

```
In [1]: # Custom vision
from azure.cognitiveservices.vision.customvision.training import CustomVisionTrainingClient
from azure.cognitiveservices.vision.customvision.prediction import CustomVisionPredictionClient
from msrest.authentication import ApiKeyCredentials
```

```
In [2]: prediction_key='7846d19dd63f40f491f77e0b64dbca12'
resource_id='/subscriptions/54c4256e-bb50-4fbd-895d-da32982a5dad/resourceGroups/ashish/providers/Microsoft.Cogni
```

```
In [3]: endpoint='https://iiiisafaf.cognitiveservices.azure.com/'
```

```
In [10]: check=ApiKeyCredentials(in_headers={'Prediction-key':prediction_key})
prediction_check=CustomVisionPredictionClient(endpoint,check)
```

```
In [18]: publish_iteration_name='Iteration1'
# Testing
project_id="384cf96e-afb9-4683-b4da-83c2eeda1dc2"
with open(r"C:\Users\Jaswanth Reddy\Desktop\Image dataset\api_fruit_vegetable\vegetable\0.jpg", 'rb') as images:
    result=prediction_check.classify_image(project_id,publish_iteration_name,images.read())

for a in result.predictions:
    #print(a.tag_name)
    #print(a.probability)
    print('name: ',a.tag_name,'probability :',a.probability)
```

```
name: vegetables probability : 0.99999999
name: fruits probability : 1.6457321e-07
```

```
In [ ]: from azure.cognitiveservices.vision.computervision import CustomVisionClient
```

```
In [22]: import os
import requests
import matplotlib.pyplot as plt
from io import BytesIO
from PIL import Image
```

```
In [24]: subscription_key="2c70fd24d8794a218d4a5d9cf1007f35"
```

```
In [26]: endpoint="https://wersf.cognitiveservices.azure.com/"
analyze_image=endpoint+"vision/v3.1/analyze?%s"
```

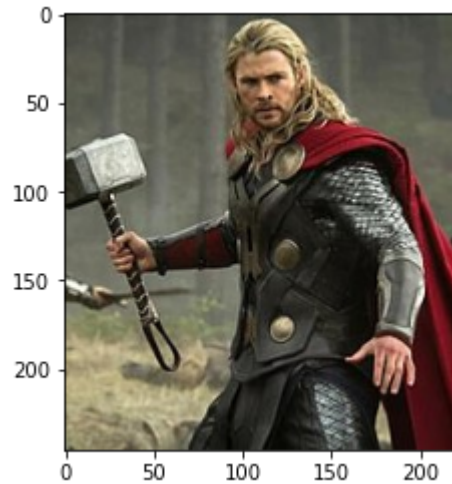
```
In [25]: image_url="https://upload.wikimedia.org/wikipedia/en/thumb/3/3c/Chris_Hemsworth_as_Thor.jpg/220px-Chris_Hemsworth.jpg"
headers={'Ocp-Apim-Subscription-Key':subscription_key}
params={'visualFeatures':'Categories,Description,Color'}
data={'url':image_url}
```

```
In [27]: response=requests.post(analyze_image,headers=headers,params=params,json=data)
analysis=response.json()
print(analysis)
```

```
{'categories': [{'name': 'others_', 'score': 0.00390625}, {'name': 'people_', 'score': 0.63671875, 'detail': {'celebrities': [{'name': 'Chris Hemsworth', 'confidence': 0.8439303040504456, 'faceRectangle': {'left': 95, 'top': 27, 'width': 36, 'height': 36}}]}], 'color': {'dominantColorForeground': 'Black', 'dominantColorBackground': 'Grey', 'dominantColors': ['Grey', 'Black'], 'accentColor': '9E2D3B', 'isBwImg': False, 'isBWImg': False}, 'description': {'tags': ['person', 'outdoor'], 'captions': [{'text': 'Chris Hemsworth in a garment holding a gun', 'confidence': 0.4867195188999176}]}, 'requestId': '31fdf8d4-ce19-4c26-b1b1-8ea3f961e226', 'metadata': {'height': 246, 'width': 220, 'format': 'Jpeg'}}
```

```
In [33]: img=Image.open(BytesIO(requests.get(image_url).content))  
plt.imshow(img)
```

```
Out[33]: <matplotlib.image.AxesImage at 0x21653277940>
```



```
In [34]: description=analysis['description']  
print(description)
```

```
{'tags': ['person', 'outdoor'], 'captions': [{'text': 'Chris Hemsworth in a garment holding a gun', 'confidence': 0.4867195188999176}]}
```

In [35]: *# thumbnail*

```
headers={'Ocp-Apim-Subscription-Key':subscription_key}
params={'width':'50','height':'50','smartCropping':'true'}
data={'url':image_url}
```

In [37]: `analyze_image=endpoint+"vision/v3.1/generateThumbnail"`
`response=requests.post(analyze_image,headers=headers,params=params,json=data)`

In [38]: `result=Image.open(BytesIO(response.content))`

In [39]: `print(result)`

```
<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=50x50 at 0x21652A7FEE0>
```

In [40]: `result.save("thor.png")`
`result.show()`

In [56]: *# Local image*

```
analyze_image=endpoint+"vision/v3.1/analyze"
image_url=r"C:\Users\Jaswanth Reddy\Pictures\Images downloaded\black_widow.jpeg"
image_converted=open(image_url,'rb').read()

headers={'Ocp-Apim-Subscription-Key':subscription_key,
          'Content-Type':'application/octet-stream'}
params={'visualFeatures':'Categories,Description,Color'}
```

In [59]:

```
response=requests.post(analyze_image,headers=headers,params=params,data=image_converted)
analysis=response.json()
image = Image.open(BytesIO(image_converted))
plt.imshow(image)
```

Out[59]: <matplotlib.image.AxesImage at 0x2165435f670>



In []: