Deep Azure - Final Project

Computer Vision API

Full Report – Shailesh Beri

Feb 10, 2018

Table of Contents

Introduction

Finding actionable information from images, generating captions, identifying objects in images and interpreting text in images has been an important breakthrough in Cognitive services

Azure cloud-based Computer Vision API provides developers with access to advanced algorithms for processing images and returning analysis and description of the images

With the Computer Vision API users can analyze images to:

- Tag images based on content.
- Categorize images.
- Identify the type and quality of images.
- Detect human faces and return their coordinates.
- Recognize domain-specific content.
- Generate descriptions of the content.
- Use optical character recognition to identify printed text found in images.
- Recognize handwritten text.
- Distinguish color schemes.
- Flag adult content.
- Crop photos to be used as thumbnails.

Requirements

- Supported input methods: Raw image binary in the form of an application/octet stream or image URL.
- Supported image formats: JPEG, PNG, GIF, BMP.
- Image file size: Less than 4 MB.
- Image dimension: Greater than 50 x 50 pixels

By uploading an image or specifying an image URL, Microsoft Computer Vision algorithms can analyze visual content in different ways based on inputs and user choices

Various Applications in this space range from 1) Identification of faces or celebrities in images, 2) Blocking images with adult content, 3) Text in Image recognition, 4) Optical character recognition (OCR)

This project is focused on using two Computer Vision API capabilities 1) Image Analysis and 2) Image Description. Intent is to use skeletal API and modify/customize code to create reusable functions, test various image types, analyze the results and perform qualitative analysis of the API performance

Problem Statement

Need a Computer Vision Reusable Feature which performs Computer Image Analysis, identifies objects in the image and Describes these objects in a single function call.

Image Analysis and Description capabilities shall include Tagging, Categorization, Image Types, Face Recognition, Perceived color scheme, Accent Color and if the image is Black and White

Description of Data, URL(s) and Sources of Data sets

Data Description

Data used for this project is various computer images found on the internet. To support the key objective of qualitative analysis of the Computer Vision API a variety of images were obtained from the Internet and were tested to verify results

Data Set

URL(s) of various images from the internet (See below) were used for testing

https://ausopen.com/sites/default/files/201801/28/o_federer_f_rla_28012018_42.jpg

https://ausopen.com/sites/default/files/201801/28/o_federer_f_rla_28012018_35.jpg

https://cdn.cnn.com/cnnnext/dam/assets/180202172405-01-week-in-politics-0204-restricted-super-169.jpg

https://oxfordportal.blob.core.windows.net/vision/Analysis/3.jpg

https://c1.staticflickr.com/7/6013/5918998899_3051a519f6_b.jpg

https://travel.usnews.com/statictravel/images/destinations/44/statue_and_skyline_getty_triggerphoto.jpg

https://www.gettyimages.com/detail/photo/new-york-city-aerial-skyline-at-dusk-usa-royalty-free-image/494545485?esource=SEO_GIS_CDN_Redirect

https://upload.wikimedia.org/wikipedia/commons/c/c8/Taj_Mahal_in_March_2004.jpg

http://www3.pictures.zimbio.com/gi/Celebrities+At+The+Lakers+Game+ZEcLGLE6U2Ll.jpg

http://www.stogieboys.com/media/690x380-Hawaii-Sunset.jpg

https://static.nascar.com/content/dam/nascar/articles/2016/9/14/main/rules-main2.jpg/jcr:content/renditions/original

https://photos.smugmug.com/02Sports-2/Warriors/Warriors-stun-Thunder-in-Game-/i-4DDRNhm/0/a0e14f53/L/SJM-WARRIORS-0529-148-L.jpg

https://i.redd.it/p7hwbrx35bwy.jpg

http://www.iboxphotography.co.uk/wp-content/siteuploads/2015/07/London-Bridge-Black-and-White.jpg

https://thumbs.gfycat.com/MeanIlliterateBlackbear-max-1mb.gif

Description of Hardware

Local Machines

- Windows 10 Home; Dell Laptop
- Windows 7 Professional; Dell Laptop

Azure Cloud Services

- Computer Vision API created in “West US” region

Description of Software

Programming Language and Platform

Programming Language – Python 3.6

Platform – Anaconda – Jupyter

Azure Computer Vision API(s)

API(s) used for project

Image Analysis (URL)

```
_url = 'https://{}.api.cognitive.microsoft.com/vision/v1.0/analyze'.format(_region)
```

Image Description (URL)

```
_urldes = 'https://{}.api.cognitive.microsoft.com/vision/v1.0/describe?%s'.format(_region)
```

Computer Vison Features supported by Analysis and Description API(s)

These API(s) perform these Image function – Categorization of objects in Image, Color details including dominant colors, if the image is Black and White only, Image Metadata, Tagging, Description of action in image with confidence level, Facial recognition (Celebrities)

Common Variables for both API(s)

```
# Common Variables
_region = 'westus' #Here you enter the region of  subscription
_key = 'a6fd23a9ba584b4892780e337bb8ad66' #Here paste primary key for Computer Vision API
_maxNumRetries = 10
```

GitHub Public Folder

Project Artifacts including code was uploaded to this Public GitHub Repository

https://github.com/shaileshberi/Azure-Computer-Vision

PIP

PIP install opencv-python

Python Packages used and their relevance

Packages used for Image Analysis

```
import time
import requests
import cv2
import operator
import numpy as np
from __future__ import print_function
```

Additional Packages required for Image Description

```
import http.client, urllib.request, urllib.parse, urllib.error, base64
```

Package to display results in Jupyter Notebook during Run

```
import matplotlib.pyplot as plt
%matplotlib inline
```

Additional Computer Vision API(s)

- OCR (Optical Character Recognition)
- Recognize Handwritten Texts
- Generate Thumbnails
- Flagging Adult Content

Installation & Configuration steps

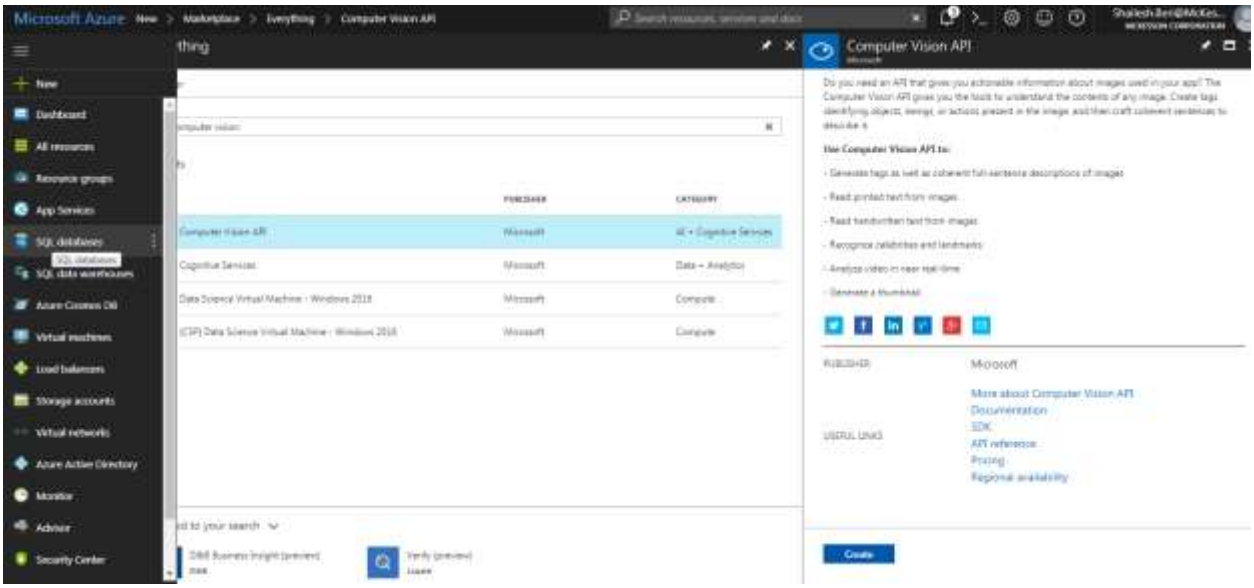
Computer Vision API

Create new resource group “SB_FinalProject” in “WestUS” Region

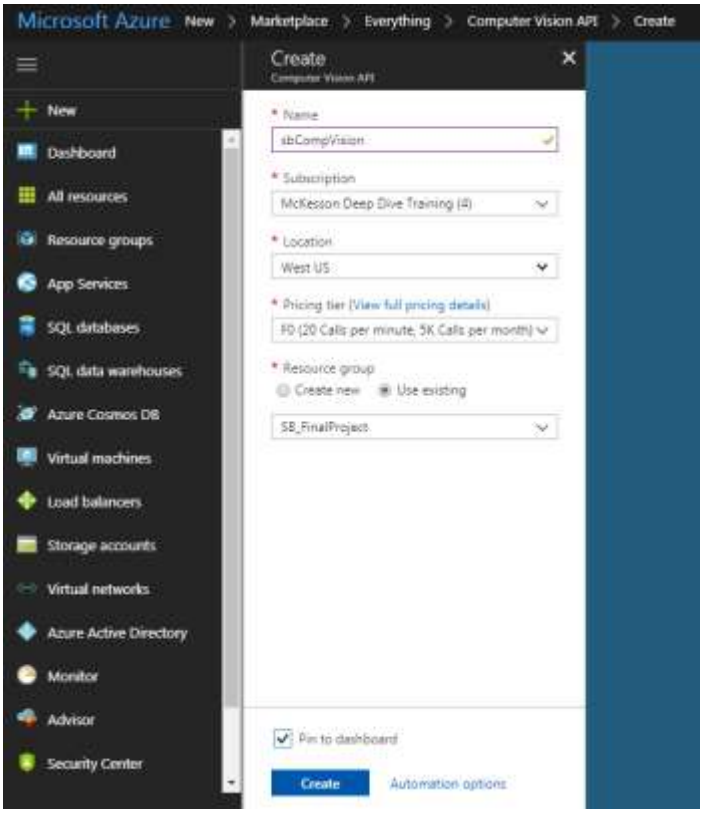
Significance of region – Computer Vision API is available only in the following US regions

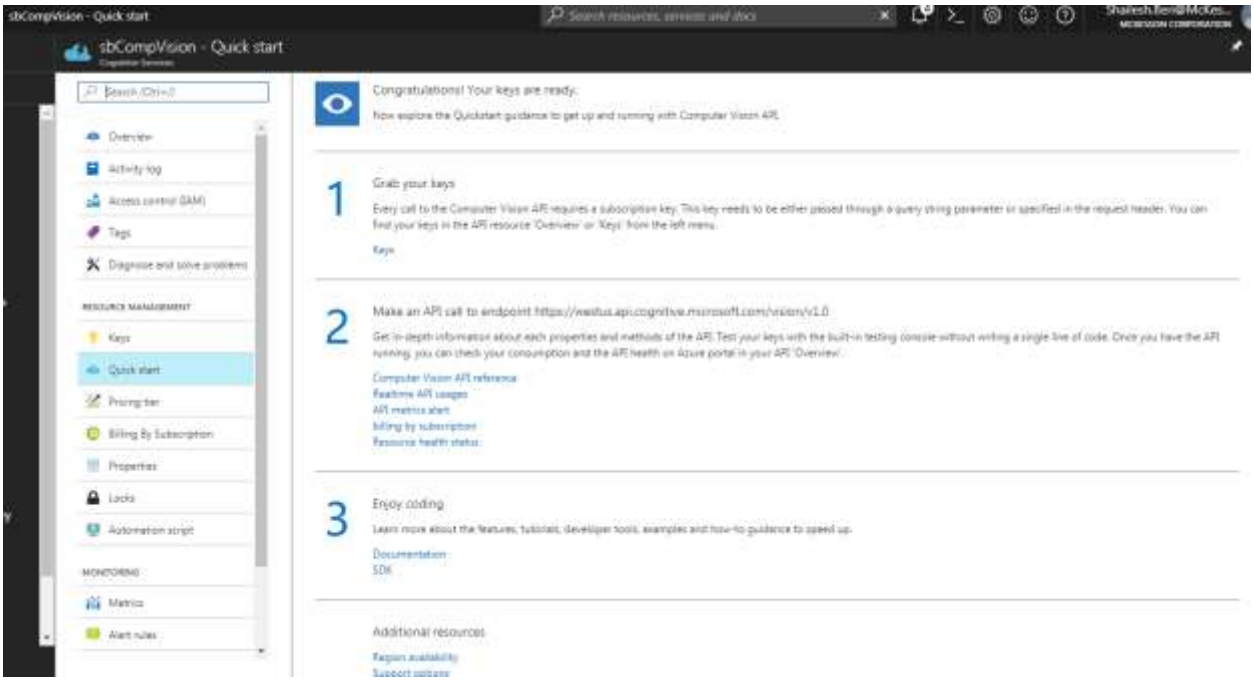
- West US westus.api.cognitive.microsoft.com
- East US 2 eastus2.api.cognitive.microsoft.com
- West Central US westcentralus.api.cognitive.microsoft.com

Sign-Up for Computer Vision API

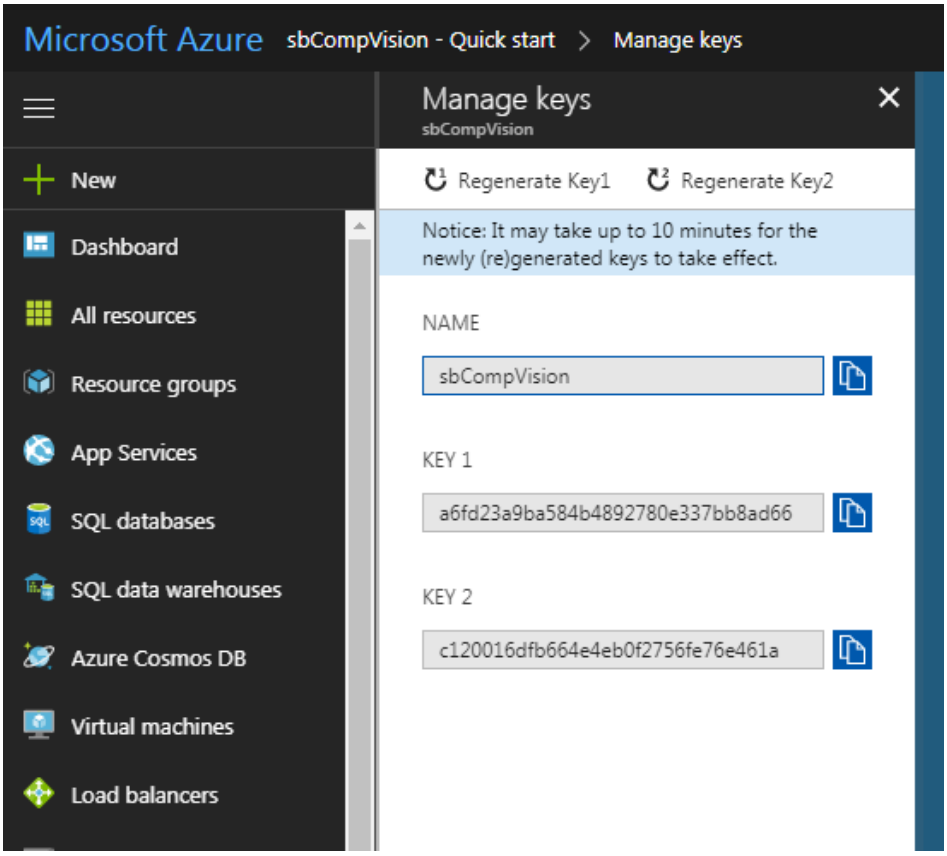


Hit Create and populate the required fields (Make sure to create the API in a region it is supported)

on 



Grab the keys for Computer Vision API



sbCompVision

a6fd23a9ba584b4892780e337bb8ad66 (Key 1)

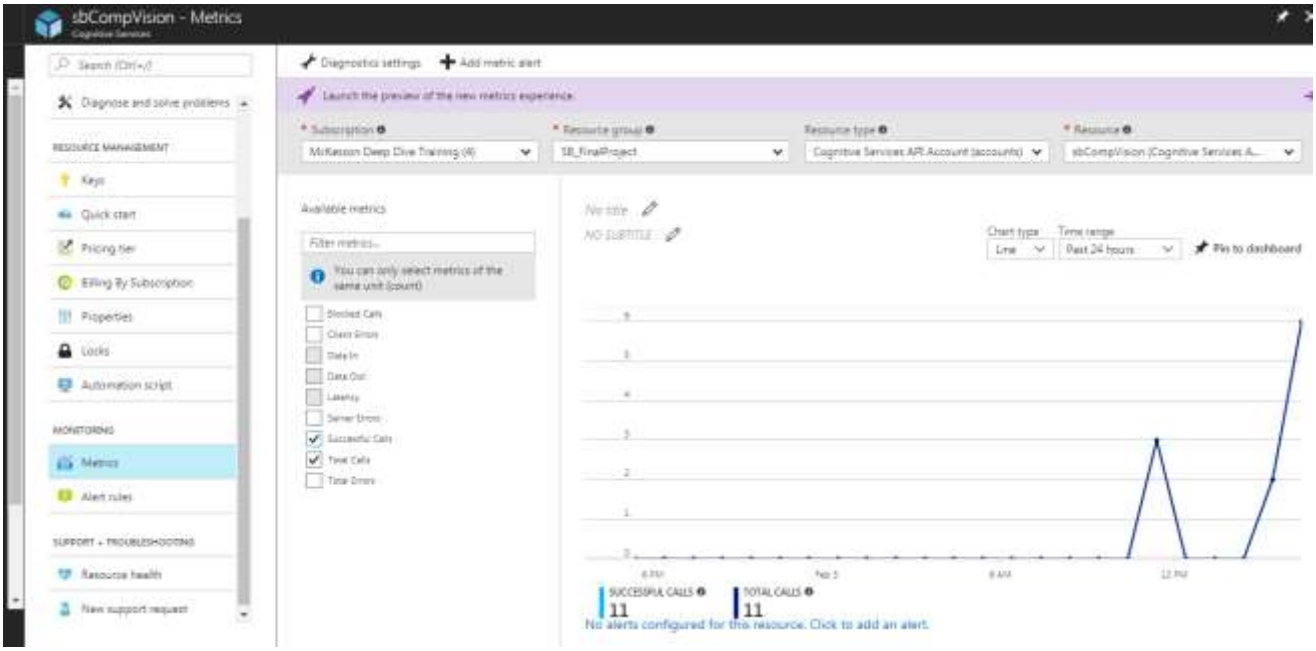
c120016dfb664e4eb0f2756fe76e461a (Key 2)

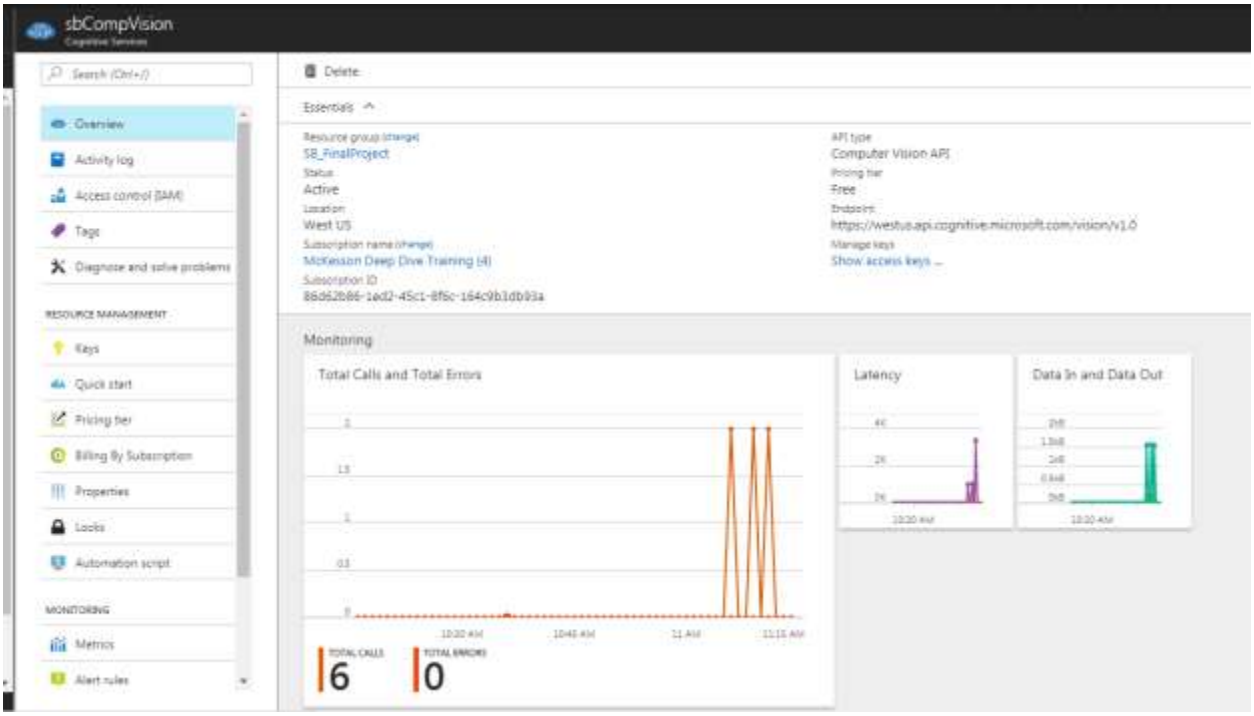
PIP Installs

PIP install opencv-python /* To support CV2 package */

```
C:\Users\eiexxt8>pip install opencv-python
Collecting opencv-python
  Downloading opencv_python-3.4.0.12-cp36-cp36m-win_amd64.whl (33.3MB)
    100% |################################| 33.4MB 729kB/s
Requirement already satisfied: numpy>=1.11.3 in c:\users\eiexxt8\appdata\local\c
ontinuum\anaconda3\lib\site-packages (from opencv-python)
Installing collected packages: opencv-python
Successfully installed opencv-python-3.4.0.12

C:\Users\eiexxt8>
```

Monitor API Calls using Metrics for the Computer Vision API and checking the overview





Local Machine Run

Using Python (Jupyter Notebook) call the API(s) by passing image URL(s) to process and receive back Image Analysis and Image Objects Description

Azure function Option

The Computer Vision Feature can be deployed as Azure Function in the FaaS model with option to use one or more Computer Vision API(s)

Source Code

```
# This Python skeletal API program for Image Analyis has been downloaded from GitHub
https://github.com/Microsoft/Cognitive-Vision-Python and

# Modifed for Final Project of Deep Azure Training by customizing and organizing the code and adding Image Description API
capability - Shailesh Beri# Image Processing - Image Analysis and Image Description

# Image File URL is input and the image is processed

# Supported image formats: JPEG, PNG, GIF, BMP

# Image file size: Less than 4MB

# Image dimension: Greater than 50 x 50 pixels


# Image Analysis

import time

import requests

import cv2

import operator

import numpy as np

from __future__ import print_function


# Image Description

import http.client, urllib.request, urllib.parse, urllib.error, base64


# Import library to display results

import matplotlib.pyplot as plt

%matplotlib inline

# Display images within Jupyter


# Text Recognition



# Common Variables


_region = 'westus' #Here you enter the region of subscription

_key = 'a6fd23a9ba584b4892780e337bb8ad66' #Here paste primary key for Computer Vision API

_maxNumRetries = 10



# Image Analysis URL
```

```
_url = 'https://{}.api.cognitive.microsoft.com/vision/v1.0/analyze'.format(_region)

# Image Description URL

_urldes = 'https://{}.api.cognitive.microsoft.com/vision/v1.0/describe?%s'.format(_region)


_urlt = 'https://westcentralus.api.cognitive.microsoft.com/vision/v1.0/RecognizeText'


# Helper functions

# Function to process Image file to analyze image

def processRequestImage( json, data, headers, params ):

    """
    Helper function to process the request to Project Oxford

    Parameters:
    json: Used when processing images from its URL. See API Documentation
    data: Used when processing image read from disk. See API Documentation
    headers: Used to pass the key information and the data type request
    """

    retries = 0
    result = None

    while True:

        # Debug
        # import pdb; pdb.set_trace()
        # Debug

        response = requests.request( 'post', _url, json = json, data = data, headers = headers, params = params )
```

```
        if response.status_code == 429:

            print( "Message: %s" % ( response.json() ) )

            if retries <= _maxNumRetries:
                time.sleep(1)
                retries += 1
                continue
            else:
                print( 'Error: failed after retrying!' )
                break

        elif response.status_code == 200 or response.status_code == 201:

            if 'content-length' in response.headers and int(response.headers['content-length']) == 0:
                result = None
            elif 'content-type' in response.headers and isinstance(response.headers['content-type'], str):
                if 'application/json' in response.headers['content-type'].lower():
                    result = response.json() if response.content else None
                elif 'image' in response.headers['content-type'].lower():
                    result = response.content
        else:
            print( "Error code: %d" % ( response.status_code ) )
            print( "Message: %s" % ( response.json() ) )

        break

    return result



# Function to process Image file to describe image

def processRequestImageDescription( json, data, headers, params ):

    """
    Helper function to process the request to Project Oxford
```

```
    Parameters:
    json: Used when processing images from its URL. See API Documentation
    data: Used when processing image read from disk. See API Documentation
    headers: Used to pass the key information and the data type request
    """

    retries = 0
    resultDes = None

    while True:

        # Debug
        # import pdb; pdb.set_trace()
        # Debug

        response = requests.request( 'post', _urldes, json = json, data = data, headers = headers, params = params )

        if response.status_code == 429:

            print( "Message: %s" % ( response.json() ) )

            if retries <= _maxNumRetries:
                time.sleep(1)
                retries += 1
                continue
            else:
                print( 'Error: failed after retrying!' )
                break

        elif response.status_code == 200 or response.status_code == 201:

            if 'content-length' in response.headers and int(response.headers['content-length']) == 0:
                resultDes = None
            elif 'content-type' in response.headers and isinstance(response.headers['content-type'], str):
                if 'application/json' in response.headers['content-type'].lower():
                    resultDes = response.json() if response.content else None
```

```
            elif 'image' in response.headers['content-type'].lower():
                resultDes = response.content
        else:
            print( "Error code: %d" % ( response.status_code ) )
            print( "Message: %s" % ( response.json() ) )

        break

    return resultDes

# Function to render results on image

def renderResultOnImage( result, img ):

    """Display the obtained results onto the input image"""

    R = int(result['color']['accentColor'][:2],16)
    G = int(result['color']['accentColor'][2:4],16)
    B = int(result['color']['accentColor'][4:],16)

    cv2.rectangle( img,(0,0), (img.shape[1], img.shape[0]), color = (R,G,B), thickness = 25 )

    if 'categories' in result:
        categoryName = sorted(result['categories'], key=lambda x: x['score'])[0]['name']
        cv2.putText( img, categoryName, (30,70), cv2.FONT_HERSHEY_SIMPLEX, 2, (255,0,0), 3 )

# Begin ProcessImage Function

# Analysis of an image retrieved via URL

# URL direction to image


def ProcessImage():

    # Image Analysis and Description
```

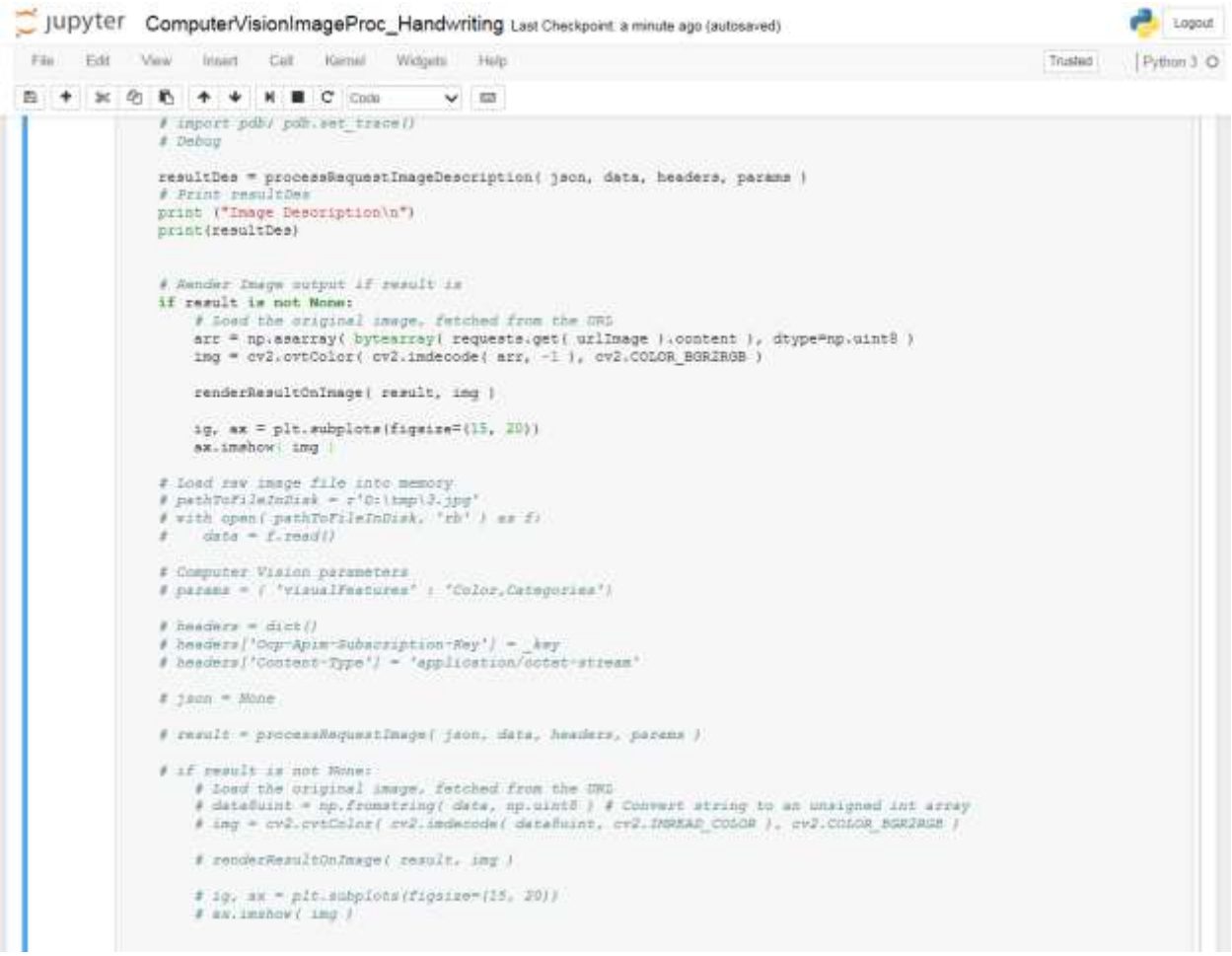
```
# Add code to get Image URL from console or Text File

#Sample URL

# urlImage = 'https://oxfordportal.blob.core.windows.net/vision/Analysis/3.jpg'
urlImage = 'https://ausopen.com/sites/default/files/201801/28/o_federer_f_rla_28012018_42.jpg'
# urlImage = 'https://ausopen.com/sites/default/files/201801/28/o_federer_f_rla_28012018_35.jpg'

# Get Image URL from console input

urlImage = input ("Enter URL of Image : ")

# import pdb; pdb.set_trace()

# Computer Vision Image Analysis parameters and headers

params = { 'visualFeatures' : 'Color,Categories'}
headers = dict()
headers['Ocp-Apim-Subscription-Key'] = _key
headers['Content-Type'] = 'application/json'
json = { 'url': urlImage }
data = None


# Debug
# import pdb; pdb.set_trace()
# Debug

result = processRequestImage( json, data, headers, params )
# Print result
print ("Image Analysis\n")
print(result)
print ("\n")

# Computer Vision Image Description parameters and headers

params = urllib.parse.urlencode({
```

```
# Request parameters
'maxCandidates': '10',
})

# headers = dict()
headers['Ocp-Apim-Subscription-Key'] = _key
headers['Content-Type'] = 'application/json'
json = { 'url': urlImage }
data = None



# Debug
# import pdb; pdb.set_trace()
# Debug

resultDes = processRequestImageDescription( json, data, headers, params )
# Print resultDes
print ("Image Description\n")
print(resultDes)



# Render Image output if result is
if result is not None:
    # Load the original image, fetched from the URL
    arr = np.asarray( bytearray( requests.get( urlImage ).content ), dtype=np.uint8 )
    img = cv2.cvtColor( cv2.imdecode( arr, -1 ), cv2.COLOR_BGR2RGB )

    renderResultOnImage( result, img )

    ig, ax = plt.subplots(figsize=(15, 20))
    ax.imshow( img )

# Load raw image file into memory
# pathToFileInDisk = r'D:\tmp\3.jpg'
# with open( pathToFileInDisk, 'rb' ) as f:
#    data = f.read()
```

```
    # Computer Vision parameters
    # params = { 'visualFeatures' : 'Color,Categories'}

    # headers = dict()
    # headers['Ocp-Apim-Subscription-Key'] = _key
    # headers['Content-Type'] = 'application/octet-stream'

    # json = None

    # result = processRequestImage( json, data, headers, params )

    # if result is not None:
        # Load the original image, fetched from the URL
        # data8uint = np.fromstring( data, np.uint8 ) # Convert string to an unsigned int array
        # img = cv2.cvtColor( cv2.imdecode( data8uint, cv2.IMREAD_COLOR ), cv2.COLOR_BGR2RGB )

        # renderResultOnImage( result, img )

        # ig, ax = plt.subplots(figsize=(15, 20))
        # ax.imshow( img )


# End ProcessImage Function

# main function

if __name__ == '__main__':
    print ( "Processing Azure Cognitive Services Computer Vision with  SDK for Python" )
    ProcessImage()
    print (" Image Processing Completed")
```

Compile and Run code.

In Jupyter notebook keep cursor in the cell containing code and hit execute



Enter URL of Image when prompted by the program

```
Processing Azure Cognitive Services Computer Vision with  SDK for Python
Enter URL of Image : https://photos.smugmug.com/025sports-2/Warriors/Warriors-stun-Thunder-in-Game-/i-4DDRNhm/0/a0e14f
53/L/SJM-WARRIORS-0529-148-L.jpg
```

Results and Visualization

```
Image Analysis

{'categories': [{'name': 'people_crowd', 'score': 0.546875}], 'color': {'dominantColorForeground': 'Black', 'dominant
ColorBackground': 'Black', 'dominantColors': ['Black', 'Brown'], 'accentColor': '9F642C', 'isBwImg': False}, 'request
Id': 'd73383a5-7d4e-462c-bf0d-c5ad92fb7031', 'metadata': {'height': 534, 'width': 800, 'format': 'Jpeg'}}


Image Description

{'description': {'tags': ['person', 'sport', 'game', 'basketball', 'court', 'player', 'playing', 'ball', 'man', 'grou
p', 'holding', 'standing', 'female', 'young', 'people', 'walking', 'wearing', 'woman', 'crowd', 'soccer', 'street'],
'captions': [{'text': 'a group of men playing a game of basketball', 'confidence': 0.840567729389681}, {'text': 'a gr
oup of people playing a game of basketball', 'confidence': 0.839567729389681}, {'text': 'a group of people playing ba
sketball on a court', 'confidence': 0.838567729389681}, {'text': 'a group of young men playing a game of basketball',
'confidence': 0.670308292944782}, {'text': 'a group of people playing a basketball game', 'confidence': 0.66930829294
4782}, {'text': 'a group of men playing a basketball game', 'confidence': 0.668308292944782}, {'text': 'a group of me
n playing a game of basketball on a court', 'confidence': 0.667308292944782}, {'text': 'a group of people standing on
a basketball court', 'confidence': 0.666308292944782}, {'text': 'a group of people on a basketball court', 'confidenc
e': 0.665308292944782}, {'text': 'a group of basketball players on the court', 'confidence': 0.664308292944782}]}, 'r
equestId': '60f5cd50-1e85-4026-b65d-434d6c17e211', 'metadata': {'height': 534, 'width': 800, 'format': 'Jpeg'}}
 Image Processing Completed
```

Data for Testing and Testing Results

Cleansing, Upload and Transformations of Data

The images from sources were used as it exists on the sites and URL

Example data Image (s) for Testing, Image URL and Testing Results

Test Case 1

https://ausopen.com/sites/default/files/201801/28/o_federer_f_rla_28012018_35.jpg

```
Processing Azure Congnitive Services Computer Vision with  SDK for Python
Image Analysis

{'categories': [{'name': 'sky_object', 'score': 0.74609375}, {'name': 'others_', 'scor
e': 0.00390625}, {'name': 'outdoor_', 'score': 0.00390625}], 'color': {'dominantColorF
oreground': 'Blue', 'dominantColorBackground': 'Blue', 'dominantColors': ['Blue'], 'ac
centColor': '0D48A1', 'isBwImg': False}, 'requestId': 'a9bcf992-45eb-4000-8c0c-185f7ba
18654', 'metadata': {'height': 900, 'width': 1600, 'format': 'Jpeg'}}
> <ipython-input-1-71b1f0959973>(237)ProcessImage()
-> resultDes = processRequestImageDescription( json, data, headers, params )
(Pdb) c
Image Description

{'description': {'tags': ['tennis', 'ball', 'person', 'man', 'sport', 'game', 'racket'
, 'hitting', 'water', 'player', 'swinging', 'male', 'grass', 'court', 'blue', 'playing
', 'ready', 'shirt', 'holding', 'standing', 'air'], 'captions': [{'text': 'a male tenn
is player swinging a racket at a ball', 'confidence': 0.8933657818480079}, {'text': 'a
tennis player swinging a racket at a ball', 'confidence': 0.8923657818480079}, {'text'
: 'a man hitting a tennis ball', 'confidence': 0.8913657818480079}, {'text': 'a man hi
tting a ball with a tennis racket', 'confidence': 0.8903657818480079}, {'text': 'a man
hitting a tennis ball with a racket', 'confidence': 0.8893657818480079}, {'text': 'a m
ale tennis player hitting a ball with a racket', 'confidence': 0.8883657818480079}, {'
text': 'a man is hitting a ball with a tennis racket', 'confidence': 0.887365781848007
9}, {'text': 'a man swinging a racket at a tennis ball', 'confidence': 0.8863657818480
078}, {'text': 'a tennis player hitting a ball with a racket', 'confidence': 0.8853657
818480078}, {'text': 'a man hitting a tennis ball with his racket', 'confidence': 0.88
43657818480078}]}, 'requestId': '9476d855-86a6-4791-8431-19d055da81ac', 'metadata': {'
height': 900, 'width': 1600, 'format': 'Jpeg'}}
 Image Processing Completed
```

Test Case 2

https://ausopen.com/sites/default/files/201801/28/o_federer_f_rla_28012018_42.jpg

```
Processing Azure Congnitive Services Computer Vision with  SDK for Python
Image Analysis

{'categories': [{'name': 'people_', 'score': 0.7890625}], 'color': {'dominantColorFore
ground': 'Blue', 'dominantColorBackground': 'Black', 'dominantColors': ['Black', 'Blue
'], 'accentColor': '0E52B4', 'isBwImg': False}, 'requestId': '87d3da77-5bda-4f04-b3cf-
44d18aa306e8', 'metadata': {'height': 900, 'width': 1600, 'format': 'Jpeg'}}


> <ipython-input-1-8295f3e46e17>(238)ProcessImage()
-> resultDes = processRequestImageDescription( json, data, headers, params )
(Pdb) c
Image Description

{'description': {'tags': ['sport', 'game', 'person', 'tennis', 'racket', 'player', 'ma
n', 'holding', 'ball', 'swinging', 'hitting', 'male', 'hand', 'playing', 'ready', 'air
', 'hat', 'close', 'wearing', 'standing', 'young', 'blue', 'court', 'white'], 'caption
s': [{'text': 'Roger Federer holding a tennis racket', 'confidence': 0.932675936017759
3}, {'text': 'a close up of Roger Federer holding a tennis racket', 'confidence': 0.91
16359215386556}, {'text': 'a close up of Roger Federer swinging a tennis racket', 'con
fidence': 0.8552327782299691}, {'text': 'close up of Roger Federer holding a tennis ra
cket', 'confidence': 0.8542327782299691}, {'text': 'Roger Federer swinging a tennis ra
cket', 'confidence': 0.8532327782299691}, {'text': 'Roger Federer holding a tennis rac
ket at a ball', 'confidence': 0.8522327782299691}, {'text': 'a tennis player swinging
a racket at a ball', 'confidence': 0.8512327782299691}, {'text': 'Roger Federer holdin
g a tennis racket in his hand', 'confidence': 0.8502327782299691}, {'text': 'a close u
p of Roger Federer hitting a ball with a tennis racket', 'confidence': 0.8149999226292
449}, {'text': 'a close up of Roger Federer with a tennis racket', 'confidence': 0.813
9999226292449}]}, 'requestId': 'f40130b7-3697-4587-aaf4-d3bed97a4c84', 'metadata': {'h
eight': 900, 'width': 1600, 'format': 'Jpeg'}}
 Image Processing Completed
```

Test Case 3

https://cdn.cnn.com/cnnnext/dam/assets/180202172405-01-week-in-politics-0204-restricted-super-169.jpg

```
Processing Azure Congnitive Services Computer Vision with  SDK for Python

Enter URL of Image : https://cdn.cnn.com/cnnnext/dam/assets/180202172405-01-week-in-politics-0204-restricted-super-169.jpg

Image Analysis

{'categories': [{'name': 'people_group', 'score': 0.27734375}], 'color': {'dominantColorForeground': 'Black', 'dominantColorBackground': 'Grey', 'dominantColors': ['Black', 'Grey'], 'accentColor': '6B2726', 'isBwImg': False}, 'requestId': 'be2a686a-13e2-4af4-95bb-172a0c0a2ec2', 'metadata': {'height': 619, 'width': 1100, 'format': 'Jpeg'}}

Image Description

{'description': {'tags': ['person', 'building', 'suit', 'man', 'table', 'business', 'standing', 'dressed', 'woman', 'sitting', 'wearing', 'people', 'holding', 'talking', 'group', 'red', 'wedding'], 'captions': [{'text': 'Mike Pence, Donald J. Trump are posing for a picture', 'confidence': 0.9797977818008676}, {'text': 'Mike Pence, Donald J. Trump are posing for a picture', 'confidence': 0.9787977818008676}, {'text': 'Mike Pence, Donald J. Trump are posing for a picture', 'confidence': 0.8332103240349907}, {'text': 'Mike Pence, Donald J. Trump are posing for a picture', 'confidence': 0.8322103240349907}, {'text': 'Mike Pence, Donald J. Trump are posing for a picture', 'confidence': 0.8312103240349907}, {'text': 'Mike Pence, Donald J. Trump are posing for a picture', 'confidence': 0.8302103240349907}, {'text': 'Mike Pence, Donald J. Trump are posing for a picture', 'confidence': 0.8292103240349907}, {'text': 'Mike Pence, Donald J. Trump are posing for a picture', 'confidence': 0.8282103240349907}, {'text': 'Mike Pence, Donald J. Trump are posing for a picture', 'confidence': 0.8272103240349907}, {'text': 'Mike Pence, Donald J. Trump in a suit and tie', 'confidence': 0.8262103240349907}]}, 'requestId': '05b940ed-f58f-4743-a5e9-7fcb953a2b7b', 'metadata': {'height': 619, 'width': 1100, 'format': 'Jpeg'}}

 Image Processing Completed
```

Test Case 4

https://oxfordportal.blob.core.windows.net/vision/Analysis/3.jpg

```
Processing Azure Congnitive Services Computer Vision with  SDK for Python

Enter URL of Image : https://oxfordportal.blob.core.windows.net/vision/Analysis/3.jpg

Image Analysis

{'categories': [{'name': 'trans_trainstation', 'score': 0.98828125}], 'color': {'domin
antColorForeground': 'Black', 'dominantColorBackground': 'Black', 'dominantColors': ['
Black'], 'accentColor': '484B83', 'isBwImg': False}, 'requestId': 'adecffec-1307-4657-
81b9-cbeff8bd9729', 'metadata': {'height': 1155, 'width': 1500, 'format': 'Jpeg'}}

Image Description

{'description': {'tags': ['train', 'platform', 'building', 'station', 'track', 'walkin
g', 'subway', 'board', 'pulling', 'holding', 'people', 'man', 'standing', 'waiting', '
luggage', 'woman', 'umbrella'], 'captions': [{'text': 'a person waiting for a train at
a train station', 'confidence': 0.6548414048450634}, {'text': 'a person walking next t
o a train station', 'confidence': 0.6538414048450634}, {'text': 'a person standing nex
t to a train station', 'confidence': 0.6528414048450634}, {'text': 'people walking nea
r a train station', 'confidence': 0.6518414048450634}, {'text': 'a subway train at a t
rain station', 'confidence': 0.5726738774749787}, {'text': 'a train pulling into a sta
tion', 'confidence': 0.5435602682475842}, {'text': 'a person walking past a train stat
ion', 'confidence': 0.5425602682475842}, {'text': 'a couple of people that are standin
g in a train station', 'confidence': 0.49921281808046075}, {'text': 'a close up of a t
rain station', 'confidence': 0.49821281808046075}, {'text': 'a person waiting for a tr
ain at a station', 'confidence': 0.49721281808046075}]}, 'requestId': '4270f058-7a94-4
1a6-a65d-cee9eb38ca36', 'metadata': {'height': 1155, 'width': 1500, 'format': 'Jpeg'}}

 Image Processing Completed
```

Test Case 5

https://c1.staticflickr.com/7/6013/5918998899_3051a519f6_b.jpg

```
Processing Azure Congnitive Services Computer Vision with  SDK for Python

Enter URL of Image : https://c1.staticflickr.com/7/6013/5918998899_3051a519f6_b.jpg

Image Analysis

{'categories': [{'name': 'building_', 'score': 0.75390625}, {'name': 'outdoor_', 'scor
e': 0.1015625}], 'color': {'dominantColorForeground': 'Blue', 'dominantColorBackground
': 'Blue', 'dominantColors': ['Blue'], 'accentColor': '0039CC', 'isBwImg': False}, 're
questId': 'a95bfd91-e291-4b8e-817c-3f35fc9207df', 'metadata': {'height': 680, 'width':
1024, 'format': 'Jpeg'}}

Image Description

{'description': {'tags': ['water', 'outdoor', 'building', 'bridge', 'large', 'boat', '
river', 'blue', 'front', 'big', 'clock', 'sitting', 'city', 'top', 'tower', 'harbor',
'cloudy', 'yellow', 'standing'], 'captions': [{'text': 'a large bridge over some water
', 'confidence': 0.7079410837096002}, {'text': 'a large bridge over some water in fron
t of a building', 'confidence': 0.6366183495493438}, {'text': 'a large bridge over a b
ody of water', 'confidence': 0.6356183495493438}, {'text': 'a bridge over water with a
city in the background', 'confidence': 0.6346183495493438}, {'text': 'a large bridge o
ver water', 'confidence': 0.6336183495493438}, {'text': 'a large bridge over a river',
'confidence': 0.6326183495493438}, {'text': 'a large bridge over water with a city in
the background', 'confidence': 0.6316183495493438}, {'text': 'a bridge over a body of
water', 'confidence': 0.6306183495493438}, {'text': 'a bridge over water in front of a
building', 'confidence': 0.6296183495493438}, {'text': 'a large bridge over a river in
a city', 'confidence': 0.6286183495493438}]}, 'requestId': '1d207559-4084-41cb-b5f7-88
5b481009a2', 'metadata': {'height': 680, 'width': 1024, 'format': 'Jpeg'}}

 Image Processing Completed
```

Test Case 6

https://travel.usnews.com/static-travel/images/destinations/44/statue_and_skyline_getty_triggerphoto.jpg

```
Processing Azure Congnitive Services Computer Vision with  SDK for Python

Enter URL of Image : https://travel.usnews.com/static-travel/images/destinations/44/statue_and_skyline_getty_triggerphoto.jpg

Image Analysis

{'categories': [{'name': 'building_', 'score': 0.19140625}, {'name': 'outdoor_', 'score': 0.05859375}, {'name': 'outdoor_city', 'score': 0.39453125}], 'color': {'dominantColorForeground': 'White', 'dominantColorBackground': 'White', 'dominantColors': ['White'], 'accentColor': '2B80A0', 'isBwImg': False}, 'requestId': '5780e135-8fda-4eef-bf93-d5d18997b8f8', 'metadata': {'height': 385, 'width': 577, 'format': 'Jpeg'}}


Image Description

{'description': {'tags': ['water', 'outdoor', 'building', 'city', 'background', 'ship', 'river', 'large', 'boat', 'top', 'sitting', 'front', 'harbor', 'lake', 'riding', 'body', 'green', 'blue', 'ocean', 'floating', 'table', 'man', 'island', 'dock', 'bird', 'tall', 'tower', 'bridge', 'white', 'clock', 'standing', 'flying'], 'captions': [{'text': 'a large body of water with a city in the background', 'confidence': 0.9641173768312217}, {'text': 'a body of water with a city in the background', 'confidence': 0.9528259234838581}, {'text': 'a large ship in a body of water with a city in the background', 'confidence': 0.8401263389031705}, {'text': 'a building next to a body of water with a city in the background', 'confidence': 0.8391263389031705}, {'text': 'a ship in a body of water with a city in the background', 'confidence': 0.8381263389031705}, {'text': 'a bridge over a body of water with a city in the background', 'confidence': 0.8055174881481382}, {'text': 'a large body of water with city buildings in the background', 'confidence': 0.8045174881481382}, {'text': 'a large body of water and a city in the background', 'confidence': 0.8035174881481382}, {'text': 'a large ship in the water with a city in the background', 'confidence': 0.8025174881481382}, {'text': 'water next to a body of water with a city in the background', 'confidence': 0.8015174881481382}]}, 'requestId': '1721430e-20d7-4fc8-9ef2-d2189db889e6', 'metadata': {'height': 385, 'width': 577, 'format': 'Jpeg'}}


---------------------------------------------------------------------------
error                                     Traceback (most recent call last)
<ipython-input-1-e93d090e00a6> in <module>()
    289 if __name__ == '__main__':
    290     print ( "Processing Azure Congnitive Services Computer Vision with  SDK for Python" )
--> 291     ProcessImage()
    292     print (" Image Processing Completed")
    293

<ipython-input-1-e93d090e00a6> in ProcessImage()
    249         # Load the original image, fetched from the URL
    250         arr = np.asarray( bytearray( requests.get( urlImage ).content ), dtype=np.uint8 )
--> 251         img = cv2.cvtColor( cv2.imdecode( arr, -1 ), cv2.COLOR_BGR2RGB )
    252
    253         renderResultOnImage( result, img )
```

```
error: C:\projects\opencv-python\opencv\modules\imgproc\src\color.cpp:11079: error: (-
215) scn == 3 || scn == 4 in function cv::cvtColor
```

Test Case 7

https://www.gettyimages.com/detail/photo/new-york-city-aerial-skyline-at-dusk-usa-royalty-free-image/494545485?esource=SEO_GIS_CDN_Redirect

```
Processing Azure Cognitive Services Computer Vision with  SDK for Python

Enter URL of Image : https://www.gettyimages.com/detail/photo/new-york-city-aerial-skyline-at-dusk-usa-royalty-free-image/494545485?esource=SEO_GIS_CDN_Redirect

Error code: 400

Message: {'code': 'InvalidImageUrl', 'requestId': '786a3ff6-5abb-477b-a4b5-88823ea0a5e5', 'message': 'Image URL is not accessible.'}

Image Analysis


None


Error code: 400

Message: {'code': 'InvalidImageUrl', 'requestId': 'b70a8bbe-69fb-449d-b374-2dd85cf4c435', 'message': 'Image URL is not accessible.'}

Image Description


None

 Image Processing Completed
```

Test Case 8

https://upload.wikimedia.org/wikipedia/commons/c/c8/Taj_Mahal_in_March_2004.jpg

```
Processing Azure Cognitive Services Computer Vision with  SDK for Python

Enter URL of Image : https://upload.wikimedia.org/wikipedia/commons/c/c8/Taj_Mahal_in_March_2004.jpg

Image Analysis

{'categories': [{'name': 'building_', 'score': 0.80078125}, {'name': 'outdoor_', 'score': 0.01953125}], 'color': {'dominantColorForeground': 'Grey', 'dominantColorBackground': 'Blue', 'dominantColors': ['Grey', 'White'], 'accentColor': '2C5D9F', 'isBwImg': False}, 'requestId': '0638636c-c8a2-4ed8-9ec8-011ee4399ff2', 'metadata': {'height': 1681, 'width': 2040, 'format': 'Jpeg'}}

Image Description

{'description': {'tags': ['outdoor', 'grass', 'building', 'church', 'park', 'bench', 'standing', 'front', 'people', 'white', 'sidewalk', 'group', 'large', 'sitting', 'walking', 'stone', 'old', 'man', 'water', 'green', 'field', 'red', 'city', 'parked', 'train', 'riding', 'street'], 'captions': [{'text': 'a group of people standing in front of a building', 'confidence': 0.9199449513416805}, {'text': 'a group of people standing in front of a church', 'confidence': 0.907419318789255}, {'text': 'a group of people standing in front of a white building', 'confidence': 0.888499562395904}, {'text': 'a group of people standing in front of a large building', 'confidence': 0.887499562395904}, {'text': 'a group of people walking in front of a building', 'confidence': 0.886499562395904}, {'text': 'a group of people in front of a building', 'confidence': 0.885499562395904}, {'text': 'a group of people standing in front of a large church', 'confidence': 0.8757014410155697}, {'text': 'a group of people standing in front of a large white building', 'confidence': 0.8747014410155697}, {'text': 'a group of people walking in front of a church', 'confidence': 0.8737014410155697}, {'text': 'a group of people that are standing in front of a building', 'confidence': 0.8572352657242961}]}, 'requestId': '059bca1f-b80b-4792-94b7-e77de5b144f0', 'metadata': {'height': 1681, 'width': 2040, 'format': 'Jpeg'}}

 Image Processing Completed
```

Test Case 9

http://www3.pictures.zimbio.com/gi/Celebrities+At+The+Lakers+Game+ZEcLGLE6U2Ll.jpg

```
Processing Azure Cognitive Services Computer Vision with  SDK for Python

Enter URL of Image : http://www3.pictures.zimbio.com/gi/Celebrities+At+The+Lakers+Game
+ZEcLGLE6U2Ll.jpg

Image Analysis

{'categories': [{'name': 'people_group', 'score': 0.73046875}], 'color': {'dominantCol
orForeground': 'Black', 'dominantColorBackground': 'Black', 'dominantColors': ['Black'
], 'accentColor': '46415D', 'isBwImg': False}, 'requestId': 'b167774e-3d73-4f40-9a9b-6
dced4b27807', 'metadata': {'height': 475, 'width': 594, 'format': 'Jpeg'}}

Image Description

{'description': {'tags': ['person', 'sitting', 'man', 'people', 'indoor', 'table', 'gr
oup', 'woman', 'holding', 'cake', 'older', 'smiling', 'cutting', 'bench', 'large', 'cr
owd', 'standing', 'young', 'food', 'riding', 'player'], 'captions': [{'text': 'Gordon
Ramsay, David Beckham are posing for a picture', 'confidence': 0.9211064233040244}, {'
text': 'Gordon Ramsay, David Beckham sitting in front of a crowd', 'confidence': 0.912
8235329641989}, {'text': 'Gordon Ramsay, David Beckham sitting together smiling for th
e camera', 'confidence': 0.8726999662871983}, {'text': 'Gordon Ramsay, David Beckham a
re posing for a picture', 'confidence': 0.8716999662871983}, {'text': 'Gordon Ramsay,
David Beckham sitting at a table', 'confidence': 0.8577580085806669}, {'text': 'Gordon
Ramsay, David Beckham sitting and standing in front of a crowd', 'confidence': 0.85675
80085806669}, {'text': 'Gordon Ramsay, David Beckham sitting around each other', 'conf
idence': 0.8557580085806669}, {'text': 'Gordon Ramsay, David Beckham sitting around a
table', 'confidence': 0.8542811701971863}, {'text': 'Gordon Ramsay, David Beckham sitt
ing in chairs', 'confidence': 0.8532811701971863}, {'text': 'Gordon Ramsay, David Beck
ham sitting together smiling for the picture', 'confidence': 0.8522811701971863}]}, 'r
equestId': 'e4f8ac75-09b9-4f96-81d4-436948d46313', 'metadata': {'height': 475, 'width'
: 594, 'format': 'Jpeg'}}

 Image Processing Completed
```

Test Case 10

http://www.stogieboys.com/media/690x380-Hawaii-Sunset.jpg

```
Processing Azure Cognitive Services Computer Vision with  SDK for Python

Enter URL of Image : http://www.stogieboys.com/media/690x380-Hawaii-Sunset.jpg

Image Analysis

{'categories': [{'name': 'abstract_', 'score': 0.00390625}, {'name': 'others_', 'score
': 0.00390625}, {'name': 'outdoor_', 'score': 0.0234375}], 'color': {'dominantColorFor
eground': 'Black', 'dominantColorBackground': 'Green', 'dominantColors': ['Green'], 'a
ccentColor': '1784B4', 'isBwImg': False}, 'requestId': 'fefda354-d017-4260-8c1e-f3ed4b
5c9764', 'metadata': {'height': 380, 'width': 690, 'format': 'Jpeg'}}


Image Description

{'description': {'tags': ['water', 'outdoor', 'palm', 'tree', 'plant', 'grass', 'river
', 'lake', 'body', 'beach', 'pond', 'pool', 'shore', 'sitting', 'front', 'ocean', 'dri
nking', 'boat', 'herd', 'large', 'bench', 'umbrella', 'island', 'group', 'swimming', '
floating', 'dog', 'people', 'eating', 'standing', 'giraffe', 'tower'], 'captions': [{'
text': 'a palm tree in front of a body of water', 'confidence': 0.9715302860601285}, {
'text': 'a group of palm trees next to a body of water', 'confidence': 0.9636493498896
506}, {'text': 'a beach with a palm tree in front of a body of water', 'confidence': 0
.9626493498896506}, {'text': 'a palm tree next to a body of water', 'confidence': 0.96
16493498896506}, {'text': 'a palm tree in front of a large body of water', 'confidence
': 0.9606493498896506}, {'text': 'a body of water in front of a palm tree', 'confidenc
e': 0.9596493498896506}, {'text': 'a group of palm trees and a body of water', 'confid
ence': 0.9572439357882007}, {'text': 'a beach with a palm tree next to a body of water
', 'confidence': 0.9562439357882007}, {'text': 'a palm tree sitting next to a body of
water', 'confidence': 0.9552439357882007}, {'text': 'a body of water next to a palm tr
ee', 'confidence': 0.9542439357882007}]}, 'requestId': '9cbaa3e5-ed52-4e15-85a4-f4034d
27dcdc', 'metadata': {'height': 380, 'width': 690, 'format': 'Jpeg'}}

 Image Processing Completed
```

Test Case 11

https://static.nascar.com/content/dam/nascar/articles/2016/9/14/main/rules-main2.jpg/jcr:content/renditions/original

```
Processing Azure Cognitive Services Computer Vision with  SDK for Python

Enter URL of Image : https://static.nascar.com/content/dam/nascar/articles/2016/9/14/main/rules-main2.jpg/jcr:content/renditions/original

Image Analysis

{'categories': [{'name': 'others_', 'score': 0.015625}, {'name': 'outdoor_', 'score': 0.0234375}, {'name': 'outdoor_road', 'score': 0.62890625}], 'color': {'dominantColorForeground': 'Grey', 'dominantColorBackground': 'Black', 'dominantColors': ['Grey', 'Black'], 'accentColor': '416777', 'isBwImg': False}, 'requestId': '17d4fa7c-cbfc-4e3b-ae57-800035c76ef9', 'metadata': {'height': 520, 'width': 922, 'format': 'Jpeg'}}

Image Description

{'description': {'tags': ['road', 'car', 'highway', 'sitting', 'lined', 'track', 'riding', 'parking', 'row', 'many', 'man', 'traffic', 'plane', 'group', 'driving', 'street', 'people', 'parked', 'airplane'], 'captions': [{'text': 'a car is lined up on a highway', 'confidence': 0.853414406104719}, {'text': 'a car is lined up in a parking lot', 'confidence': 0.796944794866151}, {'text': 'a car is lined up in a row', 'confidence': 0.7891770849251388}, {'text': 'a car is lined up in a row in a parking lot', 'confidence': 0.7283589482822601}, {'text': 'a car is lined up on the side of a road', 'confidence': 0.7273589482822601}, {'text': 'a car is lined up on the highway', 'confidence': 0.7263589482822601}, {'text': 'a car is lined up in a parking lot next to a highway', 'confidence': 0.7253589482822601}, {'text': 'a car is lined up on the side of the road', 'confidence': 0.7243589482822601}, {'text': 'a car is lined up against a highway', 'confidence': 0.7233589482822601}, {'text': 'a car is lined up with each other in a parking lot', 'confidence': 0.6951678390418559}]}, 'requestId': '95903d7a-6e1f-4e66-8c46-aacd63e2a826', 'metadata': {'height': 520, 'width': 922, 'format': 'Jpeg'}}

 Image Processing Completed
```

Test Case 12

https://photos.smugmug.com/02Sports-2/Warriors/Warriors-stun-Thunder-in-Game-/i-4DDRNhm/0/a0e14f53/L/SJM-WARRIORS-0529-148-L.jpg

```
Processing Azure Cognitive Services Computer Vision with  SDK for Python

Enter URL of Image : https://photos.smugmug.com/02Sports-2/Warriors/Warriors-stun-Thunder-in-Game-/i-4DDRNhm/0/a0e14f53/L/SJM-WARRIORS-0529-148-L.jpg

Image Analysis

{'categories': [{'name': 'people_crowd', 'score': 0.546875}], 'color': {'dominantColor
Foreground': 'Black', 'dominantColorBackground': 'Black', 'dominantColors': ['Black',
'Brown'], 'accentColor': '9F642C', 'isBwImg': False}, 'requestId': 'd73383a5-7d4e-462c
-bf0d-c5ad92fb7031', 'metadata': {'height': 534, 'width': 800, 'format': 'Jpeg'}}

Image Description

{'description': {'tags': ['person', 'sport', 'game', 'basketball', 'court', 'player',
'playing', 'ball', 'man', 'group', 'holding', 'standing', 'female', 'young', 'people',
'walking', 'wearing', 'woman', 'crowd', 'soccer', 'street'], 'captions': [{'text': 'a
group of men playing a game of basketball', 'confidence': 0.840567729389681}, {'text':
'a group of people playing a game of basketball', 'confidence': 0.839567729389681}, {'
text': 'a group of people playing basketball on a court', 'confidence': 0.838567729389
681}, {'text': 'a group of young men playing a game of basketball', 'confidence': 0.67
0308292944782}, {'text': 'a group of people playing a basketball game', 'confidence':
0.669308292944782}, {'text': 'a group of men playing a basketball game', 'confidence':
0.668308292944782}, {'text': 'a group of men playing a game of basketball on a court',
'confidence': 0.667308292944782}, {'text': 'a group of people standing on a basketball
court', 'confidence': 0.666308292944782}, {'text': 'a group of people on a basketball
court', 'confidence': 0.665308292944782}, {'text': 'a group of basketball players on t
he court', 'confidence': 0.664308292944782}]}, 'requestId': '60f5cd50-1e85-4026-b65d-4
34d6c17e211', 'metadata': {'height': 534, 'width': 800, 'format': 'Jpeg'}}

 Image Processing Completed
```

Test Case 13

https://i.redd.it/p7hwbrx35bwy.jpg

```
Processing Azure Cognitive Services Computer Vision with  SDK for Python

Enter URL of Image : https://i.redd.it/p7hwbrx35bwy.jpg

Image Analysis

{'categories': [{'name': 'people_show', 'score': 0.93359375}], 'color': {'dominantColo
rForeground': 'Black', 'dominantColorBackground': 'Black', 'dominantColors': ['Black']
, 'accentColor': 'CA1701', 'isBwImg': False}, 'requestId': '1aa734eb-e226-41f8-bc47-55
da47d08809', 'metadata': {'height': 438, 'width': 700, 'format': 'Jpeg'}}

Image Description

{'description': {'tags': ['indoor', 'table', 'sitting', 'monitor', 'black', 'small', '
top', 'screen', 'food', 'keyboard', 'computer', 'flat', 'red', 'desk', 'bowl', 'glass'
, 'white', 'star', 'plate', 'phone'], 'captions': [{'text': 'a flat screen tv sitting
in a bowl', 'confidence': 0.14251875311468432}, {'text': 'a flat screen tv sitting in
a bowl on a table', 'confidence': 0.08255995105986577}, {'text': 'a flat screen tv', '
confidence': 0.08155995105986577}, {'text': 'a screen shot of a bowl', 'confidence': 0
.08055995105986577}, {'text': 'a flat screen tv sitting in the bowl', 'confidence': 0.
07955995105986577}, {'text': 'a screen shot of a glass bowl', 'confidence': 0.07855995
105986577}, {'text': 'a flat screen tv sitting on a table', 'confidence': 0.0775599510
5986577}, {'text': 'a screen shot of a computer', 'confidence': 0.07655995105986577},
{'text': 'a flat screen tv sitting in a glass bowl', 'confidence': 0.07151429250049618
}, {'text': 'a flat screen tv sitting in the bowl on the table', 'confidence': 0.06874
635728590266}]}, 'requestId': 'dda636f0-6bf4-492a-9fd4-045494b19421', 'metadata': {'he
ight': 438, 'width': 700, 'format': 'Jpeg'}}

 Image Processing Completed
```

Test Case 14

http://www.iboxphotography.co.uk/wp-content/siteuploads/2015/07/London-Bridge-Black-and-White.jpg

```
Processing Azure Cognitive Services Computer Vision with  SDK for Python

Enter URL of Image : http://www.iboxphotography.co.uk/wp-content/siteuploads/2015/07/London-Bridge-Black-and-White.jpg

Image Analysis

{'categories': [{'name': 'building_', 'score': 0.2109375}, {'name': 'outdoor_', 'score': 0.33984375}], 'color': {'dominantColorForeground': 'White', 'dominantColorBackground': 'Grey', 'dominantColors': ['Grey', 'White'], 'accentColor': '575755', 'isBwImg': True}, 'requestId': 'c7dfacd4-4d61-42d3-b6af-125fdf9d777e', 'metadata': {'height': 800, 'width': 1200, 'format': 'Jpeg'}}

Image Description

{'description': {'tags': ['outdoor', 'water', 'bridge', 'building', 'photo', 'river', 'white', 'large', 'black', 'old', 'front', 'background', 'clock', 'city', 'view', 'tower', 'boat', 'top', 'lake', 'traveling', 'tall', 'standing', 'body'], 'captions': [{'text': 'a large bridge over a body of water', 'confidence': 0.7955908301311526}, {'text': 'a bridge over a body of water', 'confidence': 0.7945908301311526}, {'text': 'a bridge over a body of water with a city in the background', 'confidence': 0.7797356916584433}, {'text': 'a black and white photo of a bridge', 'confidence': 0.7787356916584433}, {'text': 'a bridge over a large body of water', 'confidence': 0.7777356916584433}, {'text': 'a white bridge over a body of water', 'confidence': 0.7135561765271775}, {'text': 'a large bridge over some water', 'confidence': 0.7125561765271775}, {'text': 'a large bridge over a river in a city', 'confidence': 0.7115561765271775}, {'text': 'a large bridge over a river', 'confidence': 0.7105561765271775}, {'text': 'a bridge over a river in a city', 'confidence': 0.7095561765271775}]}, 'requestId': 'fcea7851-6235-4846-9c0b-b06e0290af35', 'metadata': {'height': 800, 'width': 1200, 'format': 'Jpeg'}}

 Image Processing Completed
```

Test Case 15

https://thumbs.gfycat.com/MeanIlliterateBlackbear-max-1mb.gif

This is image stream scenario

```
Processing Azure Cognitive Services Computer Vision with  SDK for Python

Enter URL of Image : https://thumbs.gfycat.com/MeanIlliterateBlackbear-max-1mb.gif

Image Analysis

{'categories': [{'name': 'building_', 'score': 0.2578125}, {'name': 'others_', 'score'
: 0.0078125}, {'name': 'outdoor_', 'score': 0.03515625}], 'color': {'dominantColorFore
ground': 'White', 'dominantColorBackground': 'White', 'dominantColors': ['White'], 'ac
centColor': 'AF1C57', 'isBwImg': False}, 'requestId': '4bb7f748-7905-4043-9c91-8b179c2
9b1b7', 'metadata': {'height': 336, 'width': 504, 'format': 'Gif'}}

Image Description

{'description': {'tags': ['road', 'outdoor', 'building', 'decker', 'street', 'bus', 'd
ouble', 'red', 'driving', 'city', 'traveling', 'side', 'large', 'riding', 'busy', 'sto
p', 'traffic', 'clock', 'green', 'tall', 'parked'], 'captions': [{'text': 'a double de
cker bus driving down a city street', 'confidence': 0.9576062816305586}, {'text': 'a d
ouble decker bus driving down a street', 'confidence': 0.9565365147973518}, {'text': '
a red double decker bus driving down a street', 'confidence': 0.9423683057226069}, {'t
ext': 'a red double decker bus driving down a city street', 'confidence': 0.9413683057
226069}, {'text': 'a double decker bus driving down a busy street', 'confidence': 0.94
03683057226069}, {'text': 'a double decker bus on a city street', 'confidence': 0.9393
683057226069}, {'text': 'a double decker bus driving down a busy city street', 'confid
ence': 0.9383683057226069}, {'text': 'a red double decker bus driving down a busy stre
et', 'confidence': 0.9325329532263785}, {'text': 'a double decker bus driving down the
street', 'confidence': 0.9315329532263785}, {'text': 'a red double decker bus on a cit
y street', 'confidence': 0.9305329532263785}]}, 'requestId': '8932e8f0-710a-4a8d-9d5c-
49b9b744b621', 'metadata': {'height': 336, 'width': 504, 'format': 'Gif'}}
```

Due to stream of images a single image cannot be rendered out and results in error

```
---------------------------------------------------------------------------
error                                     Traceback (most recent call last)
<ipython-input-1-9d27f89a7af6> in <module>()
    289 if __name__ == '__main__':
    290     print ( "Processing Azure Cognitive Services Computer Vision with  SDK for
Python" )
--> 291     ProcessImage()
    292     print (" Image Processing Completed")
    293

<ipython-input-1-9d27f89a7af6> in ProcessImage()
    249         # Load the original image, fetched from the URL
    250         arr = np.asarray( bytearray( requests.get( urlImage ).content ), dtype
=np.uint8 )
--> 251         img = cv2.cvtColor( cv2.imdecode( arr, -1 ), cv2.COLOR_BGR2RGB )
    252
    253         renderResultOnImage( result, img )

error: C:\projects\opencv-python\opencv\modules\imgproc\src\color.cpp:11079: error: (-
215) scn == 3 || scn == 4 in function cv::cvtColor
```

Test Case 16

http://static.guim.co.uk/sys-images/Guardian/Pix/pictures/2014/10/7/1412679646851/Krispy-Kreme-doughnuts-014.jpg

```
Processing Azure Cognitive Services Computer Vision with  SDK for Python

Enter URL of Image : http://static.guim.co.uk/sys-images/Guardian/Pix/pictures/2014/10
/7/1412679646851/Krispy-Kreme-doughnuts-014.jpg

Image Analysis

{'categories': [{'name': 'food_', 'score': 0.5859375}], 'color': {'dominantColorForegr
ound': 'Brown', 'dominantColorBackground': 'Brown', 'dominantColors': ['Brown', 'White
'], 'accentColor': '3C1C05', 'isBwImg': False}, 'requestId': '51ed7a72-696e-4bb3-b433-
db2b33893707', 'metadata': {'height': 1536, 'width': 2560, 'format': 'Jpeg'}}

Image Description

{'description': {'tags': ['doughnut', 'donut', 'food', 'different', 'chocolate', 'fill
ed', 'assorted', 'box', 'top', 'covered', 'toppings', 'sitting', 'variety', 'white', '
full', 'topped', 'many', 'plate', 'various', 'pink', 'table', 'colorful', 'several', '
lot', 'holding'], 'captions': [{'text': 'a box filled with different kinds of donuts',
'confidence': 0.9047755900482618}, {'text': 'a box filled with different types of choc
olate covered donut', 'confidence': 0.7771123907433646}, {'text': 'a box filled with d
ifferent types of donuts', 'confidence': 0.7761123907433646}, {'text': 'a box filled w
ith different types of doughnuts', 'confidence': 0.7751123907433646}, {'text': 'a choc
olate covered donut', 'confidence': 0.7741123907433646}, {'text': 'a box filled with d
ifferent kinds of doughnuts', 'confidence': 0.7731123907433646}, {'text': 'a box fille
d with different types of chocolate covered doughnut', 'confidence': 0.729840194399297
8}, {'text': 'a box full of different kinds of donuts', 'confidence': 0.72884019439929
78}, {'text': 'a pink box filled with different kinds of donuts', 'confidence': 0.7278
401943992978}, {'text': 'a box of assorted donuts', 'confidence': 0.7268401943992978}]
}, 'requestId': '59e1fac8-d27f-432b-b249-372ec52b7ed6', 'metadata': {'height': 1536, '
width': 2560, 'format': 'Jpeg'}}

 Image Processing Completed
```

Summary

Lessons Learned

The Computer Vison API(s) are a powerful tool to get 360 degree view of an image, its characteristics, persons, objects and texts in the image

Azure Computer API capability is good and flexible. There is room for improvement by enhancing the API through machine learning as more and more images are analyzed and the image knowledge base grows

Issues / Benefits (Pros & Cons),

There are more Pros vs. Cons;

Pros

- A single Computer Vision API service to be deployed in Azure to access all the functions by making calls to different underlying API(s)
- API(s) return all information back in JSON format which can be directly stored in database
- Multiple Image formats are supported (JPEG, GIF, BMP. PNG)
- Option to perform enhanced image analysis from 86 category taxonomy

Cons

- Image size is restricted to 4 MB
- More AI and Machine learning capabilities should be provided

References

URLs for sample code or technical info from online sources.

https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/tutorials/pythontutorial

Skeletal API(s)

https://github.com/Microsoft/Cognitive-Vision-Python

URLs to YouTube Videos

2-Minute Video

https://youtu.be/s7AtUIe5kks

15-Minute Video