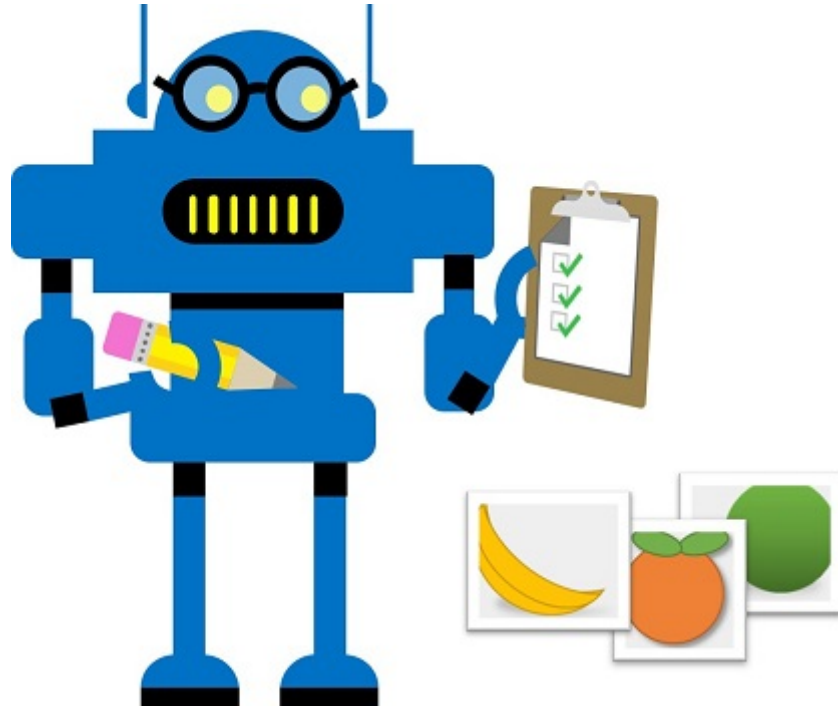


이미지 분류하기(Classification)

Computer Vision Cognitive 서비스는 이미지를 작업할 때 유용하게 사전 제작된 모델을 제공하지만 종종 컴퓨터 비전을 위한 자신만의 모델을 교육해야 한다. 예를 들어, Northwind Traders 소매 회사가 계산대에서 카메라로 찍은 이미지를 기반으로 고객이 구매하고자 하는 식료품 품목을 식별하는 자동 계산 시스템을 생성하려고 한다고 가정해 보자. 이렇게 하려면 이미지를 분류하여 구입하려는 항목을 식별할 수 있는 분류 모델을 교육해야 한다.



Azure에서 **Custom Vision** Cognitive 서비스를 사용하여 이미 존재하는 이미지를 기초로하여 이미지 분류모델을 학습시킬 수 있다. 이미지 분류 모델을 생성하기 위해서는 두가지 요소가 있다. 첫번째로 이미 존재하는 이미지를 사용하여 각기 다른 클래스로 인지하도록 모델을 학습시켜야 한다. 그리고 나서 모델이 학습하였다면 응용프로그램이 이용하도록 서비스를 게시해야 한다.

Custom Vision 리소스 만들기

Custom Vision 서비스를 사용하려면 모델을 *training* 하는 데 사용할 수 있는 Azure 리소스와 애플리케이션을 위해 모델을 게시할 수 있는 리소스가 필요하다. 두 작업 중 하나 또는 둘 다에 대한 리소스는 일반적인 **Cognitive 서비스** 리소스 또는 특별한 **CustomVision** 리소스가 될 수 있다. 이러한 각 작업에 대해 동일한 Cognitive 서비스 리소스를 사용하거나, 각 작업에 대해 서로 다른 리소스(동일한 지역)를 사용하여 비용을 별도로 관리할 수 있다.

새로운 **Custom Vision** 리소스를 만들기 위해서는 다음과 같은 순서로 수행하면 된다.

1. 새 브라우저 탭에서 <https://portal.azure.com> (<https://portal.azure.com>)에서 Azure 포털을 열고 Azure 구독과 연결된 Microsoft 계정을 사용하여 로그인한다.
2. + 리소스 만들기 버튼을 선택하고, *Custom Vision*을 검색한 뒤에 다음과 같이 **Custom Vision** 리소스를 생성한다:
 - 만들기 옵션: 둘 다
 - 구독: Azure 구독 옵션
 - 리소스 그룹: 유일한 이름으로 리소스 그룹 생성하기
 - 이름: 유일한 이름(영문과 숫자 적용)
 - 학습 위치: 가능한 위치 선택

- 교육 가격 책정 계층: F0
- 예측 위치: 학습위치와 동일한 지역 선택
- 예측 가격 책정위치: F0

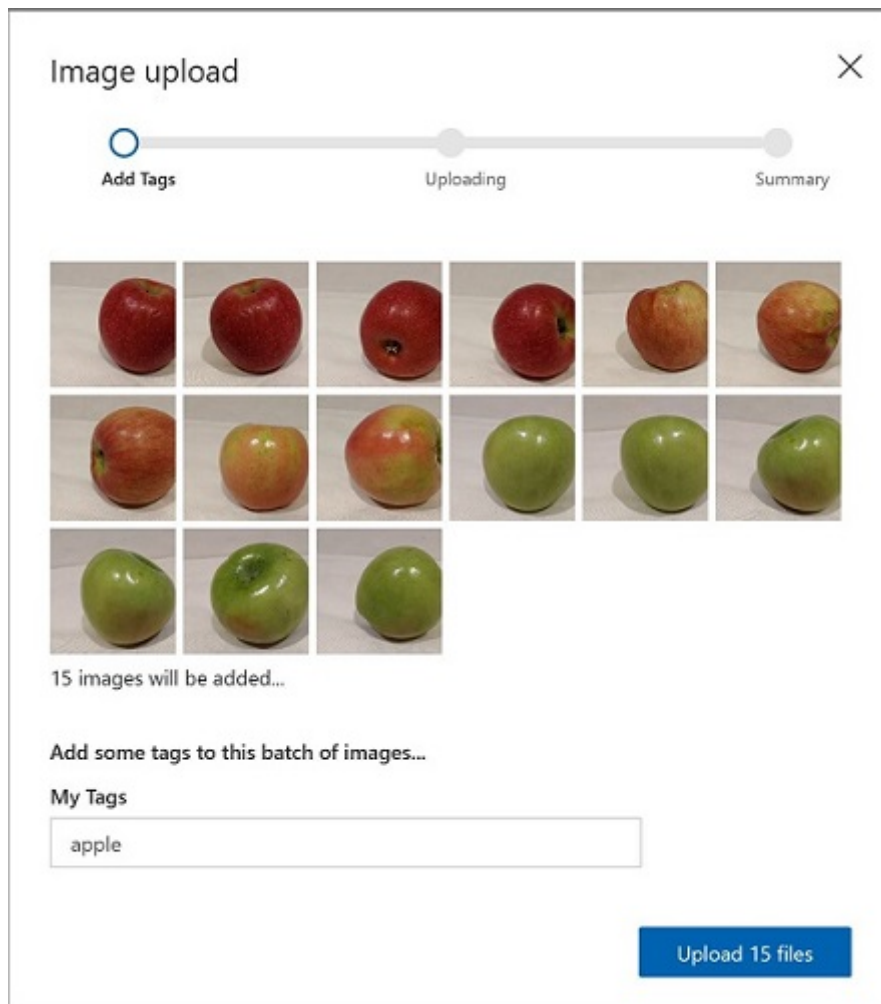
알림: 여러분들이 이미 F0 Custom Vision 서비스를 사용하고 있다면 여기서는 **S0**를 선택함.

3. 리소스가 생성되길 기다리고 두개의 Custom Vision 리소스가 생성 되어야 하는데 하나는 학습을 위해서이고 하나는 예측을 위해서이다. 이런 것들은 여러분이 생성한 리소스 그룹에 종속되어 보여진다.

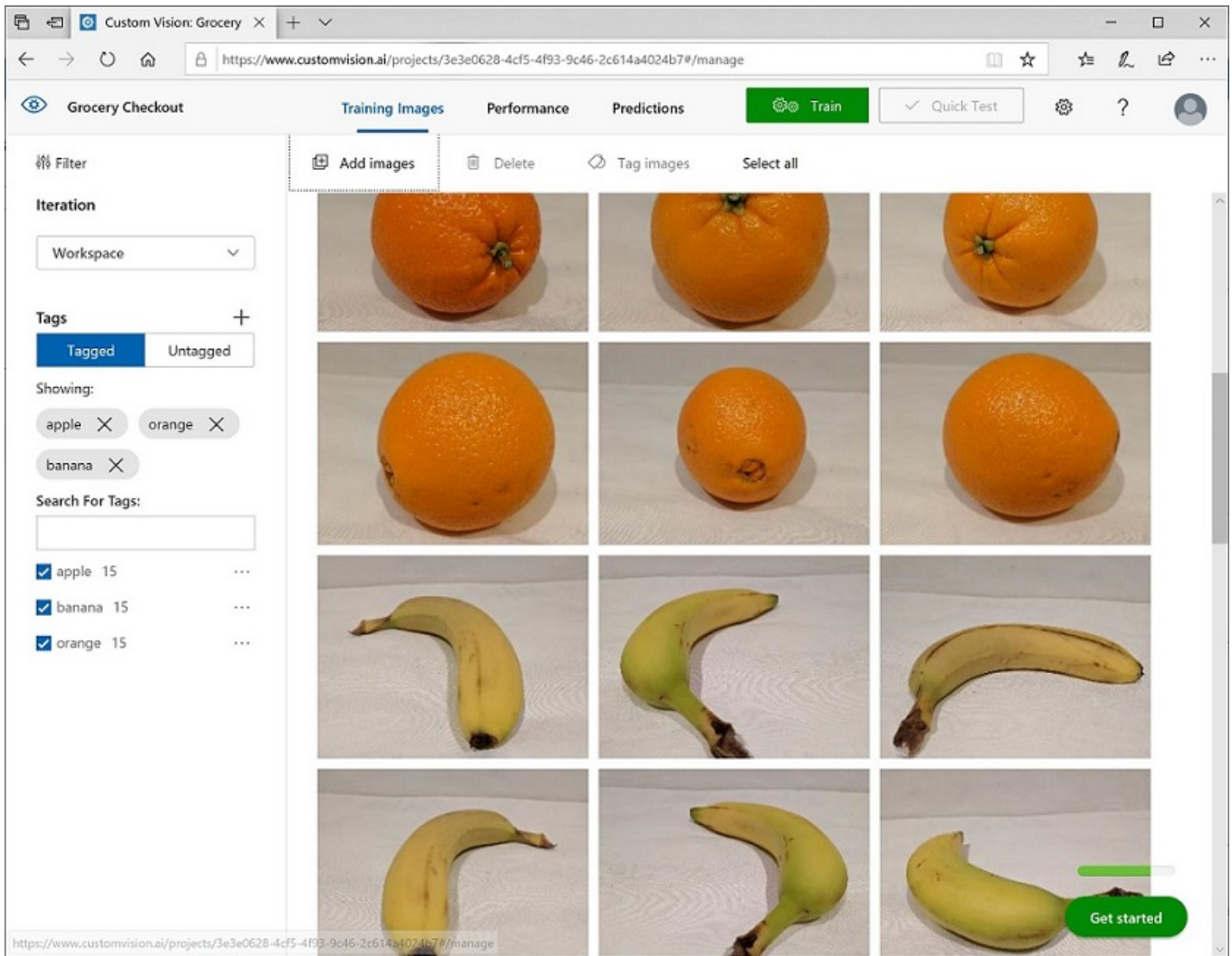
Custom Vision 프로젝트 만들기

객체 감지 모델을 학습시키기 위해 학습 리소스를 바탕으로 하여 Custom Vision 객체를 생성하여야 한다. 이렇게 하기 위해서는 Custom Vision 포털을 사용하게 될 것이다.

1. 학습 이미지 파일을 다운로드 하여 압축을 해제한다. (<https://aka.ms/fruit-images> (<https://aka.ms/fruit-images>)).
2. 브라우저의 다른 탭을 열고, Custom Vision portal 사이트 <https://customvision.ai> (<https://customvision.ai>)를 입력한다. 입력 창에 Azure 구독에 사용된 Microsoft 계정 정보를 입력하고 서비스 계약서에 동의한다..
3. Custom Vision 포털에서, 다음과 같은 내용으로 새로운 프로젝트를 입력한다.
 - **Name:** Grocery Checkout
 - **Description:** 채소에 대한 이미지 분류
 - **Resource:** 앞에서 만든 Custom Vision 리소스
 - **Project Types:** Classification
 - **Classification Types:** Multiclass (single tag per image)
 - **Domains:** Food
4. **[+] Add images**를 클릭하고 앞에서 압축을 해제한 **apple** 폴더에 있는 모든 파일을 선택한다. 이미지를 업로드하고 나서 이미지에 **apple**로 태그를 단다.



1. 앞에서와 같은 순서로 **banana** 폴더의 이미지에 태그를 *banana*로 달고 , **orange** 폴더에 있는 이미지들에 *orange*라는 태그를 부여한다.
2. Custom Vision 프로젝트 내에 업로드한 이미지들을 확인해본다-각 분야마다 15개의 이미지들이 존재한다.예를들면 다음과 같다.



1. Custom Vision 프로젝트에서 이미지들 위에 있는 **Train**을 클릭하여 태그가 된 이미지들 이용하여 분류 모델을 학습시킨다. **Quick Training** 옵션을 클릭하고 학습 순서가 끝나기를 기다린다(이 작업은 1분 이상 걸릴 수 있다).
2. 모델이 학습되었다면 *Precision*, *Recall*, 그리고 *AP* 성능 Metrics를 확인한다 - 이런 값들은 분류 모델의 정확도를 예측하는데 사용되며 모두 높게 나와야 좋다.

모델을 테스트하기

응용 프로그램에 사용하기 위해 게시하는 작업 전에 모델을 테스트해보아야 한다.

1. 성능 Metrics위에 있는 **Quick Test**를 클릭한다.
2. **Image URL** 상자에서, <https://aka.ms/apple-image> 라 입력하고 → 를 클릭한다.
3. 모델에 의해 예측된 결과를 확인한다-아래와 같이 **apple**에 대한 Probability 점수가 가장 높아야 한다:

Quick Test




Image URL

ka.ms/apple-image

or

Browse local files

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

Using model trained in

Iteration

Iteration 1

Predictions

Tag	Probability
apple	99.9%
orange	0%
banana	0%

1. **Quick Test** 창을 닫는다.

이미지 분류 모델을 게시하고 사용하기

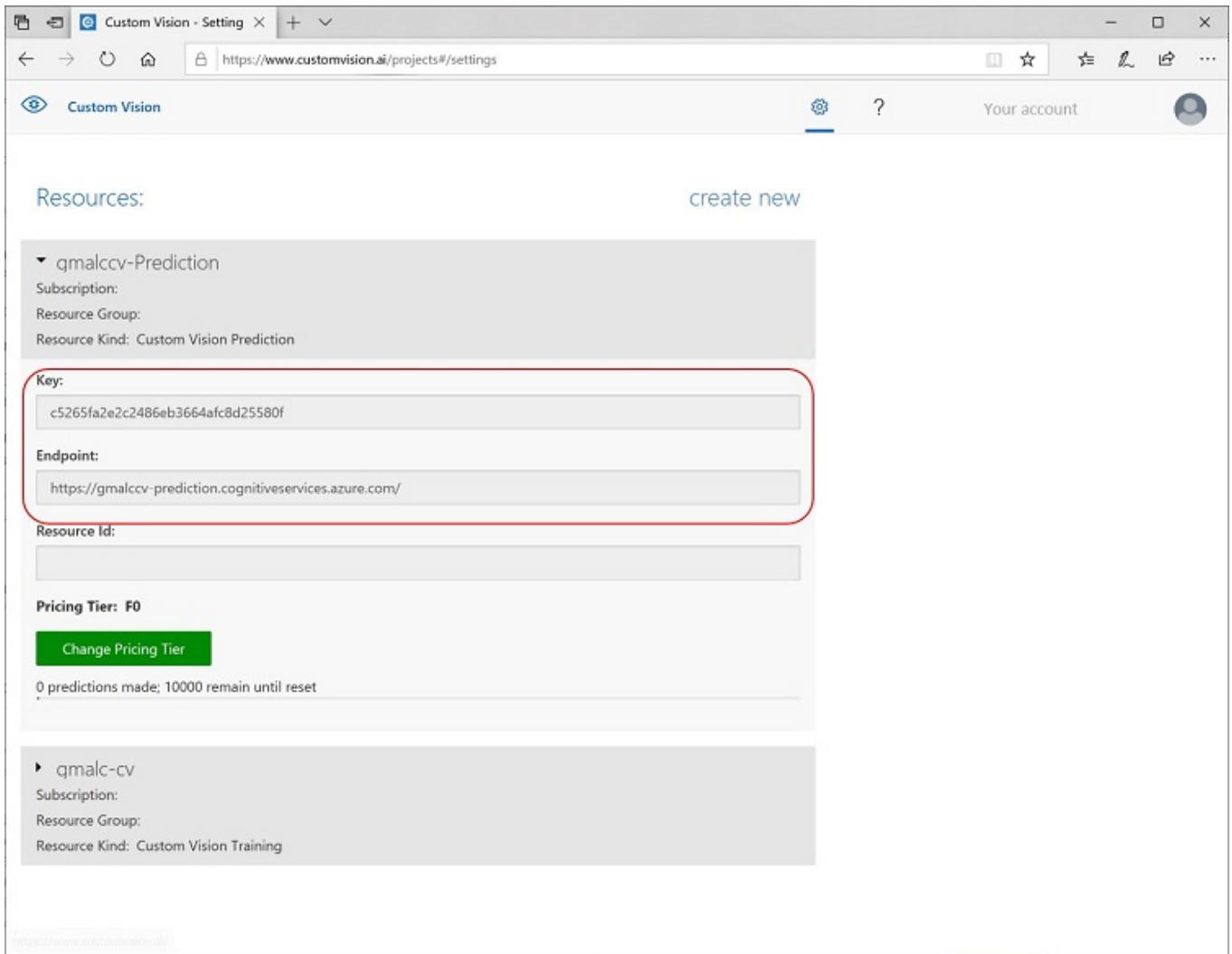
이제 학습시킨 모델을 게시하고 클라이언트 응용프로그램에서 사용할 수 있게 되었다.

1. 다음과 같은 내용을 입력하고 ✓ **Publish**를 클릭하여 학습시킨 모델을 게시한다.
 - **Model name:** groceries
 - **Prediction Resource:** 앞에서 생성한 예측 리소스.
2. 게시후에 프로젝트의 설정을 확인하기 위해 **Performance**페이지의 왼쪽 오른쪽에 있는 **settings** (⚙) 아이콘을 클릭한다. 그런 후 (왼쪽에 있는)**General** 에서, **Project Id**를 복사하여 아래의 코드셀에 붙여 넣는다 (**YOUR_PROJECT_ID**를 대체한다).

The screenshot displays the 'Project Settings' for a Custom Vision project named 'Grocery Checkout'. The 'Project Id' is highlighted with a red box. The 'Resources' section on the right provides details about the subscription and pricing tier.

_Note: 이 학습이 초기에 **Custom Vision** 리소스를 사용하는 대신 **Cognitive Services** 리소스를 사용했다면, 프로젝트 설정의 오른쪽에 있는 키와 엔드포인트를 복사해서 아래의 코드셀에 붙여 넣고 결과를 확인하기 위해서 실행한다. 그렇지 않으면, Custom Vision 예측 리소스에 대한 키와 엔드포인트를 얻기 위해 아래와 같은 작업을 완료한다.

1. **AProject Settings** 설정 페이지의 위쪽 왼쪽에 있는 **Projects Gallery** (👁) 아이콘을 클릭하여 Custom Vision 포털 홈페이지로 이동하면 프로젝트 목록이 나타난다.
2. Custom Vision 포털 홈페이지에서 위쪽 오른쪽에서 **settings** (⚙) 아이콘을 클릭하여 Custom Vision 서비스에 대한 설정을 본다. 그리고 나서 **Resources** 밑에 **prediction** 리소스를 클릭하고 resource (학습리소스가 아님) 키와 엔드포인트값을 복사해서 아래의 **YOUR_KEY** 와 **YOUR_ENDPOINT**를 대체한다.



1. 프로젝트 ID, 키, 엔드포인트값 변수 값을 설정한 후에 **Run cell** (▶) 버튼(셀의 왼쪽에 있음)을 클릭하여 실행합니다.

In []:

```
project_id = 'YOUR_PROJECT_ID'
cv_key = 'YOUR_KEY'
cv_endpoint = 'YOUR_ENDPOINT'

model_name = 'groceries' # 이 모델 이름은 모델 게시작업때 설정한 이름과 대소문자까지 일치해야 한다!
print('Ready to predict using model {} in project {}'.format(model_name, project_id))
```

Python에서 Custom Vision 서비스를 사용하기 위해서는 Azure Cognitive Services Custom Vision 패키지를 설치해야 한다.

In []:

```
!pip install azure-cognitiveservices-vision-customvision
```


이제 Custom Vision 클라이언트가 키와 엔드포인트를 사용하여 만든 Custom Vision 분류 모델에 연결할 수 있다.

게시한 모델을 사용하여 테스트 이미지를 잘 분석하는지 알아보기 위해 아래와같은 코드를 실행한다.

Note: 코드의 너무 자세한 내용에 대해서 걱정하지 말라. 이건 Computer Vision SDK for Python를 사용하여 /data/image-classification/test-fruit폴더에 있는 각이미지에 대한 분류 예측을 위한 것이다.

In []:

```
from azure.cognitiveservices.vision.customvision.prediction import CustomVisionPredictionClient
from msrest.authentication import ApiKeyCredentials
import matplotlib.pyplot as plt
from PIL import Image
import os
%matplotlib inline

# data/vision/test 폴더에 있는 이미지 들을 가져온다.
test_folder = os.path.join('data', 'image-classification', 'test-fruit')
test_images = os.listdir(test_folder)

# 예측 서비스의 인스턴스를 생성한다.
credentials = ApiKeyCredentials(in_headers={"Prediction-key": cv_key})
custom_vision_client = CustomVisionPredictionClient(endpoint=cv_endpoint, credentials=credentials)

# 결과를 화면에 표시하도록 그림을 생성한다.
fig = plt.figure(figsize=(16, 8))

# 이미지를 가져와서 각 이미지에 대한 예측 분류값을 보여준다.
print('Classifying images in {} ...'.format(test_folder))
for i in range(len(test_images)):
    # 이미지를 열고 Custom Vision 모델을 사용하여 분류하도록 준비한다.
    image_contents = open(os.path.join(test_folder, test_images[i]), "rb")
    classification = custom_vision_client.classify_image(project_id, model_name, image_contents.read())
    # 예측 값은 각 이미지 태그에 대한 예측값을 포함하고 확률을 내림차순으로 정리해서 보여준다.
    prediction = classification.predictions[0].tag_name
    # 예측 클래스가 포함된 이미지를 나타낸다.
    img = Image.open(os.path.join(test_folder, test_images[i]))
    a=fig.add_subplot(len(test_images)/3, 3,i+1)
    a.axis('off')
    imgplot = plt.imshow(img)
    a.set_title(prediction)
plt.show()
```

다행히 여러분의 이미지 분류 모델은 정확하게 이미지 있는 채소를 바르게 찾아 낸다.

추가 학습

Custom Vision 서비스는 이 학습에서 본 것들 보다 더 능률적인 것으로 보여준다. 예를 들면 Custom Vision 서비스를 사용하여 객체 감지 모델을 생성하는데 Custom Vision 모델을 사용할 수 있다(이것은 이미지 내에 있는 객체들을 구분할 뿐만 아니라 이미지에 포함되어 있는 개체의 위치를 보여주는 튀어오르는 상자를 구분할 수 있다).

Custom Vision cognitive 서비스에 대해서 더 많은 것을 알고 싶다면 , [Custom Vision documentation](https://docs.microsoft.com/azure/cognitive-services/custom-vision-service/home) (<https://docs.microsoft.com/azure/cognitive-services/custom-vision-service/home>)를 참조하면 된다.