

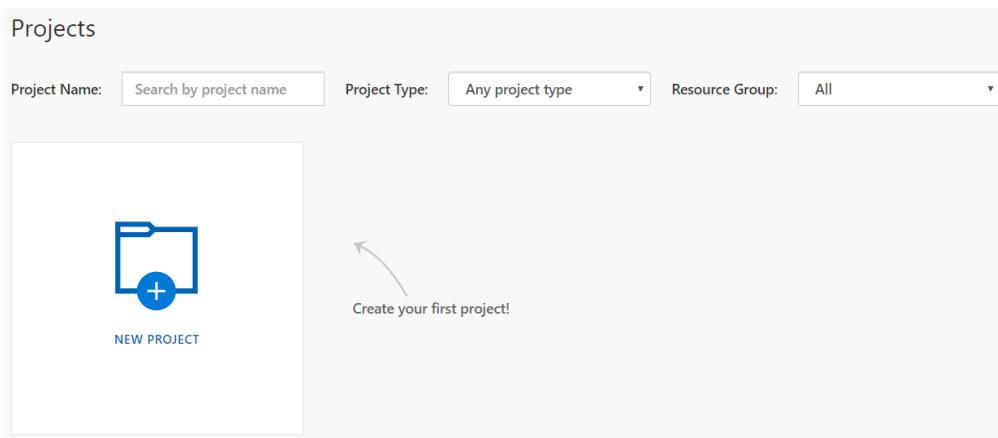
# AI 初探-如何使用Microsoft Custom Vision服務

在這個主題中, 我們要教大家如何使用 Microsoft Custom Vision 服務. Custom Vision提供了物件辨認的服務,使用者只要提供一些想要辨認的圖片, 機器就可以透過訓練學習出要辨認的物體. 這個服務等於是把訓練模型的這件事交給專家(Microsoft)處理, 微軟也提供了一個便利的介面讓使用者上傳自己的資料.

以下的操作都是建立在已經擁有 Microsoft Azure 帳號並開通 CusomVision 服務的情況下. 首先到 <https://www.customvision.ai> (<https://www.customvision.ai>) 並登入.

登入帳號為 {你的itouch帳號}@O365st.cycu.edu.tw (例如 : 99999@O365sy.cycu.edu.tw), 密碼是你的itouch密碼.

登入後會有操作提示, 提醒使用者開啟一個新的專案.



**動作: 請建立一個專案, 取名叫做group-{xx} (xx是你的組別)**

開啟後要選擇一個專案名稱(自取), 幾個屬性我們稍微說明一下.

- Resource Group: 代表提供這個服務的虛擬機器, 請根據你的電腦名稱選擇正確的群組名稱(例如, 電腦編號是cycu002的組別請選擇cvst02)
- Project Types: 想要學習的問題. Classification 是分類問題, Object detection 是偵測圖片中的物體. 如果是 detection, 必須標記出想要偵測的物體在圖片的哪一個位置.
- Classification Types: 如果一張圖片中有一個以上的分類就要選 Multilabel, 只有一種物件就選 Multiclass, 本範例中請選擇 Multiclass
- Domains: 這就好像模型的先備知識一樣, 如果選對了機會學習的比較好(也就是比較準). 如果想要一個通用的模型就是選 general. 後面帶有 compact 字樣的表示是採用比較精簡的模型, 通常是為了給運算能力較差的裝置使用的. 後果就是辨識率會稍微差一些, 但對於不複雜的問題還是有不錯的辨識度. 而且 compact 模型可以匯出成 Tensorflow 的 .pb 檔, 就可以把模型檔案下載下來後在其他的裝置上執行

選完以後就可以建立專案.

## Create new project

X

### Name\*

cats-vs-dogs

### Description

Enter project description

### Resource Group

[create new](#)

test\_group [S0]



[Manage Resource Group Permissions](#)

### Project Types (i)

- Classification
- Object Detection

### Classification Types (i)

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

### Domains: (i)

- General
- Food
- Landmarks
- Retail
- General (compact)
- Food (compact)
- Landmarks (compact)
- Retail (compact)

### Export Capabilities: (i)

- Basic platforms (Tensorflow, CoreML, ONNX, ...)
- Vision AI Dev Kit

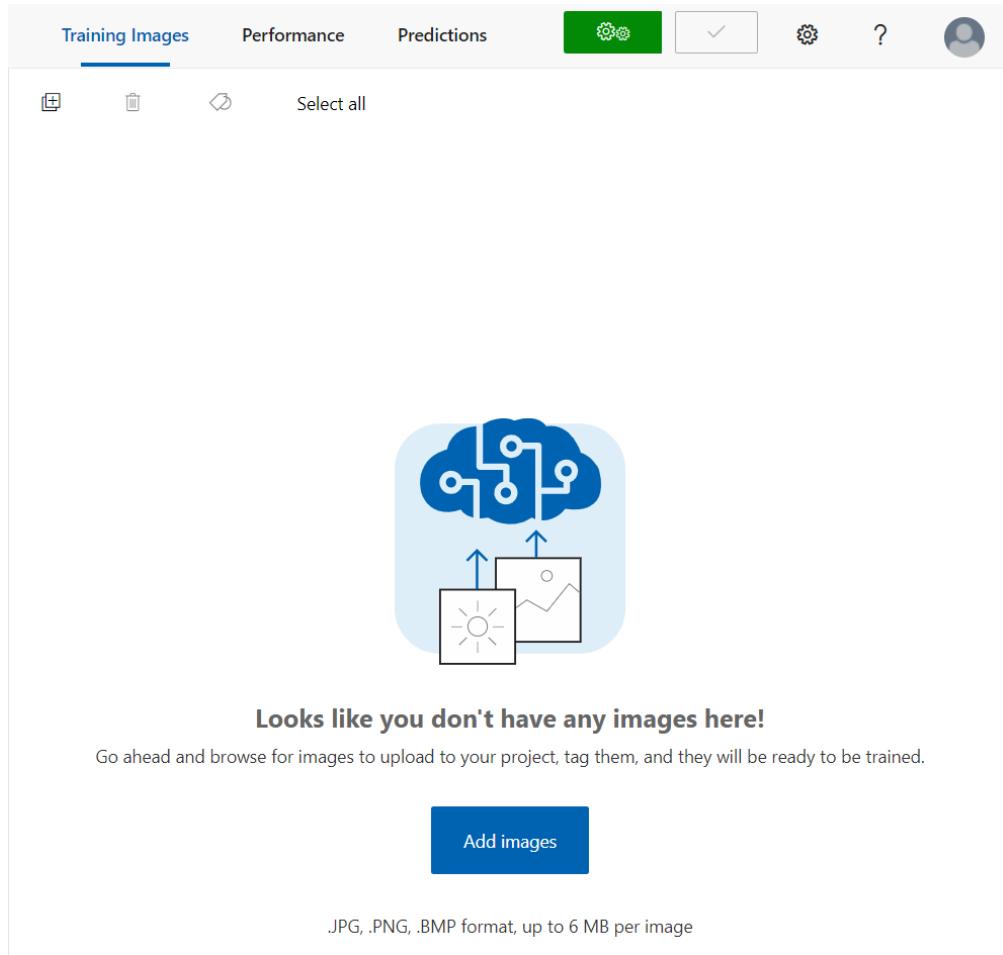
[Cancel](#)

[Create project](#)

建好專案後就可以開始上傳資料。這邊我們用Kaggle上的經典問題:[貓狗大戰](https://www.kaggle.com/c/dogs-vs-cats)(<https://www.kaggle.com/c/dogs-vs-cats>)的資料集做練習。這個資料集總共有50000張貓跟狗的圖片，這邊我們已準備了一個比較小範圍的資料集，只有1000張的貓狗圖片集(在dataset資料夾內)。

首先,我們先示範如何上傳訓練用的圖片. 在開啟專案後的上方會有三個分頁標籤,分別是 Training Images, Performance 跟 Predictions. 在 Training Images 的分頁上,你可以點擊 Add Images 按鈕來加入一些圖片到專案中.

註: 如果你已經上傳過一些圖片,在這個分頁中會顯示已經上傳的圖片. 如果你要上傳更多的圖片,你可以點選 Training Images下方的「+」圖案.



**動作:請從 dataset/cat-vs-dogs 資料夾中的 cat 資料夾選取 cat.0.jpg ~ cat.4.jpg 共五張圖片並加入專案**

選完圖片後會出現 Image Upload 這個畫面,此時請在下面 My Tags 輸入框中設定這些照片的標籤(tag)名稱.  
註:標籤名稱原本可以自訂,但建議訂成好懂的名稱,或是配合後續的其他程式需要的名稱,這邊我們定義貓的標籤叫cat,狗的標籤叫dog.

## Image upload

X



Add Tags



Uploading

Summary



1000 images will be added...

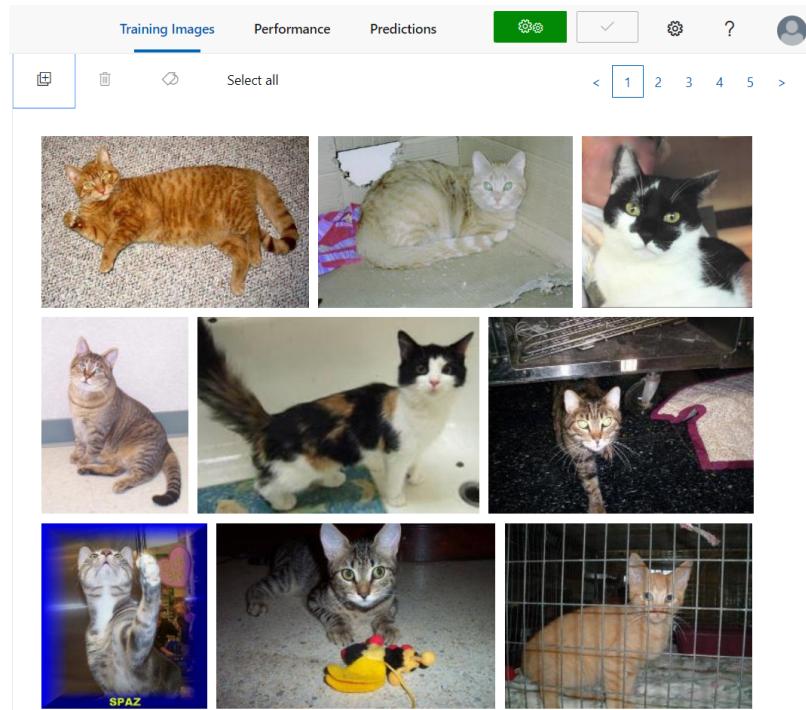
Add some tags to this batch of images...

My Tags

cat

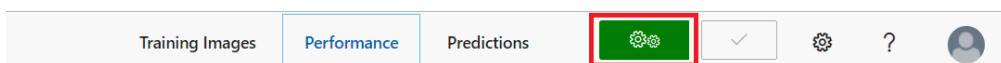
Upload 1000 files

添加完貓的照片後再用同樣的方式增加狗的照片. 可以點擊左上角的「+」按鈕繼續增加照片並附上標籤.



**動作:請從 dataset/cat-vs-dogs 資料夾中的 dog 資料夾選取 dog.0.jpg ~ dog.4.jpg 共五張圖片並加入專案**

上傳完貓跟狗的照片後，點選右上角的綠色按鈕開始訓練。訓練方式有兩種，這邊請先選擇 Quick training。  
註: advanced training是用來做更精準的辨認，但因為耗時較長且需要較多設定，這邊先行跳過。



## Choose Training Type

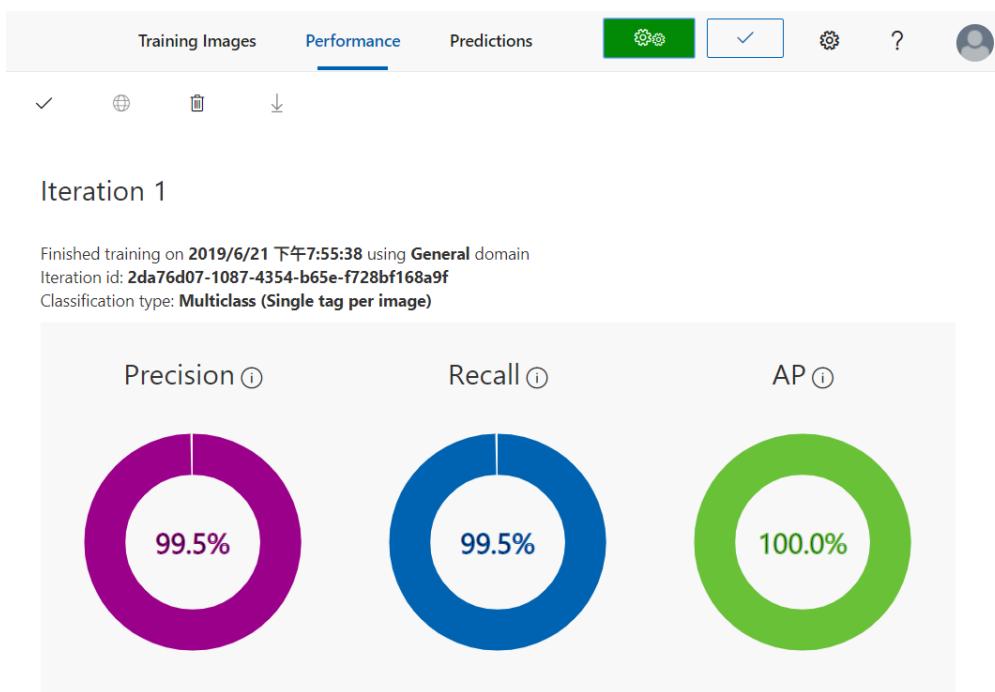


### Training Types (i)

- Quick Training
- Advanced Training

Train

開始訓練後請等待個幾分鐘(等待時間會依問題的複雜程度跟當下的網路狀態會有所不同,從一分鐘到十分鐘都是有可能的),訓練完畢後在Performance分頁可以看到模型分類的結果.



### Performance Per Tag

Tag	Precision	Recall	A.P.	Image count
cat	100.0%	99.0%	100.0%	1000
dog	99.0%	100.0%	100.0%	1000

在Performance頁面有三個屬性來顯示訓練的成果.

- Precision - 精確率, 意思是在所有被模型判斷成該物體的資料中真正正確的物件所佔的比例
- Recall - 召回率, 在所有該物件的資料中被模型判斷出來的比例
- AP - average precision的縮寫. ap 的評估是用在偵測(detection)問題上, 我們目前是處理分類問題, 因此這個數字沒有意義.

例如, 我有100張狗跟100張貓的照片, 其中有110張被機器判斷成狗, 這些被判斷成狗的照片裡面真正是狗的照片有90張, 那 precision 就是 90/110, 而 recall 是 90/100.

訓練完畢後, 你如果想要快速的確認一下訓練模型的好壞, 你可以選訓練按鈕右邊的勾勾(Quick Test)即時的測試模型的辨認結果.



在 Quick Test 頁面, 你有兩種選擇. 第一種是提供網路上的圖片的網址給 Image URL, Custom Vision 服務會去取得那張圖片後進行辨認. 另一種方法是選取自己電腦上的圖片(Browse local files)來測試辨認結果.

下方的 Predictions 會顯示模型預測的各種標籤的機率, 例如這邊範例中模型認為圖片中是狗的機率是99.9%.

Tag	Probability
dog	99.9%
cat	0%

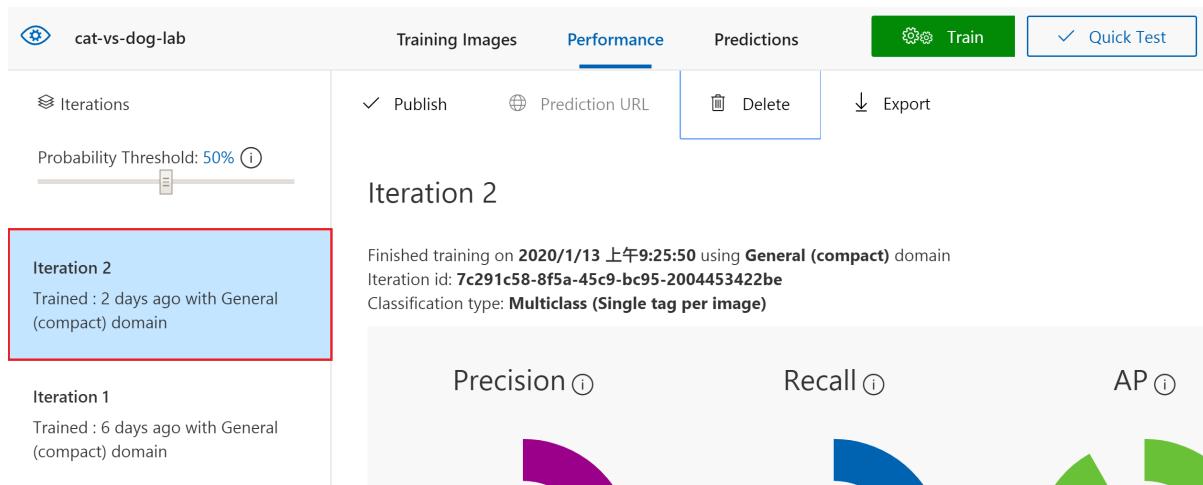
**動作:請從 dataset/cat-vs-dogs 資料夾中的 test 資料夾選取幾張圖片並觀察辨認結果**

到此你已經有能力可以訓練一個物件辨認的AI模型. 請試著根據以下的操作指示, 完成數個問題.

## 問題一：訓練影像的數量對辨認的結果會有影響嗎？

請完成下列操作並在學習單上填上你的實驗結果。

1. 把test資料夾中的所有測試圖片都用 Quick Test 測試一次。請問模型辨認正確的貓有幾張？狗又有幾張？
2. 請把誤判的多的那一個類別的圖片數量增加一倍(5張)。如果你的模型是貓誤判的多，請增加貓的圖片。反之，請增加狗的圖片(記得，請不要加入重複的圖片)
3. 重新訓練模型。訓練完畢後你會看到在Performance頁面的左邊會有出現一個 iteration 2，表示是第二次(用15張圖片)訓練的結果。



1. 再次用 Quick Test 測試test資料夾的所有圖片，看看那些圖片會機器會辨認錯。
2. 請增加訓練的圖片到有20張貓跟20張狗的影像(總共40張圖片)並再次訓練模型(iteration 3)
3. 機器看錯的比例是否有下降？(請觀察，同一張圖片機器認為是貓跟狗的機率的變化)

最後測試：請找一張機器辨認錯誤的圖片，試試看需要多少張的訓練圖片機器才會正確辨認那張圖。(請保持貓跟狗的訓練圖片數量一致)

提示：如果對模型的分類結果不滿意，你可以再加入更多的照片，或是調整資料集的內容後重新訓練。

## 問題二：如果圖片裡是別的東西，會發生什麼事？

1. 找一張裡面沒有貓也沒有狗的照片，機器判斷成什麼？
2. 根據上一個問題，討論可能的原因並思考要怎麼改善這個問題。

提示：在 Add images 的時候，你可以設定一個微軟提供的標籤叫做 negative。你覺得它的功能是什麼？

## 下載你的模型

如果你之前選擇的是 compact 模型，訓練完後會有 Export 的選項可以選擇，這邊可以讓我們把模型下載到本機端。



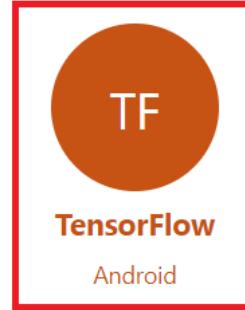
選 Export 後請選擇 Tensorflow

Choose your platform



CoreML

iOS 11



TensorFlow

Android



ONNX

Windows ML



Dockerfile

Azure IoT Edge, Azure Functions,  
AzureML



Vision AI Dev Kit ⓘ

在紅框位置先選Export, CustomVision就會開始打包模型, 打包完後就可以下載.

Choose your platform

X



Download

How to use?

[TensorFlow documentation](#)

[Sample code](#)

[Licenses](#)

下載完後的檔案會是一個壓縮檔, 解壓縮後裡面有兩個檔案 model.pb 跟 labels.txt. model.pb 檔舊式模型的參數, labels.txt 記錄了類別的名稱.

此時你已經可以把這麼檔案拿去別的程式中使用了.

## 總結

這一個單元你學會了如何操作 Custom Vision 訓練模型並下載模型

## 延伸閱讀：透過 Web API (網路應用程序) 呼叫使用你的模型

使用web api呼叫的方式的優點是你可以在任何應用(例如, 網頁或是手機App)都可以使用同樣的服務，而不需要一直到處複製模型檔案. 因此這邊示範一下如何使用web api呼叫的方式使用你訓練的模型.

首先, 在Custom Vision服務中, 你必須先發布你的模型, 這樣才能使用網路應用程序. 請在 Performance 頁面的下方點擊 Publish 後給它取一個名字.

Training Images    Performance    Predictions    Train    Quick Test

✓ Publish    Prediction URL    Delete    Export

Iteration 2

Publish name  X

Publish Cancel

要如何使用這個模型呢? Custom Vision 是設計成 Web api call 的方式, 也就是說微軟會提供一個網址(稱為 endpoint), 輸入這個網址, 把我們要測試的圖片丟給它, 就會傳回辨認的結果. 我們可以點擊箭頭指的地方來獲得 endpoint.

Training Images    Performance    Predictions    Train    ?    ✓

Iteration 1

Finished training on 2019/6/21 下午7:55:38 using General domain  
Iteration id: 2da76d07-1087-4354-b65e-f728bf168a9f  
Classification type: Multiclass (Single tag per image)

Precision ⓘ    Recall ⓘ    AP ⓘ

99.5%    99.5%    100.0%

#### Performance Per Tag

Tag	Precision	Recall	A.P.	Image count
cat	100.0%	99.0%	100.0%	1000
dog	99.0%	100.0%	100.0%	1000

點擊以後就會出現使用的 endpoint, 以及相關的參數設定.

## How to use the Prediction API



If you have an image URL:

```
https://southeastasia.api.cognitive.microsoft.com/customvision/v3.0/Prediction/ff01
```

Set `Prediction-Key` Header to: [REDACTED]

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

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

If you have an image file:

```
https://southeastasia.api.cognitive.microsoft.com/customvision/v3.0/Prediction/ff01
```

Set `Prediction-Key` Header to: [REDACTED]

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

Set Body to: <image file>

Got it!

大家可以把下面的程式區塊跟上面的圖對照一下，就可以知道設定的方式。因為不同的帳號，endpoint 跟 `Prediction-Key` 都不太一樣，因此讀者需要把下面的資料置換成自己帳號的設定的 endpoint 跟 `prediction-key`。

In [6]:

```
url = 'https://southeastasia.api.cognitive.microsoft.com/customvision/v3.0/Prediction/ff019354-7ae7-4d4f-bed7-214a3f58965b/classify/iterations/cats-vs-dog/image'  
headers = {  
    'Content-Type': 'application/octet-stream',  
    'Prediction-Key': '{YOUR_PREDICTION_KEY}',  
}
```

到此Web api算是設定完畢。在使用這個服務之前，我們要先準備好測試的圖片。如同之前的範例，我們從網路上找一些照片來測試。下面的程式碼可以從網路上的某個網址取得一張圖片(就好像在 Quick Test 中使用 Image URL 的方式一樣)

In [2]:

```
from PIL import Image
from io import BytesIO
import matplotlib.pyplot as plt
import requests

# you can get new images by replacing the string in get() function.
response = requests.get("https://www.petmd.com/sites/default/files/what-does-it-mean-when-
cat-wags-tail.jpg")

if not response.ok:
    print(response)
im_data = response.content

im = Image.open(BytesIO(im_data))
plt.imshow(im)
plt.axis('off')
plt.show()
```



取下來的資料我們放在 `im_data` 這個變數裡面, 請執行下面的程式片段呼叫 Custom Vision 服務並把 `im_data` 丟給 Custom Vision 服務. 成功的話會看到 *successfully get prediction result* 字樣.  
註:這邊你不需要對影像作任何處理(正規化, 縮放等), Custom Vision服務會自動幫你處理.

In [7]:

```
response = requests.post(url=url,data =im_data,headers=headers)

if response.ok:
    print('successfully get prediction result.')
```

successfully get prediction result.

回傳回來的預測資料會是 json 格式. 這是一個網路資料交換的標準格式. 你可以到這裡 (<https://zh.wikipedia.org/wiki/JSON>) 看到更多關於這種格式的資訊. 它的結構組成非常簡單, 是一種 key-value 的配對方式, 我們可以用下面的程式片段來把結果轉換成 python 支援的字典(dict)格式.

In [8]:

```
import json
j = json.loads(response.content)
print(json.dumps(j, indent=4))

{
    "id": "ab2a0942-5ec1-4982-a5de-ce9324191630",
    "project": "ff019354-7ae7-4d4f-bed7-214a3f58965b",
    "iteration": "2da76d07-1087-4354-b65e-f728bf168a9f",
    "created": "2019-06-21T13:11:23.007Z",
    "predictions": [
        {
            "probability": 1.0,
            "tagId": "61748b60-665e-48cf-bc16-d96936701fee",
            "tagName": "cat"
        },
        {
            "probability": 1.67265544e-14,
            "tagId": "8ce2410e-9f2d-462c-92cc-f7145771b62d",
            "tagName": "dog"
        }
    ]
}
```

直接觀察上面的列印結果，我們就會看到預測的結果被放在 predictions 下，每個類別都有一個預測的機率，所以我們單純的挑出機率最大的那個就是最後的預測結果。

In [10]:

```
prob = 0
tagName = 'None'
for obj in j['predictions']:
    if obj['probability'] > prob:
        tagName = obj['tagName']
        prob = obj['probability']

im = Image.open(BytesIO(im_data))
plt.imshow(im)
plt.title('Custom Vision says: {:.2f}% {}'.format(prob*100) + tagName)
plt.axis('off')
plt.show()
```

Custom Vision says: 100.00% cat

