



中原大學 雲端計算平台實務

12/02-作業報告

Microsoft Azure AI Fundamentals:

Explore computer vision

資訊碩一 11177035 林彥輝

授課教師：鍾武君 教授

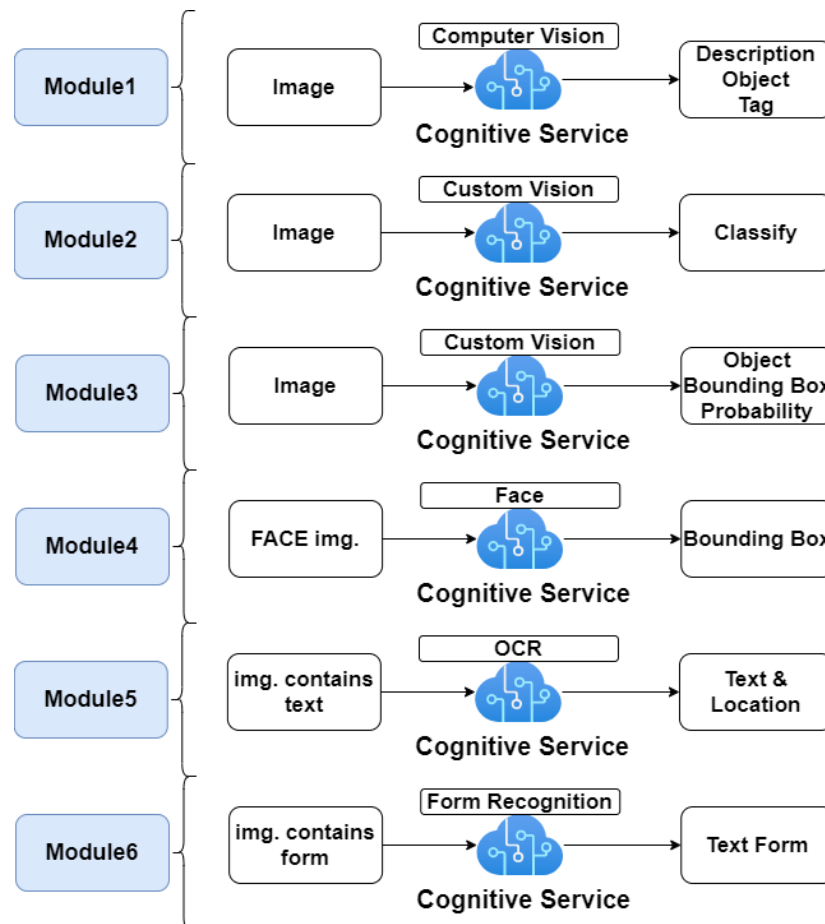
中華民國一一一年十二月

1. Model Intro

Microsoft Azure AI Fundamentals: Explore computer vision

<https://docs.microsoft.com/en-us/learn/paths/explore-computer-vision-microsoft-azure/>

2. Summary Homework Assignment



1. Prepare: Create a Cognitive Service in Azure & Get Keys and Endpoints

Hide Keys

KEY 1

e88c611a33ec415c8e3f3fe416256865

Copy to clipboard

KEY 2

5e08bf886e4040a3b82bdf2845474bc0

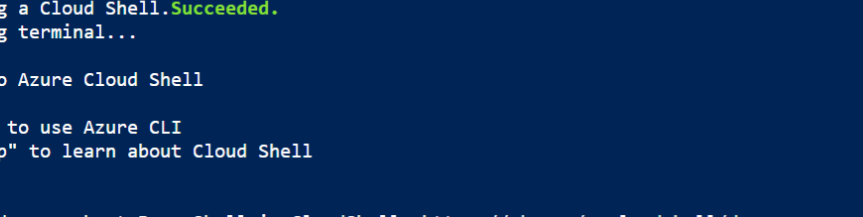
Location/Region ⓘ

brazilsouth

Endpoint

<https://11177035cognitiveservices.cognitiveservices.azure.com/>

<https://github.com/MicrosoftLearning/AI-900-AIFundamentals>



The screenshot shows a PowerShell terminal window with a dark blue background. At the top, there is a title bar with 'PowerShell' and several icons. The terminal output is as follows:

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

MOTD: Read more about PowerShell in CloudShell: https://aka.ms/pscloudshell/docs

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/azureuser> git clone https://github.com/MicrosoftLearning/AI-900-AIFundamentals ai-900
Cloning into 'ai-900'...
remote: Enumerating objects: 1226, done.
remote: Counting objects: 100% (1226/1226), done.
remote: Compressing objects: 100% (614/614), done.
remote: Total 1226 (delta 652), reused 1151 (delta 592), pack-reused 0
Receiving objects: 100% (1226/1226), 95.95 MiB | 16.67 MiB/s, done.
Resolving deltas: 100% (652/652), done.
```

此模組透過 Cognitive Service 使用 pre-train model 進行圖片分析。

```

analyze-image.ps1
1 $key="e88c611a33ec415c8e3f3fe416256865"
2 $endpoint="https://11177035cognitiveservices.cognitiveservices.azure.com/"
3
4
5 # Code to call Computer Vision service for image analysis
6 $img_file = "store-camera-1.jpg"
7 if ($args.count -gt 0 -And $args[0] -in ("store-camera-1.jpg", "store-camera-2.jpg", "sto
8 {
9     $img_file = $args[0]
10 }
11
12 $img = "https://raw.githubusercontent.com/MicrosoftLearning/AI-900-AIFundamentals/main/de
13
14 $headers = @{}
15 $headers.Add( "Ocp-Apim-Subscription-Key", $key )
16 $headers.Add( "Content-Type", "application/json" )
17
18 $body = "{ 'url' : '$img' }"
19
20 write-host "Analyzing image..."
21 $result = Invoke-RestMethod -Method Post `
22     -Uri "$endpoint/vision/v3.2/analyze?visualFeatures=Categories,Description,Objec
23     -Headers $headers `

```

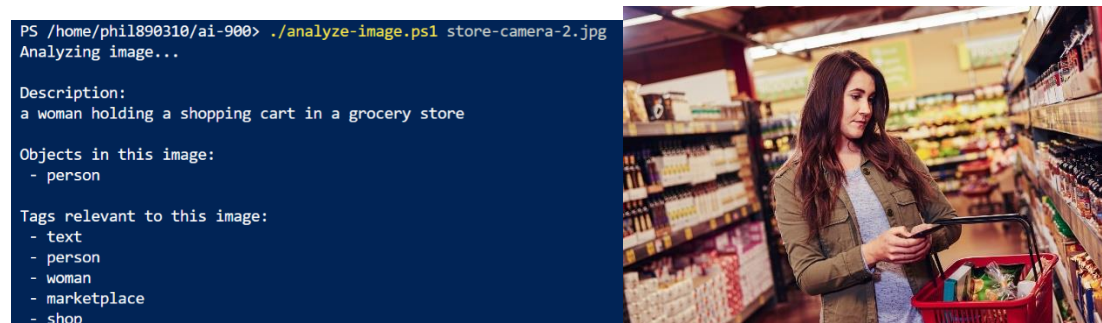
Lines 1-23 為程式碼主要重點，剩餘程式碼為 print function。在 Lines 1-2 需要帶入 Cognitive Service 的 Key 以及 Endpoint，作為調用服務所需的路徑及憑證；Lines 5-11 選定欲分析照片，照片檔名取決於呼叫程式碼的 arg；Line 12 根據照片檔名下載圖片；Lines 14-18 宣告調用 RestfulAPI 所需標頭檔；Lines 21-23

進行調用，透過 Post 方式傳輸內容，內容特徵選定 Computer Vision3.2 版，並且分析特徵：Categories、Description、Objects，透過 Document[1]可查閱更多特徵。

Result-1:



Result-2:



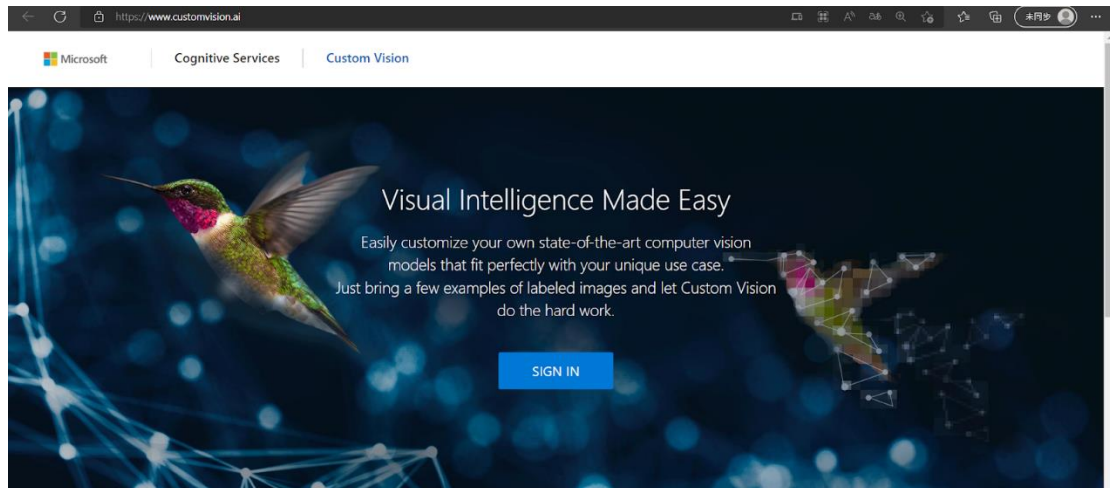
Result-3:



透過 API 可以調用 Cognitive Service 中的 CV 服務，使用 Pre-Train Model 對圖片各種特徵進行分析，如 Description、Objects、Tags。在 Description 的特徵會描述圖片的內容，附加可能的情境；Object 會使用物件辨識找出圖片中的物件；Tags 則是根據圖片中的內容給予更精確的單詞。其餘仍有更多的特徵如 Adult 輔助檢測有無色情或暴力性質；Brands 檢測照片中有無 logo；Categories 能將圖片歸類到預定好的主題；Color 分析強調色、主色；Faces 分析人臉特徵。

Module 2: Classify images with the Custom Vision service

此主題透過 Custom Vision 訓練 Computer Vision Model，有別於一般的 Cognitive Service，Custom Vision 能自定義訓練內容，透過 GUI 的方式簡單訓練一個 AI 模型。

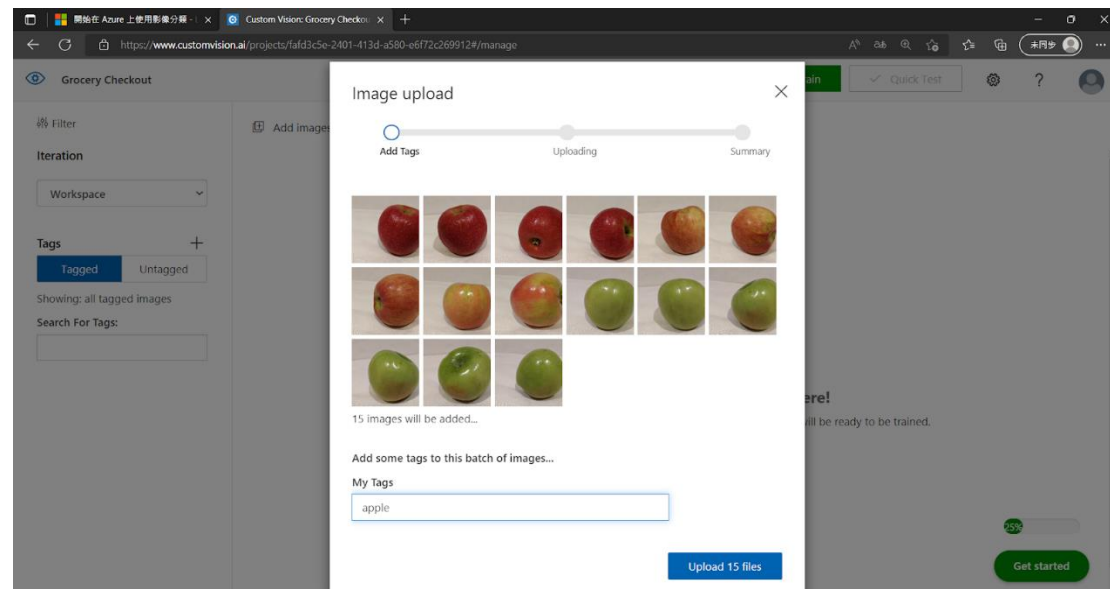


首先，在 Custom Vision 中創建一個 Project，綁定 Azure Resource 並且定義調用訓練所需優化參數。在此 Module 中選擇 Classification 搭配 Multiclass 的訓練方式，並且使用 Food 的 Domain 對模型進行優化。

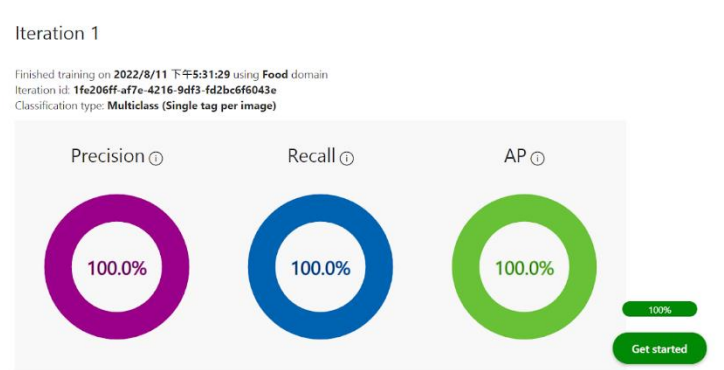
Azure 針對不同的 Domain 都有不同的特性，在 Document [2] 中，General 對於廣泛的訓練任務擁有廣泛的優化，是最通用的選項；General[A1]為追求更高的準確度，需要更多的 Inference Time，適合用於比較困難的資料集；General[A2]則與 General[A1]相反，能擁有更少的 Inference Time，適合用於多數資料集；其餘的 Domain 則相對偏重特定領域，例如 Food Domain 針對菜餚水果分析做出優

化；Landmark 針對知名景點地標進行優化；Retail 針對零售（如在裙子、褲子、襯衫）進行更精確的分析；以及 Retail(compact)，針對 Edge Device 的運算能力約束進行優化。

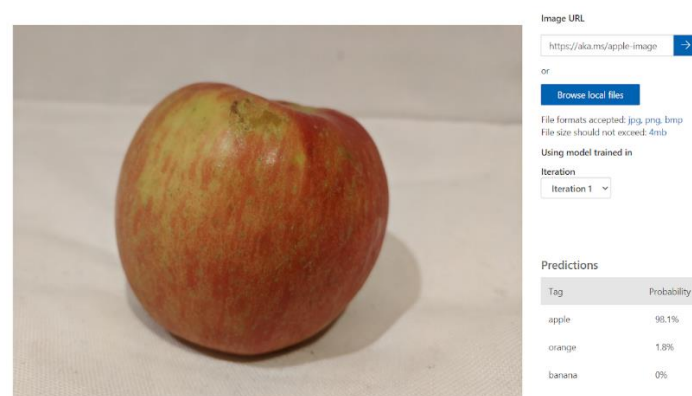
在此 Module 中，使用了水果資料集[3]，並且搭配 Food Domain 用來分蘋果、香蕉以及橘子。



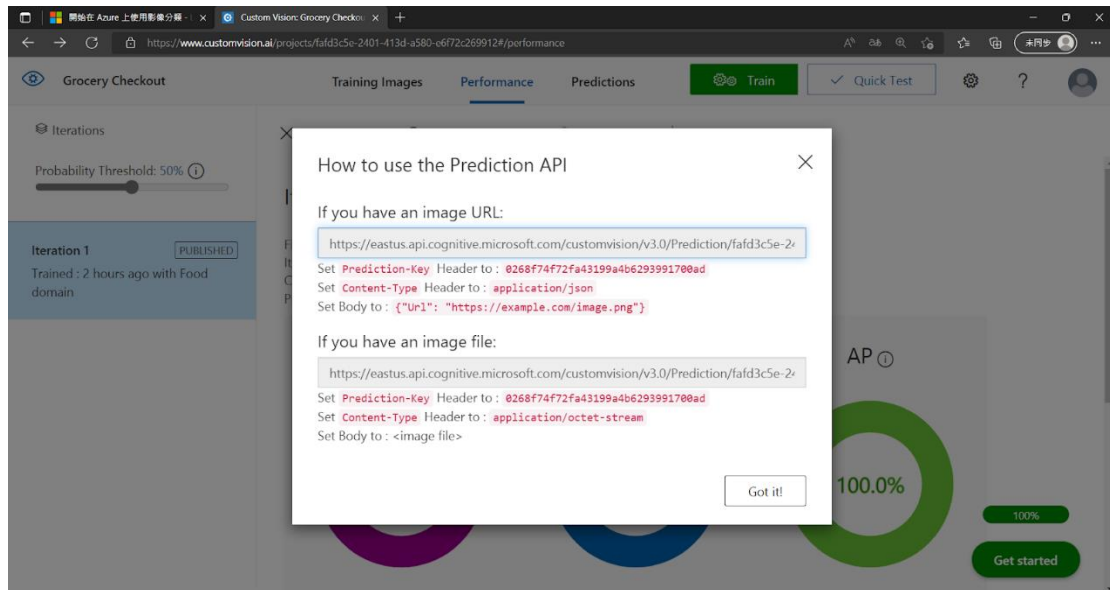
訓練完成後，Azure 會自動針對模型進行測試，得出 Precision、Recall、AP 的評估指標。



Custom Vision 官方也提供 Quick Test，可以即時做 Test 做額外的確認，當一切測試皆達到開發人員的要求時，再將模型發布。



Custom Vision 提供兩種 URL 做 Publish，分別是使用 image url 作為圖片輸入的方式，或是使用傳輸 image 作為輸入。



在 classify-image.ps1 中，替代 predictionUrl 以及 predictionKey 即可執行。

```
classify-image.ps1
1 $predictionUrl="https://11177035cognitiveservices.cognitiveservices.azure.com/"
2 $predictionKey = "e88c611a33ec415c8e3f3fe416256865"
3
4
5 # Code to call Custom Vision service for image classification
6
7 $img_num = 1
8 if ($args.count -gt 0 -And $args[0] -in (1..3))
9 {
10     $img_num = $args[0]
11 }
12
13 $img = "https://raw.githubusercontent.com/MicrosoftLearning/AI-900-AIFundamentals/main/data/vision/fruit-$( $img_num ).jpg"
14
15 $headers = @{}
16 $headers.Add( "Prediction-Key", $predictionKey )
17 $headers.Add( "Content-Type","application/json" )
18
19 $body = '{"url': '$img'}"
20
21 write-host "Analyzing image..."
22 $result = Invoke-RestMethod -Method Post `
23     -Uri $predictionUrl `
24     -Headers $headers `
25     -Body $body | ConvertTo-Json -Depth 5
26
27 $prediction = $result | ConvertFrom-Json
28
29 Write-Host ("`n",$prediction.predictions[0].tagName, "`n")
```

Lines 1-25 為程式碼主要重點，剩餘程式碼為 print function。在 Lines 1-2 帶入 predictionUrl 以及 predictionKey，作為調用服務所需的路徑及憑證；Lines 7-13 選定欲分類照片，照片檔名取決於呼叫程式碼的 arg；Line 13 根據照片檔名下載圖片；Lines 15-17 宣告調用 RestfulAPI 所需標頭檔，按照 Custom Vision 所要求格式進行設定；Lines 22-25 進行調用，透過 Post 方式傳輸圖片以及分析結果。進階查看 Document [4]，回傳結果也包含 Probability 的參數可以顯示。

```
PS /home/phil890310/ai-900> ./classify-image.ps1 1
Analyzing image...

apple

PS /home/phil890310/ai-900> ./classify-image.ps1 2
Analyzing image...

banana

PS /home/phil890310/ai-900> ./classify-image.ps1 3
Analyzing image...

orange
```



結果如上圖所示，呼叫程式搭配參數，不同的參數代表不同的 test image 編號，結果也成功正確預測分類結果。

Module 3: Detect objects in images with the Custom Vision service

摘要：使用 Custom Vision 訓練自訂模型進行物件偵測。

與 Module2 相似，使用 Custom Vision 訓練自訂模型，但在此 Module 練習的是物件偵測，因此在 Create Project 時選擇 Object Detection 的選項，並設定 General Domain。

Name*

Grocery Detection

Description

Object detection for groceries.

Resource*

GroceryCheckout [S0]

[create new](#)

[Manage Resource Permissions](#)

Project Types ⓘ

☐ Classification

☒ Object Detection

Domains:

☐ General [A1]

☒ General

☐ Logo

☐ Products on Shelves

☐ General (compact) [S1]

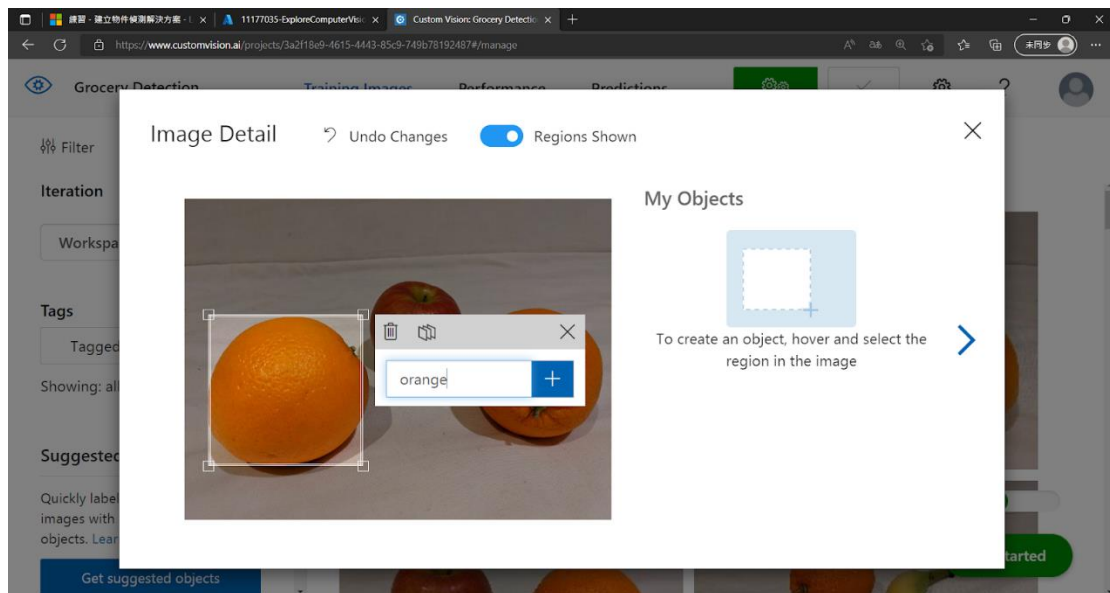
☐ General (compact)

Pick the domain closest to your scenario. Compact domains are lightweight models that can be exported to iOS/Android and other platforms. [Learn More](#)

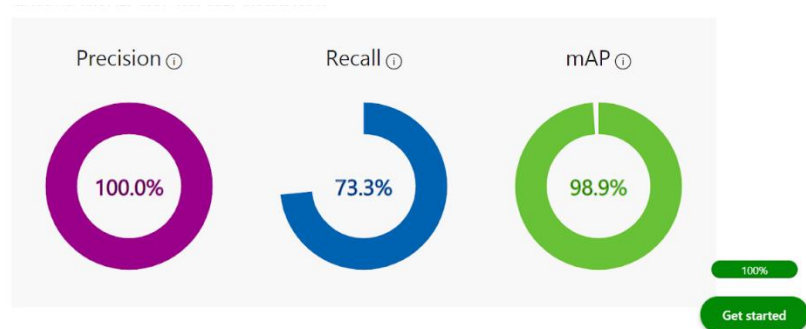
Cancel

Create project

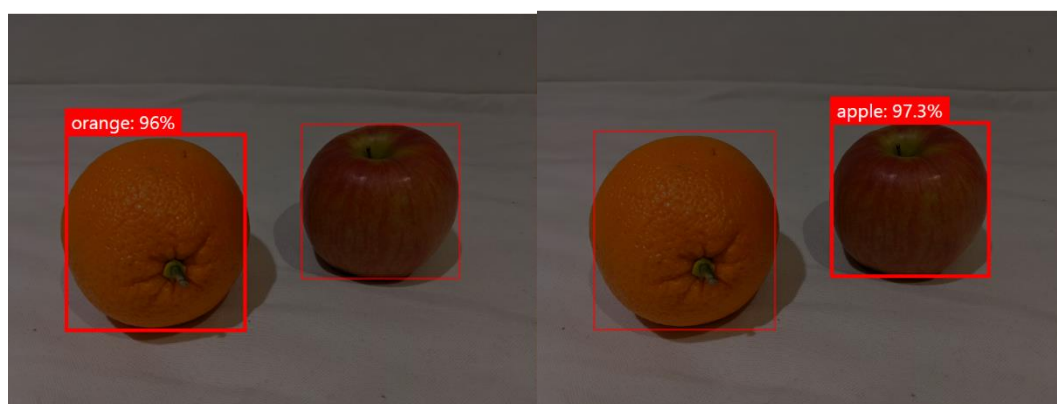
搭配水果圖片資料集 [5] 訓練圖片，資料集中擁有蘋果、橘子、香蕉的混和資料，把資料上傳至 Custom Vision 中，並且手動標記 Label。



標記完 Label 之後進行訓練，訓練完的結果 Azure 會自動進行 Precision、Recall、mAP 的模型分析。



訓練模型可以當下做 Test 做額外的確認，當一切測試皆達到開發人員的要求時，再將模型發布。



與前一 Module 相同，Prediction API 提供不同 Inference 的方式供圖片做預測。

How to use the Prediction API



If you have an image URL:

```
https://eastus.api.cognitive.microsoft.com/customvision/v3.0/Prediction/3a2f18e9-4
```

Set **Prediction-Key** Header to: 0268f74f72fa43199a4b6293991700ad

Set **Content-Type** Header to: application/json

Set Body to: {"Url": "https://example.com/image.png"}

If you have an image file:

```
https://eastus.api.cognitive.microsoft.com/customvision/v3.0/Prediction/3a2f18e9-4
```

Set **Prediction-Key** Header to: 0268f74f72fa43199a4b6293991700ad

Set **Content-Type** Header to: application/octet-stream

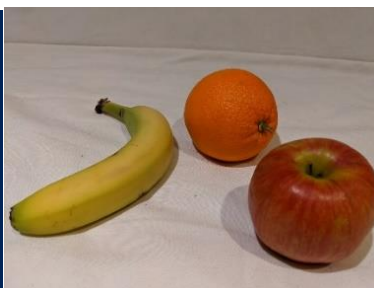
Set Body to: <image file>

```
detect-objects.ps1
1 $predictionUrl="https://11177035cognitiveservices.cognitiveservices.azure.com/"
2 $predictionKey = "e88c611a33ec415c8e3f3fe416256865"
3
4
5 # Code to call Custom Vision service for image detection
6
7 $img = "https://raw.githubusercontent.com/MicrosoftLearning/AI-900-AIFundamentals/main/data/vision/produce.jpg"
8
9 $headers = @{}
10 $headers.Add( "Prediction-Key", $predictionKey )
11 $headers.Add( "Content-Type","application/json" )
12
13 $body = '{"url' : '$img'}"
14
15 write-host "Analyzing image..."
16 $result = Invoke-RestMethod -Method Post `
17     -Uri $predictionUrl `
18     -Headers $headers `
19     -Body $body | ConvertTo-Json -Depth 5
20
21 $prediction = $result | ConvertFrom-Json
22
23 $items = $prediction.predictions
24
25 foreach ($item in $items)
26 {if ($item.probability -gt .9)
27 {
28     Write-Host ("`n",$item.tagName, "`n")
29 }
30 }
```

Lines 1-19 為程式碼主要重點，剩餘程式碼為 print function。在 Lines 1-2 帶入 predictionUrl 以及 predictionKey，作為調用服務所需的路徑及憑證；Line 7 下載圖片；Lines 9-13 宣告調用 RestfulAPI 所需標頭檔，按照 Custom Vision 所要求格式進行設定；Lines 16-19 進行調用，透過 Post 方式傳輸圖片以及分析結果。最終，在 Lines 25-30 將機率大於[6] (-gt) 0.9 的標籤顯示於螢幕中。

```
PS /home/phil890310/ai-900> ./detect-objects.ps1
Analyzing image...
```

```
apple
orange
banana
```



最終結果也成功辨識出照片中的所有水果物件並且將結果印出。

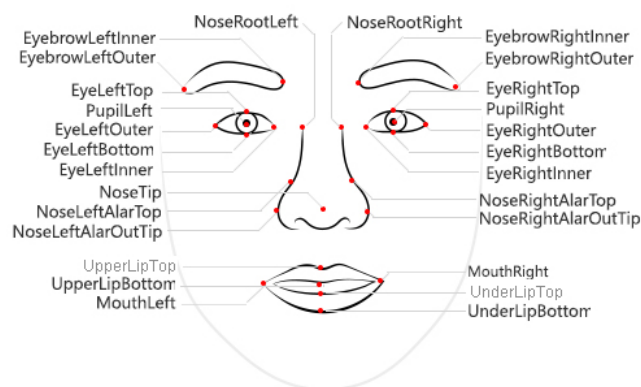
Module 4: Detect and analyze faces with the Face service

此部分主要探討 CV 中的 FaceAPI 對於人臉領域的服務，但受限於 Responsible AI [7] 的限制，Microsoft 官方嚴格禁止將服務出售給美國警察部門做使用，私人使用的用戶需要向官方提交註冊表並簽屬相關法律條款。

```
find-faces.ps1
1 $key="e88c611a33ec415c8e3f3fe416256865"
2 $endpoint="https://11177035cognitiveservices.azure.com/"
3
4
5
6 # Code to call Face service for face detection
7 $img_file = "store-camera-1.jpg"
8 if ($args.count -gt 0 -And $args[0] -in ("store-camera-1.jpg", "store-camera-2.jpg", "store-camera-3.jpg"))
9 {
10     $img_file = $args[0]
11 }
12
13 $img = "https://raw.githubusercontent.com/MicrosoftLearning/AI-900-AIFundamentals/main/data/vision/$img_file"
14
15 $headers = @{}
16 $headers.Add( "Ocp-Apim-Subscription-Key", $key )
17 $headers.Add( "Content-Type", "application/json" )
18
19 $body = '{"url": "$img"}'
20
21 write-host "Analyzing image..."
22 $result = Invoke-RestMethod -Method Post `
23     -Uri "$endpoint/face/v1.0/detect?detectionModel=detection_01" `
24     -Headers $headers `
25     -Body $body | ConvertTo-Json -Depth 5
26
27 $analysis = ($result | ConvertFrom-Json)
28 Write-Host ("From June 21st 2022, Face service capabilities that return personally identifiable features are restricted. See https://azur
29 foreach ($face in $analysis)
30 {
31     Write-Host("Face location: $($face.faceRectangle)`n")
32 }
33
```

Lines 1-26 為程式碼主要重點，剩餘程式碼為 print function。在 Lines 1-2 需要帶入 Cognitive Service 的 Key 以及 Endpoint，作為調用服務所需的路徑及憑證；Lines 6-13 選定欲分析照片，照片檔名取決於呼叫程式碼的 arg；Line 13 根據照片檔名下載圖片；Lines 14-19 宣告調用 RestfulAPI 所需標頭檔以及傳輸內容；Lines 22-25 進行調用，透過 Post 方式傳輸內容，使用特定的“detection_01”模型進行分析。最後在 31 行的印出臉部位置的 bounding box 位置。

在 Document [8] 中，進階閱讀可以發現調用此 API 可以獲得更多資訊，例如 faceLandmark（如下圖）資訊，以及 faceAttributes（年紀、性別、微笑、戴眼鏡）等特徵。



```
PS /home/phil890310/ai-900> ./find-faces.ps1 store-camera-1.jpg
Analyzing image...
```

```
From June 21st 2022, Face service capabilities that return personally identifiable features are restricted.
See https://azure.microsoft.com/blog/responsible-ai-investments-and-safeguards-for-facial-recognition/ for details.
This code is restricted to returning the location of any faces detected:
```

```
Face location: @{{top=133; left=339; width=94; height=94}}
```



```
From June 21st 2022, Face service capabilities that return personally identifiable features are restricted.
See https://azure.microsoft.com/blog/responsible-ai-investments-and-safeguards-for-facial-recognition/ for details.
This code is restricted to returning the location of any faces detected:
```

```
Face location: @{{top=171; left=454; width=95; height=95}}
```



程式結果如上，結果顯示在每張照片中，人臉位置的 Bounding Box 位置。

Module 5: Read text with the Computer Vision service

此模組使用 OCR(Optical Character Recognition) API 對於圖片文字的辨識服務。

```
ocr.ps1
1 $key="e88c611a33ec415c8e3f3fe416256865"
2 $endpoint="https://11177035cognitiveservices.cognitiveservices.azure.com/"
3
4
5 # Code to call OCR service for text in image analysis
6 $img_file = "advert.jpg"
7 if ($args.count -gt 0 -And $args[0] -in ("advert.jpg", "letter.jpg", "note.jpg"))
8 {
9     $img_file = $args[0]
10 }
11
12 $img = "https://raw.githubusercontent.com/MicrosoftLearning/AI-900-AIFundamentals/main/data/vision/$img_file"
13
14 $headers = @{}
15 $headers.Add( "Ocp-Apim-Subscription-Key", $key )
16 $headers.Add( "Content-Type","application/json" )
17
18 $body = '{"url' : '$img'}"
19
20 write-host "Analyzing Image..."
21 $result = Invoke-WebRequest -Method Post `
22     -Uri "$endpoint/vision/v3.2/read/analyze?language=en" `
23     -Headers $headers `
24     -Body $body
```

範例程式中，Lines 1-24 為程式碼主要重點，剩餘程式碼為 print function。在 Lines 1-2 需要帶入 Cognitive Service 的 Key 以及 Endpoint，作為調用服務所需的路徑及憑證；Lines 6-10 選定欲分析照片，照片檔名取決於呼叫程式碼的 arg；Line 12 根據照片檔名下載圖片；Lines 14-18 宣告調用 RestfulAPI 所需標頭檔以及傳輸內容；Lines 21-24 進行調用，透過 Post 方式傳輸內容，調用 read API [9]，結果如下圖顯示，返回所有圖片中的單詞 Bounding Box 位置。


```
PS /home/phil890310/ai-900> ./ocr.ps1 advert.jpg
Analyzing image...

Text: Northwind | Text Bounding Box: 74,76,529,98
Text: Traders | Text Bounding Box: 639,70,368,91
Text: Fresh | Text Bounding Box: 74,281,138,48
Text: produce, | Text Bounding Box: 235,277,234,61
Text: friendly | Text Bounding Box: 71,360,207,57
Text: service | Text Bounding Box: 296,358,185,47
Text: Open | Text Bounding Box: 1176,1069,141,55
Text: 7 | Text Bounding Box: 1338,1066,29,43
Text: days | Text Bounding Box: 1388,1062,117,56
Text: a | Text Bounding Box: 1524,1072,26,33
Text: week | Text Bounding Box: 1570,1055,142,48
```



```
PS /home/phil890310/ai-900> ./ocr.ps1 letter.jpg
Analyzing image...

Text: January | Text Bounding Box: 1212,23,153,29
Text: 23rd | Text Bounding Box: 1391,18,72,27
Text: 2020 | Text Bounding Box: 1488,21,84,24
Text: For | Text Bounding Box: 33,113,63,22
Text: the | Text Bounding Box: 122,111,63,24
Text: attention | Text Bounding Box: 212,110,198,25
Text: of: | Text Bounding Box: 435,111,57,24
Text: The | Text Bounding Box: 32,156,63,24
Text: manager | Text Bounding Box: 120,163,155,24
Text: Northwind | Text Bounding Box: 31,200,201,25
Text: Traders | Text Bounding Box: 256,201,152,24
Text: 123 | Text Bounding Box: 34,246,60,24
Text: Any | Text Bounding Box: 120,248,67,29
Text: Street | Text Bounding Box: 212,248,130,22
Text: Bellevue, | Text Bounding Box: 32,291,191,30
Text: WA | Text Bounding Box: 254,293,45,22
Text: Dear | Text Bounding Box: 33,383,86,22
Text: Sir | Text Bounding Box: 145,380,63,25
Text: or | Text Bounding Box: 234,388,41,17
Text: Madam | Text Bounding Box: 289,381,126,30
```

For the attention of:
The manager
Northwind Traders
123 Any Street
Bellevue, WA

Dear Sir or Madam,

I am writing to thank you for the fantastic service I received at your store on January 20th. The store assistant who helped me was extremely pleasant and attentive; and took the time to find all of the fresh produce I needed.

I've always found the quality of the produce in your store to be high, and the prices to be competitive; and the helpfulness of your employees is another reason I will continue to remain a loyal Northwind Traders customer.

Sincerely,

A Customer
A. Customer

January 23rd 2020

Module 6: Analyze receipts with the Form Recognizer

service

此模組使用 Form Recognizer 辨識更進階的 OCR 應用，Form Recognizer 能進一步組織 OCR 的內容，適用於文件辨識（文件、發票、表單、名片）。

```
form-recognizer.ps1
1 $key="e88c611a33ec415c8e3f3fe416256865"
2 $endpoint="https://11177035cognitiveservices.cognitiveservices.azure.com/"
3
4
5 # Create the URL where the raw receipt image can be found
6 $img = "https://raw.githubusercontent.com/MicrosoftLearning/AI-900-AIFundamentals/main/data/vision/receipt.jpg"
7
8 # Create the header for the REST POST with the subscription key
9 # In this example, the URL of the image will be sent instead of
10 # the raw image, so the Content-Type is JSON
11 $headers = @{}
12 $headers.Add( "Ocp-Apim-Subscription-Key", $key )
13 $headers.Add( "Content-Type", "application/json" )
14
15 # Create the body with the URL of the raw image
16 $body = '{"source": "$img"}'
17
18 # Call the receipt analyze method with the header and body
19 # Must call the Invoke-WebRequest to have access to the header
20 Write-Host "Sending receipt..."
21 $response = Invoke-WebRequest -Method Post `
22     -Uri "$endpoint/formrecognizer/v2.1/prebuilt/receipt/analyze" `
23     -Headers $headers `
24     -Body $body
25 Write-Host "...Receipt sent."
```

範例程式中，Lines 1-2 需要帶入 Cognitive Service 的 Key 以及 Endpoint，作為調用服務所需的路徑及憑證；Line 6 下載圖片；Lines 11-16 宣告調用 Restful API 所需標頭檔以及傳輸內容；Lines 20-24 進行調用，透過 Post 方式傳輸內容，調用 formrecognizer API [10]，使用 receipt 的 pre-train 模型得到第一步分析。

```

27 # Extract the URL from the response of the receipt analyzer
28 # to call the API to getting the analysis results
29 $resultUrl = $($response.Headers['operation-location'])
30
31 # Create the header for the REST GET with only the subscription key
32 $resultHeaders = @{}
33 $resultHeaders.Add( "Ocp-Apim-Subscription-Key", $key )
34
35 # Get the receipt analysis results, passing in the resultUrl
36 # Continue to request results until the analysis is "succeeded"
37 Write-Host "Getting results..."
38 Do {
39     $result = Invoke-RestMethod -Method Get `
40         -Uri $resultUrl `
41         -Headers $resultHeaders | ConvertTo-Json -Depth 10
42     $analysis = ($result | ConvertFrom-Json)
43 } while ($analysis.status -ne "succeeded")
44 Write-Host "...Done`n"

```

Lines 29-44 調用 formrecognizer API [11]將第一步的圖片分析提取成為第二步的結構化數據。

```

47 # Access the relevant fields from the analysis
48 $analysisFields = $analysis.analyzeResult.documentResults.fields
49
50 # Print out all of the properties of the receipt analysis
51 Write-Host ("Receipt Type: ", $($analysisFields.ReceiptType.valueString))
52 Write-Host ("Merchant Address: ", $($analysisFields.MerchantAddress.text))
53 Write-Host ("Merchant Phone: ", $($analysisFields.MerchantPhoneNumber.text))
54 Write-Host ("Transaction Date: ", $($analysisFields.TransactionDate.valueDate))
55 Write-Host ("Receipt Items: ")
56
57 # Access the individual items from the analysis
58 $receiptItems = $($analysisFields.Items.valueArray)
59
60 for (($idx = 0); $idx -lt $receiptItems.Length; $idx++) {
61     $item = $receiptItems[$idx]
62     Write-Host ("Item #", ($idx+1))
63     Write-Host (" - Name: ", $($item.valueObject.Name.valueString))
64     Write-Host (" - Price: ", $($item.valueObject.TotalPrice.valueNumber))
65 }
66
67 Write-Host ("Subtotal: ", $($analysisFields.Subtotal.text))
68 Write-Host ("Tax: ", $($analysisFields.Tax.text))
69 Write-Host ("Total: ", $($analysisFields.Total.text))

```

Lines 47-69 皆是 print function，參考 formrecognizer Document [11]的內容，將”receipt”種類的分析結果全數印出，結果如下圖所示。

```

PS /home/phil890310/ai-900> ./form-recognizer.ps1
Sending receipt...
...Receipt sent.
Getting results...
...Done

Receipt Type: Itemized
Merchant Address: 123 Main Street
Merchant Phone: 555-123-4567
Transaction Date: 2020-02-17
Receipt Items:
Item # 1
  - Name: Apple
  - Price: 0.9
Item # 2
  - Name: Orange
  - Price: 0.8
Subtotal: $1.70
Tax: $0.17
Total: $1.87

```

```

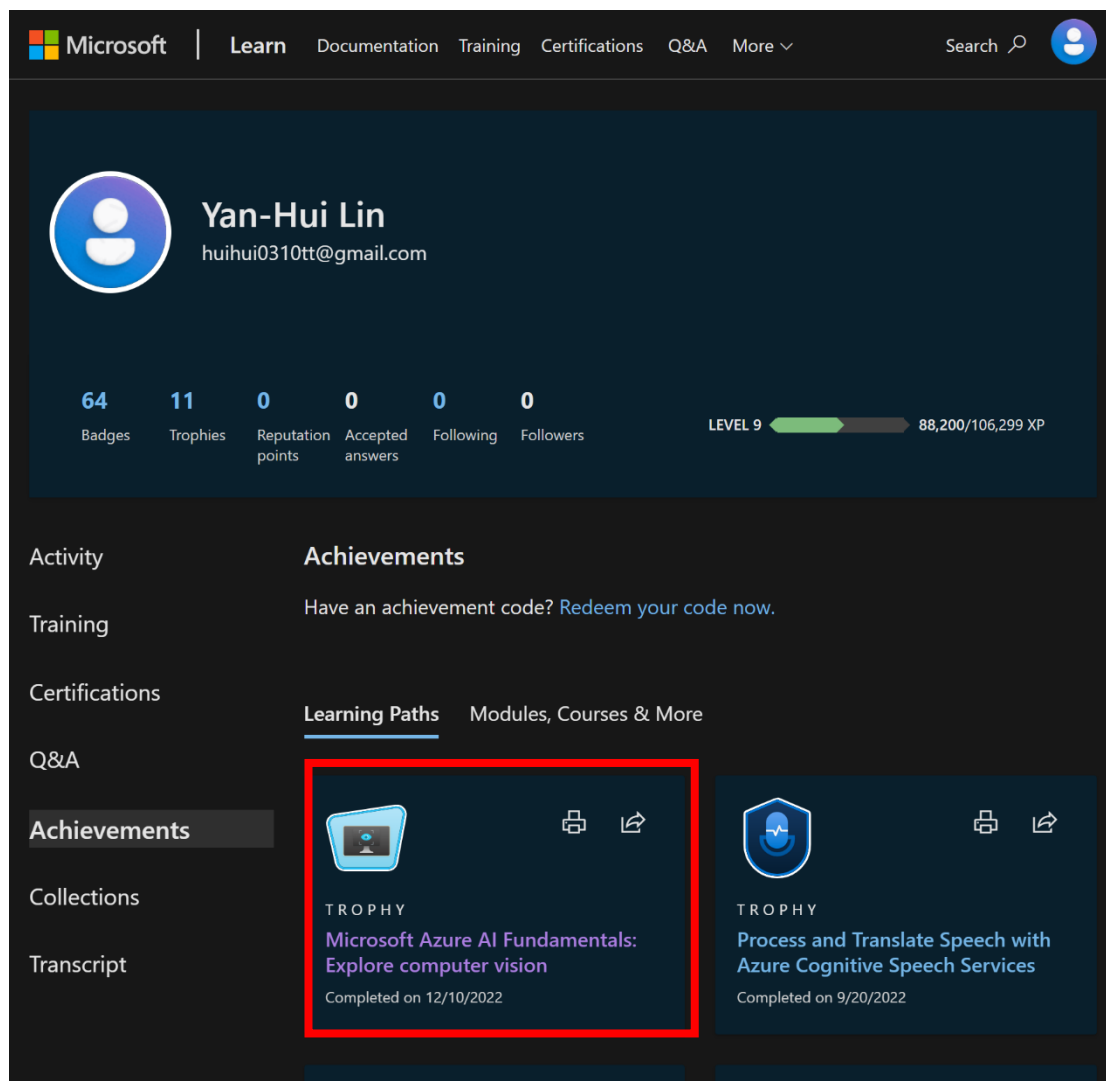
Northwind Traders
123 Main Street
555-123-4567
2/17/2020 13:07
-----
1 Apple      $0.90
1 Orange     $0.80
-----
Sub-Total    $1.70
Tax          $0.17
Total        $1.87

```

相關的分析欄為也完整的定義於 Document 當中。

receipt			
Field	Type	Description	Example
MerchantName	string	Name of the merchant issuing the receipt	Contoso
MerchantPhoneNumber	phoneNumber	Listed phone number of merchant	907-654-3210
MerchantAddress	address	Listed address of merchant	123 Main St Redmond WA 98052
Total	number	Full transaction total of receipt	\$14.34
TransactionDate	date	Date the receipt was issued	June 06, 2019
TransactionTime	time	Time the receipt was issued	4:49 PM
Subtotal	number	Subtotal of receipt, often before taxes are applied	\$12.34
TotalTax	number	Tax on receipt, often sales tax or equivalent	\$2.00
Tip	number	Tip included by buyer	\$1.00
Items	array		

Take screenshots of Badges and Trophies



Learned from the Learning Path

透過這個教程，了解到 Azure 對於 AI 開發的強大之處，開發者無需具備 AI 知識 透過 RESTful API 使用這些 AI 的服務，提供開發者可以簡單新增 AI 的功能到自己的應用上。

透過這次的 Learning Path 已經對這門課的 Final Project 有了初步的構想，Microsoft Document 對於這些 Azure AI 工具有充分的描述，包含如何使用 SDK 以及 Sample Code 描述，這些內容提供 Final Project 有更多延伸發揮的空間。

3. Problems

整體教程相當容易上手，除了在 FaceAPI 的部分受到功能限制。

FeedBack

FaceAPI 的部分因為 Responsible AI 的關係，希望可以做出教程中的改動，例如可以刪減 Exercise 的環節，使用文字描述如何調用 API，否則整體使用體驗與單純的 Object Detection 沒有太大的差異；亦或是在 Sandbox 上開啟些許功能，並限定檢測的圖片。

Cognitive Service 中還有看見另外的服務，例如 Decision 可以輔助檢測 Application 的異常流量，希望也可以一併加入 Learning Path 當中。

Reference

- [1] <https://westcentralus.dev.cognitive.microsoft.com/docs/services/computer-vision-v3-2/operations/56f91f2e778daf14a499f21b>
- [2] <https://learn.microsoft.com/en-us/azure/cognitive-services/custom-vision-service/select-domain>
- [3] <https://aka.ms/fruit-images>
- [4] <https://learn.microsoft.com/en-us/azure/cognitive-services/custom-vision-service/quickstarts/image-classification?tabs=cli&pivots=programming-language-csharp>
- [5] <https://aka.ms/fruit-objects>
- [6] <https://learn.microsoft.com/zh-tw/powershell/scripting/learn/deep-dives/everything-about-if?view=powershell-7.3>
- [7] <https://learn.microsoft.com/en-us/legal/cognitive-services/computer-vision/limited-access-identity?context=%2Fazure%2Fcognitive-services%2Fcomputer-vision%2Fcontext%2Fcontext>
- [8] <https://westus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbb8d/operations/563879b61984550f30395236>
- [9] <https://centraluseuap.dev.cognitive.microsoft.com/docs/services/computer-vision-v3-2/operations/5d986960601faab4bf452005>
- [10] <https://westus.dev.cognitive.microsoft.com/docs/services/form-recognizer-api-2022-08-31/operations/AnalyzeDocument>
- [11] <https://westus.dev.cognitive.microsoft.com/docs/services/form-recognizer-api-2022-08-31/operations/GetAnalyzeDocumentResult>