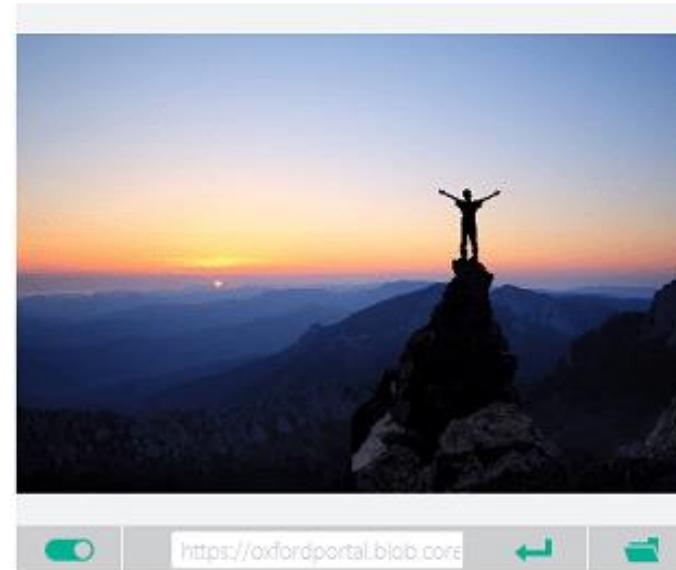
偵測色彩配置

- 電腦視覺服務

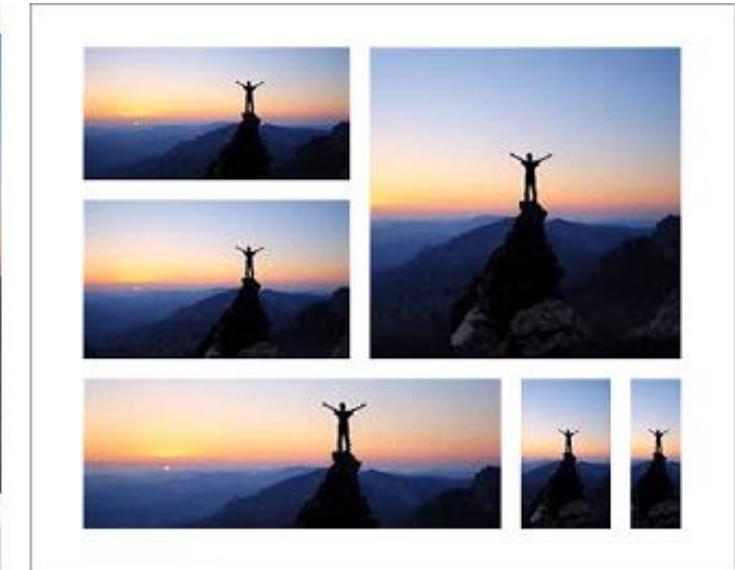
```
"result": {  
    "landmarks": [ {  
        "name": "Pyramid of Khafre",  
        "confidence": 0.970619797706604  
    } ]
```



特定領域偵測



智慧裁切縮圖



電腦視覺服務總結

- 電腦視覺服務

品牌偵測



```
"brands": [
  {
    "name": "Microsoft",
    "rectangle": {
      "x": 20,
      "y": 97,
      "w": 62,
      "h": 52
    }
  }
]
```

手寫文字辨識



```
Nutrition Facts Amount Per Serving
[144.0, 0.0, 1238.0, 211.0, 1224.0, 280.0, 130.0, 57.0]
Serving size: 1 bar (40g)
[110.0, 58.0, 598.0, 157.0, 587.0, 206.0, 100.0, 108.0]
Serving Per Package: 4
[83.0, 108.0, 548.0, 206.0, 538.0, 256.0, 72.0, 157.0]
Total Fat 13g
[683.0, 213.0, 1000.0, 286.0, 989.0, 332.0, 672.0, 260.0]
Saturated Fat 1.5g
[695.0, 295.0, 1120.0, 394.0, 1108.0, 447.0, 683.0, 347.0]
Amount Per Serving
[29.0, 207.0, 491.0, 309.0, 478.0, 367.0, 16.0, 265.0]
Trans Fat 0g
[668.0, 363.0, 954.0, 435.0, 940.0, 488.0, 655.0, 416.0]
calories 190
[8.0, 293.0, 265.0, 350.0, 254.0, 396.0, 0.0, 339.0]
Cholesterol 0mg
[593.0, 424.0, 1007.0, 526.0, 993.0, 580.0, 579.0, 479.0]
ories from Fat 110
[9.0, 377.0, 398.0, 464.0, 388.0, 509.0, 0.0, 421.0]
Sodium 20mg
[561.0, 497.0, 913.0, 588.0, 899.0, 643.0, 547.0, 552.0]
nt Daily Values are based on
[7.0, 476.0, 521.0, 598.0, 511.0, 640.0, 0.0, 518.0]
Vitamin A 50%
[525.0, 597.0, 776.0, 657.0, 766.0, 699.0, 514.0, 640.0]
calorie diet.
[12.0, 534.0, 196.0, 576.0, 187.0, 615.0, 4.0, 574.0]
```

- 定價資料

- <https://azure.microsoft.com/zh-tw/pricing/details/cognitive-services/computer-vision/>

- 不同地區的價格可能不同
- 請至上述網址輸入使用服務的地區位置查詢真實價格。

| 執行個體 | 每秒交易數 (TPS)** | 功能 | 價格 |
|--------------------|---------------|--|--|
| 免費 - Web/容器 | 每分鐘 20 | | 每月免費 5,000 筆交易 |
| S1 - Web/容器 | 10 TPS | 標記 臉部 GetThumbnail 色彩 影像類型 GetAreaOfInterest | 0-1M 筆交易 – NT\$30.055 每 1,000 筆交易 1M-5M 筆交易 – NT\$24.044 每 1,000 筆交易 5M-10M 筆交易 – NT\$19.536 每 1,000 筆交易 10M-100M 筆交易 – NT\$19.536 每 1,000 筆交易 100M+ 筆交易 – NT\$19.536 每 1,000 筆交易 |
| OCR | | 0-1M 筆交易 – NT\$45.082 每 1,000 筆交易 1M-5M 筆交易 – NT\$30.055 每 1,000 筆交易 5M-10M 筆交易 – NT\$19.536 每 1,000 筆交易 10M-100M 筆交易 – NT\$19.536 每 1,000 筆交易 100M+ 筆交易 – NT\$19.536 每 1,000 筆交易 | |
| 描述* 辨識文字* 讀取 | | NT\$75.136 每 1,000 筆交易 | |

- 開啟Demo_9-3.ipynb
- 執行印刷文字辨識服務操作
- 執行手寫文字辨識服務操作
- 執行成人內容偵測服務操作

- 了解如何使用印刷和手寫文字辨識服務
- 了解如何使用成人內容偵測服務服務
- 了解電腦視覺服務支援的區域與定價



- Lab01: 印刷文字辨識服務操作
- Lab02: 手寫文字辨識服務操作
- Lab03: 成人內容偵測服務操作

Estimated time:
20 minutes



Module 10

臉部辨識服務

學習目標：

- 10-1: Azure 臉部辨識服務介紹、認知表情辨識服務介紹
- 10-2: 臉部偵測服務介紹
- 10-3: 臉部驗證服務介紹

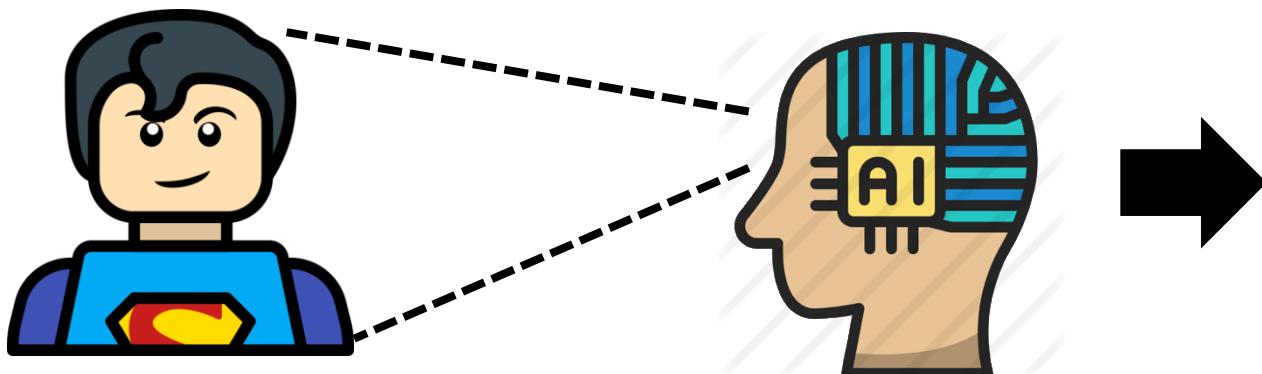
學習目標：

10-1:臉部辨識服務介紹

- 臉部辨識服務介紹
- 建置Azure臉部辨識服務環境
- 認知表情辨識服務介紹

臉部辨識服務介紹

- 臉部辨識
 - Azure提供的分析影像中臉部的相關資訊功能有：
 - 認知表情辨識
 - 臉部偵測
 - 臉部驗證



Smile

Superman

Hair:{ bald: 0.14 }

Emotion:{ happiness:0.96 }

FacialHair: {beard:0.1}

hairColor: {Grey:0.78}

- 於認知服務主控台點選新增
 - 進到Marketplace後選擇AI + 機器學習服務
 - 右側Cognitive Services區域選擇查看更多
 - 點選臉部辨識圖像



A screenshot of the Azure Marketplace. The top navigation bar shows '首頁 > 認知服務 > Marketplace'. The main area is titled 'Marketplace' and contains a sidebar with links: '我已儲存的清單', '最近建立', '服務提供者', and a '類別' dropdown set to 'AI + 機器學習服務'. Below the sidebar, there are two service cards: 'Bing 自動建議' (Bing AutoSuggest) and 'Bing 自訂搜尋' (Bing Custom Search). Both services are provided by Microsoft and have brief descriptions and a blue heart icon at the bottom right.

建置 Azure 臉部辨識服務環境

- 建立臉部辨識服務

- 點選建立
- 填寫資料：名稱、位置、定價層、新建資源群組

首頁 > 認知服務 > Marketplace > 臉部

臉部
Microsoft



臉部 儲存以便稍後使用
Microsoft

建立

概觀 **方案**

在您的應用程式中內嵌臉部辨識，以實現流暢且極為安全的使用者體驗。這不需要機器學習的專業技術。此服務會在 Azure 全球性存儲庫（可達 1 百萬人）中進行人員比對；表情辨識能感知多種反應，例如快樂、輕蔑、中立和恐懼；還能檢測到臉部。

實用的連結

- 深入了解臉部 API
- 文件
- API 參考
- 定價
- 區域可用性

首頁 > 認知服務 > Marketplace > 臉部 > 建立

建立

臉部

名稱 *
facialRecogExample

訂用帳戶 *
免費試用版

位置 *
(US) 美國西部

定價層 (檢視完整定價詳細資料) *
F0 (20 每分鐘呼叫次數, 30K 每月呼叫次數)

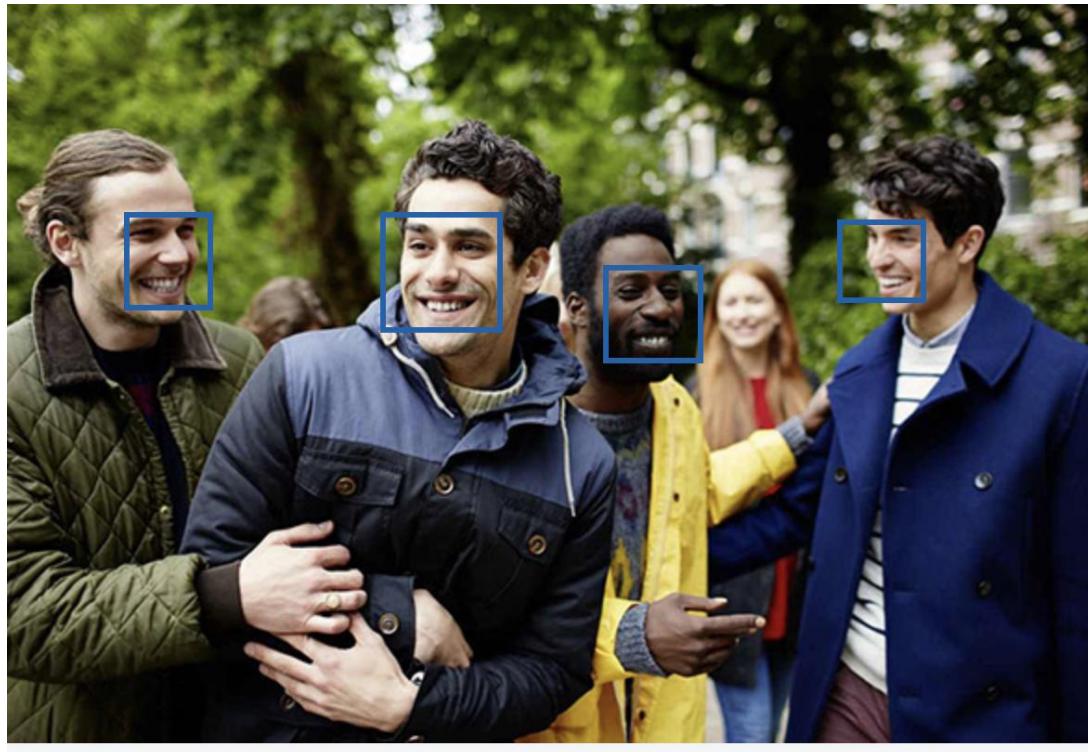
資源群組 *
CogAI

新建

- 完成建立臉部辨識服務
 - 完成部署後可以在認知服務中控台看到建立的服務內容

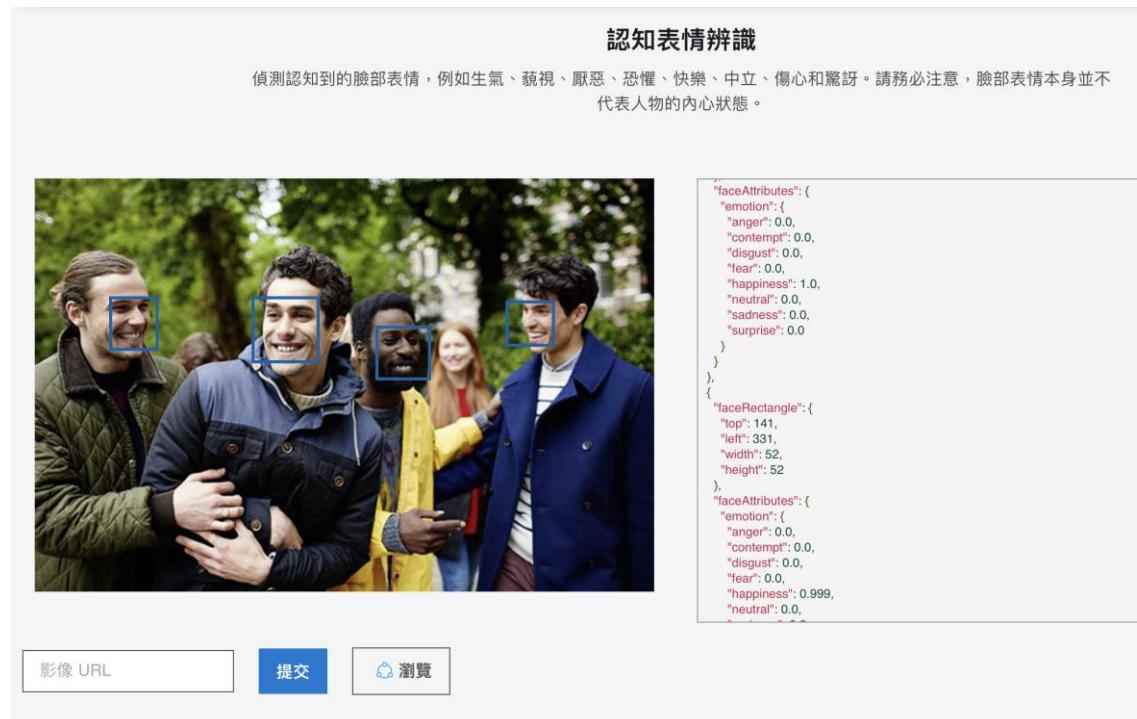
The screenshot shows the Microsoft Cognitive Services Face deployment overview page. The title bar reads "Microsoft.CognitiveServicesFace | 概觀". The main content area displays a success message: "您的部署已完成" (Deployment completed). It provides details about the deployment: name (Microsoft.CognitiveServicesFace), account type (Free Trial), resource group (CogAI), start time (10/4/2020 下午2:14:10), and a shared access key (5b9029a2-ff17-4d96-af25-e6146e73590d). Below this, there are sections for "部署詳細資料 (下載)" (Deployment details) and "後續步驟" (Next steps). A "前往資源" (Go to resources) button is located at the bottom left. On the right side, there are links to "資訊安全中心" (Information Security Center), "免費 Microsoft 教學課程" (Free Microsoft learning courses), and "洽詢專家" (Consultant experts). The left sidebar includes navigation links for "概觀", "輸入", "輸出", and "範本".

- 偵測認知到的臉部表情，例如生氣、藐視、厭惡、恐懼、快樂、中立、傷心和驚訝。



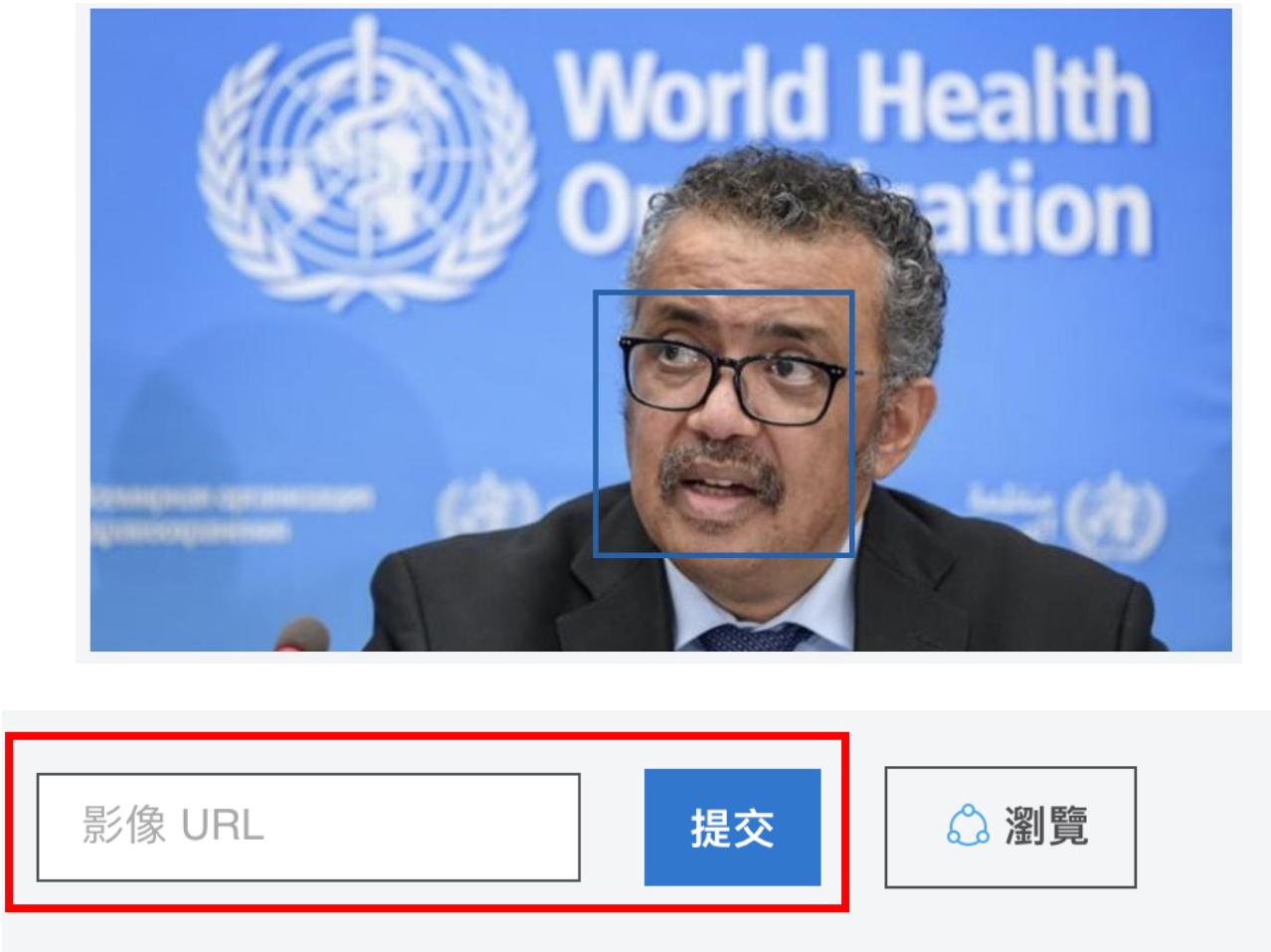
```
"faceAttributes": {  
    "emotion": {  
        "anger": 0.0,  
        "contempt": 0.0,  
        "disgust": 0.0,  
        "fear": 0.0,  
        "happiness": 1.0,  
        "neutral": 0.0,  
        "sadness": 0.0,  
        "surprise": 0.0  
    }  
}
```

- 點選以下Azure臉部辨識Demo網址
 - <https://azure.microsoft.com/zh-tw/services/cognitive-services/face/#features>



認知表情辨識服務介紹

- 試著於影像URL自行輸入影像網址並觀察回傳結果



偵測結果：
偵測到 1 個臉部

JSON :

```
[  
 {  
   "faceRectangle": {  
     "top": 303,  
     "left": 543,  
     "width": 288,  
     "height": 288  
   },  
   "faceAttributes": {  
     "emotion": {  
       "anger": 0.002,  
       "contempt": 0.002,  
       "disgust": 0.002,  
       "fear": 0.003,  
       "happiness": 0.002,  
       "neutral": 0.831,  
       "sadness": 0.008,  
       "surprise": 0.149  
     }  
   }  
 }]
```

學習目標：

10-2:臉部偵測服務介紹

- 認識臉部偵測服務
- 使用臉部偵測服務

認識臉部偵測服務

- 臉部偵測可偵測影像中的人臉，並將相關資訊擷取後傳回，如姿勢、性別、年齡、眼鏡等



- 支援的地區與價格請參考以下連結
 - <https://azure.microsoft.com/zh-tw/pricing/details/cognitive-services/face-api/>

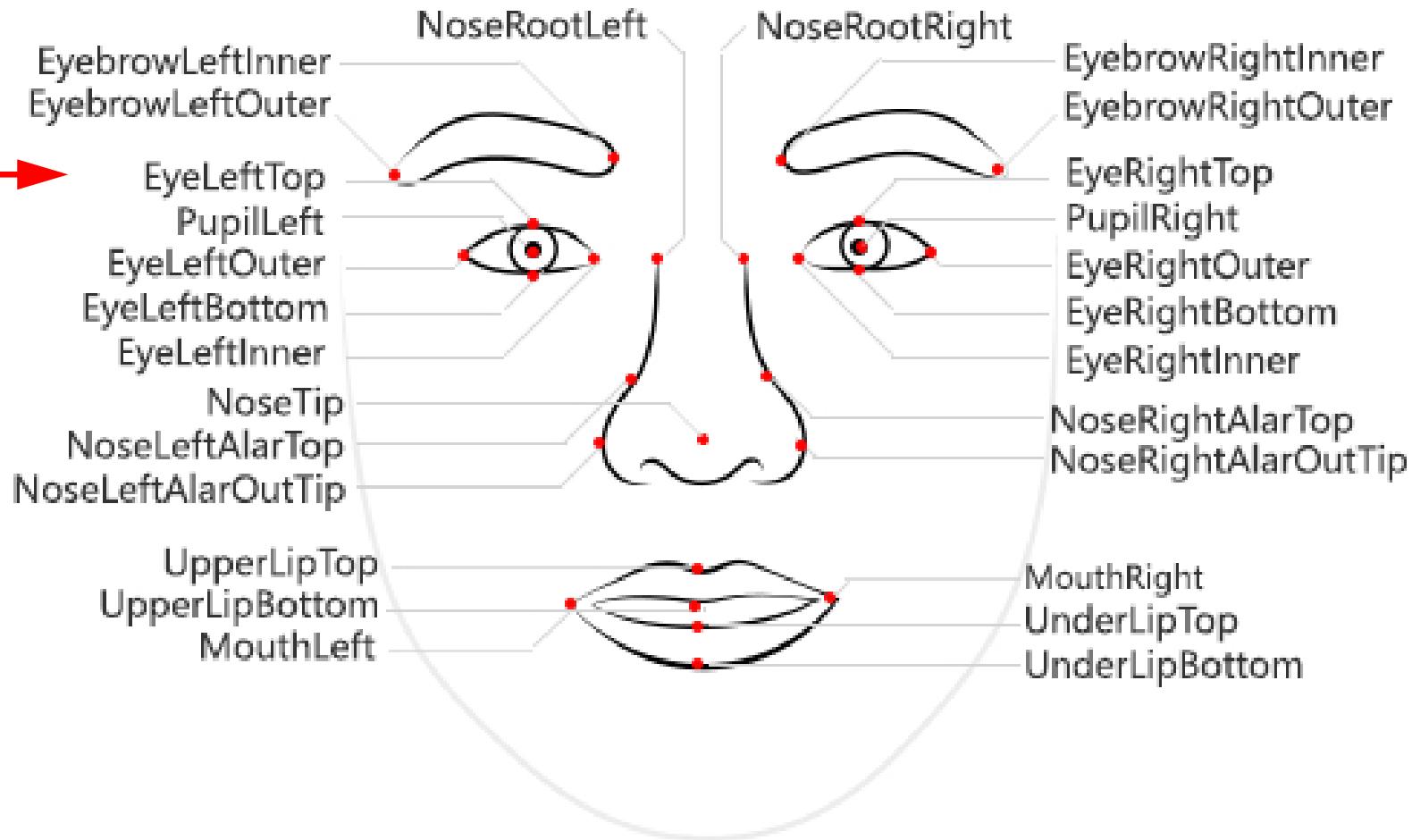
- 點選以下Azure臉部偵測Demo網址
 - <https://azure.microsoft.com/zh-tw/services/cognitive-services/face/#features>

```
"color": "red",
"confidence": 0.02
},
],
},
"smile": 1.0,
"headPose": {
"pitch": 3.4,
"roll": -13.7,
"yaw": -5.8
},
"gender": "male",
"age": 52.0,
"facialHair": {
"moustache": 0.1,
"beard": 0.1,
"sideburns": 0.1
},
"glasses": "NoGlasses",
"makeup": {
"eyeMakeup": false,
"lipMakeup": false
},
"emotion": {
"anger": 0.0,
"contempt": 0.0,
"disgust": 0.0,
"fear": 0.0,
"happiness": 1.0
}
```

影像 URL 提交 瀏覽 偵測模型：detection_01

認識臉部偵測服務

- 臉部偵測回傳：
 - 臉部識別碼
 - 臉部特徵點
 - 各特徵點的位置
 - 臉部屬性
 - 年齡、臉部清晰與否
 - 情感、面部毛髮
 - 性別、眼鏡、頭髮
 - 化妝、視覺噪音
 - 微笑、物件遮擋人臉



Copyright (c) Microsoft. All rights reserved

使用臉部偵測服務

- 安裝用戶端程式庫

- pip install --upgrade azure-cognitiveservices-vision-face
- 匯入相關模組
- 從認知服務台複製key與Region

The screenshot shows the Azure Cognitive Services facial recognition quickstart page. The top navigation bar includes '首頁' and 'facialRecogExample | 快速入門'. The main content area has a title 'facialRecogExample | 快速入門' with a '認知服務' icon. A search bar at the top says '搜尋 (Cmd +/)'.

The left sidebar lists several options: '概觀' (Overview), '活動記錄' (Activity logs), '存取控制 (IAM)', '標籤' (Tags), and '診斷並解決問題' (Diagnose and solve problems). Below this is a '資源管理' (Resource management) section and a '快速入門' (Quick start) button.

The main content area contains the following steps:

1. 探索快速入門指南，開始啟動並執行臉部。
取得 API 金鑰和端點以驗證您的應用程式，並開始傳送呼叫給服務:
Key1 ⓘ
[Redacted]
端點 ⓘ
[Redacted]
- 所有臉部呼叫和 docker 容器啟用都需要金鑰。請在要求標頭 (Web API)、臉部用戶端 (SDK) 或命令列 (CLI) 中包含此金鑰。
- 在 API 主控台中試用此服務 - 需要 API 金鑰並選取您的位置: westus

- 設定
 - 將key與endpoint輸入程式中相對應參數中
 - 設定欲偵測的影像網址
 - 參考以下兩種方法

使用臉部偵測服務

- 範例：
 - 使用SDK回傳偵測結果

使用SDK偵測影像中的人臉



In [10]:

```
1 from io import BytesIO
2 import os
3 from PIL import Image, ImageDraw
4 import requests
5
6 from azure.cognitiveservices.vision.face import FaceClient
7 from azure.cognitiveservices.vision.face.models import FaceAttributeType
8 from msrest.authentication import CognitiveServicesCredentials
9
10 ...
11 Authenticate the Face service
12 ...
13 KEY = '8b361049be23449391869bdb83839edb'
14 ENDPOINT = 'https://facialrecogexample.cognitiveservices.azure.com/'
15
16 # Create an authenticated FaceClient.
17 face_client = FaceClient(ENDPOINT, CognitiveServicesCredentials(KEY))
18
19 ...
20 Detect face(s) with attributes in a URL image
21 ...
22 # Image of face(s)
23 face1_url = 'https://raw.githubusercontent.com/Microsoft/Cognitive-Face-Windows/master/Data/detection1.jpg'
24 face1_name = os.path.basename(face1_url)
25
26
27 # List of url images
28 url_images = [face1_url]
29
30
31 # Attributes you want returned with the API call, a list of FaceAttributeType enum (string format)
32 face_attributes = ['age', 'gender', 'headPose', 'smile', 'facialHair', 'glasses', 'emotion']
33
34 # Detect a face with attributes, returns a list[DetectedFace]
35 for image in url_images:
36     detected_faces = face_client.face.detect_with_url(url=image, return_face_attributes=face_attributes)
37     if not detected_faces:
38         raise Exception(
39             'No face detected from image {}'.format(os.path.basename(image)))
40
```

- 範例：
 - 使用REST API回傳偵測結果

使用REST API回傳偵測結果

```
1 import requests
2 import json
3
4 KEY = '8b361049be23449391869bdb83839edb'
5 ENDPOINT = 'https://facialrecogexample.cognitiveservices.azure.com/face/v1.0/detect'
6
7 # set to your own subscription key value
8 subscription_key = KEY
9 assert subscription_key
10
11 # replace <My Endpoint String> with the string from your endpoint URL
12 face_api_url = ENDPOINT #'https://<My Endpoint String>.com/face/v1.0/detect'
13
14
15 image_url = 'https://raw.githubusercontent.com/Microsoft/Cognitive-Face-Windows/master/Data/detection1.jpg'
16
17 headers = {'Ocp-Apim-Subscription-Key': subscription_key}
18
19 params = {
20     'returnFaceId': 'true',
21     'returnFaceLandmarks': 'false',
22     'returnFaceAttributes': 'age,gender,headPose,smile,facialHair,glasses,emotion,hair,makeup,occlusion,accessor
23 }
24
25 response = requests.post(face_api_url, params=params,
26                             headers=headers, json={"url": image_url})
27 print(json.dumps(response.json()))
```

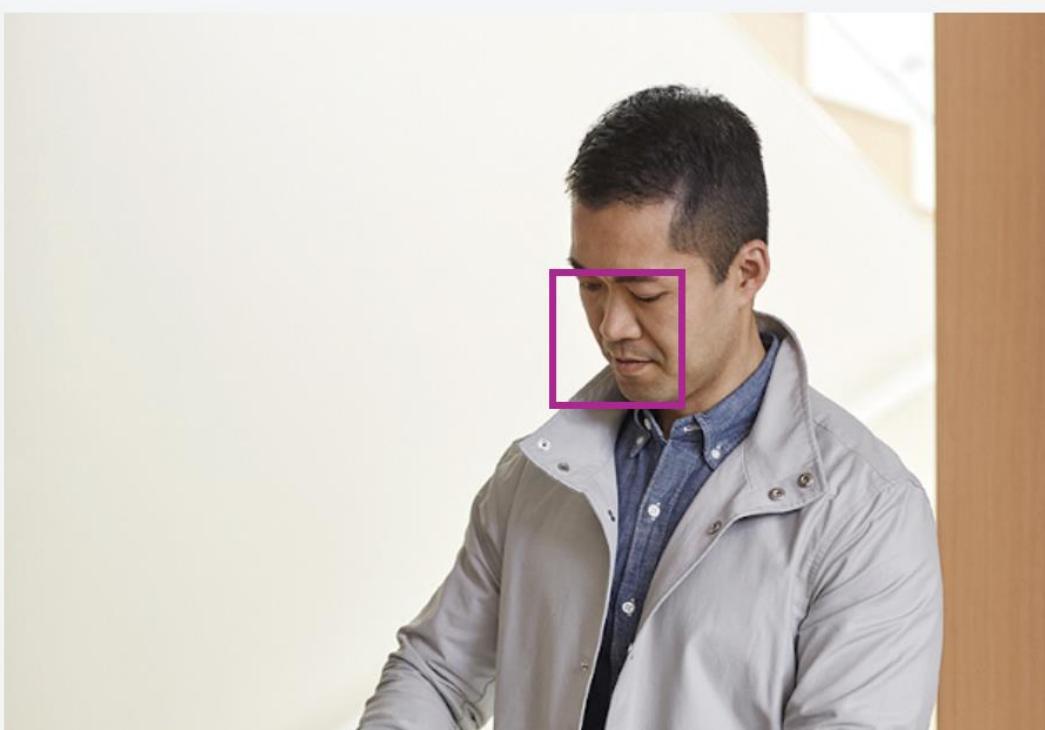
```
{"faceId": "cc854911-0267-4676-b941-91f83b55a10a", "faceRectangle": {"top": 124, "left": 459, "width": 227, "height": 227}, "faceAttributes": {"smile": 1.0, "headPose": {"pitch": -12.0, "roll": -13.6, "yaw": 5.0}, "gender": "female", "age": 24.0, "facialHair": {"moustache": 0.0, "beard": 0.0, "sideburns": 0.0}, "glasses": "ReadingGlasses", "emotion": {"anger": 0.0, "contempt": 0.0, "disgust": 0.0, "fear": 0.0, "happiness": 1.0, "neutral": 0.0, "sadness": 0.0, "surprise": 0.0}, "blur": {"blurLevel": "low", "value": 0.0}, "exposure": {"exposureLevel": "goodExposure", "value": 0.48}, "noise": {"noiseLevel": "low", "value": 0.11}, "makeup": {"eyeMakeup": true, "lipMakeup": true}, "accessories": [{"type": "glasses", "confidence": 1.0}], "occlusion": {"foreheadOccluded": false, "eyeOccluded": false, "mouthOccluded": false}, "hair": {"bald": 0.1, "invisible": false, "hairColor": [{"color": "brown", "confidence": 0.99}, {"color": "black", "confidence": 0.45}, {"color": "blond", "confidence": 0.42}, {"color": "red", "confidence": 0.41}, {"color": "gray", "confidence": 0.14}, {"color": "other", "confidence": 0.12}]}}}
```

10-3:臉部驗證服務介 紹

學習目標：

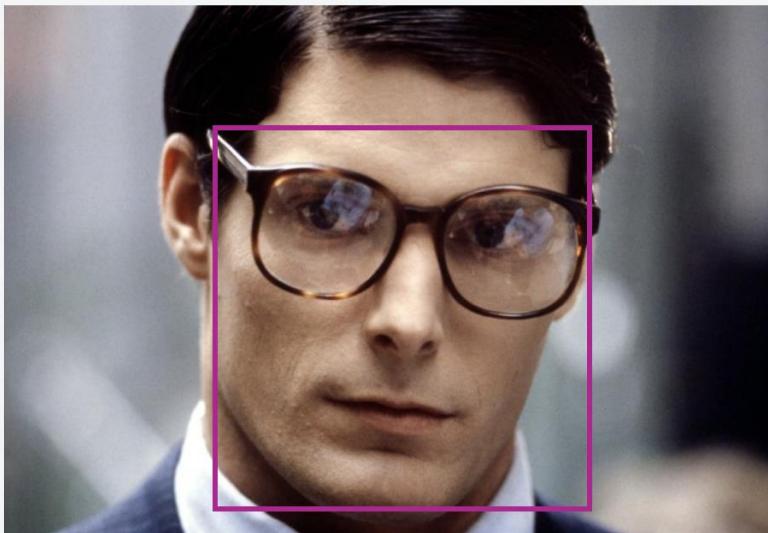
- 認識臉部驗證服務
- 使用臉部驗證服務

- 查看兩張影像中的人像是否屬於同一人

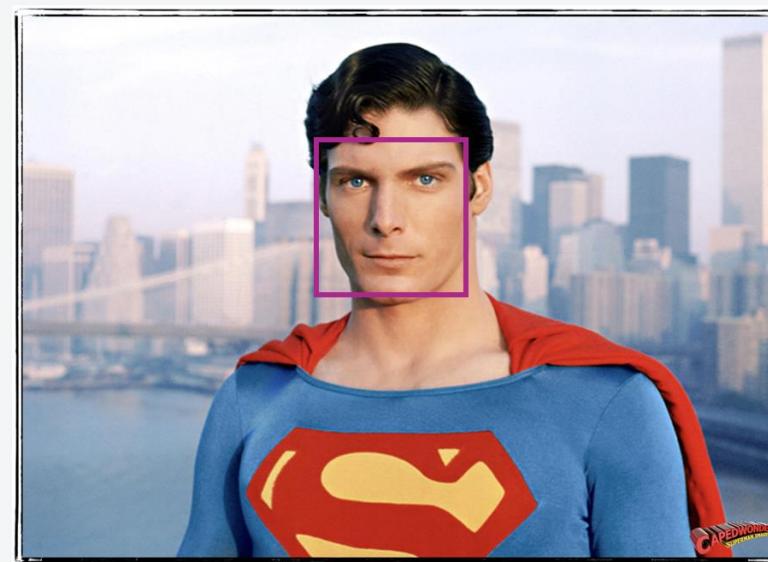


驗證結果：這兩張臉屬於同一個人。信心為 0.93468。

- 點選臉部偵測提到的Azure網址：
 - 試著於影像URL自行輸入影像網址並觀察回傳結果



Clark Kent



Superman

驗證結果：這兩張臉屬於同一個人。信心為 0.9187。

- 如臉部偵測服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 於Python應用程式中設定API key、Region

- 範例：

- 建立與訓練臉部影像群組

臉部辨識

建立與訓練臉部影像群組

```
1 import asyncio
2 import io
3 import glob
4 import os
5 import sys
6 import time
7 import uuid
8 import requests
9 from urllib.parse import urlparse
10 from io import BytesIO
11 from PIL import Image, ImageDraw
12 from matplotlib import pyplot as plt
13 from azure.cognitiveservices.vision.face import FaceClient
14 from msrest.authentication import CognitiveServicesCredentials
15 from azure.cognitiveservices.vision.face.models import TrainingStatusType, Person, SnapshotObjectType, Operation
16 ...
17 ...
18 Authenticate the Face service
19 ...
20 KEY = '8b361049be23449391869bdb83839edb'
21 ENDPOINT = 'https://facialrecogexample.cognitiveservices.azure.com/'
22 ...
23 # Create an authenticated FaceClient.
24 face_client = FaceClient(ENDPOINT, CognitiveServicesCredentials(KEY))
25 ...
26 # Used in the Person Group Operations, Snapshot Operations, and Delete Person Group examples.
27 # You can call list_person_groups to print a list of preexisting PersonGroups.
28 # SOURCE_PERSON_GROUP_ID should be all lowercase and alphanumeric. For example, 'mygroupname' (dashes are OK).
29 PERSON_GROUP_ID = 'my-unique-person-group'
30 ...
31 # Used for the Snapshot and Delete Person Group examples.
32 TARGET_PERSON_GROUP_ID = str(uuid.uuid4()) # assign a random ID (or name it anything)
33 ...
34 ...
35 Create the PersonGroup
36 ...
37 # Create empty Person Group. Person Group ID must be lower case, alphanumeric, and/or with '-','_'.
38 print('Person group:', PERSON_GROUP_ID)
39 face_client.person_group.create(person_group_id=PERSON_GROUP_ID, name=PERSON_GROUP_ID)
40
```

- 從影像群組中進行人員識別

Identify a face

In [5]:

```
1  '''
2 Identify a face against a defined PersonGroup
3  '''
4 # Group image for testing against
5 group_photo = 'test-image-person-group.jpg'
6 IMAGES_FOLDER = os.path.join(os.path.dirname(os.path.realpath(group_photo)))
7
8 # Get test image
9 test_image_array = glob.glob(os.path.join(IMAGES_FOLDER, group_photo))
10 image = open(test_image_array[0], 'r+b')
11
12 # Detect faces
13 face_ids = []
14 faces = face_client.face.detect_with_stream(image)
15 for face in faces:
16     face_ids.append(face.face_id)
17
18 # Identify faces
19 results = face_client.face.identify(face_ids, PERSON_GROUP_ID)
20 print('Identifying faces in {}'.format(os.path.basename(image.name)))
21 if not results:
22     print('No person identified in the person group for faces from {}'.format(os.path.basename(image.name)))
23 for person in results:
24     print('Person for face ID {} is identified in {} with a confidence of {}.'.format(person.face_id, os.path.basename(image.name), person.confidence))
```

```
Identifying faces in test-image-person-group.jpg
Person for face ID afa5ea71-cd25-4615-ac1b-8dd94582e3f7 is identified in test-image-person-group.jpg with a confidence of 0.92387.
Person for face ID 62abb154-16a5-4af1-8a2b-e84f3e6bb7b7 is identified in test-image-person-group.jpg with a confidence of 0.93316.
```

- 驗證兩張影像中的臉是否屬於同一人

Verify Faces from different images

```
1 # Base url for the Verify and Facelist/Large Facelist operations
2 IMAGE_BASE_URL = 'https://csdx.blob.core.windows.net/resources/Face/Images/'
3
4 # Create a list to hold the target photos of the same person
5 target_image_file_names = ['Family1-Dad1.jpg', 'Family1-Dad2.jpg']
6 # The source photos contain this person
7 source_image_file_name1 = 'Family1-Dad3.jpg'
8 source_image_file_name2 = 'Family1-Son1.jpg'
9
10 # Detect face(s) from source image 1, returns a list[DetectedFaces]
11 detected_faces1 = face_client.face.detect_with_url(IMAGE_BASE_URL + source_image_file_name1)
12
13 # Add the returned face's face ID
14 source_image1_id = detected_faces1[0].face_id
15 print('{0} face(s) detected from image {1}'.format(len(detected_faces1), source_image_file_name1))
16
17 # Detect face(s) from source image 2, returns a list[DetectedFaces]
18 detected_faces2 = face_client.face.detect_with_url(IMAGE_BASE_URL + source_image_file_name2)
19
20 # Add the returned face's face ID
21 source_image2_id = detected_faces2[0].face_id
22 print('{0} face(s) detected from image {1}'.format(len(detected_faces2), source_image_file_name2))
23
24 # List for the target face IDs (uuids)
25 detected_faces_ids = []
26 # Detect faces from target image url list, returns a list[DetectedFaces]
27 for image_file_name in target_image_file_names:
28     detected_faces = face_client.face.detect_with_url(IMAGE_BASE_URL + image_file_name)
29     # Add the returned face's face ID
30     detected_faces_ids.append(detected_faces[0].face_id)
31     print('{0} face(s) detected from image {1}'.format(len(detected_faces), image_file_name))
32
33
34 # Verification example for faces of the same person. The higher the confidence, the more identical the faces in
35 # Since target faces are the same person, in this example, we can use the 1st ID in the detected_faces_ids list
36 verify_result_same = face_client.face.verify_face_to_face(source_image1_id, detected_faces_ids[0])
37 print('Faces from {0} & {1} are of the same person, with confidence: {2}'
38       .format(source_image_file_name1, target_image_file_names[0], verify_result_same.confidence))
39 if verify_result_same.is_identical
40 else 'Faces from {0} & {1} are of a different person, with confidence: {2}'
41       .format(source_image_file_name1, target_image_file_names[0], verify_result_same.confidence))
```

- 開啟Demo_10-3.ipynb
- 執行Azure臉部辨識Web服務操作
- 執行臉部偵測服務操作
- 執行臉部辨識服務操作

- Azure臉部辨識介紹
- Azure臉部辨識相關方法
- 了解如何使用臉部偵測服務
- 了解如何使用臉部辨識服務



- Lab01: 建立Azure臉部辨識服務
- Lab02: 執行臉部偵測服務操作
- Lab03: 執行臉部辨識服務操作

Estimated time:
20 minutes



Module 11

語音服務

學習目標：

- 11-1: Azure 語音服務介紹
- 11-2: 語音轉文字服務介紹
- 11-3: 文字轉語音服務介紹

11-1:語音服務介紹

- 語音服務介紹
- 建置Azure語音服務環境

- 語音服務的語音轉換文字（也稱為語音辨識）可將音訊串流的即時轉譯為文字。
- 語音服務範例
 - <https://azure.microsoft.com/zh-tw/services/cognitive-services/speech-to-text/#features>

The screenshot shows a user interface for testing speech-to-text features. On the left, there's a list of example sentences with audio icons:

- 範例句子 1
- 範例句子 2
- 範例句子 3
- 範例句子 4
- 範例句子 5
- 範例句子 6

In the center, under the heading "基準" (Standard), is the text: "Show me all flights from Pittsburgh to Boston is directly connecting that Martin pitch burgh after 7."

To the right, under the heading "客製化的語音" (Customized Speech), is the text: "Show me all flights from Pittsburgh to Boston both directed connecting that depart Pittsburgh after 7 PM."

At the bottom, there's a control bar with icons for volume, play/pause, and a progress bar showing 0:11.

- 語音服務由以下功能組成
 - 語音轉文字、批次轉譯、多裝置交談、對話轉譯、建立自訂語音模型、文字轉換語音、建立自訂語音、語音翻譯、語音助理
 - 後面章節將會說明使用方式

- 於認知服務主控台點選新增
 - 進到 Marketplace 後選擇 AI + 機器學習服務
 - 右側 Cognitive Services 區域選擇查看更多
 - 點選語音圖像



The screenshot shows the Azure Marketplace search results for 'Speech'. The search bar at the top contains '搜尋 Marketplace' and the category 'Bot Service'. The results section is titled 'Bot Service' and shows two items: 'Web App Bot' by Microsoft and 'Bot Channels Registration' by Microsoft. Both items have a blue heart icon. Below this, under 'Cognitive Services', there are four items: 'Bing 自動建議' (Bing AutoSuggest), 'Bing 自訂搜尋' (Bing Custom Search), 'Bing 實體搜尋' (Bing Entity Search), and 'Bing 搜尋' (Bing Search). Each item has a blue heart icon. There are also '查看更多' (View more) links for both the 'Bot Service' and 'Cognitive Services' sections.

- 建立語音服務
 - 點選建立
 - 填寫資料：名稱、位置、定價層、新建資源群組

The screenshot shows the 'Create a new resource' step in the Azure Speech service setup. It includes:

- Title:** 語音 Microsoft
- Create button:** 建立
- Description:** 將可聽的語音轉譯為可讀且可搜尋的文字。將即時語音翻譯新增至應用程式與服務。近乎即時地將文字轉換為音訊。使用慣用的程式設計語言，來快速建置啟用語音的應用程式與服務。自訂語音系統來針對特定案例將品質最佳化。
- 实用的連結:** 深入了解整合式語音, 文件, 語音辨識參考, 文字轉換語音參考, 定價, 區域可用性

首頁 > 認知服務 > Marketplace > 語音 > 建立

建立

語音

名稱 *
moduleSpeechReg

訂用帳戶 *
免費試用版

位置 *
(US) 美國西部 2

定價層 (檢視完整定價詳細資料) *
F0

資源群組 *
CogAI

新建

建置 Azure 語音服務環境

- 完成部署後可以在認知服務中控台看到建立的服務內容

The screenshot shows two main sections of the Azure portal.

Top Section (Deployment Status):

- Header: Microsoft.CognitiveServicesSpeechServices | 概觀
- Message: 已成功部署 (Deployment successful)
- Details: 開始時間: 28/3/2020 上午12:01:58, 相互關聯識別碼: e04748e0-625f-4922-ad1b-52898489b8bf
- Buttons: 前往資源 (Go to resources), 釘選到儀表板 (Pin to dashboard)

Bottom Section (Cognitive Services Dashboard):

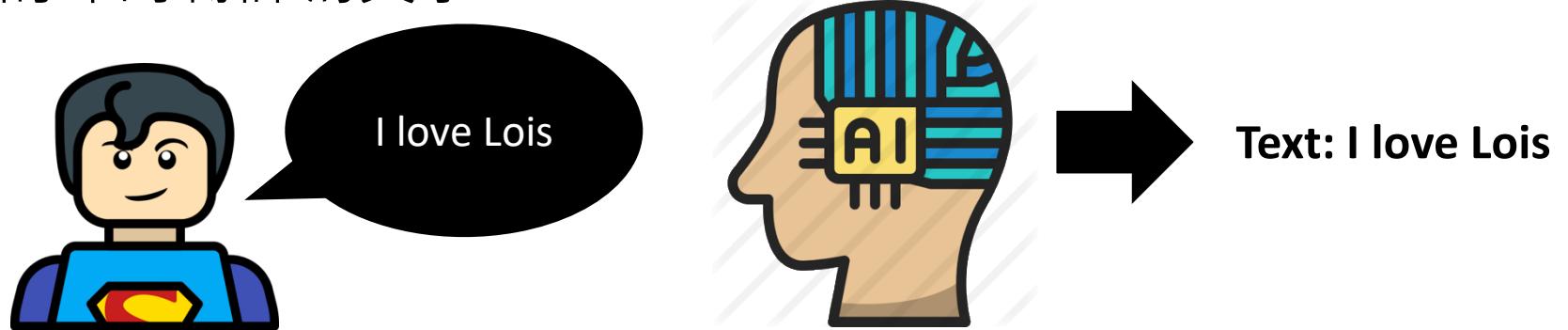
- Header: Microsoft.CognitiveServicesSpeechServices | 概觀
- Left sidebar: 概觀, 入輸出, 範本
- Central area:
 - Message: 您的部署已完成 (Your deployment is complete)
 - Deployment details: 部署名稱: Microsoft.CognitiveServicesSpeechServices, 訂用帳戶: 免費試用版, 資源群組: CogAI
 - Links: 部署詳細資料 (下載) (Deployment details (Download)), 後續步驟 (Next steps)
 - Buttons: 前往資源 (Go to resources)
- Right sidebar:
 - 資訊安全中心 (Information security center): 保護應用程式及基礎結構, 前往 Azure 資訊安全中心 >
 - 免費 Microsoft 教學課程 (Free Microsoft learning courses): 立即開始學習 >

11-2:語音轉文字服務 介紹

學習目標：

- 認識語音轉文字服務
- 使用語音轉文字服務

- 語音服務的語音轉換文字（也稱為語音辨識）
 - 可將音訊串流的即時轉譯為文字。



- 語音服務支援的語言和區域總表請到以下網址查詢
 - <https://docs.microsoft.com/zh-tw/azure/cognitive-services/speech-service/language-support>

| Locale | Language | 支援 | 自訂 |
|--------|----------------|----|------|
| ar-AE | 阿拉伯文（阿拉伯聯合大公國） | 是 | 否 |
| ar-BH | 阿拉伯文（巴林），現代化標準 | 是 | 語言模型 |
| ar-EG | 阿拉伯文 (埃及) | 是 | 語言模型 |

- 安裝用戶端程式庫
 - pip install azure-cognitiveservices-speech
 - 如果安裝時 (macOS) 發生問題，請先以下列指令更新pip
 - python3 -m pip install --upgrade pip
- 於Python中載入Speech SDK
 - Import azure.cognitiveservices.speech as speechsdk

- 設定SpeechConfig設定檔內容
 - 從認知服務台複製key與Region

首頁 > moduleSpeechReg | 快速入門

moduleSpeechReg | 快速入門

認知服務

搜尋 (Cmd +/)

概觀

活動記錄

存取控制 (IAM)

標籤

診斷並解決問題

資源管理

快速入門

恭喜! 立即探索快速入門指導，以啟動並執行語音。

查看 Azure 認知服務 (預覽) 中 Docker 容器的新支援

以下是金鑰以及您要開始使用的端點:

Key1 ⓘ

端點 ⓘ

`https://westus2.api.cognitive.microsoft.com/sts/v1.0/issuetoken`

整合式語音的每次 Web API 呼叫及語音轉換文字或文字轉換語音的每次 Docker 容器啟用都需要上方的金鑰。從左側功能表的 [金鑰] 索引標籤中找到您的金鑰。

執行適用於語音服務的 Docker 容器 (預覽)



- 設定key、endpoint、host、Region

```
speech_key, service_region = "YourSubscriptionKey", "YourServiceRegion"  
speech_config = speechsdk.SpeechConfig(subscription=speech_key, region=service_region)
```

- 設定欲分析語音位置與設定辨識器

```
audio_filename = "whatsttheweatherlike.wav"  
audio_input = speechsdk.AudioConfig(filename=audio_filename)  
speech_recognizer = speechsdk.SpeechRecognizer(speech_config=speech_config, audio_config=audio_input)
```

- 更改來源語音語系

```
speech_config.speech_recognition_language="de-DE"
```

- 使用片語清單提高準確度
 - 建立清單

```
phrase_list_grammar = speechsdk.PhraseListGrammar.from_recognizer(reco)
phrase_list_grammar.addPhrase("Move to Ward")
phrase_list_grammar.addPhrase("Move to Bill")
phrase_list_grammar.addPhrase("Move to Ted")
```

- 清除清單內容

```
phrase_list_grammar.clear()
```

- 辨識一次
 - 一次處理一個句子，可辨識句子長度為20秒

```
result = speech_recognizer.recognize_once()

if result.reason == speechsdk.ResultReason.RecognizedSpeech:
    print("Recognized: {}".format(result.text))
elif result.reason == speechsdk.ResultReason.NoMatch:
    print("No speech could be recognized: {}".format(result.no_match_details))
elif result.reason == speechsdk.ResultReason.Canceled:
    cancellation_details = result.cancellation_details
    print("Speech Recognition canceled: {}".format(cancellation_details.reason))
    if cancellation_details.reason == speechsdk.CancellationReason.Error:
        print("Error details: {}".format(cancellation_details.error_details))
```

- 連續辨識
 - 處理超過一個句子

```
def recognized_cb(evt):
    if evt.result.reason == speechsdk.ResultReason.RecognizedSpeech:
        # Do something with the recognized text
        # evt.result.text

speech_recognizer.recognized.connect(recognized_cb)

# Start continuous speech recognition
speech_recognizer.start_continuous_recognition()

# Stop continuous speech recognition
speech_recognizer.stop_continuous_recognition()
```

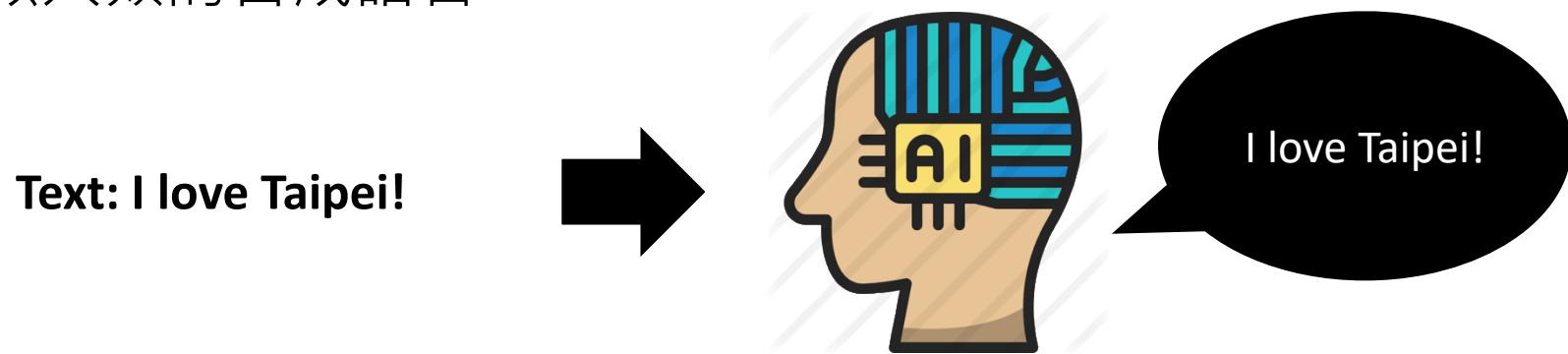
學習目標：

11-3:文字轉語音服務 介紹

- 文字轉語音服務介紹
- 使用文字轉語音服務服務

- 介紹

- 語音服務中的文字轉語音服務使應用程式、工具或設備能夠將文本轉換為類似人類的合成語音。



- 語系支援

- 細節請參考下面連結取的所使用的地區位置是否有相關服務與支援
 - <https://docs.microsoft.com/zh-tw/azure/cognitive-services/speech-service/language-support#text-to-speech>

- 如語音轉文字服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 於Python應用程式中設定API key、Region

- 從鍵盤輸入字句轉成合成語音

```
# Creates a speech synthesizer using the default speaker as audio output.
speech_synthesizer = speechsdk.SpeechSynthesizer(speech_config=speech_config)

# Receives a text from console input.
print("Type some text that you want to speak...")
text = input()

# Synthesizes the received text to speech.
# The synthesized speech is expected to be heard on the speaker with this line executed.
result = speech_synthesizer.speak_text_async(text).get()

# Checks result.
if result.reason == speechsdk.ResultReason.SynthesizingAudioCompleted:
    print("Speech synthesized to speaker for text {}".format(text))
elif result.reason == speechsdk.ResultReason.Canceled:
    cancellation_details = result.cancellation_details
    print("Speech synthesis canceled: {}".format(cancellation_details.reason))
    if cancellation_details.reason == speechsdk.CancellationReason.Error:
        if cancellation_details.error_details:
            print("Error details: {}".format(cancellation_details.error_details))
print("Did you update the subscription info?")
```

- 從鍵盤輸入字句轉成音訊檔案

```
# Creates an audio configuration that points to an audio file.  
# Replace with your own audio filename.  
audio_filename = "helloworld.wav"  
audio_output = speechsdk.AudioOutputConfig(filename=audio_filename)  
  
# Creates a synthesizer with the given settings  
speech_synthesizer = speechsdk.SpeechSynthesizer(speech_config=speech_config, audio_config=audio_output)  
  
# Synthesizes the text to speech.  
# Replace with your own text.  
text = "Hello world!"  
result = speech_synthesizer.speak_text_async(text).get()  
  
# Checks result.  
if result.reason == speechsdk.ResultReason.SynthesizingAudioCompleted:  
    print("Speech synthesized to {} for text {}".format(audio_filename, text))  
elif result.reason == speechsdk.ResultReason.Canceled:  
    cancellation_details = result.cancellation_details  
    print("Speech synthesis canceled: {}".format(cancellation_details.reason))  
    if cancellation_details.reason == speechsdk.CancellationReason.Error:  
        if cancellation_details.error_details:  
            print("Error details: {}".format(cancellation_details.error_details))  
print("Did you update the subscription info?")
```

- 開啟Demo_11-3.ipynb
- 執行Azure語音Web服務操作
- 執行語音轉文字服務操作
- 執行文字轉語音服務操作

- Azure語音介紹
- Azure語音分析相關方法
- 了解如何使用語音轉文字服務
- 了解如何使用文字轉語音服務



- Lab01: 建立Azure語音服務
- Lab02: 執行語音轉文字服務操作
- Lab03: 執行文字轉語音服務操作

Estimated time:
20 minutes



Module 12

語音服務應用

學習目標：

- 12-1:語音翻譯服務介紹
- 12-2:說話者辨識服務介紹
- 12-3:語音服務總結

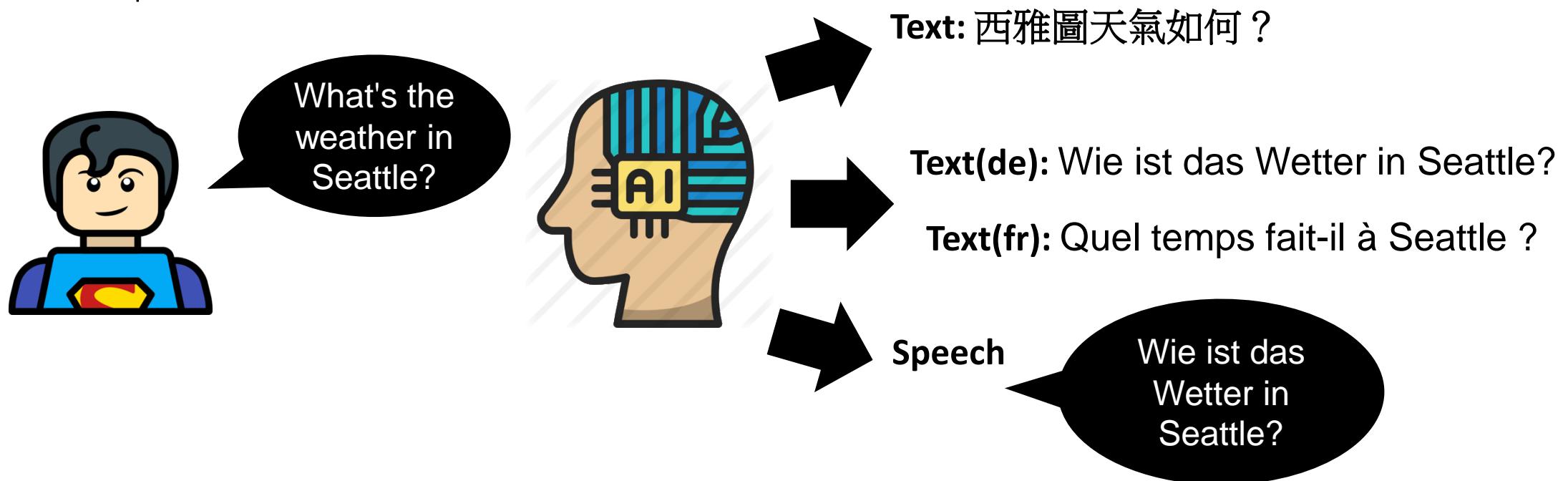
12-1:語音翻譯服務介 紹

學習目標：

- 認識語音翻譯服務
- 使用語音翻譯服務

- 語音翻譯

- 借助語音 SDK，應用程式、工具和設備可以提供音訊的翻譯輸出。
- 支援語音轉成翻譯文字、語音轉成多國翻譯文字、語音轉成多國語音輸出



- 語音翻譯語系限制

- 請參考以下網址找尋欲翻譯的原始音源是否屬於支援的語系
- <https://docs.microsoft.com/zh-tw/azure/cognitive-services/speech-service/language-support#speech-translation>

文字語言

| 文字語言 | 語言代碼 |
|-------------|------|
| 南非荷蘭文 | af |
| 阿拉伯文 | ar |
| 孟加拉文 | bn |
| 波士尼亞文 (拉丁文) | bs |
| 保加利亞文 | bg |
| 粵語 (繁體中文) | yue |

- 如語音轉文字服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 於Python應用程式中設定API key、Region

使用語音翻譯服務

- 語音轉成翻譯文字

```
def translate_speech_to_text():

    # Creates an instance of a speech translation config with specified subscription key and service region.
    # Replace with your own subscription key and region identifier from here: https://aka.ms/speech/sdkregion
    translation_config = speechsdk.translation.SpeechTranslationConfig(subscription=speech_key, \
        region=service_region)

    # Sets source and target languages.
    # Replace with the languages of your choice, from list found here: https://aka.ms/speech/sttt-languages
    fromLanguage = 'en-US'
    toLanguage = 'de'
    translation_config.speech_recognition_language = fromLanguage
    translation_config.add_target_language(toLanguage)

    # Creates a translation recognizer using an audio file as input.
    recognizer = speechsdk.translation.TranslationRecognizer(translation_config=translation_config)

    # Starts translation, and returns after a single utterance is recognized. The end of a
    # single utterance is determined by listening for silence at the end or until a maximum of 15
    # seconds of audio is processed. It returns the recognized text as well as the translation.
    # Note: Since recognize_once() returns only a single utterance, it is suitable only for single
    # shot recognition like command or query.
    # For long-running multi-utterance recognition, use start_continuous_recognition() instead.
    print("Say something...")
    result = recognizer.recognize_once()

    # Check the result
    if result.reason == speechsdk.ResultReason.TranslatedSpeech:
        print("RECOGNIZED '{}': {}".format(fromLanguage, result.text))
        print("TRANSLATED into {}: {}".format(toLanguage, result.translations['de']))
    elif result.reason == speechsdk.ResultReason.RecognizedSpeech:
        print("RECOGNIZED: {} (text could not be translated)".format(result.text))
    elif result.reason == speechsdk.ResultReason.NoMatch:
        print("NOMATCH: Speech could not be recognized: {}".format(result.no_match_details))
    elif result.reason == speechsdk.ResultReason.Canceled:
        print("CANCELED: Reason={}".format(result.cancellation_details.reason))
        if result.cancellation_details.reason == speechsdk.CancellationReason.Error:
            print("CANCELED: ErrorDetails={}".format(result.cancellation_details.error_details))

translate_speech_to_text()
```

- 語音轉成多國翻譯文字

```
def translate_speech_to_text():

    # Creates an instance of a speech translation config with specified subscription key and service region.
    # Replace with your own subscription key and region identifier from here: https://aka.ms/speech/sdkregion
    translation_config = speechsdk.translation.SpeechTranslationConfig(subscription=speech_key, \
        region=service_region)

    # Sets source and target languages.
    # Replace with the languages of your choice, from list found here: https://aka.ms/speech/sttt-languages
    fromLanguage = 'en-US'
    translation_config.speech_recognition_language = fromLanguage
    translation_config.add_target_language('de')
    translation_config.add_target_language('fr')

    # Creates a translation recognizer using and audio file as input.
    recognizer = speechsdk.translation.TranslationRecognizer(translation_config=translation_config)

    # Starts translation, and returns after a single utterance is recognized. The end of a
    # single utterance is determined by listening for silence at the end or until a maximum of 15
    # seconds of audio is processed. It returns the recognized text as well as the translation.
    # Note: Since recognize_once() returns only a single utterance, it is suitable only for single
    # shot recognition like command or query.
    # For long-running multi-utterance recognition, use start_continuous_recognition() instead.
    print("Say something...")
    result = recognizer.recognize_once()

    # Check the result
    if result.reason == speechsdk.ResultReason.TranslatedSpeech:
        print("RECOGNIZED '{0}': {1}".format(fromLanguage, result.text))
        print("TRANSLATED into '{0}': {1}".format('de', result.translations['de']))
        print("TRANSLATED into '{0}': {1}".format('fr', result.translations['fr']))
    elif result.reason == speechsdk.ResultReason.RecognizedSpeech:
        print("RECOGNIZED: {} (text could not be translated)".format(result.text))
    elif result.reason == speechsdk.ResultReason.NoMatch:
        print("NOMATCH: Speech could not be recognized: {}".format(result.no_match_details))
    elif result.reason == speechsdk.ResultReason.Canceled:
        print("CANCELED: Reason={}".format(result.cancellation_details.reason))
        if result.cancellation_details.reason == speechsdk.CancellationReason.Error:
            print("CANCELED: ErrorDetails={}".format(result.cancellation_details.error_details))

translate_speech_to_text()
```

使用語音翻譯服務

● 語音轉成翻譯語言

```
def translate_speech_to_speech():

    # Creates an instance of a speech translation config with specified subscription key and service region.
    # Replace with your own subscription key and region identifier from here: https://aka.ms/speech/sdkregion
    translation_config = speechsdk.translation.SpeechTranslationConfig(subscription=speech_key, \
                                                                        region=service_region)

    # Sets source and target languages.
    # Replace with the languages of your choice, from list found here: https://aka.ms/speech/sttt-languages
    fromLanguage = 'en-US'
    toLanguage = 'de'
    translation_config.speech_recognition_language = fromLanguage
    translation_config.add_target_language(toLanguage)

    # Sets the synthesis output voice name.
    # Replace with the languages of your choice, from list found here: https://aka.ms/speech/tts-languages
    translation_config.voice_name = "de-DE-Hedda"

    # Creates a translation recognizer using an audio file as input.
    recognizer = speechsdk.translation.TranslationRecognizer(translation_config=translation_config)

    # Prepare to handle the synthesized audio data.
    def synthesis_callback(evt):
        size = len(evt.result.audio)
        print('AUDIO SYNTHESIZED: {} byte(s) {}'.format(size, '(COMPLETED)' if size == 0 else ''))

    recognizer.synthesizing.connect(synthesis_callback)

    # Starts translation, and returns after a single utterance is recognized. The end of a
    # single utterance is determined by listening for silence at the end or until a maximum of 15
    # seconds of audio is processed. It returns the recognized text as well as the translation.
    # Note: Since recognize_once() returns only a single utterance, it is suitable only for single
    # shot recognition like command or query.
    # For long-running multi-utterance recognition, use start_continuous_recognition() instead.
    print("Say something...")
    result = recognizer.recognize_once()

    # Check the result
    if result.reason == speechsdk.ResultReason.TranslatedSpeech:
        print("RECOGNIZED '{}': {}".format(fromLanguage, result.text))
        print("TRANSLATED into {}: {}".format(toLanguage, result.translations['de']))
    elif result.reason == speechsdk.ResultReason.RecognizedSpeech:
        print("RECOGNIZED: {} (text could not be translated)".format(result.text))
    elif result.reason == speechsdk.ResultReason.NoMatch:
        print("NOMATCH: Speech could not be recognized: {}".format(result.no_match_details))
    elif result.reason == speechsdk.ResultReason.Canceled:
        print("CANCELED: Reason={}".format(result.cancellation_details.reason))
        if result.cancellation_details.reason == speechsdk.CancellationReason.Error:
            print("CANCELED: ErrorDetails={}".format(result.cancellation_details.error_details))

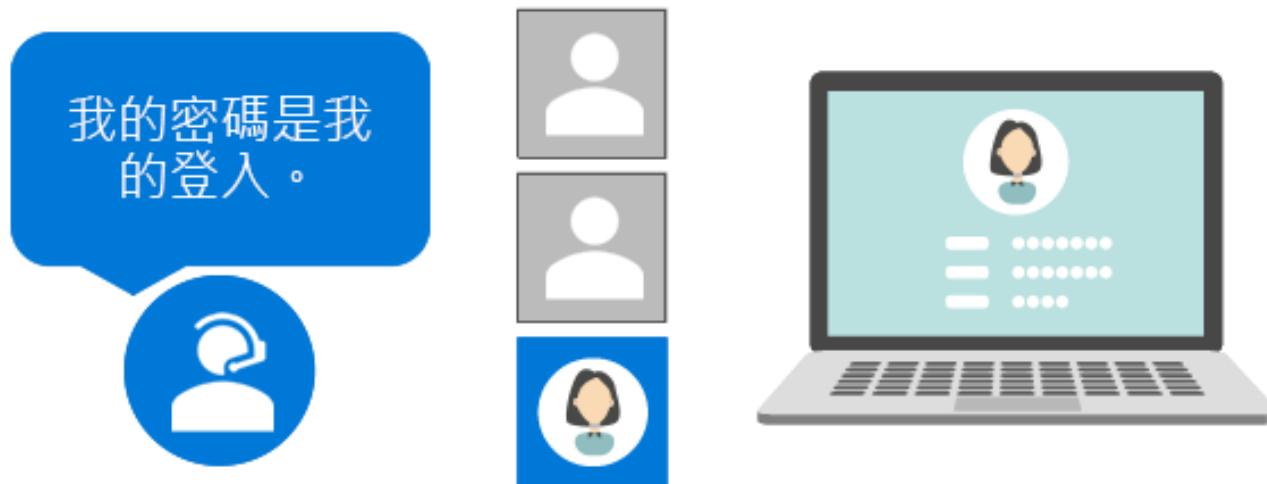
translate_speech_to_speech()
```

學習目標：

12-2:說話者辨識服務 介紹

- 了解說話者辨識概念
- 使用說話者辨識服務

- 說話者辨識功能可辨認說話者
 - 說話者驗證：
 - 使用其聲音或語音來自動檢查並驗證使用者。
 - 說話者辨識：
 - 將聲音與一組可能說話者進行比較，來自動辨識音訊檔案中的說話者。



- 每個人說話的聲音都獨一無二，稱之聲紋

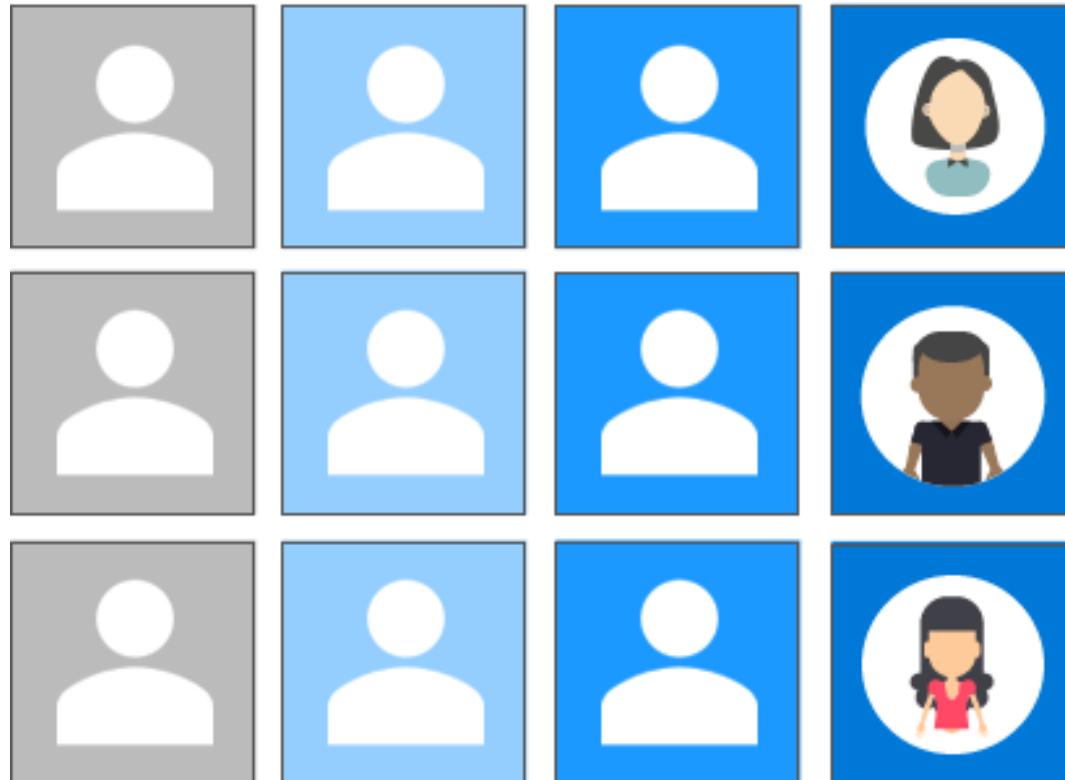
- 說話者辨識 API 提供

- 驗證：

- 檢查這兩種聲音屬於同一個人的可能性。

- 識別和辨識：

- 判斷某個聲音是否符合其他已知聲音。



- 註冊

- 擷取說話者聲音和語音模式來形成「聲音模型」以供日後比較聲紋。



- 注意

- 「說話者辨識」與「語音辨識」之間有所差異。語音辨識會嘗試判斷「說話內容」，而說話者辨識會嘗試識別「內容的說話者」。

使用說話者辨識服務

- 範例：點選音訊，每一陣子服務會辨認出說話者
 - <https://azure.microsoft.com/zh-tw/services/cognitive-services/speaker-recognition/>



文字 JSON

總統 Barack Obama
是所選音訊中識別的說話者。

□ 停止

▷ 音訊 2

□ 停止

▷ 音訊 4

▷ 音訊 5

▷ 音訊 6

學習目標：

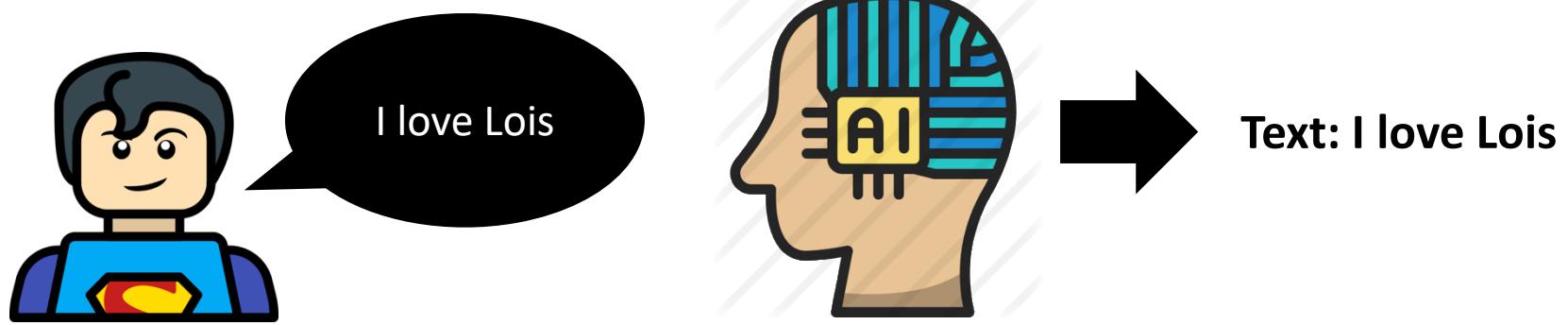
12-3:語音服務總結

- 語音服務總結
- 語音服務定價

語音服務總結

- 語音轉文字

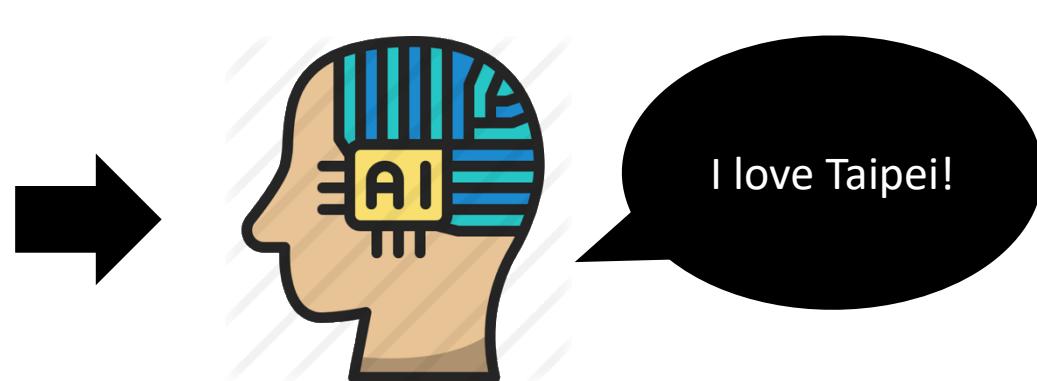
- 可將音訊串流的即時轉譯為文字。



- 文字轉語音

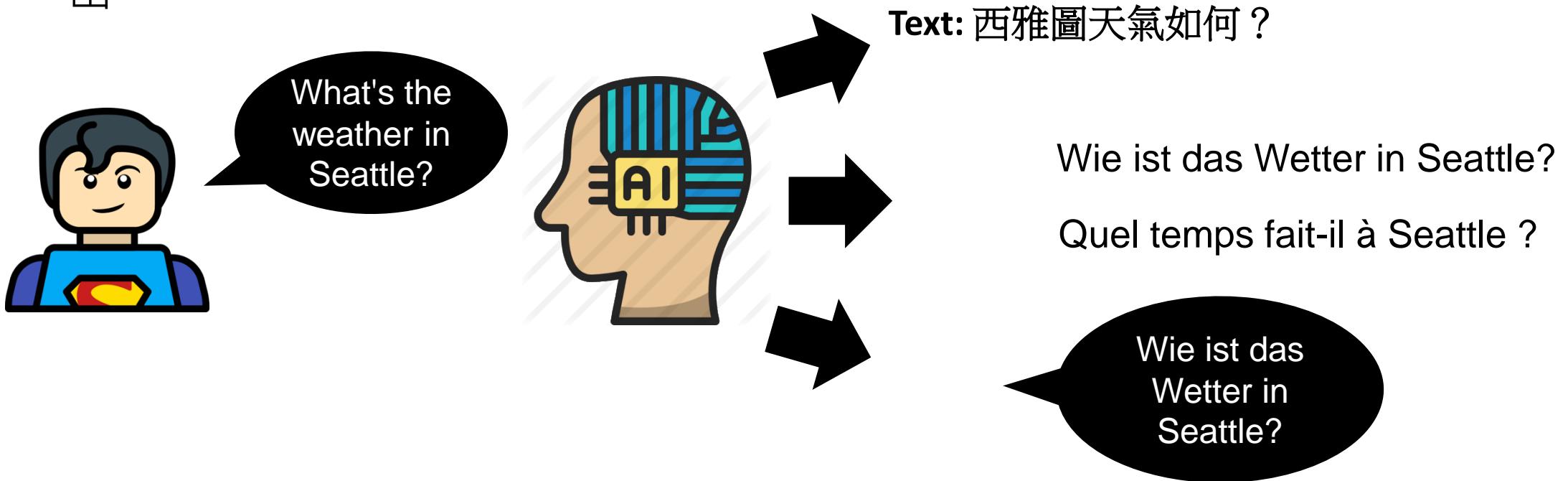
- 語音服務中的文字轉語音服務使應用程式、工具或設備能夠將文本轉換為類似人類的合成語音。

Text: I love Taipei!



- 語音翻譯

- 借助語音 SDK，應用程式、工具和設備可以提供音訊的翻譯輸出。
- 支援語音轉成翻譯文字、語音轉成多國翻譯文字、語音轉成多國語音輸出



語音服務總結

- 說話者辨識
 - 透過驗證、識別與辨識來查詢說話者是不是已經註冊的使用者。



- 定價資料

- <https://azure.microsoft.com/zh-tw/pricing/details/cognitive-services/speech-services/>

- 不同地區的價格可能不同
- 請至上述網址輸入使用服務的地區位置查詢真實價格。

The screenshot shows the Microsoft Azure Pricing page for the Speech Services. It includes dropdown menus for selecting the service (語音服務), region (美國西部), and currency (新台幣 (NT\$)). The main content is a table detailing the pricing for various speech-related services across different categories: Text-to-Speech, Speech-to-Text, and Translation.

| 執行個體 | 類別 | 功能 | 價格 |
|--------------------------------------|---------------------------|---|------------------|
| 免費 - Web/容器 1 個並行要求 ¹ | 語音轉換文字 | 標準 | 每月免費 5 音訊小時 |
| | 自訂 | 每月免費 5 音訊小時 端點裝載：每月免費 1 個模型 ² | |
| | 交談轉譯多聲道音訊 ^{預覽 3} | 每月免費 5 音訊小時 | |
| | 文字轉換語音 | 標準 | 每月免費 5 百萬個字元 |
| | 神經網路 | 每月免費 0.5 百萬個字元 | |
| | 自訂 | 每月免費 5 百萬個字元 端點裝載：每月免費 1 個模型 | |
| | 語音翻譯 | 標準 | 每月免費 5 音訊小時 |
| 標準 - Web/容器 20 個並行要求 ¹ | 語音轉換文字 | 標準 | 每音訊小時 NT\$30.055 |

- 開啟Demo_12-3.ipynb
- 執行語音翻譯服務操作
- 執行說者話識別web示範

- 了解如何使用語音翻譯服務
- 認識說話者辨識服務
- 了解語音服務支援的區域與定價



- Lab01: 語音轉成翻譯文字服務操作
- Lab02: 語音轉成多國翻譯文字服務操作
- Lab03: 語音轉成多國語音服務操作

Estimated time:
20 minutes



Module 13

語言服務

學習目標：

- 13-1: Azure 語言服務介紹
- 13-2: 文字情感分析服務介紹
- 13-3: 文字語言偵測服務介紹

學習目標：

13-1:Azure語言服務 介紹

- 語言相關服務介紹
- 建置Azure語言服務環境

- 從非結構化文字擷取意義
 - 允許應用程式使用預先建立的指令碼處理自然語言、評估情感，以及了解如何辨識使用者想要的內容

See Language Understanding in action

What the user says (utterances)



What LUIS returns

```
{  
  "query": "Where is the nearest club?",  
  "prediction": {  
    "topIntent": "LocationFinder",  
    "intents": {  
      "LocationFinder": {  
        "score": 0.8996477  
      },  
      "None": {  
        "score": 0.08387748  
      },  
      "BookFlight": {  
        "score": 0.00309361285  
      }  
    }  
  }  
}
```

- 服務應用
 - 偵測並翻譯、語言理解、偵測情感、關鍵片語、協助讀者用視覺提示理解文字、透過預先設定好的資料建立對話
 - 後面章節將會說明使用方式

建置Azure語言服務環境

- 於認知服務主控台點選新增
 - 進到Marketplace後選擇AI + 機器學習服務
 - 右側Cognitive Services區域選擇查看更多
 - 點選文字分析圖像



首頁 > 認知服務 > Marketplace

Bing 實體搜尋 place

我已儲存的清單
最近建立
服務提供者
類別
開始使用
AI + 機器學習服務
分析
區塊鍵
計算
容器
資料庫
開發人員工具
DevOps
身分識別
整合
物聯網
管理工具
媒體
混合環境
網路
安全性
軟體即服務 (SaaS)
儲存體
Web

搜尋 Marketplace
定價：所有
作業系統：所有
發行者：所有

AI + 機器學習服務 > Cognitive Services

| | | | |
|--|---|--|--|
| Bing 自動建議 Microsoft 新增智慧型預先鍵入功能來協助使用者更快完成查詢 | Bing 自訂搜尋 Microsoft 建立沒有廣告結果的自訂搜尋引擎 | Bing 實體搜尋 Microsoft 辨識和分類具名實體，並根據這些實體尋找搜尋結果 | Bing 搜尋 Microsoft 利用領先的搜尋引擎 Bing 來為使用者提供安全且無廣告的搜尋 |
| 電腦視覺 Microsoft 能夠分析影像中內容的 AI 服務 | Content Moderator Microsoft 能夠偵測不當內容的 AI 服務 | 自訂視覺 Microsoft 供您將電腦視覺套用到特定案例的 AI 服務與完善的平臺 | 臉部 Microsoft 此 AI 服務可分析影像中的臉部 |
| 語音 Microsoft 能讓您為應用程式內嵌近似人類語音功能的 AI 服務 | QnA Maker Microsoft 此 AI 服務可讓您在幾分鐘內從資料建立 Bot | 文字分析 Microsoft 此 AI 服務可讓您使用情感分析、具名實體辨識、語言偵測及提供多種語言的關鍵片語擴取，從自然語言 | Translator Text (翻譯文字) Microsoft 此 AI 服務可讓您輕鬆翻譯超過 60 種語言的文字 |

- 建立文字分析服務

- 點選建立
- 填寫資料：名稱、位置、定價層、新建資源群組

首頁 > 認知服務 > Marketplace > 文字分析

文字分析 Microsoft

文字分析  儲存以便稍後使用 Microsoft

建立

概觀 方案

使用進階自然語言處理，從非結構化文字中獲得見解。使用情感分析找出客戶對品牌的看法。使用關鍵片語檢取來尋找與主題相關的片語，並以語言偵測來識別文字的語言。使用具名實體辨識來偵測文字中的實體並加以分類。

實用的連結
深入了解文字分析 API
文件
程式碼範例
API 參考
最新功能
定價
區域可用性

首頁 > 建立

建立

Text Analytics

名稱 *

TxtAnalyze

訂用帳戶 *

免費試用版

位置 *

(US) 美國西部

定價層 (檢視完整定價詳細資料) *

F0 (5K 每 30 天交易數)

資源群組 *

CogAI

新建

建置 Azure 語音服務環境

- 完成部署後可以在認知服務中控台看到建立的服務內容

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and a user profile icon. The main content area has a blue header bar with the text "Microsoft Azure". Below this, the breadcrumb navigation shows "首頁 > 認知服務 > TxtAnalyze".

The left sidebar is titled "認知服務" (Cognitive Services) and lists several resources: "Luis_resource_name", "moduleComputerVision", "moduleSpeechReg", and "TxtAnalyze".

The right pane displays the details for the "TxtAnalyze" resource. The title bar says "TxtAnalyze" under "認知服務".

The "概觀" (Overview) section contains the following information:

- 資源群組 (變更) : CogAI
- 狀態 : 作用中
- 位置 : 美國西部
- 訂用帳戶 (變更) : 免費試用版
- 訂用帳戶識別碼 : 9a37f675-345c-4dd8-baad-4b7cf2748b3f
- 標籤 (變更) : 按一下這裡即可新增標籤

The "配額資訊" (Quota Information) section shows:

- Free tier 總計: 5000 Calls
- 剩下 Free tier 個: 5000 Calls
- Free tier 重設週期: 30.00 天

學習目標：

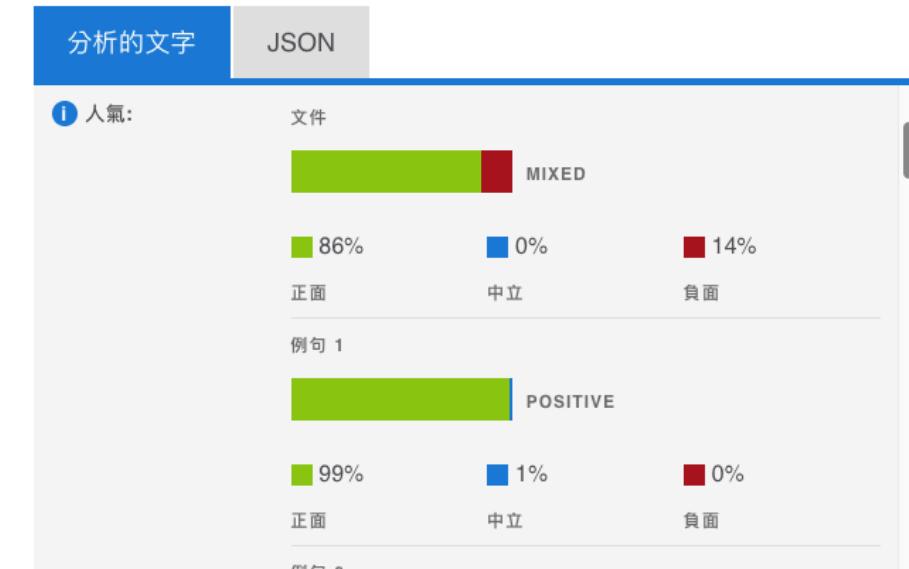
13-2:文字情感分析服務介紹

- 認識文字情感分析服務
- 使用文字情感分析服務

- 從文字分析情感
 - 文字分析 API 的情感分析功能會評估文字，並傳回每個句子的情感分數和標籤。
 - <https://azure.microsoft.com/zh-tw/services/cognitive-services/text-analytics/>

We went to Contoso Steakhouse located at midtown NYC last week for a dinner party, and we adore the spot! They provide marvelous food and they have a great menu. The chief cook happens to be the owner (I think his name is John Doe) and he is super nice, coming out of the kitchen and greeted us all. We enjoyed very much dining in the place! The Sirloin steak I ordered was tender and juicy, and the place was impeccably clean. You can even pre-order from their online menu at www.contososteakhouse.com, call 312-555-0176 or send email to order@contososteakhouse.com! The only complaint I have is the food didn't come fast enough. Overall I highly recommend it!

分析



- 此功能可用來偵測社交媒體、客戶評價及討論論壇中的正面和負面情感。
- API 所使用的 AI 模型由服務所提供之，您只需要傳送要分析的內容即可。
- 語系限制
 - 目前全面支援的語系為：英文、中文、法文、德文、西班牙文
 - 更多支援細節請參考以下網址
<https://docs.microsoft.com/zh-tw/azure/cognitive-services/text-analytics/language-support>

使用文字情感分析服務

- 安裝用戶端程式庫
 - pip install azure-ai-textanalytics
- 設定key、endpoint
 - 於認知服務台取得文字分析的API key與endpoint



- 將取得的API key與endpoint輸入

```
key = "<paste-your-text-analytics-key-here>"  
endpoint = "<paste-your-text-analytics-endpoint-here>"
```

- 驗證用戶端

```
from azure.ai.textanalytics import TextAnalyticsClient, TextAnalyticsApiKeyCredential  
  
def authenticate_client():  
    ta_credential = TextAnalyticsApiKeyCredential(key)  
    text_analytics_client = TextAnalyticsClient(  
        endpoint=endpoint, credential=ta_credential)  
    return text_analytics_client  
  
client = authenticate_client()
```

- 情感分析

```
def sentiment_analysis_example(client):

    document = ["I had the best day of my life. I wish you were there with me."]
    response = client.analyze_sentiment(inputs=document)[0]
    print("Document Sentiment: {}".format(response.sentiment))
    print("Overall scores: positive={0:.2f}; neutral={1:.2f}; negative={2:.2f}\n".format(
        response.confidence_scores.positive,
        response.confidence_scores.neutral,
        response.confidence_scores.negative,
    ))
    for idx, sentence in enumerate(response.sentences):
        print("[Length: {}]\n".format(sentence.grapheme_length))
        print("Sentence {} sentiment: {}\n".format(idx+1, sentence.sentiment))
        print("Sentence score:\nPositive={0:.2f}\nNeutral={1:.2f}\nNegative={2:.2f}\n".format(
            sentence.confidence_scores.positive,
            sentence.confidence_scores.neutral,
            sentence.confidence_scores.negative,
        ))
sentiment_analysis_example(client)
```

13-3:文字語言偵測服務介紹

學習目標：

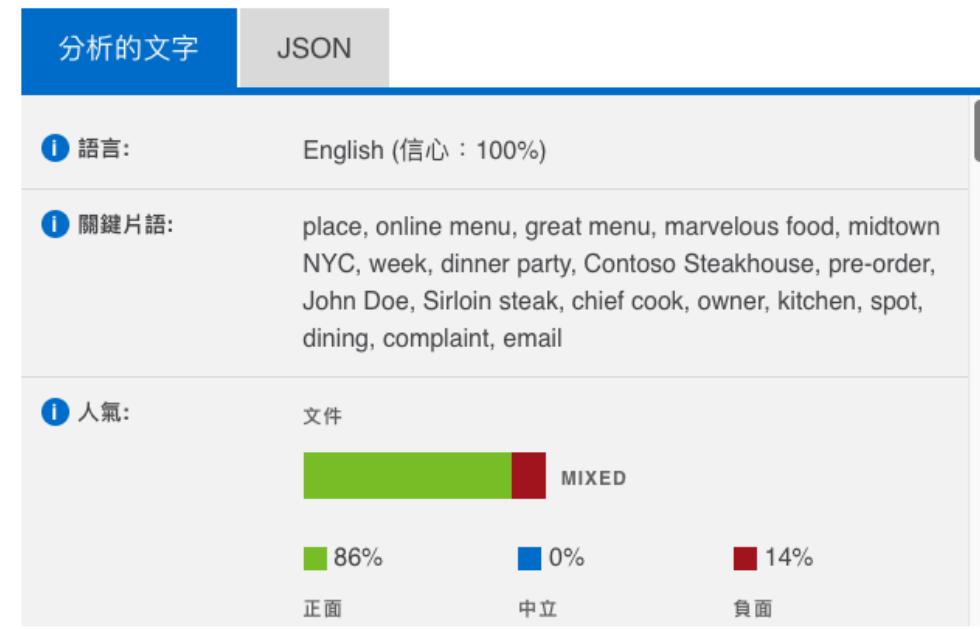
- 認識文字語言偵測服務
- 使用文字語言偵測服務

- 從文字分析語言類型

- 文字分析 API 的語言偵測功能會評估文字，並傳回偵測到的語言。
- <https://azure.microsoft.com/zh-tw/services/cognitive-services/text-analytics/>

We went to Contoso Steakhouse located at midtown NYC last week for a dinner party, and we adore the spot! They provide marvelous food and they have a great menu. The chief cook happens to be the owner (I think his name is John Doe) and he is super nice, coming out of the kitchen and greeted us all. We enjoyed very much dining in the place! The Sirloin steak I ordered was tender and juicy, and the place was impeccably clean. You can even pre-order from their online menu at www.contososteakhouse.com, call 312-555-0176 or send email to order@contososteakhouse.com! The only complaint I have is the food didn't come fast enough. Overall I highly recommend it!

分析



- 限制
 - 支援120種語言，詳細語系清單支援請參考以下網址。
 - <https://docs.microsoft.com/zh-tw/azure/cognitive-services/text-analytics/language-support>

- 如文字情感分析服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 於Python應用程式中設定API key、Endpoint
 - 於Python應用程式中設定驗證用戶端

- 語言偵測範例程式碼

```
def language_detection_example(client):
    try:
        document = ["Ce document est rédigé en Français."]
        response = client.detect_language(inputs = document, country_hint = 'us')[0]
        print("Language: ", response.primary_language.name)

    except Exception as err:
        print("Encountered exception. {}".format(err))
language_detection_example(client)
```

- 開啟Demo_13-3.ipynb
- 執行Azure語言Web示範
- 執行文字情感分析服務操作
- 執行文字語言偵測服務操作

- Azure語言介紹
- 了解如何使用文字情感分析服務
- 了解如何使用文字語言偵測服務



- Lab01: 建立Azure語言服務
- Lab02: 文字情感分析服務操作
- Lab03: 文字語言偵測服務操作

Estimated time:
20 minutes



Module 14

語言服務應用1

- 14-1:關鍵字詞擷取
服務介紹
- 14-2:實體辨識服務
介紹
- 14-3:文本翻譯服務
介紹

學習目標：

14-1: 關鍵字詞擷取服務介紹

- 認識關鍵字詞擷取服務
- 使用關鍵字詞擷取服務

- 關鍵片語評估非結構化的文字，傳回要點清單。
 - 輸入：“The food was delicious and there were wonderful staff” ，
 - 傳回：“food” 和 “wonderful staff” 。

We went to Contoso Steakhouse located at midtown NYC last week for a dinner party, and we adore the spot! They provide marvelous food and they have a great menu. The chief cook happens to be the owner (I think his name is John Doe) and he is super nice, coming out of the kitchen and greeted us all. We enjoyed very much dining in the place! The Sirloin steak I ordered was tender and juicy, and the place was impeccably clean. You can even pre-order from their online menu at www.contososteakhouse.com, call 312-555-0176 or send email to order@contososteakhouse.com! The only complaint I have is the food didn't come fast enough. Overall I highly recommend it!

分析



- 語系限制與文字情感分析服務一樣：
 - 僅支援英文、中文、法文、德文、西文
 - 詳細內容請參考以下網址：
<https://docs.microsoft.com/zh-tw/azure/cognitive-services/text-analytics/language-support>

- 如文字情感分析服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 於Python應用程式中設定API key、Endpoint
 - 於Python應用程式中設定驗證用戶端

- 關鍵字詞擷取範例程式碼

```
def key_phrase_extraction_example(client):  
  
    try:  
        document = ["My cat might need to see a veterinarian."  
  
                    response = client.extract_key_phrases(inputs= document)[0]  
  
                    if not response.is_error:  
                        print("\tKey Phrases:")  
                        for phrase in response.key_phrases:  
                            print("\t\t", phrase)  
                    else:  
                        print(response.id, response.error)  
  
    except Exception as err:  
        print("Encountered exception. {}".format(err))  
  
key_phrase_extraction_example(client)
```

14-2: 實體辨識服務介紹

學習目標：

- 認識實體辨識服務
- 使用實體辨識服務

- 實體辨識服務可以：
 - 透過查詢網路知識庫來分辨同字不同義的能力

We went to Contoso Steakhouse located at midtown NYC last week for a dinner party, and we adore the spot! They provide marvelous food and they have a great menu. The chief cook happens to be the owner (I think his name is John Doe) and he is super nice, coming out of the kitchen and greeted us all. We enjoyed very much dining in the place! The Sirloin steak I ordered was tender and juicy, and the place was impeccably clean. You can even pre-order from their online menu at www.contososteakhouse.com, call 312-555-0176 or send email to order@contososteakhouse.com! The only complaint I have is the food didn't come fast enough. Overall I highly recommend it!

| 分析的文字 | JSON | Wikipedia link |
|--|---|--|
| <p>分析的文字內容</p> <p>連結實體:</p> <p>We went to Contoso Steakhouse located at midtown NYC last week for a dinner party, and we adore the spot! They provide marvelous food and they have a great menu. The chief cook happens to be the owner (I think his name is John Doe) and he is super nice, coming out of the kitchen and greeted us all. We enjoyed very much dining in the place! The Sirloin steak I ordered was tender and juicy, and the place was impeccably clean. You can even pre-order from their online menu at www.contososteakhouse.com, call 312-555-0176 or send email to order@contososteakhouse.com! The only complaint I have is the food didn't come fast enough. Overall I highly recommend it!</p> | <p>分析的 JSON 數據</p> <pre>[{"text": "We went to Contoso Steakhouse located at midtown NYC last week for a dinner party, and we adore the spot! They provide marvelous food and they have a great menu. The chief cook happens to be the owner (I think his name is John Doe) and he is super nice, coming out of the kitchen and greeted us all. We enjoyed very much dining in the place! The Sirloin steak I ordered was tender and juicy, and the place was impeccably clean. You can even pre-order from their online menu at www.contososteakhouse.com, call 312-555-0176 or send email to order@contososteakhouse.com! The only complaint I have is the food didn't come fast enough. Overall I highly recommend it!", "entities": [{"name": "Contoso Steakhouse", "type": "Business", "start": 10, "end": 25}, {"name": "NYC", "type": "Location", "start": 26, "end": 30}, {"name": "John Doe", "type": "Person", "start": 45, "end": 55}, {"name": "Sirloin steak", "type": "Food", "start": 65, "end": 75}, {"name": "www.contososteakhouse.com", "type": "URL", "start": 85, "end": 95}]}]</pre> | <p>Wikipedia 鏈接</p> <p>We went to Contoso Steakhouse located at midtown NYC last week for a dinner party, and we adore the spot! They provide marvelous food and they have a great menu. The chief cook happens to be the owner (I think his name is John Doe) and he is super nice, coming out of the kitchen and greeted us all. We enjoyed very much dining in the place! The Sirloin steak I ordered was tender and juicy, and the place was impeccably clean. You can even pre-order from their online menu at www.contososteakhouse.com, call 312-555-0176 or send email to order@contososteakhouse.com! The only complaint I have is the food didn't come fast enough. Overall I highly recommend it!</p> |

- 實體辨識服務可以：
 - 將字詞分辨為不同類型（人、地點、連結、郵件位址...）

We went to Contoso Steakhouse located at midtown NYC last week for a dinner party, and we adore the spot! They provide marvelous food and they have a great menu. The chief cook happens to be the owner (I think his name is John Doe) and he is super nice, coming out of the kitchen and greeted us all. We enjoyed very much dining in the place! The Sirloin steak I ordered was tender and juicy, and the place was impeccably clean. You can even pre-order from their online menu at www.contososteakhouse.com, call 312-555-0176 or send email to order@contososteakhouse.com! The only complaint I have is the food didn't come fast enough. Overall I highly recommend it!

分析

分析的文字

JSON

具名實體:

Contoso Steakhouse [Organization]
Steakhouse [Location]
midtown NYC [Location]
NYC [Location-GPE]
last week [DateTime-DateRange]
dinner party [Event]
chief cook [PersonType]
owner [PersonType]
John Doe [Person]
kitchen [Location]
www.contososteakhouse.com [URL]
312-555-0176 [PhoneNumber]
email [Skill]
order@contososteakhouse.com [Email]

- 如文字情感分析服務章節所提：
 - 完成安裝程式庫
 - 於Python應用程式中載入相關套件
 - 於Python應用程式中設定API key、Endpoint
 - 於Python應用程式中設定驗證用戶端

- 實體辨識範例1程式碼

```
def entity_linking_example(client):

    try:
        document = ["""Microsoft was founded by Bill Gates and Paul Allen on April 4, 1975,
to develop and sell BASIC interpreters for the Altair 8800.
During his career at Microsoft, Gates held the positions of chairman,
chief executive officer, president and chief software architect,
while also being the largest individual shareholder until May 2014.""""]
        result = client.recognize_linked_entities(inputs= document)[0]

        print("Linked Entities:\n")
        for entity in result.entities:
            print("\tName: ", entity.name, "\tId: ", entity.data_source_entity_id, "\tUrl: ", entity.url,
                  "\n\tData Source: ", entity.data_source)
        print("\tMatches:")
        for match in entity.matches:
            print("\t\tText:", match.text)
            print("\t\tScore: {:.2f}".format(match.score), "\tLength: {}".format(match.grapheme_length))

    except Exception as err:
        print("Encountered exception. {}".format(err))
entity_linking_example(client)
```

- 實體辨識範例2程式碼
 - 從文字中辨認出個人資訊並註記

```
def entity_pii_example(client):  
  
    document = ["Insurance policy for SSN on file 123-12-1234 is here by approved."  
  
    result = client.recognize_pii_entities(inputs= document)[0]  
  
    print("Personally Identifiable Information Entities: ")  
    for entity in result.entities:  
        print("\tText: ",entity.text," \tCategory: ", entity.category," \tSubCategory: ", entity.subcategory)  
        print("\t\tLength: ", entity.grapheme_length, "\tScore: {0:.2f}".format(entity.score), "\n")  
  
entity_pii_example(client)
```

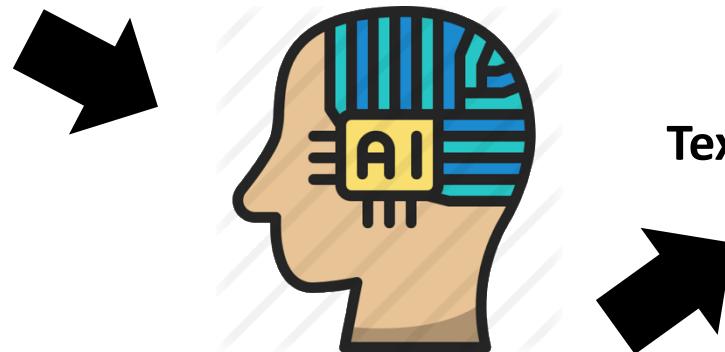
14-3:文本翻譯服務介 紹

學習目標：

- 認識文本翻譯服務
- 使用文本翻譯服務

- 文本翻譯
 - 將文字轉換成另外一種語言的同義文字

Text(en): What's the weather in Seattle ?



Text(de): Wie ist das Wetter in Seattle?

- 支援的語言清單如下網址：

<https://docs.microsoft.com/zh-tw/azure/cognitive-services/translator/language-support>

使用文本翻譯服務

- 於認知服務主控台點選新增
 - 進到Marketplace後選擇AI + 機器學習服務
 - 右側Cognitive Services區域選擇查看更多
 - 點選翻譯文字圖像



首頁 > 認知服務 > Marketplace

Marketplace

我已儲存的清單
搜尋 Marketplace
定價 : 所有
作業系統 : 所有
發行者 : 所有

AI + 機器學習服務 > Cognitive Services

| | | | |
|--|---|--|--|
| Bing 自動建議 Microsoft 新增智慧型預先鍵入功能來協助使用者更快完成查詢 | Bing 自訂搜尋 Microsoft 建立沒有廣告結果的自訂搜尋引擎 | Bing 實體搜尋 Microsoft 辨識和分類具名實體，並根據這些實體尋找搜尋結果 | Bing 搜尋 Microsoft 利用領先的搜尋引擎 Bing 來為使用者提供安全且無廣告的搜尋 |
| 電腦視覺 Microsoft 能夠分析影像中內容的 AI 服務 | Content Moderator Microsoft 能夠偵測不當內容的 AI 服務 | 自訂視覺 Microsoft 供您將電腦視覺套用到特定案例的 AI 服務與完善的平台 | 臉部 Microsoft 此 AI 服務可分析影像中的臉部 |
| 語音 Microsoft 能讓您為應用程式內嵌近似人類語音功能的 AI 服務 | QnA Maker Microsoft 此 AI 服務可讓您在幾分鐘內從資料建立 Bot | 文字分析 Microsoft 此 AI 服務可讓您使用情感分析、具名實體辨識、語言偵測及提供多種語言的關鍵片語擴取，從自然語言 | Translator Text (翻譯文字) Microsoft 此 AI 服務可讓您輕鬆翻譯超過 60 種語言的文字 |

- 建立文本翻譯服務

- 點選建立
- 填寫資料：名稱、位置、定價層、新建資源群組

Translator Text (翻譯文字)
Microsoft



概觀 方案

輕鬆地將即時文字翻譯功能整合到應用程式的網站、工具或任何需要多語言支援的解決方案，例如網站當地語系化、電子商務、客戶支援、訊息應用和實用的連結
[深入了解 Translator Text API](#)
文件
[API 參考](#)
定價
[區域可用性](#)

建立

Translator Text (翻譯文字)

名稱 *

textTranslateCogService

訂用帳戶 *

免費試用版

定價層 ([檢視完整定價詳細資料](#)) *

F0 (2M 每月最多翻譯 2 百萬個字元)

資源群組 *

CogAI

[新建](#)

使用文本翻譯服務

- 完成部署後可以在認知服務中控台看到建立的服務內容

The screenshot shows the Azure portal interface for managing cognitive services. The main title is "Microsoft.CognitiveServicesTextTranslation | 概觀". On the left, there's a sidebar with "概觀" selected, along with other options like "輸入", "輸出", and "範本". The main content area displays deployment status: "您的部署正在進行" (Deployment is in progress). It shows the deployment name: "Microsoft.CognitiveServicesTextTranslation", start time: "30/3/2020 下午4:23:06", and resource group: "CogAI". Below this, there's a section for "部署詳細資料 (下載)" (Deployment details download). The bottom part of the page lists five deployed resources: Luis_resource_name (LUIS.Authoring, US West, F0), moduleComputerVision (ComputerVision, Southeast Asia, F0), moduleSpeechReg (SpeechServices, US West 2, F0), textTranslateCogservice (TextTranslation, global, F0), and TxtAnalyze (TextAnalytics, US West, F0). The right side of the page includes links for "資訊安全中心" (Information Security Center) and "Azure 專家" (Azure Experts).

| 名稱 | 種類 | 位置 | 自訂網域 | 定價層 |
|-------------------------|-----------------|--------|---------------|-----|
| Luis_resource_name | LUIS.Authoring | 美國西部 | | F0 |
| moduleComputerVision | ComputerVision | 東南亞 | module... | F0 |
| moduleSpeechReg | SpeechServices | 美國西部 2 | | F0 |
| textTranslateCogservice | TextTranslation | global | texttransl... | F0 |
| TxtAnalyze | TextAnalytics | 美國西部 | txtanalyze | F0 |

- 安裝模組
 - pip install requests, uuid
- 於Python中匯入模組

```
# -*- coding: utf-8 -*-
import os, requests, uuid, json
```

- 設定環境變數
 - Windows中開啟命令提示字元輸入以下命令並重啟相關服務
 - Key請參考下一頁API key取得方式

console

```
setx COGNITIVE_SERVICE_KEY "your-key"
```

使用文本翻譯服務

- 設定API Key
 - 至認知服務主控台複製
 - 於Python中設定API

 **textTranslateCogService | 快速入門**
認知服務

搜尋 (Cmd +/)

概觀

活動記錄

存取控制 (IAM)

標籤

診斷並解決問題

資源管理

快速入門

Keys and Endpoint

定價層

恭喜! 您的金鑰已就緒。

立即探索快速入門指導，以啟動並執行 Translator Text (翻譯文字)。

以下是金鑰以及您要開始使用的端點:

1 Key1 

端點 
<https://texttranslatecogservice.cognitiveservices.azure.com/sts/v1.0/issuetoken>

每次呼叫翻譯工具文字 API 都需要上方的訂用帳戶金鑰。您必須透過查詢字串參數傳遞此金鑰，或在要求標頭中包含此金鑰。

2 進行 API 呼叫

取得 API 之屬性及方法的深入資訊。您無須撰寫任何程式碼，只要使用內建的測試主控台，就能測試金鑰。

API 參考

使用文本翻譯服務

- 文本翻譯範例程式碼

```
key_var_name = 'XXXXXXXXXX'  
os.environ[key_var_name] = key_var_name  
if not key_var_name in os.environ:  
    raise Exception('Please set/export the environment variable: {}'.format(key_var_name))  
subscription_key = os.environ[key_var_name]  
  
endpoint_var_name = 'https://api.cognitive.microsofttranslator.com/'  
os.environ[endpoint_var_name] = endpoint_var_name  
if not endpoint_var_name in os.environ:  
    raise Exception('Please set/export the environment variable: {}'.format(endpoint_var_name))  
endpoint = os.environ[endpoint_var_name]  
  
# If you encounter any issues with the base_url or path, make sure  
# that you are using the latest endpoint: https://docs.microsoft.com/azure/cognitive-services/translate-api/v3  
path = '/translate?api-version=3.0'  
params = '&to=de&to=it'  
constructed_url = endpoint + path + params  
  
headers = {  
    'Ocp-Apim-Subscription-Key': subscription_key,  
    'Content-type': 'application/json',  
    'X-ClientTraceId': str(uuid.uuid4())  
}  
  
# You can pass more than one object in body.  
body = [{  
    'text' : 'Hello World!'  
}]  
request = requests.post(constructed_url, headers=headers, json=body)  
response = request.json()  
  
print(json.dumps(response, sort_keys=True, indent=4, separators=(',', ': ')))
```

- 開啟Demo_14-3.ipynb
- 執行關鍵字詞擷取服務操作
- 執行實體辨識服務操作
- 執行文本翻譯服務操作

- 了解如何使用關鍵字詞擷取服務
- 了解如何使用執行實體辨識服務
- 了解如何使用文本翻譯服務



- Lab01: 關鍵字詞擷取服務操作
- Lab02: 執行實體辨識服務操作
- Lab03: 建立文本翻譯服務並執行服務操作

Estimated time:
20 minutes



Module 15

語言服務應用2

學習目標：

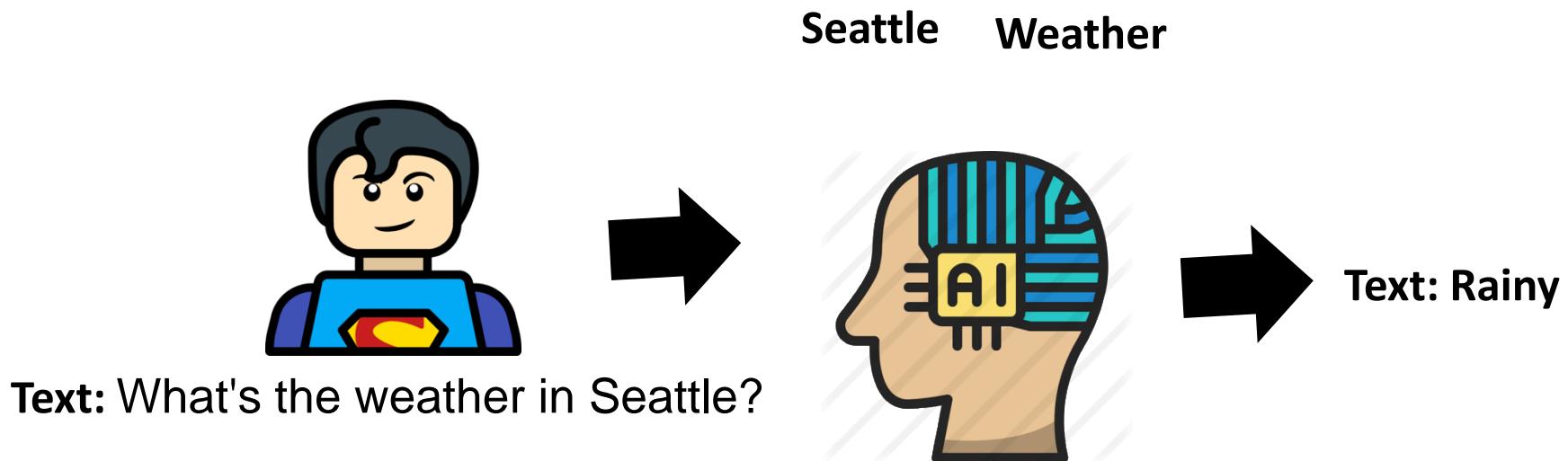
- 15-1:LUIS意圖辨識
介紹
- 15-2:設定與使用
LUIS意圖辨識
- 15-3:設定與使用
LUIS意圖辨識 2

15-1:LUIS語言意圖 辨識介紹

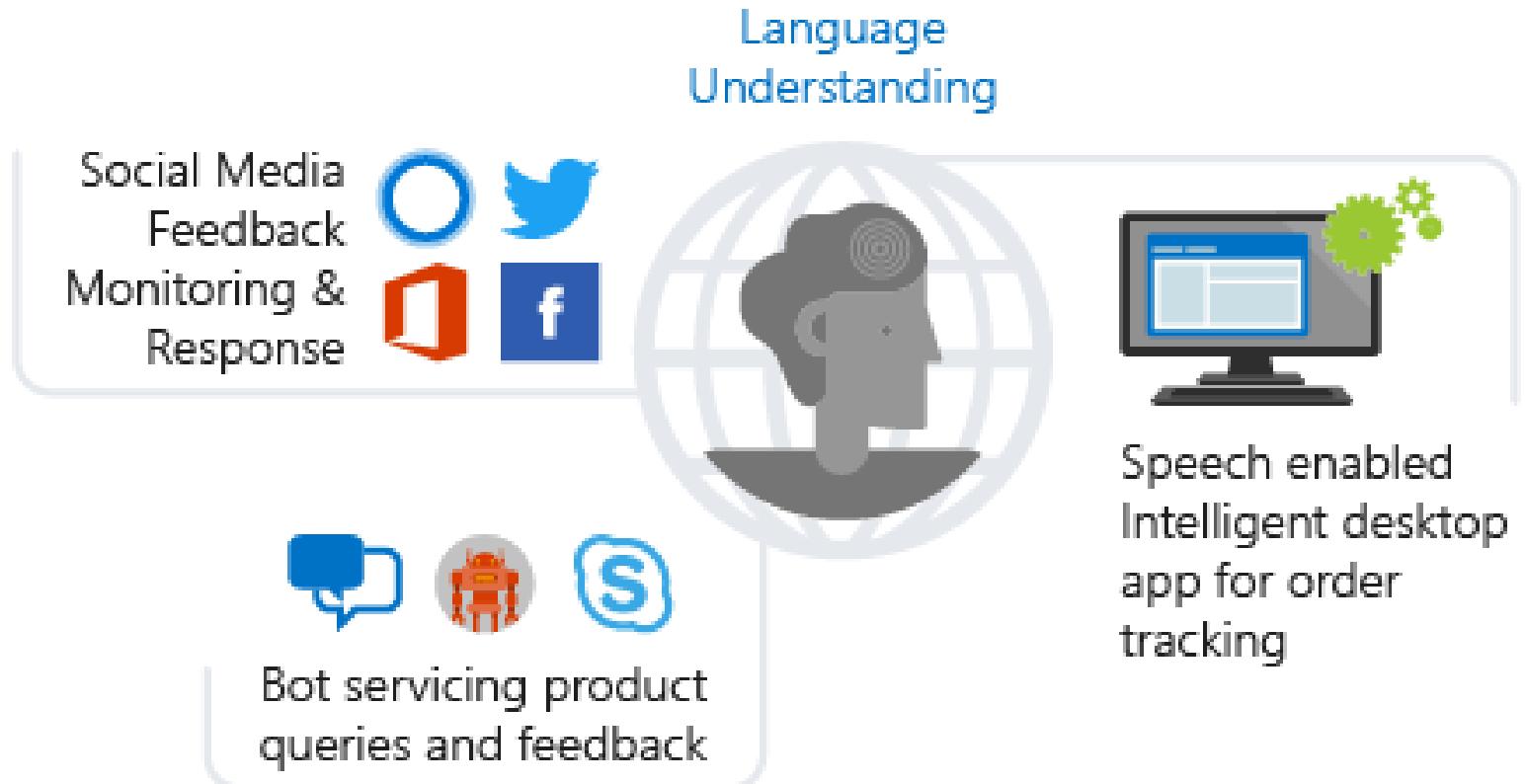
- 認識意圖辨識
- 認識LUIS服務
- 認識LUIS語言意圖
辨識服務

- 意圖辨識

- 意圖是使用者想要做的事情：訂機票、查看天氣或打電話。
- 意圖辨識就是讓AI分析自然語言並從知識庫中找出相對應語句回答



- LUIS (Language Understanding Intelligent Service)
 - 以機器學習為基礎的雲端API服務，將自然語言應用到應用程式來預測整體意義，並找出相關的詳細資訊。



- 意圖辨識與LUIS

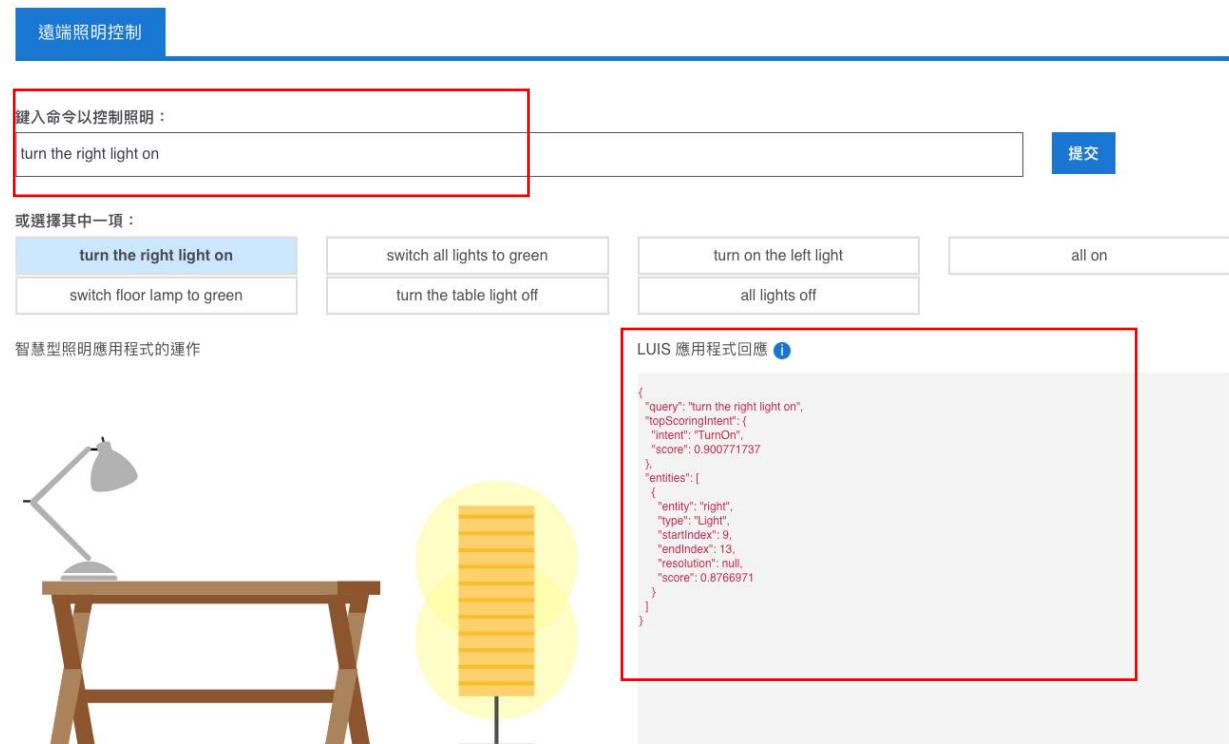
- 認知服務語言 SDK 與 Language Understanding Intelligent Service (LUIS) 整合以提供意圖辨識。
- 使用者可以使用任何他們覺得自然的字詞。然後透過機器學習，LUIS 會將使用者要求對應至已定義的意圖。

The screenshot illustrates the interaction between a user and a LUIS-based service. On the left, under 'What the user says (utterances)', a blue button displays the text 'Order me 2 pizzas'. On the right, under 'What LUIS returns', a JSON object represents the service's response:

```
{  
  "query": "Order me 2 pizzas",  
  "prediction": {  
    "topIntent": "FoodOrder",  
    "intents": {  
      "FoodOrder": {  
        "score": 0.9999981  
      },  
      "None": {  
        "score": 0.0604290478  
      },  
      "LocationFinder": {  
        "score": 0.0201619361  
      }  
    }  
}
```

認識LUIS語言意圖辨識服務

- 將自然語言內容對應到事先定義的意圖（intent）上。
- 教會AI讀懂人類語言
- 本章節將介紹語言應用。
 - 輸入文字觀察LUIS回傳的內容。
 - 範例網址：
<https://azure.microsoft.com/zh-tw/services/cognitive-services/language-understanding-intelligent-service/>



- LUIS利用三個層面來了解自然語言：
 - 表達 (utterance)
 - 使用者輸入應用程式的文字內容
 - 意圖 (intent)
 - 使用者想要執行的工作或是動作
 - 使用者表達中傳達的目的或是目標
 - 實體 (entity)
 - 開發人員想要在表達中擷取的單字或是片語
- 接下來將說明如何設定與使用LUIS語言意圖辨識服務

15-2:設定與使用 LUIS意圖辨識

學習目標：

- 設定LUIS意圖辨識服務
- 使用LUIS意圖辨識服務

- 步驟 1：在 [Azure 入口網站](#) 中建立 LUIS 資源。
- 步驟 2：使用 Python 從用戶端以 REST API 方式呼叫官方 demo 用應用程式的端點。應用程式會處理 JSON 回應，以向使用者顯示最佳答案。

設定LUIS意圖辨識服務

- 登入Azure
- 點選以下網址來建立LUIS資源
 - <https://ms.portal.azure.com/#create/Microsoft.CognitiveServicesLUISAllInOne>
 - 填寫相關資料

Basics * Tags Review + create

Language Understanding (LUIS) is a natural language processing service that enables you to understand human language in your own application, website, chatbot, IoT device, and more. After you configure and publish your LUIS model, your application can easily receive user input in natural language and take action. You don't need to understand machine learning to solve the problem of extracting meaning from input. Instead you get to focus on your own application logic and let LUIS do the heavy lifting on your behalf. After your LUIS model is built and deployed, it exports a simple HTTP endpoint that is called by your application. [Learn more](#)

建立選項 兩者 製作 Prediction

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

訂用帳戶 * 免費試用版

資源群組 * CogAI 新建

名稱 * luisAPI

Authoring Resource

Select pricing and location for Authoring Resource

製作位置 * (US) 美國西部

製作定價層 * F0 (5 Calls per second, 1M Calls per month)

Prediction Resource

Select pricing and location for Prediction Resource

Prediction location * (US) 美國西部

Prediction pricing tier * F0 (5 每秒呼叫次數, 10K 每月呼叫次數)

Review + create Next : Tags >

- 資料填寫完點選下一步直到建立

The screenshot shows the Microsoft Cognitive Services LUIS AllInOne deployment overview page. The title bar reads "首頁 > Microsoft.CognitiveServicesLUISAllInOne | 概觀". The main heading is "Microsoft.CognitiveServicesLUISAllInOne | 概觀". On the left, there is a sidebar with navigation links: "概觀" (selected), "輸入", "輸出", and "範本". The main content area displays a success message: "您的部署已完成". Below it, deployment details are listed: "部署名稱: Microsoft.CognitiveServicesLUISAllInOne", "訂用帳戶: 免費試用版", "資源群組: CogAI", "開始時間: 31/3/2020", and "相互關聯識別碼: 73bc...". There are two expandable sections: "部署詳細資料 (下載)" and "後續步驟". At the bottom, there are two buttons: "前往資源" (blue) and "Go to prediction resource" (white).

- 點選資源找到此服務的API與Endpoint位置

The screenshot shows the Microsoft Azure portal interface for the LUIS API-Authoring service. The left sidebar lists various resources, with '快速入門' (Quickstart) selected. The main content area displays three numbered steps for setting up a LUIS application:

- 1** **概觀**: Explains that LUIS authoring keys are used to build applications. It shows a placeholder for 'Key1' and the endpoint URL <https://luisapi-authoring.cognitiveservices.azure.com/>.
- 2** **Create a LUIS application**: Instructs to navigate to the LUIS portal to begin building applications. It notes that each authoring region has its own portal.
- 3** **Documentation**: Provides links to the Language Understanding Portal, Europe, and Australia, along with instructions to learn more about getting started with LUIS.

- 開啟Python應用程式
 - 安裝程式庫
 - 將API放進程式裡
 - 觀察回覆的JSON

```
import requests

try:

    key = 'YOUR-KEY' # your Runtime key
    endpoint = 'westus.api.cognitive.microsoft.com'
    appId = 'df67dcdb-c37d-46af-88e1-8b97951ca1c2'
    utterance = 'turn on all lights'

    headers = {
    }

    params ={
        'query': utterance,
        'timezoneOffset': '0',
        'verbose': 'true',
        'show-all-intents': 'true',
        'spellCheck': 'false',
        'staging': 'false',
        'subscription-key': key
    }

    r = requests.get(f'https://'{endpoint}'/luis/prediction/v3.0/apps/{appId}/slots/production/predict',\
                     headers=headers, params=params)
    print(r.json())

except Exception as e:
    print(f'{e}')
```

15-3:設定與使用 LUIS意圖辨識 2

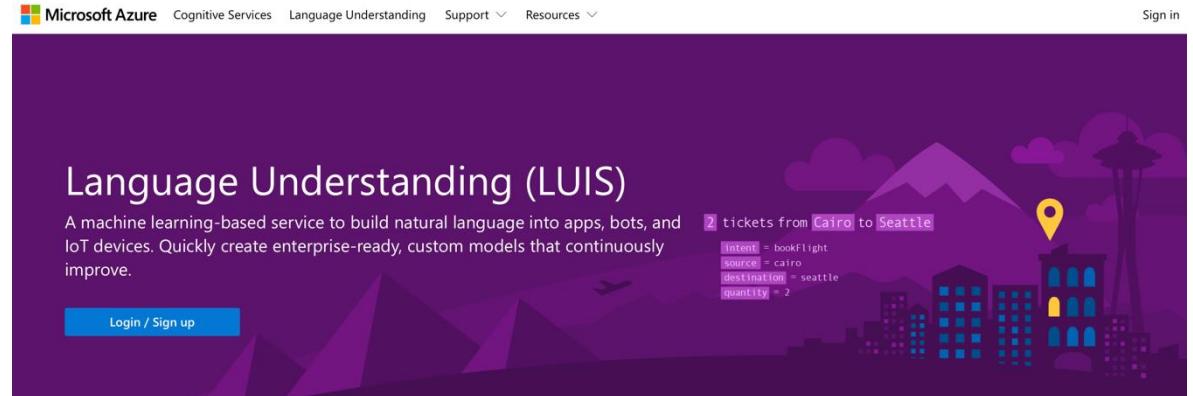
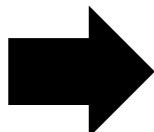
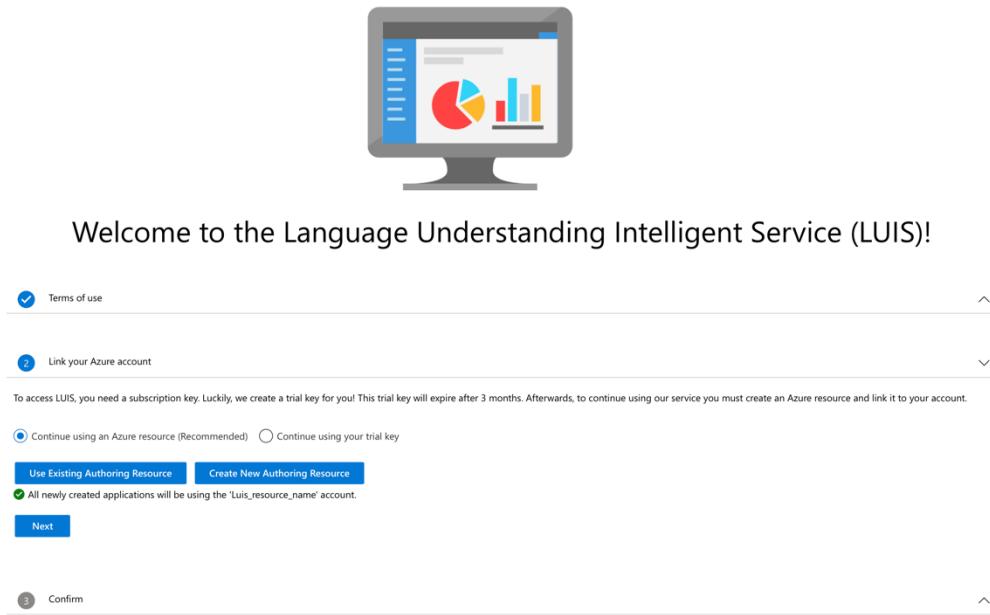
學習目標：

- 設定LUIS意圖辨識服務
- 使用LUIS意圖辨識服務

- 步驟 1：參考上一節在 [Azure 入口網站](#)中建立 LUIS資源。
- 步驟 2：在 LUIS入口網站建立應用程式並自行建立意圖
- 步驟 3：在 LUIS入口網站發佈應用程式
- 步驟 3：使用Python從用戶端以程式設計方式呼叫您知識庫的端點。用戶端應用程式會處理 JSON 回應，以向使用者顯示最佳答案。

設定LUIS意圖辨識服務

- 登入與新增LUIS服務
 - 以Azure帳號登入LUIS入口網站
 - <https://preview.luis.ai>
 - 一步一步填寫相關訊息



This is a modal dialog titled "Create new Azure Cognitive Services account". It contains four input fields with required fields marked with asterisks (*):

- "Tenant Name" field with placeholder "預設目錄".
- "Resource Name" field with placeholder "Luis_resource_name".
- "Subscription Name" field with placeholder "免費試用版".
- "Azure Resource Group Name" field with placeholder "CogAI".

At the bottom of the dialog are "Done" and "Cancel" buttons.

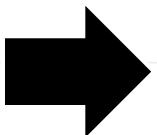
設定LUIS意圖辨識服務

- 資料填寫完畢後點選My apps
- 點選+ New app for...
- 並將新的app名稱命為Home Automation

My Apps •

+ New app for conversation ▾

App name ↑



Create new app

Name *
Home Automation

Culture *
English

** Culture is the language that your app understands and speaks, not the interface language.

Description
Type app description ...

Prediction resource
Choose prediction resource ...

** Only resources allowed for use in this region are displayed.

Done Cancel

Cognitive Services Language Understanding My apps

Welcome to Microsoft's Language Understanding (LUIS)

LUIS enables you to integrate natural language understanding into your chat-bot or other application without having to create the complex part of machine learning models. Instead, you get to focus on your own application's logic and let LUIS do the heavy lifting.

A typical LUIS app goes through the following 3 steps:

Search apps ...

Endpoint hits

{ }

設定LUIS意圖辨識服務

- 於Predefine Domains找一個將Home Automation卡片加進應用程式
- 點選Intent可以看到這個應用中常見的意圖
- 點選意圖 “TurnOff” 可以看到這個意圖涵括的語句有哪些

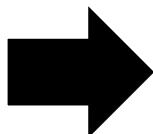
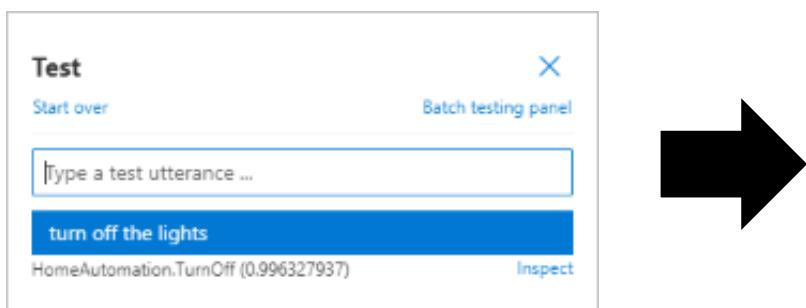
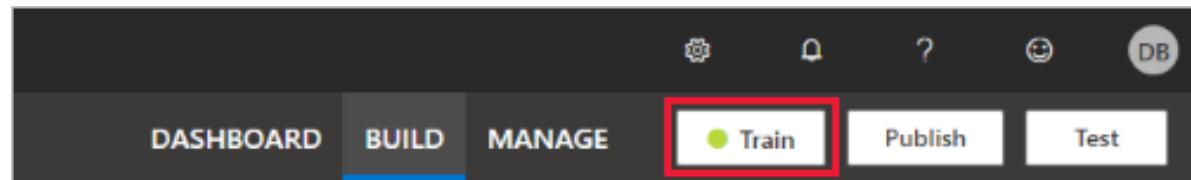
The image shows three screenshots of the Microsoft LUIS service interface illustrating the steps to set up an intent.

- Screenshot 1:** Shows the "Prebuilt Domains" section under the "App Assets" tab. A search bar at the top contains "HomeAutomation". Below it, a card for "HomeAutomation" is shown with the description: "Controlling smart home devices like lights and appliances." A large black arrow points from this screen to the next one.
- Screenshot 2:** Shows the "Intents" section under the "App Assets" tab. It lists several intents: HomeAutomation.TurnUp, HomeAutomation.TurnOn, HomeAutomation.TurnOff, HomeAutomation.TurnDown, HomeAutomation.SetDevice, and HomeAutomation.QueryState. Each intent has a count of utterances (e.g., 164 for TurnOn). A large black arrow points from this screen to the third one.
- Screenshot 3:** Shows the "HomeAutomation.TurnOff" intent details page. The left sidebar shows "Intents" selected. The main area displays example utterances such as "switch off the light", "turn off great light", "turn off living room lamp 1", and "are you able to turn off my light switches". Each utterance is associated with a score (e.g., -1.000).

設定LUIS意圖辨識服務

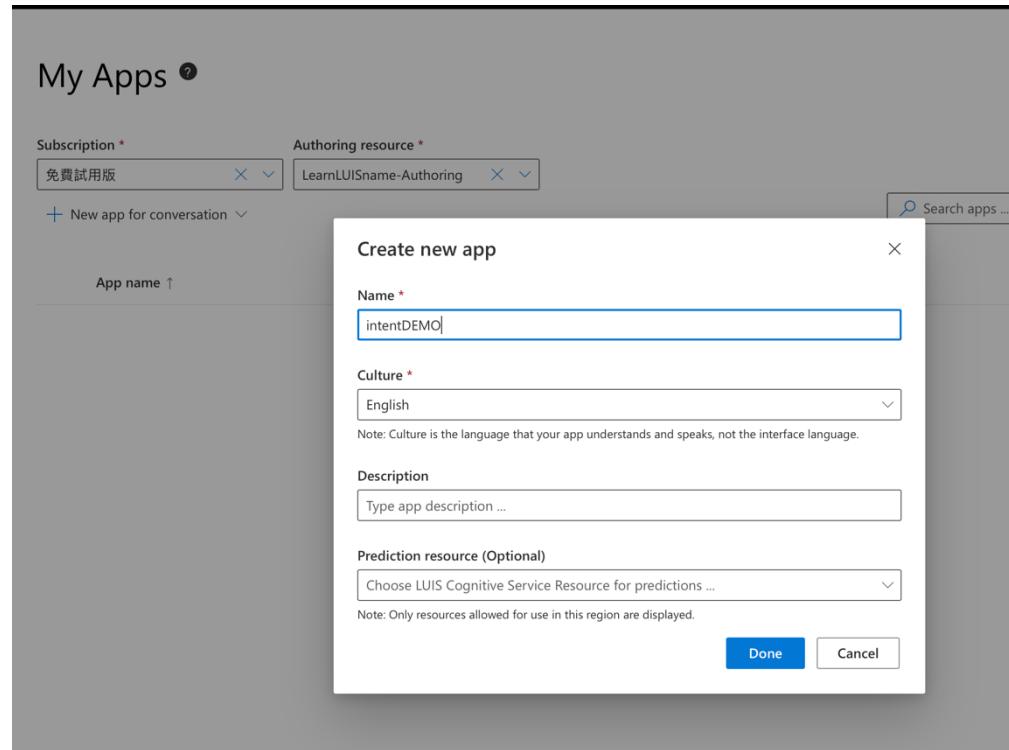
- 意圖測試

- 點選Train，使應用程式可以被測試
- 點選Test，輸入語句 “turn off the lights”
- 在右側可以看到意圖預測後的信賴分數



A screenshot of the LUIS Test results page. At the top, it says "Current: v(0.1)" and "Compare with published". The main area shows the input "turn off the lights" and the predicted intent "HomeAutomation.TurnOff (0.9979643)" with an "Inspect" link. Below this, sections include "Top scoring intent" (HomeAutomation.TurnOff (0.9979643)), "N-Depth Entities" (No predicted n-depth entities), "Composite Entities" (No predicted composite entities), "Other entities" (Text: "lights", Entity: HomeAutomation.DeviceT...), and "Top matched pattern(s)" (No matched patterns). At the bottom, there is a "Sentiment" section with the text "Enable sentiment analysis to get sentiment score".

- 製作自己的app
 - 點選My Apps，選擇Subscription與Authoring Resource後點選+
 - 填寫相關資訊，Culture一定要選與Azure上LUIS服務相同的地區
 - 點選Done



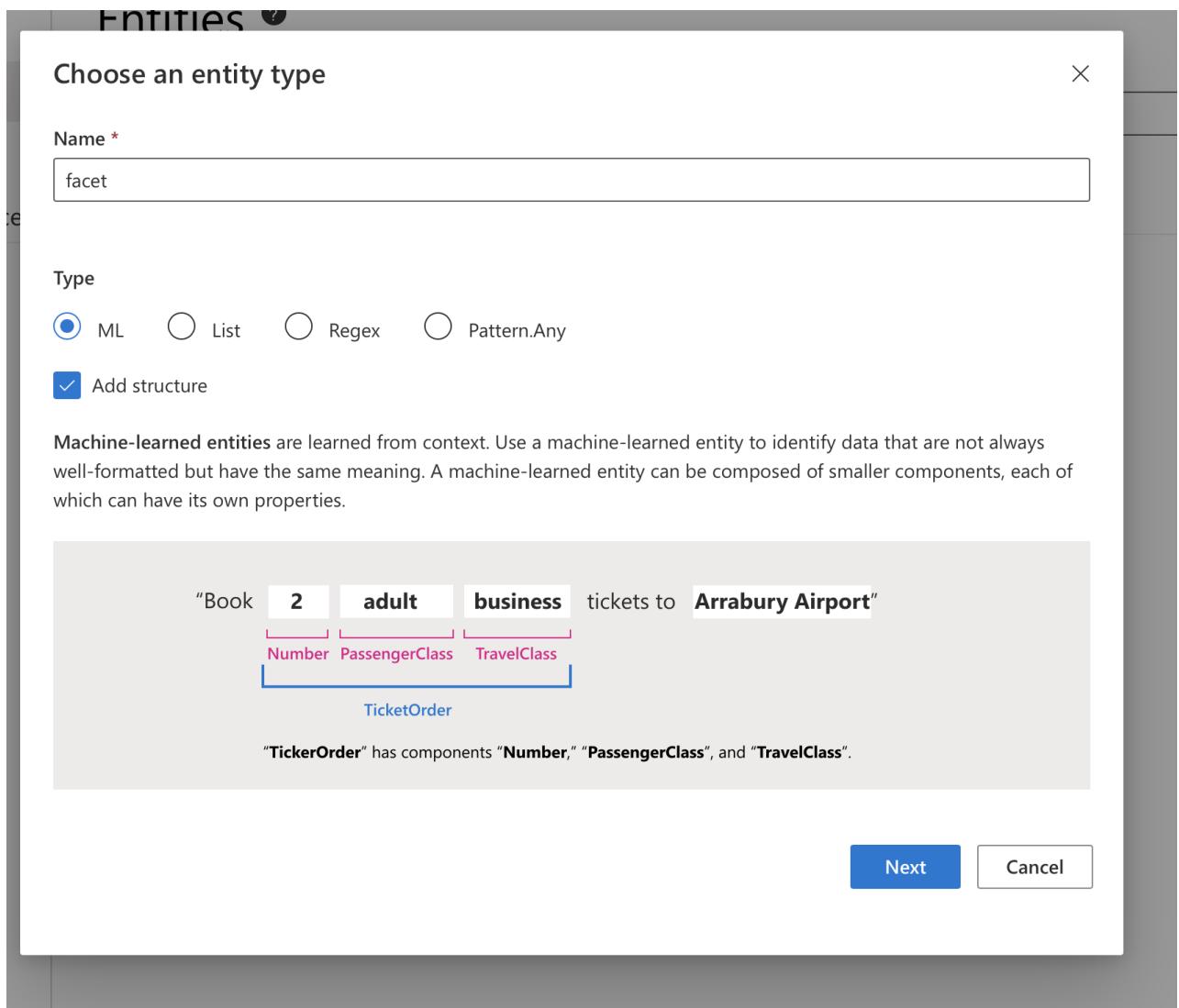
- 新增Intent

- 點選Intents，新增一個叫做SearchPics的意圖
- 輸入表達範例

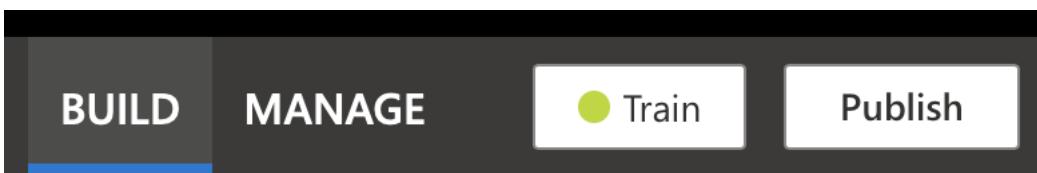
The screenshot shows the Microsoft LUIS web interface. At the top, there's a navigation bar with tabs: DASHBOARD, BUILD (which is highlighted in blue), MANAGE, Train (with a red dot), Publish, and Test. The main area has a dark header with the text "intentDEMO (V0.1) ▾". On the left, there's a sidebar with sections: App Assets (Intents, Entities, Prebuilt Domains), Improve app performance (Review endpoint utterances, Descriptors, Patterns). The main content area is titled "SearchPics" with a pencil icon. It shows a "Descriptors" section with a question mark icon and a plus sign, and a search bar with the placeholder "Search for utterance(s)". Below that is an "Example Utterance" section with a text input field containing "Enter an example of what a user might say and hit Enter.". Underneath, two examples are listed: "show me baby pictures" and "search food pictures".

設定LUIS意圖辨識服務

- 設定Entity
 - 新增一個叫做facet的Entity



- 將Intents與Entity連結
 - 回到SearchPics
 - 點選表達中的重點字如Baby，將他設定到Entity facet中
- 點選右上角Train訓練模型



The screenshot shows the LUIS Intents page for the 'SearchPics' app. The left sidebar lists 'App Assets' sections: 'Intents' (selected), 'Entities', 'Prebuilt Domains', 'Improve app performance' (with 'Review endpoint utterances'), 'Descriptors', and 'Patterns'. The main area shows the 'SearchPics' intent with its 'Descriptors' section. A search bar at the top right says 'Search for utterance(s)'. Below it, an 'Example Utterance' field contains 'Enter an example of what a user might say and hit Enter.' Three examples are listed:

- show me **baby** pictures
 - Search for or create an entity
 - ✓ facet (ML)
 - search: View in entity palette
- any train pictures
- search outdoor pictures

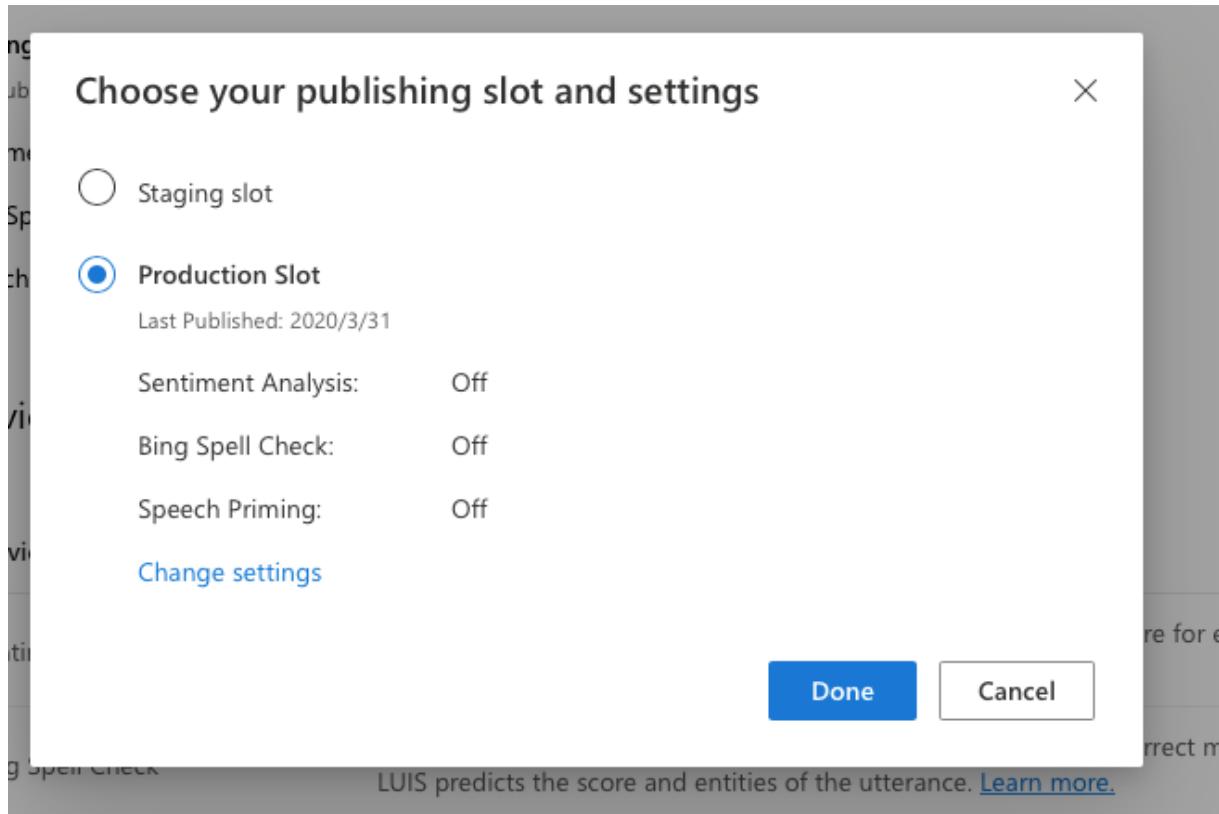
- 點選Manage，把LUIS與Azure服務連接

- 點選Azure Resources
- 點選add prediction resource
- 填寫資料

The screenshot shows the LUIS Azure Resources settings page. At the top, there are tabs for 'Prediction Resources' (which is selected) and 'Authoring Resource'. Below these tabs, there is a button labeled 'Add prediction resource'. A modal dialog box titled 'Add a prediction resource' is open. Inside the dialog, there is a message about creating an authoring resource, a link to learn more about Language Understanding resources in Azure, and several input fields: 'Tenant Name' (set to '預設目錄'), 'Subscription Name' (set to '免費試用版'), and 'LUIS resource name' (set to 'LearnLuisName'). At the bottom of the dialog are 'Done' and 'Cancel' buttons. In the background, the main page shows 'intentDEMO (V0.1)' under Application Settings, and tabs for Publish Settings, Azure Resources (which is selected), and Versions.

設定LUIS意圖辨識服務

- 點選Publish
 - 選擇Staging或是Production（記下來，後面需要寫入到程式碼中）



- 完成安裝程式庫
 - pip install azure-cognitiveservices-language-luis
- 設定環境變數
 - 開啟Windows命令提示字元輸入以下指令
 - setx LUIS_AUTHORING_KEY <replace-with-your-luis-authoring-key>
 - setx LUIS_REGION <replace-with-your-luis-region>
 - Key與Region先到LUIS控制台點選Manage查找
 - 完成後重啟電腦中會使用到的相關服務

使用LUIS意圖辨識服務

- 於Python應用程式中載入相關套件
- 於Python應用程式中將LUIS上的API key、Endpoint、AppId寫入程式裡

```
key_var_name = 'LUIS_RUNTIME_KEY'
if not key_var_name in os.environ:
    raise Exception('Please set/export the environment variable: {}'.format(key_var_name))
runtime_key = os.environ[key_var_name]
print("runtime_key: {}".format(runtime_key))

endpoint_var_name = 'LUIS_RUNTIME_ENDPOINT'
if not endpoint_var_name in os.environ:
    raise Exception('Please set/export the environment variable: {}'.format(endpoint_var_name))
runtime_endpoint = os.environ[endpoint_var_name]
print("runtime_endpoint: {}".format(runtime_endpoint))

# Use public app ID or replace with your own trained and published app's ID
# to query your own app
# public appID = 'df67dcdb-c37d-46af-88e1-8b97951ca1c2'
appID_var_name = 'LUIS_APP_ID'
os.environ[appID_var_name] = appID_var_name
if not appID_var_name in os.environ:
    raise Exception('Please set/export the environment variable: {}'.format(appID_var_name))
luisAppID = os.environ[appID_var_name]
print("luisAppID: {}".format(luisAppID))

# production or staging
slot_var_name = 'LUIS_APP_SLOT_NAME'
os.environ[slot_var_name] = slot_var_name
if not slot_var_name in os.environ:
    raise Exception('Please set/export the environment variable: {}'.format(slot_var_name))
luisSlotName = os.environ[slot_var_name]
print("luisSlotName: {}".format(luisSlotName))
```

- 範例預測程式碼與執行結果

```
def predict(app_id, slot_name):  
  
    request = { "query" : "hi, show me lovely baby pictures" }  
  
    # Note be sure to specify, using the slot_name parameter, whether your application is in staging or \  
    # production.  
    response = clientRuntime.prediction.get_slot_prediction(app_id=app_id, slot_name=slot_name, \  
                                              prediction_request=request)  
  
    print("Top intent: {}".format(response.prediction.top_intent))  
    print("Sentiment: {}".format (response.prediction.sentiment))  
    print("Intents: ")  
  
    for intent in response.prediction.intents:  
        print("\t{}".format (json.dumps (intent)))  
    print("Entities: {}".format (response.prediction.entities))
```

1 predict(luisAppID, luisSlotName)

Top intent: SearchPics
Sentiment: None
Intents:
 "SearchPics"
Entities: {'facet': ['baby']}

- 開啟Demo_15-3.ipynb
- 建立Azure LUIS意圖辨識服務
- 執行LUIS意圖辨識服務操作

- 認識LUIS意圖辨識服務
- 了解如何使用LUIS意圖辨識服務



- Lab01: 建立Azure LUIS服務
- Lab02: 建立Azure LUIS意圖辨識服務
- Lab03: 執行LUIS意圖辨識服務操作

Estimated time:
20 minutes



Module 16

語言服務應用3

學習目標：

- 16-1:QnA maker服務介紹
- 16-2:內容仲裁服務介紹
- 16-3:語言服務總結

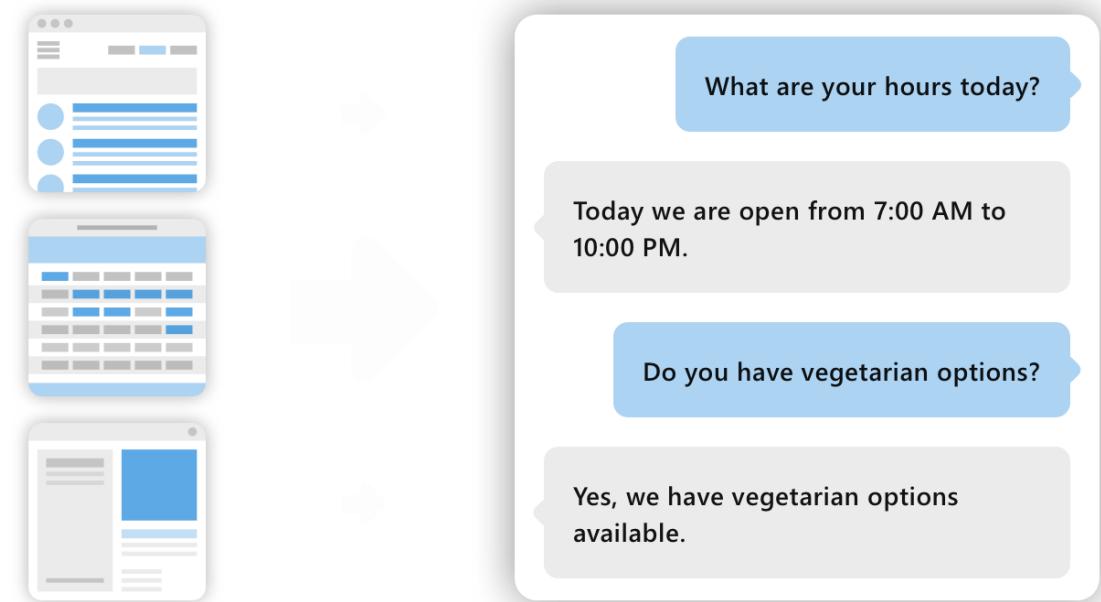
16-1:QnA maker服 務介紹

學習目標：

- 認識QnA maker服
務
- 使用QnA maker服
務

認識QnA maker服務

- QnA maker是一項雲端式 API 服務，
 - 對現有資料建立交談式的問答。
 - 從半結構化內容 (包括常見問題集、手冊和文件) 擷取問答來建立知識庫。
 - 透過 QnAs，以知識庫中的最佳答案自動回答使用者的問題。
 - 知識庫會持續從使用者行為學習。



使用QnA maker服務

- 步驟 1：在 [Azure 入口網站](#)中建立 QnA maker資源。
- 步驟 2：在 QnA maker服務入口網站新增Knowledge Base (KB)
- 步驟 3：修改KB內容並發佈
- 步驟 3：使用Python從用戶端以程式設計方式呼叫您知識庫的端點。用戶端應用程式會處理 JSON 回應，以向使用者顯示最佳答案。

使用QnA maker服務

- 點選以下連結填寫資料後於Azure中建立QnA服務
 - <https://ms.portal.azure.com/#create/Microsoft.CognitiveServicesQnAMaker>

The screenshot shows the Azure portal's 'Create' blade for creating a new QnA Maker service. The page has two main sections: 'QnA Maker' settings on the left and 'App Service' configuration on the right.

QnA Maker Settings:

- 名稱 ***: qamakeDEMO
- 訂用帳戶 ***: 免費試用版
- 定價層 (檢視完整定價詳細資料) ***: F0 (每月管理的文件數, 3 每秒交易數, 100 每分鐘交易數, 50K 每月交易)
- 資源群組 ***: CogAI

App Service Configuration:

- Azure 搜尋服務定價層 (檢視完整定價詳細資料) ***: F (3 Indexes)
- Azure 搜尋服務位置 ***: (US) 美國西部
- 應用程式名稱 ***: qamakeDEMO
- 網站位置 ***: (US) 美國西部
- 應用程式見解**: 啟用
- 應用程式見解位置 ***: (US) 美國西部

At the bottom, there are '建立' (Create) and '自動化選項' (Automation Options) buttons.

使用QnA maker服務

- 至QnA入口網站登入後點選建立KB
 - 一步一步填寫裡面資料後送出

Design sophisticated multi-turn conversations easily with follow-up prompts. [Learn more.](#)

Create a knowledge base

Create an Azure service for your QnA knowledge base and add sources that contain the question and answer pairs you would like to include. [Learn more about creating a knowledge base.](#)

STEP 1

Create a QnA service in Microsoft Azure.
Create an Azure QnA service for your KB. If you already have an Azure QnA service for this KB, skip this step. [Learn more about Azure subscriptions, pricing tiers, and keys.](#)

[Create a QnA service](#)

STEP 2

Connect your QnA service to your KB.
After you create an Azure QnA service, refresh this page and then select your Azure service using the options below

[Refresh](#)

* Microsoft Azure Directory ID

* Azure subscription name

* Azure QnA service

* Language



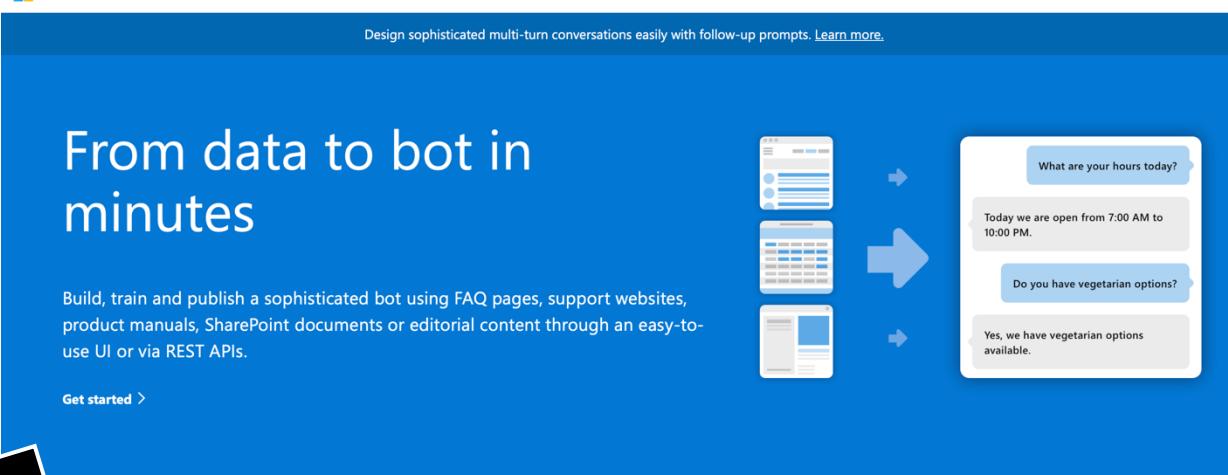
Microsoft Azure | Cognitive Services | QnA Maker | Support | Resources | Sign in

Design sophisticated multi-turn conversations easily with follow-up prompts. [Learn more.](#)

From data to bot in minutes

Build, train and publish a sophisticated bot using FAQ pages, support websites, product manuals, SharePoint documents or editorial content through an easy-to-use UI or via REST APIs.

[Get started >](#)



使用QnA maker服務

- 新增QnA對話 (Add QnA pair)
 - 完成設定後點選右上角Save and train
 - 點選Test測試內容

Knowledge base

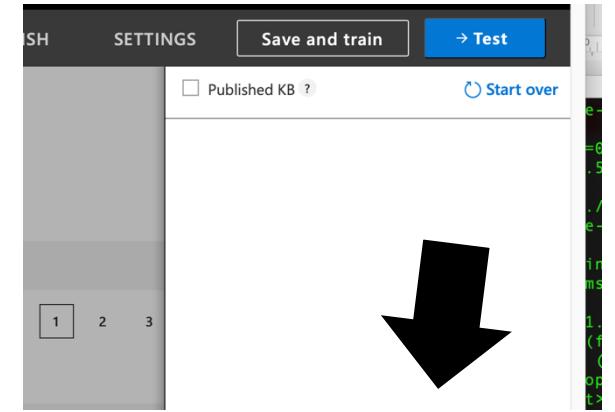
Search the knowledge base 92 QnA pairs

Add QnA pair  View options

| Context | Question | Answer |
|-------------------|---|--|
| Source: Editorial | How many Azure services are used by a knowledge base? | * Azure QnA Maker service\n* Azure Cognitive Search\n* Azure web app\n* Azure app plan |

+ Add alternative phrasing + Add follow-up prompt

Source: https://docs.microsoft.com/azure/cognitive-services/qnamaker/troubleshooting

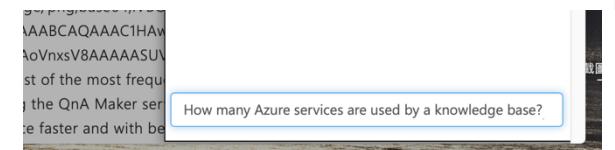


How many Azure services are used by a knowledge base?

Inspect You

- Azure QnA Maker service
- Azure Cognitive Search
- Azure web app
- Azure app plan

testKB (Test) at 6:38 PM



使用QnA maker服務



- 發佈QnA
 - 點選Publish
 - 將紅框處記錄下來

Success! Your service has been deployed. What's next?

You can always find the deployment details in your service's settings.

[Create Bot](#)

[View](#) all your bots on the Azure Portal.

Use the below HTTP request to call your Knowledgebase. [Learn more.](#)

[Postman](#) [Curl](#)

```
POST /knowledgebases/[REDACTED]/generateAnswer
Host: https://qamakedemo.azurewebsites.net/qnamaker
Authorization: EndpointKey [REDACTED]
Content-type: application/json
{"question":<Your question>"}
```

使用QnA maker服務

- 汇入程式庫

```
import http.client, urllib.parse, json, time, sys
```

- 將Host、endpoint key、route改成上一頁的內容並執行

```
# Represents the various elements used to create HTTP request URIs
# for QnA Maker operations.
# From Publish Page
# Example: YOUR-RESOURCE-NAME.azurewebsites.net
# CAUTION: This is not the exact value of HOST field
# HOST trimmed to work with http library
host = "YOUR-RESOURCE-NAME.azurewebsites.net";

# Authorization endpoint key
# From Publish Page
endpoint_key = "YOUR-ENDPOINT-KEY";

# Management APIs append the version to the route
# From Publish Page
# Example: /knowledgebases/ZZZ15f8c-d01b-4698-a2de-85b0dbf3358c/generateAnswer
# CAUTION: This is not the exact value after POST
# Part of HOST is prepended to route to work with http library
route = "/qnamaker/knowledgebases/e7015f8c-d01b-4698-a2de-85b0dbf3358c/generateAnswer";

# JSON format for passing question to service
question = "{'question': 'Is the QnA Maker Service free?', 'top': 3}";
```

```
headers = {
    'Authorization': 'EndpointKey ' + endpoint_key,
    'Content-Type': 'application/json'
}

try:
    conn = http.client.HTTPSConnection(host, port=443)
    conn.request("POST", route, question, headers)
    response = conn.getresponse()
    answer = response.read()
    print(json.dumps(json.loads(answer), indent=4))

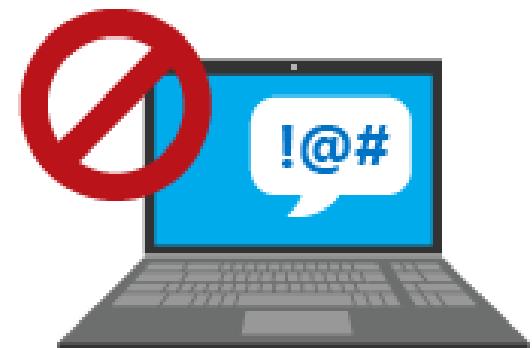
except:
    print("Unexpected error:", sys.exc_info()[0])
    print("Unexpected error:", sys.exc_info()[1])
```

學習目標：

16-2:內容仲裁服務介紹

- 認識內容仲裁服務
- 使用內容仲裁服務

- 偵測可能具冒犯性或不應出現的內容。
- 內容仲裁API
 - 不雅內容
 - 鬼語
 - 個人識別資訊（PII）
 - 電子郵件、地址、IP位址、電話號碼



- 至 Azure 主控台建立內容仲裁資源：
 - 填寫相關資料

首頁 > 新增 > Content Moderator

Content Moderator

Microsoft

Content Moderator

Microsoft

建立

概觀 方案

對影像、文字和影片使用機器輔助的內容仲裁 API 及人工審查內容的影片。使用內建的審查工具取得最佳結果。

實用的連結

深入了解 Content Moderator

文件

API 參考

定價

區域可用性

首頁 > 新增 > Content Moderator > 建立

建立

Content Moderator

名稱 *

訂用帳戶 *

免費試用版

位置 *

(US) 美國西部

定價層 (檢視完整定價詳)

F0 (1 每秒呼叫次數)

資源群組 *

CogAI

新建

首頁 > Microsoft.CognitiveServicesContentModerator | 概觀

Microsoft.CognitiveServicesContentModerator | 概觀

部署

搜尋 (Cmd +/)

概觀

輸入

輸出

範本

刪除 取消 重新部署 重新整理

✓ 您的部署已完成

部署名稱: Microsoft.CognitiveServicesContentModerator
訂用帳戶: 免費試用版
資源群組: CogAI

開始時間: 相互關聯識別

部署詳細資料 (下載)

後續步驟

前往資源

- 安裝程式庫
 - pip install --upgrade azure-cognitiveservices-vision-contentmoderator
- 建立Python應用程式
 - 汇入模組

```
import os.path
from pprint import pprint
import time
from io import BytesIO
from random import random
import uuid

from azure.cognitiveservices.vision.contentmoderator import ContentModeratorClient
import azure.cognitiveservices.vision.contentmoderator.models
from msrest.authentication import CognitiveServicesCredentials
```

- 設定環境變數

```
CONTENTMODERATOR_ENDPOINT = os.environ.get("CONTENT_MODERATOR_ENDPOINT")
subscription_key = os.environ.get("CONTENT_MODERATOR_SUBSCRIPTION_KEY")
```

- 驗證用戶端

```
client = ContentModeratorClient(
    endpoint=CONTENT_MODERATOR_ENDPOINT,
    credentials=CognitiveServicesCredentials(subscription_key)
)
```

- 將以下文字寫入同資料夾下的文字檔

- Is this a garbage email abcdef@abcd.com, phone: 4255550111, IP: 255.255.255.255, 1234 Main Boulevard, Panapolis WA 96555. Crap is the profanity here. Is this information PII? phone 2065550111

使用內容仲裁服務

- 印出內容

```
# Screen the input text: check for profanity,
# do autocorrect text, and check for personally identifying
# information (PII)
with open(os.path.join(TEXT_FOLDER, 'content_moderator_text_moderation.txt'), "rb") as text_fd:
    screen = client.text_moderation.screen_text(
        text_content_type="text/plain",
        text_content=text_fd,
        language="eng",
        autocorrect=True,
        pii=True
    )
#     assert isinstance(screen, Screen)
pprint(screen.as_dict())
```

```
{'auto_corrected_text': 'Is this a garbage email abcdef@abcd.com, phone: '
                           '4255550111, IP: 255.255.255.255, 1234 Main Boulevard, '
                           'Pentapolis WA 96555.\n'
                           'Crap is the profanity here. Is this information PII? '
                           'phone 2065550111',
 'language': 'eng',
 'normalized_text': 'garbage email abcdef@abcd.com, phone: 4255550111, IP: '
                    '255.255.255.255, 1234 Main Boulevard, Panapolis WA '
                    '96555.\n'
                    'Crap profanity . information PII? phone 2065550111',
 'original_text': 'Is this a garbage email abcdef@abcd.com, phone: 4255550111, '
                  'IP: 255.255.255.255, 1234 Main Boulevard, Panapolis WA '
                  '96555.\n'
                  'Crap is the profanity here. Is this information PII? phone '
                  '2065550111',
 'pii': {'address': [{'index': 81,
                      'text': '1234 Main Boulevard, Panapolis WA 96555'}],
         'email': [{'detected': 'abcdef@abcd.com',
                    'index': 24,
                    'sub_type': 'Regular',
                    'text': 'abcdef@abcd.com'}],
         'ipa': [{'index': 64, 'sub_type': 'IPV4', 'text': '255.255.255.255'}],
         'phone': [{'country_code': 'US', 'index': 48, 'text': '4255550111'},
                   {'country_code': 'US', 'index': 181, 'text': '2065550111'}],
         'ssn': []},
 'status': {'code': 3000, 'description': 'OK'},
 'terms': [{'index': 115, 'list_id': 0, 'original_index': 122, 'term': 'crap'}],
```

- 回覆的結果

學習目標：

16-3:語言服務總結

- 語言服務總結
- 語言服務定價

語言服務總結

• 文字分析



We went to Contoso Steakhouse located at midtown NYC last week for a dinner party, and we adore the spot! They provide marvelous food and they have a great menu. The chief cook happens to be the owner (I think his name is John Doe) and he is super nice, coming out of the kitchen and greeted us all. We enjoyed very much dining in the place! The Sirloin steak I ordered was tender and juicy, and the place was impeccably clean. You can even pre-order from their online menu at www.contososteakhouse.com, call 312-555-0176 or send email to order@contososteakhouse.com! The only complaint I have is the food didn't come fast enough. Overall I highly recommend it!

具名實體辨識

| | |
|--------------|---|
| 具名實體: | Contoso Steakhouse [Organization] Steakhouse [Location] midtown NYC [Location] NYC [Location-GPE] last week [DateTime-DateRange] dinner party [Event] chief cook [PersonType] owner [PersonType] John Doe [Person] kitchen [Location] www.contososteakhouse.com [URL] 312-555-0176 [PhoneNumber] email [Skill] order@contososteakhouse.com [Email] |
|--------------|---|

連結實體:

We went to [Contoso Steakhouse](#) located at midtown [NYC](#) last week for a dinner party, and we adore the spot! They provide marvelous food and they have a great menu. The chief cook happens to be the owner (I think his name is [John Doe](#)) and he is super nice, coming out of the kitchen and greeted us all. We enjoyed very much dining in the place! The [Sirloin steak](#) I ordered was tender and juicy, and the place was impeccably clean. You can even pre-order from their online menu at www.contososteakhouse.com, call 312-555-0176 or send email to order@contososteakhouse.com! The only complaint I have is the food didn't come fast enough. Overall I highly recommend it!

語言服務總結

- 文字分析

Text(en): What's the weather in Seattle ?

QnA maker

How many Azure services are used by
a knowledge base?

Inspect You

- Azure QnA Maker service
- Azure Cognitive Search
- Azure web app
- Azure app plan

testKB (Test) at 6:38 PM

Seattle

Weather



Text(de): Wie ist das Wetter in Seattle?

文本翻譯



Text: Rainy

意圖辨識

- 語音轉文字服務定價
 - <https://azure.microsoft.com/zh-tw/pricing/details/cognitive-services/text-analytics/>
- 文本翻譯服務定價
 - 支援的地區與費用請參考以下網址：
<https://azure.microsoft.com/zh-tw/pricing/details/cognitive-services/translator-text-api/>

- LUIS意圖辨識服務定價：
 - 地區與詳細資料細節請參考以下網址
<https://azure.microsoft.com/zh-tw/pricing/details/cognitive-services/language-understanding-intelligent-services/>
- 內容仲裁服務定價：
 - 地區與詳細資料細節請參考以下網址
<https://azure.microsoft.com/zh-tw/pricing/details/cognitive-services/content-moderator/>

- 開啟Demo_16-3.ipynb
- 建立與執行Azure QnA maker服務操作
- 執行內容仲裁服務操作

- 認識Azure QnA maker服務
- 了解如何使用QnA maker服務
- 了解如何使用內容仲裁服務



- Lab01: 建立Azure QnA maker服務
- Lab02: QnA maker服務操作
- Lab03: 內容仲裁服務操作

Estimated time:
20 minutes



Module 17

自訂視覺服務

- 17-1:自訂視覺服務介紹
- 17-2:使用自訂視覺影像分類服務
- 17-3:使用自訂視覺物件偵測服務

17-1:自訂視覺服務介 紹

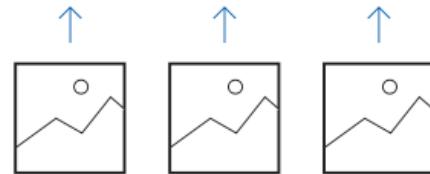
學習目標：

- 認識自訂視覺服務
- 設定自訂視覺服務

- 一般 Azure 認知服務電腦視覺
 - 由機器學習服務支援的服務和 API，讓開發人員將智慧型功能，例如相片和影片中的臉部辨識、文字中的情感分析，以及語言理解，納入到其應用程式。
- Azure 自訂視覺服務
 - 建立特定影像的影像模型，從使用者提供具有標籤的影像「學習」。
 - 例如：
 - 想知道相片中是否有花？
 - 使用大量花卉影像來訓練自訂視覺模型，
 - 接著它便可以說明下一張影像是否有花。

認識自訂視覺服務

- 上傳照片、
- 標記、
- 訓練模型、
- 判斷
- 定價詳細資料請參考以下網址
 - <https://azure.microsoft.com/zh-tw/pricing/details/cognitive-services/custom-vision-service/>



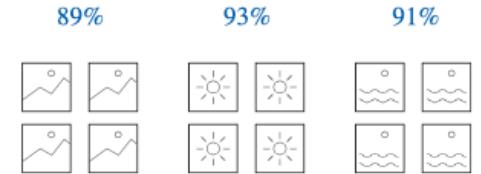
Upload Images

Bring your own labeled images, or use Custom Vision to quickly add tags to any unlabeled images.



Train

Use your labeled images to teach Custom Vision the concepts you care about.



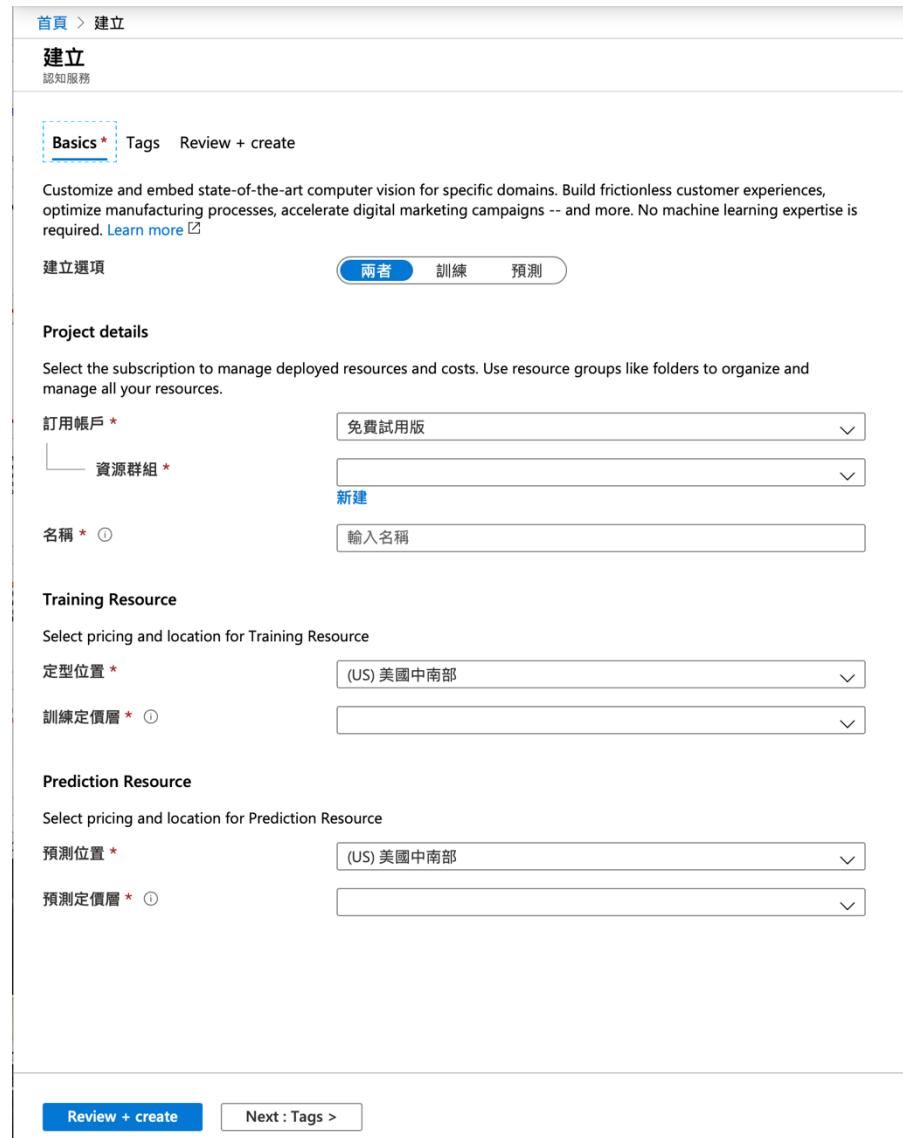
Evaluate

Use simple REST API calls to quickly tag images with your new custom computer vision model.

| 執行個體 | 每秒交易數 (TPS) | 功能 | 價格 |
|------|-------------|--|------------------------------------|
| 免費 | 2 TPS | 上傳、訓練及預測交易 最多 2 個專案 每月最多 1 小時的訓練 | 每專案 5,000 個免費訓練影像 每月 10,000 個預測 |

- 使用Web上傳照片、標記、預測

- 於Azure認知服務建立自訂視覺資源
- https://portal.azure.com/?microsoft_azuremarketplace_ItemHideKey=microsoft_azure_cognitiveservices_customvision#create/Microsoft.CognitiveServicesCustomVision
- 選擇兩者，輸入資料
- 定價與地區請參考以下網址：
<https://azure.microsoft.com/zh-tw/pricing/details/cognitive-services/custom-vision-service/>



首頁 > 建立

建立
認知服務

Basics * Tags Review + create

Customize and embed state-of-the-art computer vision for specific domains. Build frictionless customer experiences, optimize manufacturing processes, accelerate digital marketing campaigns -- and more. No machine learning expertise is required. [Learn more](#)

建立選項 兩者 訓練 預測

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

訂用帳戶 * 免費試用版

資源群組 * 新建 輸入名稱

名稱 * (US) 美國中南部

Training Resource

Select pricing and location for Training Resource

定型位置 * (US) 美國中南部

訓練定價層 * (US) 美國中南部

Prediction Resource

Select pricing and location for Prediction Resource

預測位置 * (US) 美國中南部

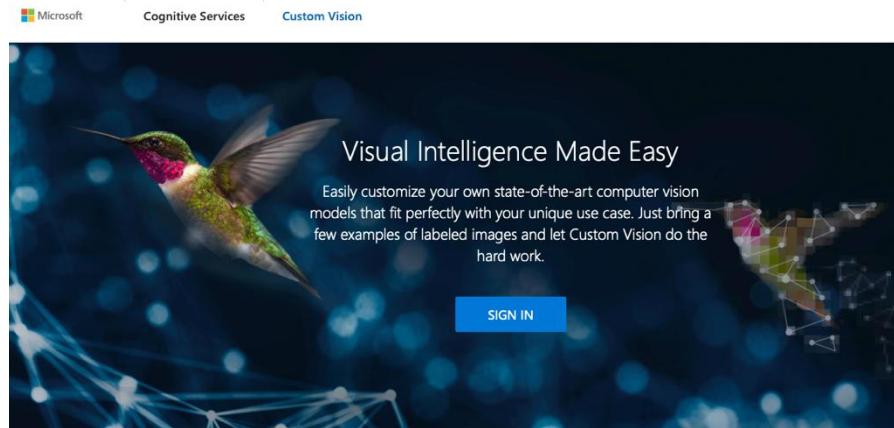
預測定價層 * (US) 美國中南部

Review + create **Next : Tags >**

設定自訂視覺服務

- 於自訂視覺入口以Azure帳號登入

- <https://www.customvision.ai/>



- 新增專案

- 資源群組選擇上一頁設定好的Azure群組
 - 接下來選擇Classification、Multilabel、General
 - 完成設定

Create new project

Name*

CVcat

Description

custom vision categorize

Resource

CVresource [FO]

[Manage Resource Permissions](#)

Project Types ⓘ

- Classification
- Object Detection

Classification Types ⓘ

- Multilabel (Multiple tags per image)
- Multiclass (Single tag per image)

Domains:

- General
- Food

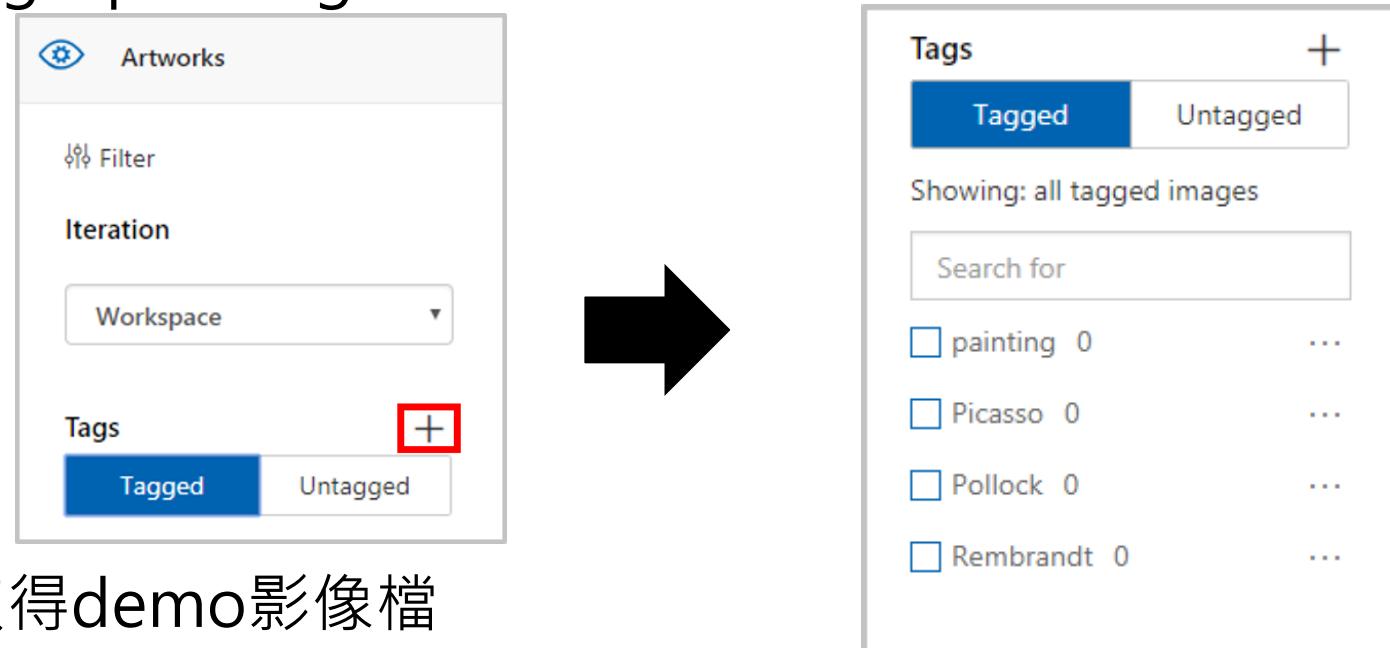
學習目標：

17-2: 使用自訂視覺影像分類服務

- 使用自訂視覺影像分類

使用自訂視覺影像分類

- 登入自訂視覺入口、點選Training Images
 - 新增四個tag：painting、Picasso、Pollock與Rembrandt

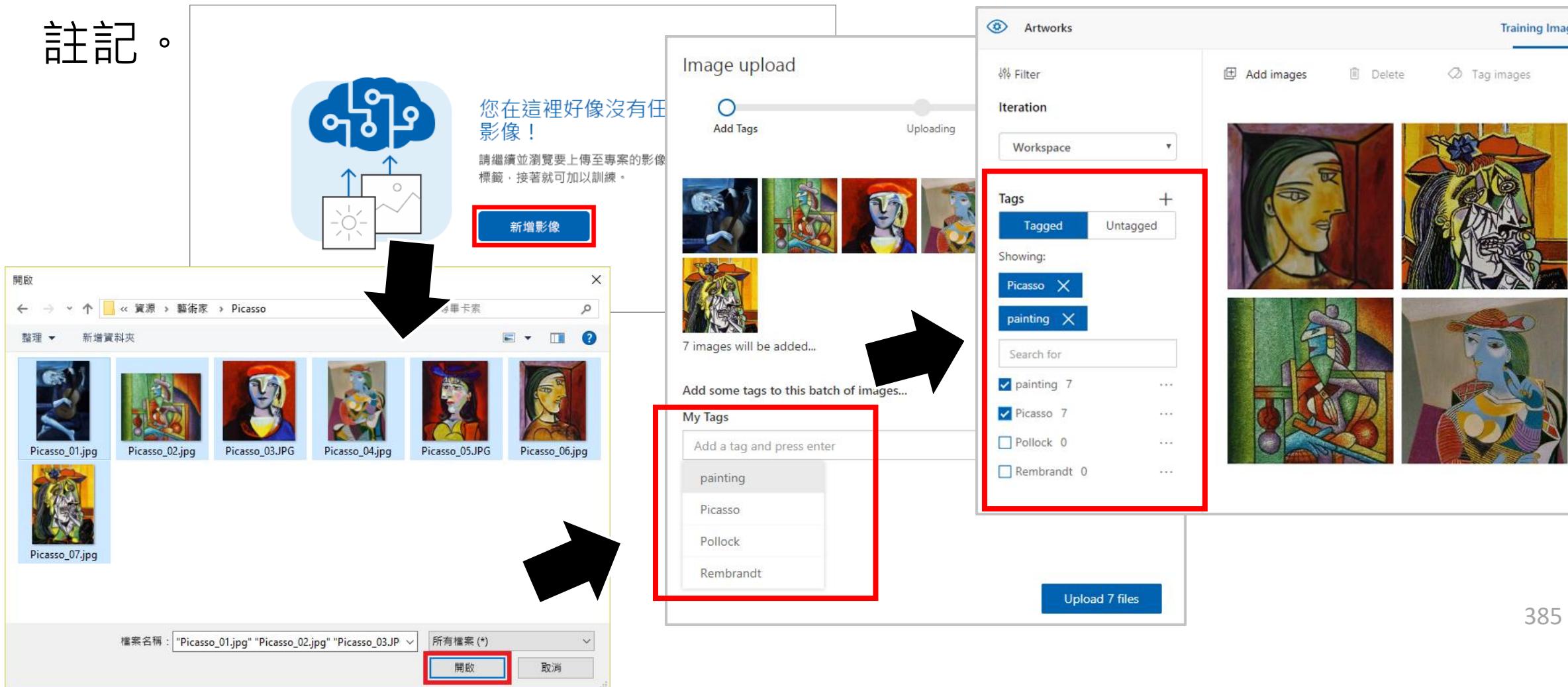


- 下載連結取得demo影像檔

<https://github.com/MicrosoftDocs/mslearn-classify-images-with-the-custom-vision-service/raw/master/cvs-resources.zip>

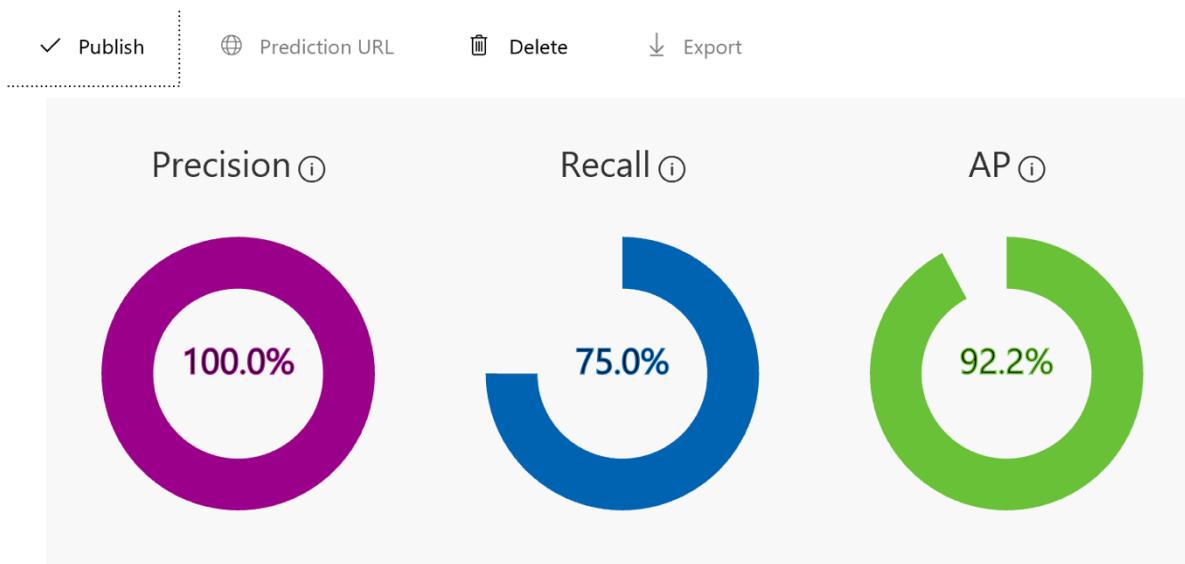
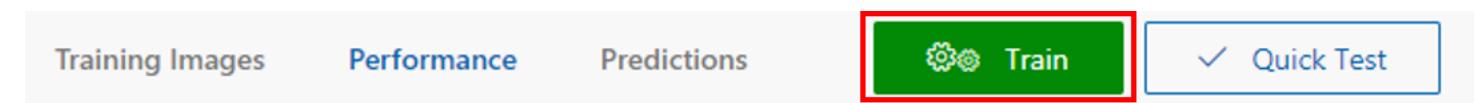
使用自訂視覺影像分類

- 以藝術家名稱不同依次上傳demo影像檔、加上上一頁製作好的註記。



使用自訂視覺影像分類

- 完成三個藝術家相片上傳與註記後
 - 點選訓練模型
 - 選擇iteration次數
 - 完成訓練模型



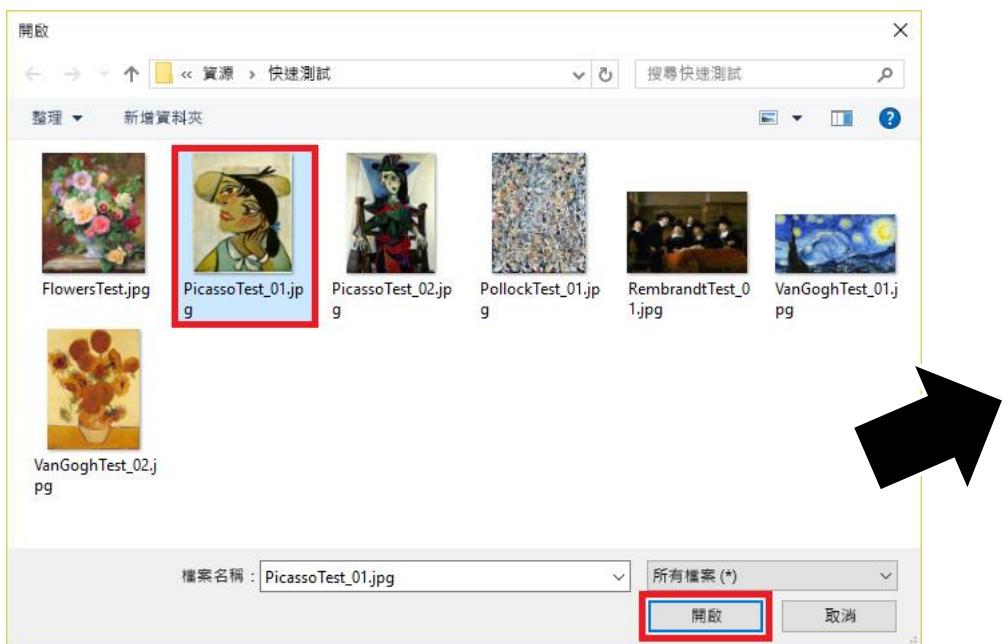
Performance Per Tag

| Tag | Precision | Recall | A.P. | Image count |
|-----------|-----------|--------|--------|-------------|
| Picasso | 100.0% | 100.0% | 100.0% | 7 |
| painting | 100.0% | 100.0% | 100.0% | 19 |
| Rembrandt | 0.0% | 0.0% | 100.0% | 6 |
| Pollock | 0.0% | 0.0% | 25.0% | 6 |

使用自訂視覺影像分類

- 測試模型

- 點選快速測試
- 上傳圖片
- 觀察測試的預測結果



Training Images Performance Predictions Train Quick Test

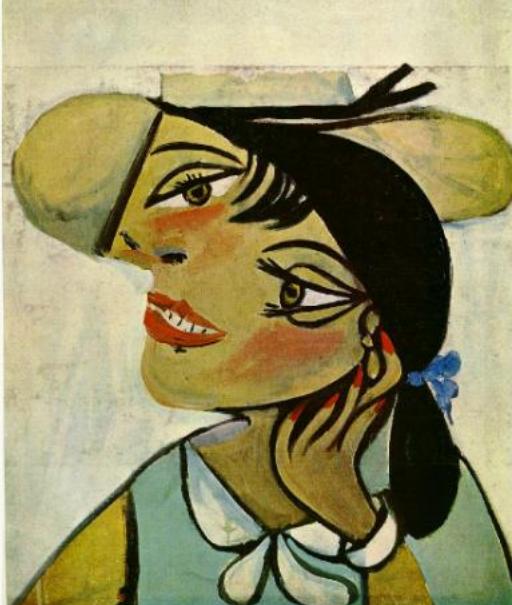
Quick Test

Image URL
Enter Image URL →
or
Browse local files

File formats accepted: jpg, png, bmp
File size should not exceed: 4mb

Predictions

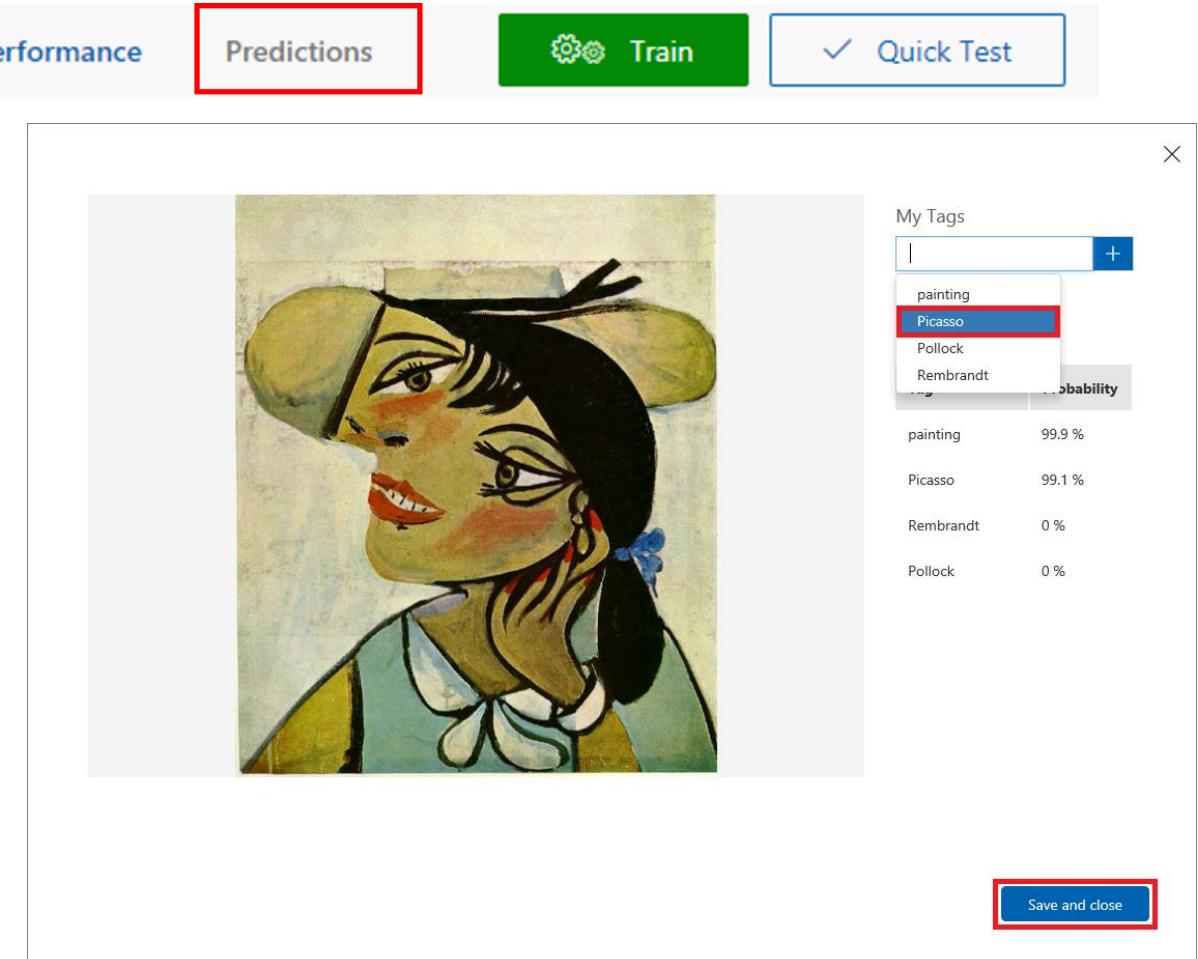
| Tag | Probability |
|-----------|-------------|
| painting | 99.9% |
| Picasso | 99.4% |
| Rembrandt | 1% |
| Pollock | 0% |



使用自訂視覺影像分類

- 修改模型

- 點選預測
- 將剛剛的預測結果標上正確標記，並儲存離開
- 完成測試



17-3:使用自訂視覺物件偵測服務

學習目標：

- 使用自訂視覺物件
偵測

使用自訂視覺物件偵測

- 安裝程式庫
 - pip install azure-cognitiveservices-vision-customvision
- 於自訂視覺網站取得相關訓練與預測的key、Endpoint、id

The screenshot shows the Azure portal interface for managing a Custom Vision resource. On the left, under 'Resources', there is a list item for 'DemoRG' with details: Subscription: Custom Vision Team, Resource Group: DemoRG, Resource Kind: Custom Vision Training. Below this, there are input fields for 'Key' (containing 'Your-Training-Key') and 'Endpoint' (containing 'https://southcentralus.api.cognitive.microsoft.com/'). These two fields are highlighted with a red border. Further down, there is a 'Resource Id' field containing a long URL, a 'Pricing Tier' field set to 'S0', and a green 'Change Pricing Tier' button. On the right, there is a detailed view of a prediction resource named 'DemoRG_prediction'. It shows the same subscription and resource group information. Below this, there are three input fields: 'Key' (containing 'Your-Prediction-Key'), 'Endpoint' (containing 'https://southcentralus.api.cognitive.microsoft.com/'), and 'Resource Id' (containing a long URL). These three fields are also highlighted with a red border. At the bottom right, there is a 'Pricing Tier: S0' section with a 'Change Pricing Tier' button and the word 'Unlimited'.

- 於Python應用程式中載入相關套件與設定key

```
from azure.cognitiveservices.vision.customvision.training import CustomVisionTrainingClient
from azure.cognitiveservices.vision.customvision.training.models import ImageFileCreateEntry, Region

ENDPOINT = "<your API endpoint>

# Replace with a valid key
training_key = "<your training key>"
prediction_key = "<your prediction key>"
prediction_resource_id = "<your prediction resource id>

publish_iteration_name = "detectModel"

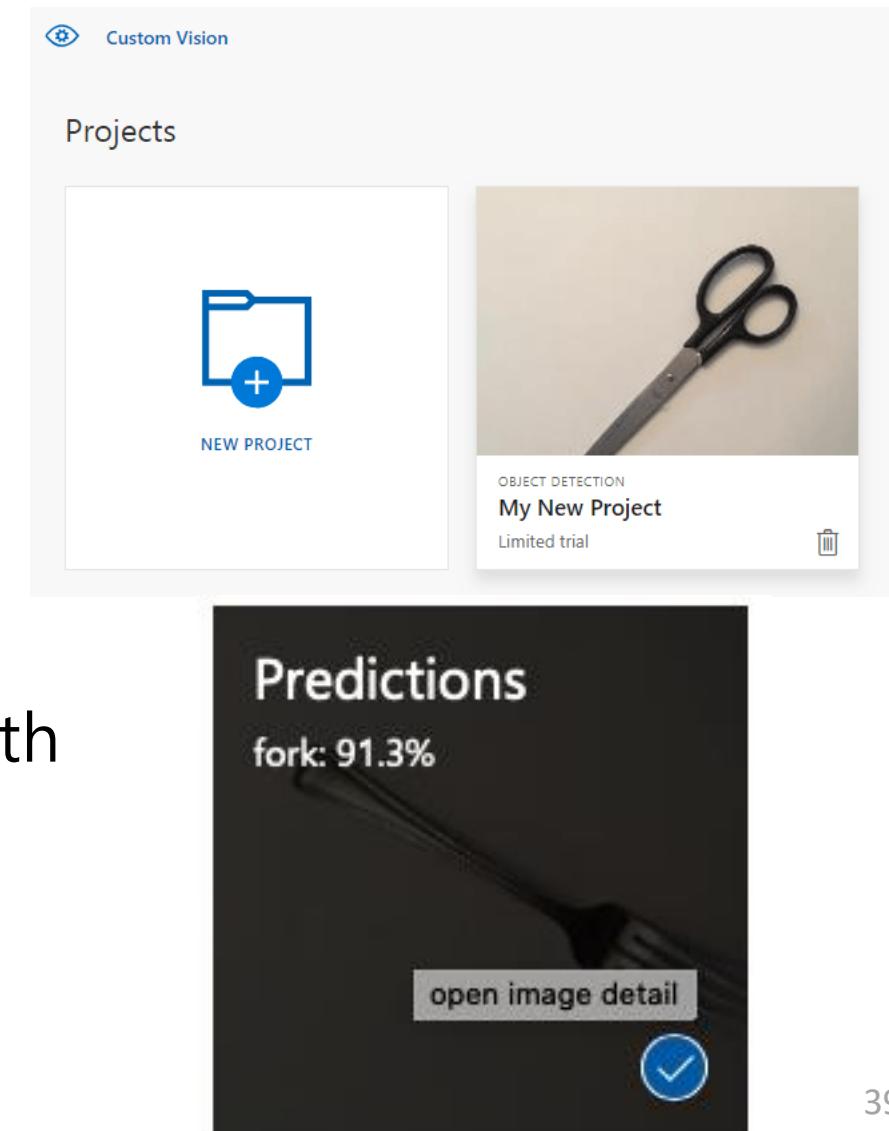
trainer = CustomVisionTrainingClient(training_key, endpoint=ENDPOINT)

# Find the object detection domain
obj_detection_domain = next(domain for domain in trainer.get_domains() if domain.type == "ObjectDetection" \
    and domain.name == "General")

# Create a new project
print ("Creating project...")
project = trainer.create_project("My Detection Project", domain_id=obj_detection_domain.id)
```

使用自訂視覺物件偵測

- 在專案中建立標記(fork, scissors)
- 將訓練圖檔上傳並一一標記影像
- 訓練專案並發布
- 將測試影像傳送至端點並回傳預測結果
 - fork: 91.36%
bbox.left = 0.09, bbox.top = 0.21, bbox.width = 0.64, bbox.height = 0.67
- 程式碼細節請參考Demo



• 定價

- 地區與詳細資料細節請參考以下網址
- <https://azure.microsoft.com/zh-tw/pricing/details/cognitive-services/custom-vision-service/>

選取供應項目：

地區：

貨幣：

自訂視覺服務可讓您輕鬆建置自訂的影像分類器並提升其精準度，以辨識影像中的特定內容。您可以使用最先進的機器學習服務，來訓練分類器辨識對您而言最重要的資訊，像是為產品影像分類，或篩選網站內容。只要上傳已加上標籤的影像即可，自訂視覺服務會接手執行困難的工作。

定價詳細資料

下列定價已反映了預覽版的折扣優惠。

| 執行個體 | 每秒交易數 (TPS) | 功能 | 價格 |
|------|-------------|--|------------------------------------|
| 免費 | 2 TPS | 上傳、訓練及預測交易 最多 2 個專案 每月最多 1 小時的訓練 | 每專案 5,000 個免費訓練影像 每月 10,000 個預測 |
| 標準 | 10 TPS | 上傳及預測交易 最多 100 個專案 | NT\$60.109 每 1,000 筆交易 |
| | | 訓練 | 每計算時數 NT\$601.084 個 |
| | | 影像儲存體 每個最多 6 MB | 每 1,000 個影像 NT\$21.038 個 |

- 開啟Demo_17-3.ipynb
- 執行自訂視覺影像分類服務操作
- 執行自訂視覺物件偵測服務操作

- 認識 Azure 自訂視覺服務
- 了解如何使用自訂視覺影像分類服務
- 了解如何使用自訂視覺物件偵測服務



- Lab01: 建立Azure自訂視覺服務
- Lab02: 自訂視覺影像分類服務操作
- Lab03: 自訂視覺物件偵測服務操作

Estimated time:
20 minutes



Module 18

專題1

學習目標：

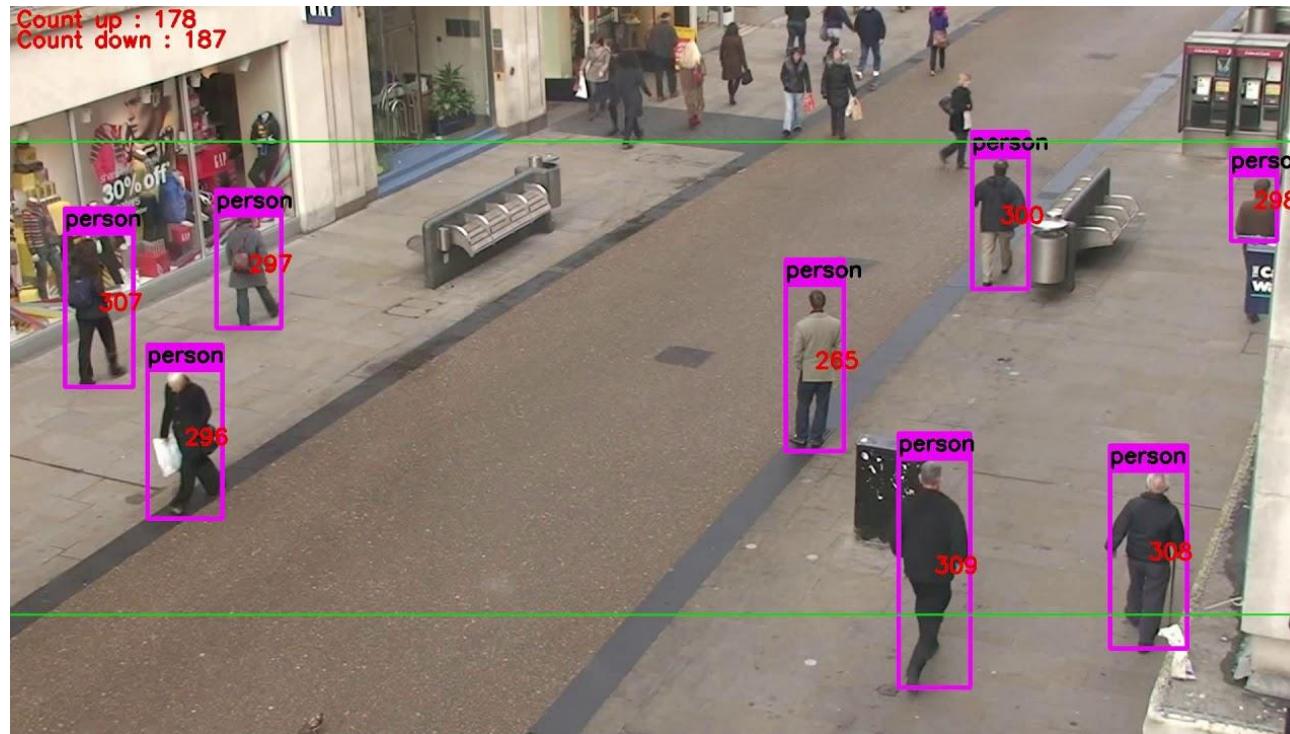
- 18-1:人流偵測
- 18-2:使用Azure做人流偵測
- 18-3:人流偵測 Demo

學習目標：

- 認識人流偵測

18-1: 人流偵測

人流偵測



18-2:使用Azure做人流偵測

學習目標：

- 使用Azure做人流偵測

使用Azure做人流偵測

- 請參考模組6

學習目標：

18-3:人流偵測Demo

- 人流偵測Demo

- <https://www.youtube.com/watch?v=mRHlxgN2TDg>
- <https://www.youtube.com/watch?v=qDK6nLdm3sQ>



Module 19

專題2

學習目標：

- 19-1:人臉辨識
- 19-2:使用Azure做人臉辨識
- 19-3:人臉辨識 Demo

- 認識人臉辨識

19-1: 人臉辨識

人臉辨識



19-2: 使用Azure做人臉辨識

- 使用Azure人臉辨識

使用Azure做人臉辨識

- 請參考模組10

學習目標：

19-3:人臉辨識Demo

- 人臉辨識Demo

- <https://www.youtube.com/watch?v=tf8sfgiUV9k>



Module 20

專題3

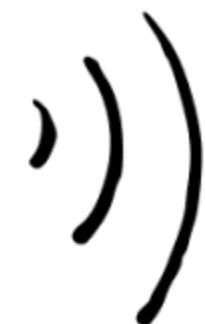
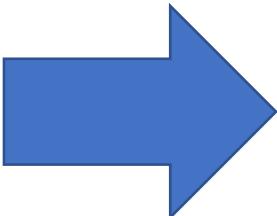
學習目標：

- 20-1:自動閱讀
- 20-2:使用Azure做自動閱讀
- 20-3:自動閱讀 Demo

- 認識自動閱讀

20-1:自動閱讀

自動閱讀



Speak Ai

OCR

TTS

- 使用Azure自動閱讀

20-2:使用Azure自動 閱讀

使用Azure做自動閱讀

- 請參考模組9

學習目標：

- 自動閱讀Demo

20-3:自動閱讀Demo

- <https://www.youtube.com/watch?v=tf8sfgiUV9k>



Module 21

專題4

- 21-1: 影片語音辨識
- 21-2: 使用Azure做
 影片語音辨識
- 21-3: 影片語音辨識
 Demo

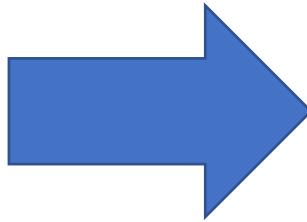
學習目標：

- 認識影片語音辨識

21-1: 影片語音辨識



youtube download



STT

20-2: 使用 Azure 影片 語音辨識

學習目標：

- 使用 Azure 影片 語音
辨識

使用Azure影片語音辨識

- 請參考模組12

學習目標：

- 影片語音辨識Demo

20-3:影片語音辨識 Demo



Module 22

專題5

學習目標：

- 22-1:文章情緒辨識
- 22-2:使用Azure做文章情緒辨識
- 22-3:文章情緒辨識 Demo

學習目標：

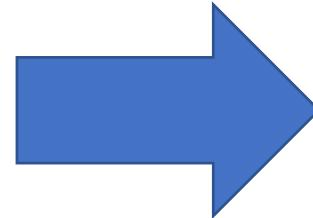
- 認識文章情緒辨識

22-1:文章情緒辨識

文章情緒辨識



article download



emotion detect

20-2: 使用 Azure 文章 情緒辨識

學習目標：

- 使用 Azure 文章情緒
辨識

- 請參考模組14-16

- 文章情緒辨識Demo

20-3:文章情緒辨識 Demo



Module 23

創意專題

學習目標：

- 23-1:創意專題
- 23-2:使用Azure做
創意專題
- 23-3:創意專題
Demo

學習目標：

- 創意專題

23-1:創意專題

- 請同學使用Azure這門課所學到的API去實作創意專題

23-2:使用Azure做創意專題

- 使用Azure做創意專題

使用Azure做創意專題

- 請參考模組4-17

學習目標：

- 創意專題Demo

23-3:創意專題Demo

- 請同學將創意專題作展示，並Demo給大家看！



Module 24

總複習

學習目標：

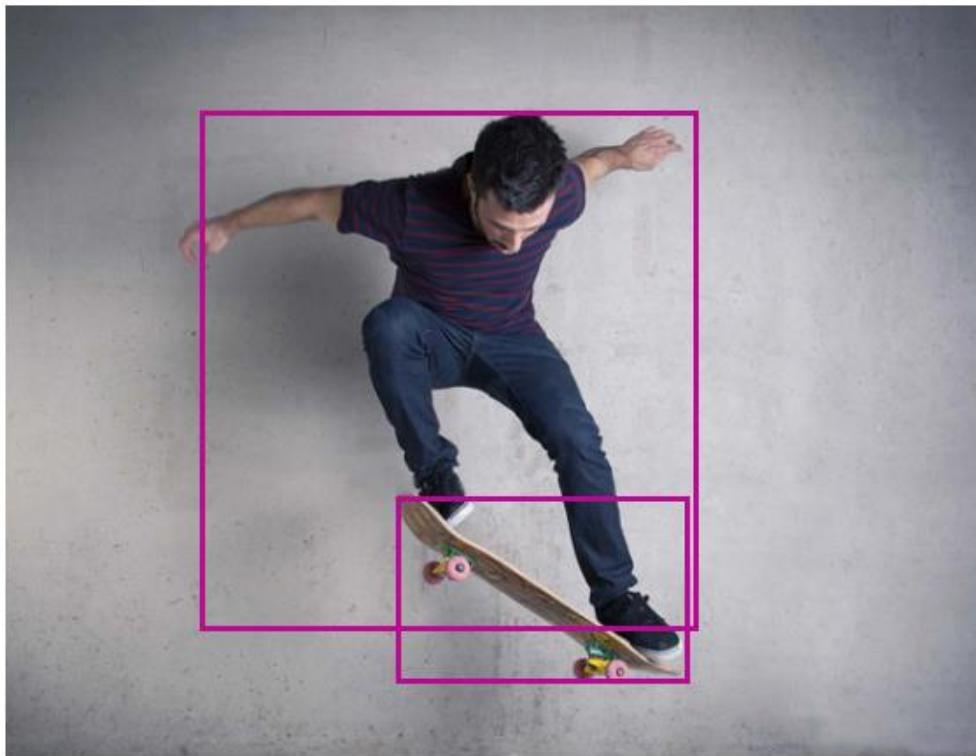
- 24-1:電腦視覺服務應用
- 24-2:臉部辨識
- 24-3:語音及語言服務

24-1:電腦視覺服務應用

學習目標：

- 認識電腦視覺服務應用

- 物件偵測會傳回每個物件（狗、貓或人物）的週框方塊座標（以像素為單位）。



| 功能名稱： | 值 |
|-------|---|
| 物件數 | [{ "rectangle": { "x": 238, "y": 299, "w": 177, "h": 117 }, "object": "Skateboard", "confidence": 0.903 }, { "rectangle": { "x": 118, "y": 63, "w": 305, "h": 321 }, "object": "person", "confidence": 0.955 }] |
| 標籤 | [{ "name": "skating", "confidence": 0.999951541 }, { "name": "snowboarding", "confidence": 0.990067363 }, { "name": "sports equipment", "confidence": 0.9774853 }, { "name": "person", "confidence": 0.9605776 }, { "name": "roller skating", "confidence": 0.945730746 }, { "name": "boardsport", "confidence": 0.9242261 }, { "name": "man", "confidence": 0.9188208 }, { "name": "outdoor", "confidence": 0.9107821 }, { "name": "riding", "confidence": 0.900007248 }, { "name": "skiing", "confidence": 0.894337356 }, { "name": "footwear", "confidence": 0.8788208 }, { "name": "sport", "confidence": 0.86974 }, { "name": "skateboard", "confidence": 0.86974 }] |

- 品牌偵測範例
 - 偵測影像中的熱門品牌
 - 下列 JSON 回應說明了在範例影像中偵測到品牌時，電腦視覺傳回的內容。



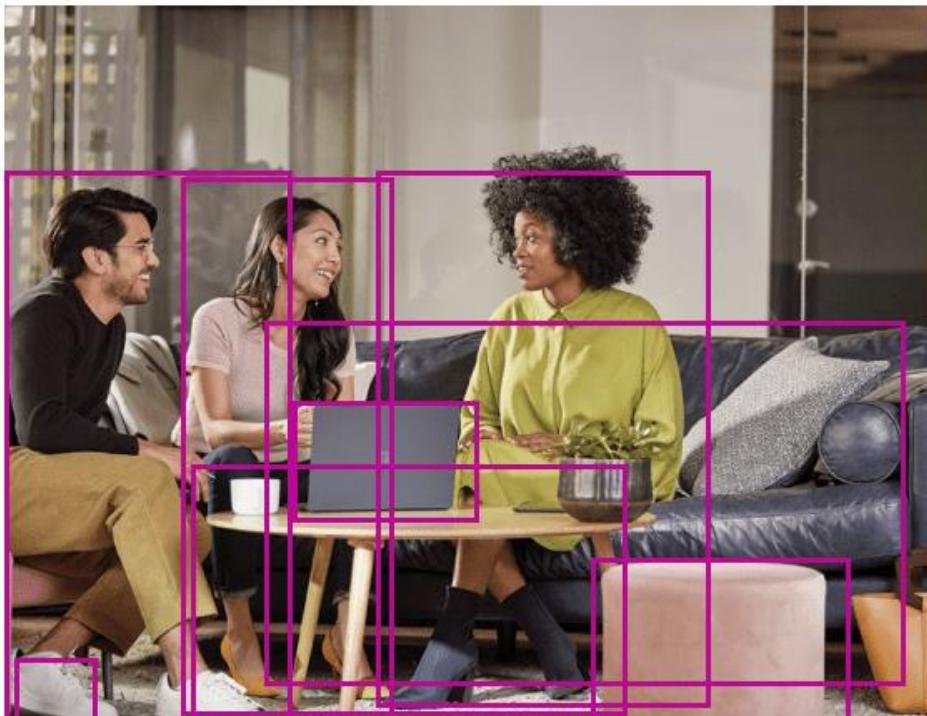
```
"brands": [  
  {  
    "name": "Microsoft",  
    "rectangle": {  
      "x": 20,  
      "y": 97,  
      "w": 62,  
      "h": 52  
    }  
  }  
]
```

- 影像分類範例
 - 下列 JSON 回應說明根據視覺功能進行範例影像分類時，電腦視覺傳回的內容。



```
{  
  "categories": [  
    {  
      "name": "people_",  
      "score": 0.81640625  
    }  
  ],  
  "requestId": "bae7f76a-1cc7-4479-8d29-48a694974705",  
  "metadata": {  
    "height": 200,  
    "width": 300,  
    "format": "Jpeg"  
  }  
}
```

- 分析影像後產生易讀的句子來描述影像內容
 - 根據不同的視覺特徵傳回數個描述, 每個描述都有信賴分數。
 - 描述清單排列順序為信賴分數最高到最低。



| | |
|---|--|
| }, { "name": "clothing", "confidence": 0.9374012 }, { "name": "furniture", "confidence": 0.9298235 }, { "name": "indoor", "confidence": 0.9139405 }, { "name": "table", "confidence": 0.7971237 }, { "name": "chair", "confidence": 0.722082436 }, { "name": "people", "confidence": 0.7205929 }, { "name": "woman", "confidence": 0.543776631 }] | |
| 描述 | { "tags": ["person", "laptop", "indoor", "computer", "sitting", "table", "man", "people", "group", "woman", "front", "using", "room", "living", "talking", "desk", "young", "food", "restaurant", "train", "standing"], "captions": [{ "text": "a group of people sitting at a table using a laptop", "confidence": 0.9693954 }] } |
| 影像格式 | "Jpeg" |
| 影像尺寸 | 430 x 558 |

- 電腦視覺會偵測影像中的人臉，並針對偵測到的臉部產生年齡、性別和矩形。
- 臉部偵測範例
 - 下列範例使用包含單一人臉的影像，示範電腦視覺傳回的 JSON 回應。



```
{  
  "faces": [  
    {  
      "age": 23,  
      "gender": "Female",  
      "faceRectangle": {  
        "top": 45,  
        "left": 194,  
        "width": 44,  
        "height": 44  
      }  
    }  
,  
  "requestId": "8439ba87-de65-441b-a0f1-c85913157ecd",  
  "metadata": {  
    "height": 200,  
    "width": 300,  
    "format": "Png"  
  }  
}
```

- 臉部辨識

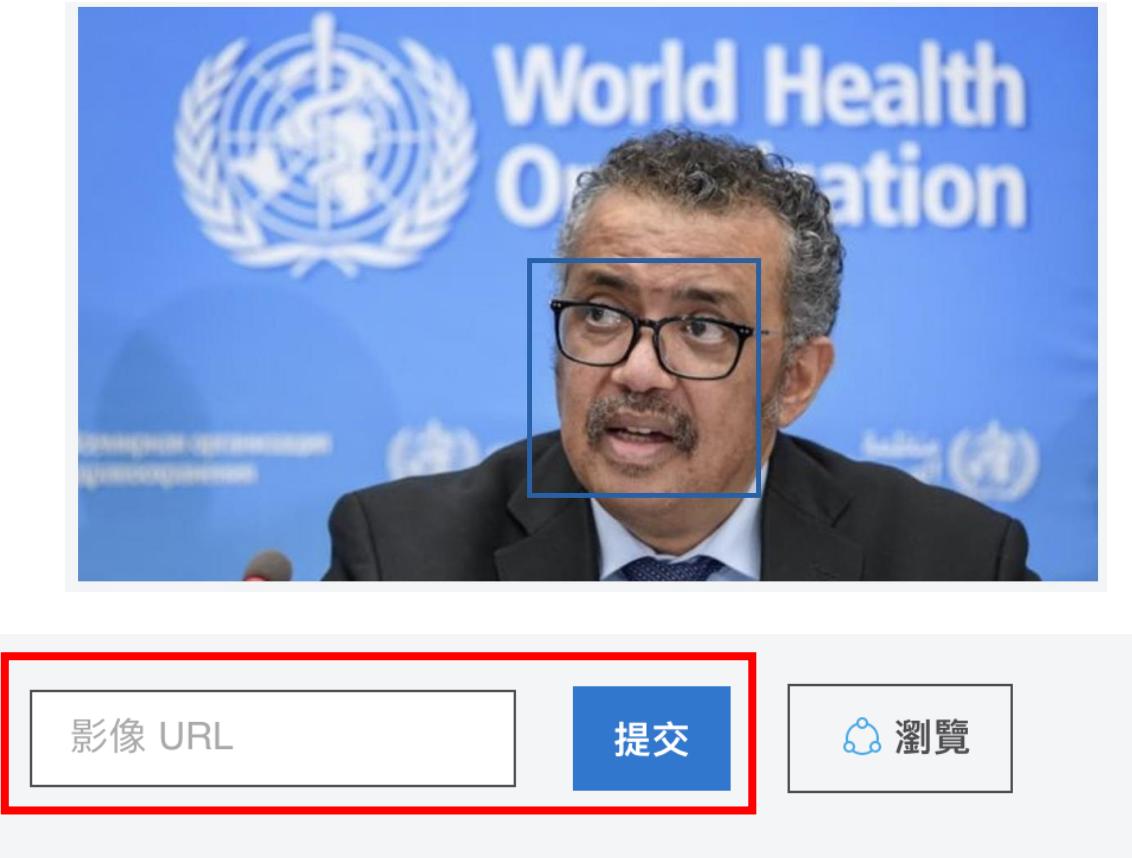
24-2:臉部辨識

- 偵測認知到的臉部表情，例如生氣、藐視、厭惡、恐懼、快樂、中立、傷心和驚訝。



```
"faceAttributes": {  
    "emotion": {  
        "anger": 0.0,  
        "contempt": 0.0,  
        "disgust": 0.0,  
        "fear": 0.0,  
        "happiness": 1.0,  
        "neutral": 0.0,  
        "sadness": 0.0,  
        "surprise": 0.0  
    }  
}
```

- 試著於影像URL自行輸入影像網址並觀察回傳結果



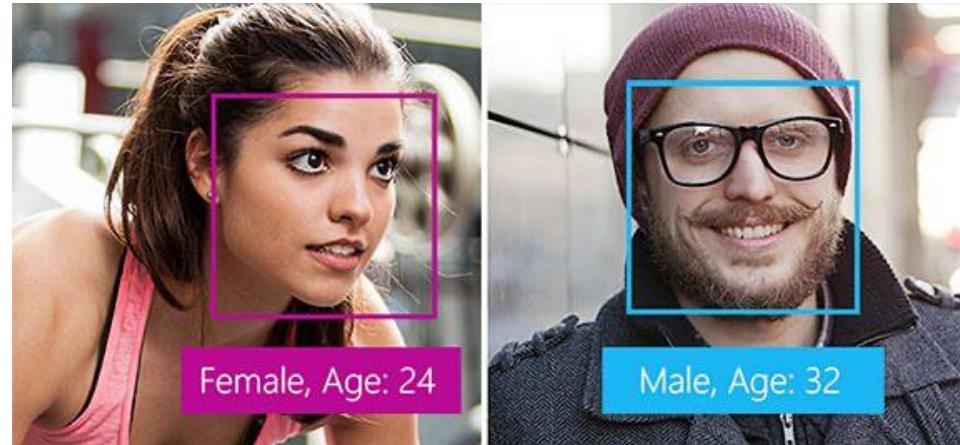
偵測結果：
偵測到 1 個臉部

JSON :

```
[  
  {  
    "faceRectangle": {  
      "top": 303,  
      "left": 543,  
      "width": 288,  
      "height": 288  
    },  
    "faceAttributes": {  
      "emotion": {  
        "anger": 0.002,  
        "contempt": 0.002,  
        "disgust": 0.002,  
        "fear": 0.003,  
        "happiness": 0.002,  
        "neutral": 0.831,  
        "sadness": 0.008,  
        "surprise": 0.149  
      }  
    }  
  }  
]
```

認識臉部偵測服務

- 臉部偵測可偵測影像中的人臉，並將相關資訊擷取後傳回，如姿勢、性別、年齡、眼鏡等



- 支援的地區與價格請參考以下連結
 - <https://azure.microsoft.com/zh-tw/pricing/details/cognitive-services/face-api/>

認識臉部驗證服務

- 查看兩張影像中的人像是否屬於同一人



驗證結果：這兩張臉屬於同一個人。信心為 0.93468。

學習目標：

- 語音及語言服務

24-3:語音及語言服務

- 語音服務的語音轉換文字（也稱為語音辨識）
 - 可將音訊串流的即時轉譯為文字。



- 語音服務支援的語言和區域總表請到以下網址查詢
 - <https://docs.microsoft.com/zh-tw/azure/cognitive-services/speech-service/language-support>

| Locale | Language | 支援 | 自訂 |
|--------|----------------|----|------|
| ar-AE | 阿拉伯文（阿拉伯聯合大公國） | 是 | 否 |
| ar-BH | 阿拉伯文（巴林），現代化標準 | 是 | 語言模型 |
| ar-EG | 阿拉伯文 (埃及) | 是 | 語言模型 |

文字轉語音服務

- 介紹

- 語音服務中的文字轉語音服務使應用程式、工具或設備能夠將文本轉換為類似人類的合成語音。

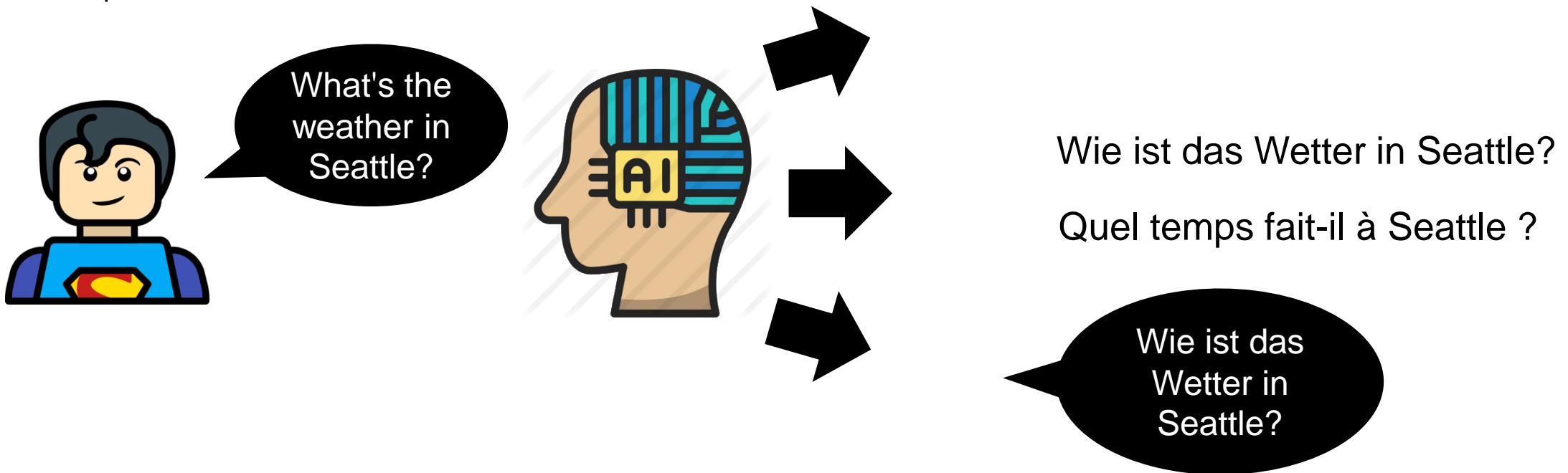


- 語系支援

- 細節請參考下面連結取的所使用的地區位置是否有相關服務與支援
 - <https://docs.microsoft.com/zh-tw/azure/cognitive-services/speech-service/language-support#text-to-speech>

- 語音翻譯

- 借助語音 SDK，應用程式、工具和設備可以提供音訊的翻譯輸出。
- 支援語音轉成翻譯文字、語音轉成多國翻譯文字、語音轉成多國語音輸出



- 從文字分析情感
 - 文字分析 API 的情感分析功能會評估文字，並傳回每個句子的情感分數和標籤。
 - <https://azure.microsoft.com/zh-tw/services/cognitive-services/text-analytics/>



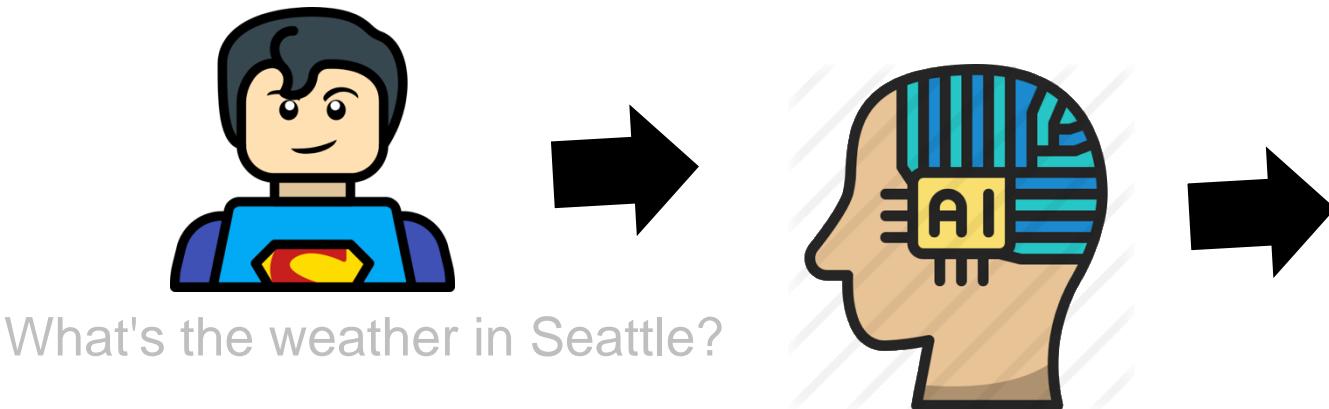
- 從文字分析語言類型

- 文字分析 API 的語言偵測功能會評估文字，並傳回偵測到的語言。
- <https://azure.microsoft.com/zh-tw/services/cognitive-services/text-analytics/>



- 意圖辨識

- 意圖是使用者想要做的事情：訂機票、查看天氣或打電話。
- 意圖辨識就是讓AI分析自然語言並從知識庫中找出相對應語句回答



認識QnA maker服務

- QnA maker是一項雲端式 API 服務，
 - 對現有資料建立交談式的問答。
 - 從半結構化內容 (包括常見問題集、手冊和文件) 擷取問答來建立知識庫。
 - 透過 QnAs，以知識庫中的最佳答案自動回答使用者的問題。
 - 知識庫會持續從使用者行為學習。

