Deakin University

SIG788- OnTrack Submission

Task 4.1 P

Submitted by

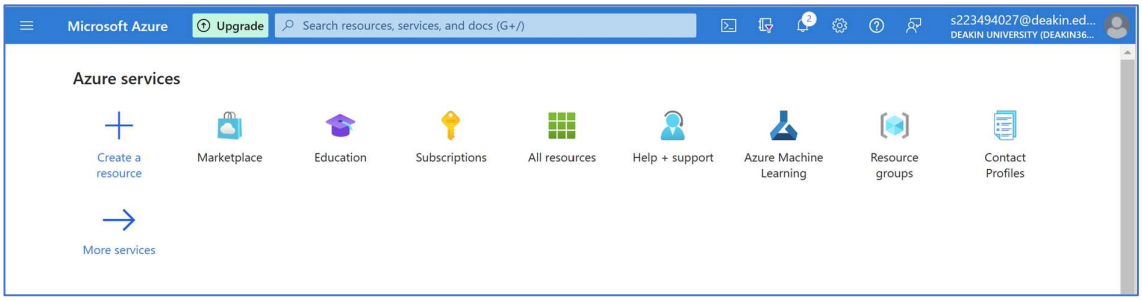
Neethu Sidhardhan
S223494027
Attempt # 2
4/7/2023

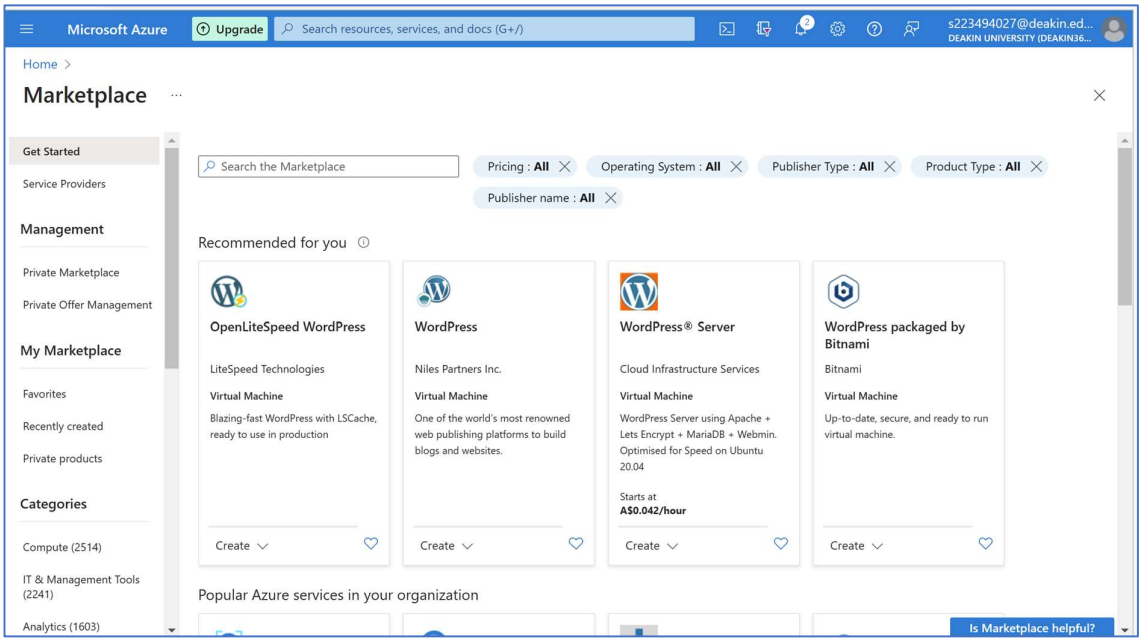
Target Grade: P

Task Details –

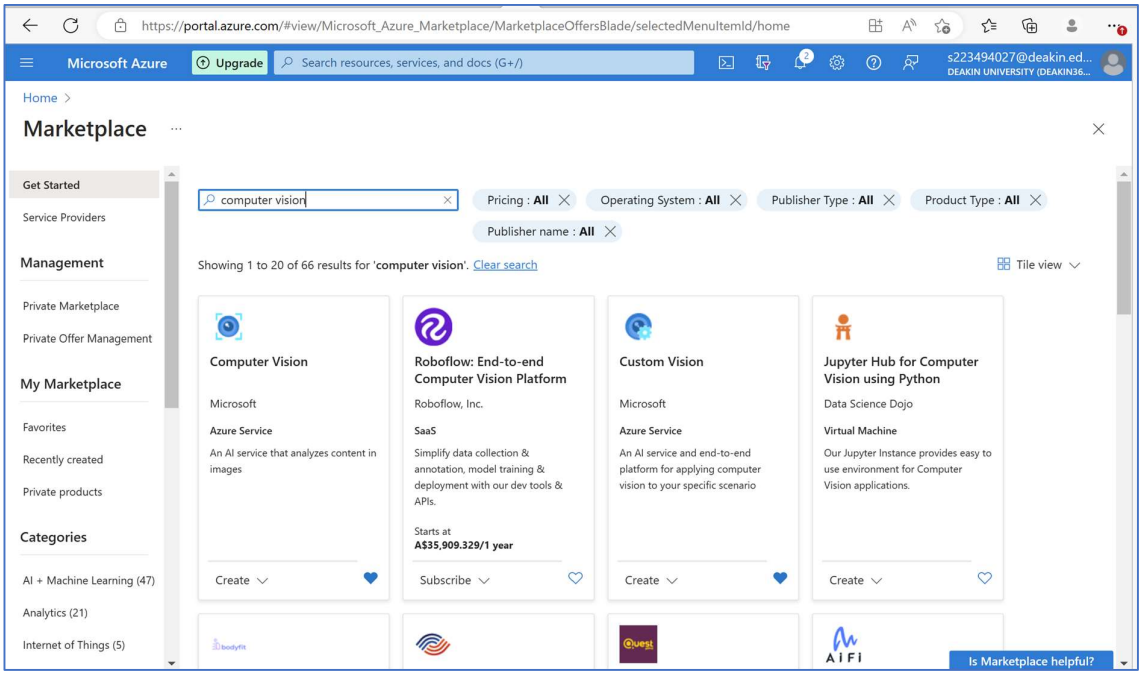
- Please explain cell by cell of your code from reading a local image to object detection, drawing a bounding box around different object. To complete this task, you need to provide the screenshot of your code and explain cell by cell of the code and explain what sort of API is being used

- Open Azure Portal > Search for Marketplace under Azure Services

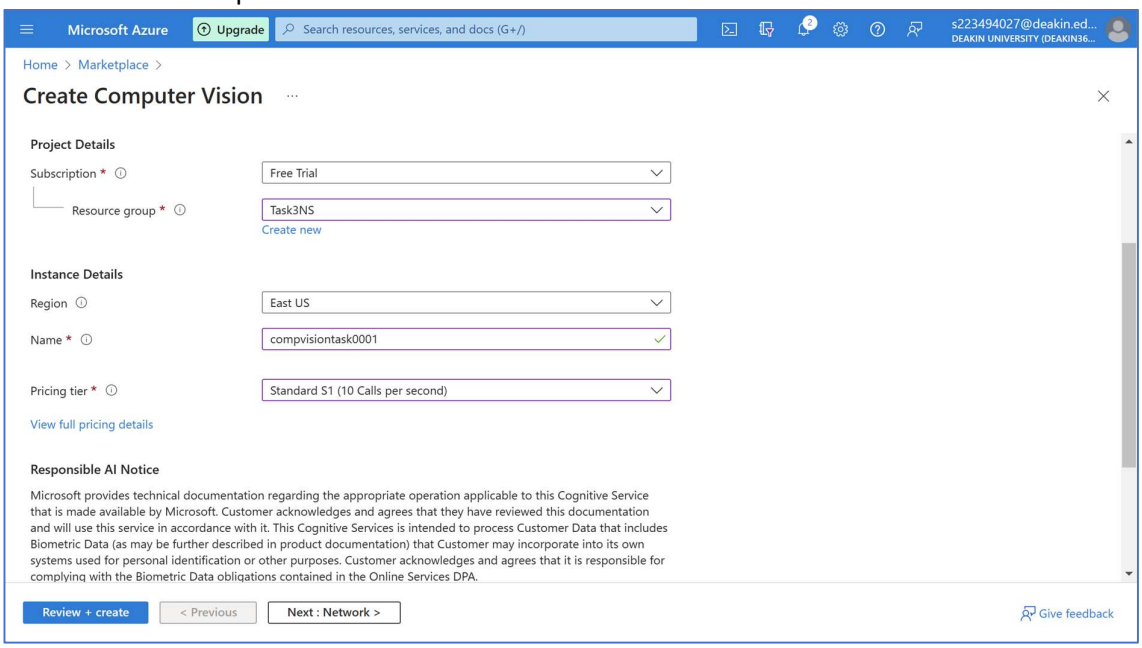




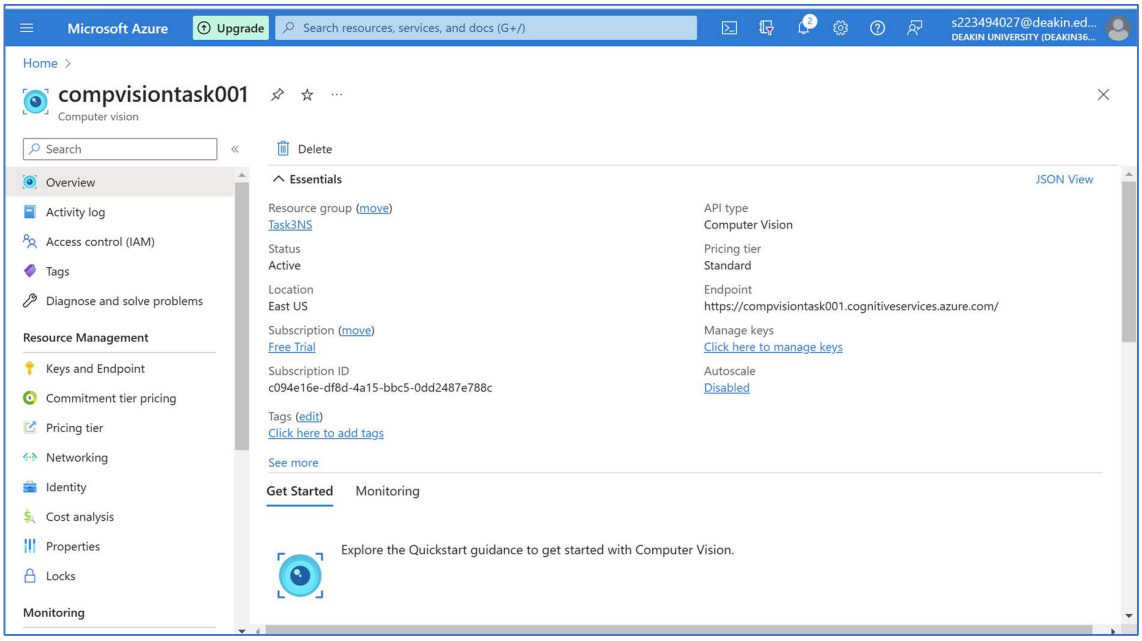
- Under Marketplace > search for Computer Vision



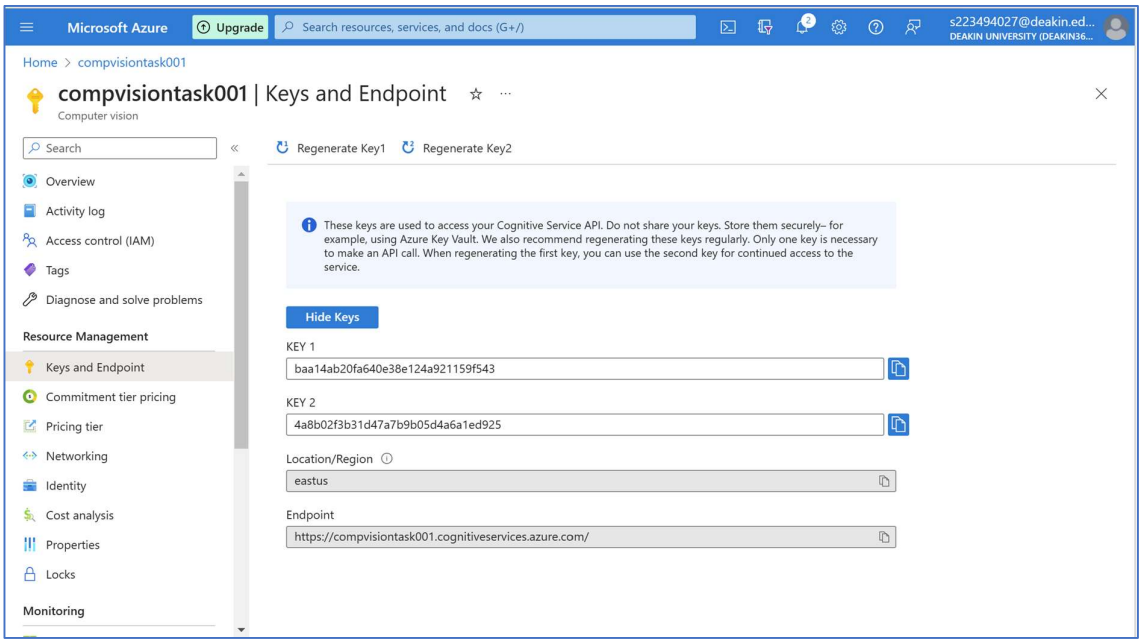
- Create Computer Vision



- Once Computer Vision is created, then check for “Keys and Endpoint” under “Resource Management” tab



- Copy the Keys and Endpoint to a notepad



- Open Jupyter Notebook.

- Load all libraries required for computer vision

```
from azure.cognitiveservices.vision.computervision import ComputerVisionClient
from azure.cognitiveservices.vision.computervision.models import OperationStatusCodes
from azure.cognitiveservices.vision.computervision.models import VisualFeatureTypes
from msrest.authentication import CognitiveServicesCredentials

from array import array
import os
from PIL import Image
import time
```

- Update the Keys and endpoint generated from Azure while creating Computer Vision resources and create the client

```
#Authenticate
#Authenticates your credentials and creates a client.

subscription_key = "baa14ab20fa640e38e124a921159f543"
endpoint = "https://compvisiontask001.cognitiveservices.azure.com/"

computervision_client = ComputerVisionClient(endpoint, CognitiveServicesCredentials(subscription_key))
```

- Uploaded the images from local drive to Github under the repository Computer Vision. Read the URL from Github where the images are uploaded.

```
# Get an image with text
read_image_url = "https://raw.githubusercontent.com/Neethusidh30/Computer_Vision1/main/sunset.jpg?token=GHSAT0AAAAAACAIT5M7OZHHL(
```

```
# Get an image with text
read_image_url = "https://raw.githubusercontent.com/Neethusidh30/Computer_Vision1/main/Flower.jpg?token=GHSAT0AAAAAACAIT5M6D64QS
```

- To read the image from local drive.

```
from os import listdir

# get the path/directory
folder_dir = "C:/Users/Dell/OneDrive/Documents/One Drive/OneDrive/Pictures/CV"
for images in os.listdir(folder_dir):

    # check if the image ends with png
    if (images.endswith(".jpg")):
        print(images)
```

```
CVFlower.jpg
Flower.jpg
IMG_20210306_141006455_PORTRAIT.jpg
Life.jpg
sunset.jpg
sunsetCV.jpg
```

Here, we have used two images for Image classification

Fig 1 Various Objects

Fig 2: Life

- To get text description of the image. Here, we can set the language on which the image to be described. The number of descriptions that can be generated is used by function “Max_description “

```
domain = "landmarks"
url = "https://raw.githubusercontent.com/Neethusidh30/Computer_Vision1/main/CV-image.jpg?token=GHSAT0AAAAAACAIT5M6GQ3BPJN62X75KW
language = "en"
max_descriptions = 3

analysis = computervision_client.describe_image(url, max_descriptions, language)

for caption in analysis.captions:
    print(caption.text)
    print(caption.confidence)
```

```
a variety of items are displayed
0.39730915427207947
```

```
domain = "landmarks"
url = "https://raw.githubusercontent.com/Neethusidh30/Computer_Vision1/main/Life.jpg?token=GHSAT0AAAAAACAIT5M76FRNDGEAE6NGMX2GZB
language = "en"
max_descriptions = 3

analysis = computervision_client.describe_image(url, max_descriptions, language)

for caption in analysis.captions:
    print(caption.text)
    print(caption.confidence)
```

```
a man and a woman sitting on a rock by a river
0.4516935646533966
```

- Analysing the images and tagging each image with appropriate categories. The feature “analyze_image” helps in analysing the images in detailed.

```
#Analyze an image

url = "https://raw.githubusercontent.com/Neethusidh30/Computer_Vision1/main/CV-image.jpg?token=GHSAT0AAAAAACAIT5M6GQ3BPJN62X75KW
image_analysis = computervision_client.analyze_image(url,visual_features=[VisualFeatureTypes.tags])

for tag in image_analysis.tags:
    print(tag)
```

```
{'additional_properties': {}, 'name': 'text', 'confidence': 0.9816043376922607, 'hint': None}
{'additional_properties': {}, 'name': 'mug', 'confidence': 0.8496596813201904, 'hint': None}
{'additional_properties': {}, 'name': 'fashion', 'confidence': 0.4777320325374603, 'hint': None}
{'additional_properties': {}, 'name': 'accessory', 'confidence': 0.4433504641056061, 'hint': None}
{'additional_properties': {}, 'name': 'design', 'confidence': 0.4095042943954468, 'hint': None}
```

```
url = "https://raw.githubusercontent.com/Neethusidh30/Computer_Vision1/main/Life.jpg?token=GHSAT0AAAAAACAIT5M76FRNDGEAE6NGMX2GZB
image_analysis = computervision_client.analyze_image(url,visual_features=[VisualFeatureTypes.tags])

for tag in image_analysis.tags:
    print(tag)
```

```
{'additional_properties': {}, 'name': 'clothing', 'confidence': 0.9942530393600464, 'hint': None}
{'additional_properties': {}, 'name': 'person', 'confidence': 0.9925233125686646, 'hint': None}
{'additional_properties': {}, 'name': 'outdoor', 'confidence': 0.9835610389709473, 'hint': None}
{'additional_properties': {}, 'name': 'human face', 'confidence': 0.9619994163513184, 'hint': None}
{'additional_properties': {}, 'name': 'tree', 'confidence': 0.9529051780700684, 'hint': None}
{'additional_properties': {}, 'name': 'ground', 'confidence': 0.9395208954811096, 'hint': None}
{'additional_properties': {}, 'name': 'mammal', 'confidence': 0.9129573702812195, 'hint': None}
{'additional_properties': {}, 'name': 'water', 'confidence': 0.8591468334197998, 'hint': None}
{'additional_properties': {}, 'name': 'lake', 'confidence': 0.8477662801742554, 'hint': None}
{'additional_properties': {}, 'name': 'jeans', 'confidence': 0.8451061248779297, 'hint': None}
{'additional_properties': {}, 'name': 'man', 'confidence': 0.841572642326355, 'hint': None}
{'additional_properties': {}, 'name': 'sitting', 'confidence': 0.725838303565979, 'hint': None}
```

- Categorize the image with confidence level

```
# Categorize an Image

print("==== Categorize an image remote ====")

url = "https://raw.githubusercontent.com/Neethusidh30/Computer_Vision1/main/CV-image.jpg?token=GHSAT0AAAAAACAIT5M6GQ3BPJN62X75KW

remote_image_features = ["categories"]
categorize_results_remote = computervision_client.analyze_image(url, remote_image_features)

print("Categories from remote image: ")
if (len(categorize_results_remote.categories)==0):
    print("No categories detected.")
else:
    for category in categorize_results_remote.categories:
        print("'{}' with confidence {:.2f}%".format(category.name, category.score *100))
```

```
==== Categorize an image remote ====
Categories from remote image:
'abstract_texture' with confidence 57.03%
```

```
# Categorize an Image

print("==== Categorize an image remote ====")

url = "https://raw.githubusercontent.com/Neethusidh30/Computer_Vision1/main/Life.jpg?token=GHSAT0AAAAAACAIT5M76FRNDGEAE6NGMX2GZB

remote_image_features = ["categories"]

categorize_results_remote = computervision_client.analyze_image(url, remote_image_features)

print("Categories from remote image: ")
if (len(categorize_results_remote.categories)==0):
    print("No categories detected.")
else:
    for category in categorize_results_remote.categories:
        print("'{}' with confidence {:.2f}%".format(category.name, category.score *100))
```

```
==== Categorize an image remote ====
Categories from remote image:
'outdoor_' with confidence 0.39%
'people_group' with confidence 23.44%
'people_many' with confidence 43.75%
```

- Analyse the object detection. Here, the objects are detected and marked as rectangle.

```
#Analysing image detection

remote_image_url_objects = 'https://raw.githubusercontent.com/Neethusidh30/Computer_Vision1/main/Life.jpg?token=GHSAT0AAAAAACAIT5

detect_objects_result_remote = computervision_client.detect_objects(remote_image_url_objects)

for object in detect_objects_result_remote.objects:
    print(object)
```

```
{'additional_properties': {}, 'rectangle': <azure.cognitiveservices.vision.computervision.models._models_py3.BoundingRect objec
t at 0x0000021326F4E850>, 'object_property': 'person', 'confidence': 0.686, 'parent': None}
{'additional_properties': {}, 'rectangle': <azure.cognitiveservices.vision.computervision.models._models_py3.BoundingRect objec
t at 0x0000021326F4EAC0>, 'object_property': 'person', 'confidence': 0.9, 'parent': None}
```

Fig 3 Image detection

Only person was detected in the image. No other objects were detected.

```
#Analysing image detection

remote_image_url_objects = 'https://raw.githubusercontent.com/Neethusidh30/Computer_Vision1/main/CV-image.jpg?token=GHSAT0AAAAAC

detect_objects_result_remote = computervision_client.detect_objects(remote_image_url_objects)

for object in detect_objects_result_remote.objects:
    print(object)
```

```
{'additional_properties': {}, 'rectangle': <azure.cognitiveservices.vision.computervision.models._models_py3.BoundingRect objec
t at 0x0000028BE0E2B670>, 'object_property': 'Wall clock', 'confidence': 0.603, 'parent': <azure.cognitiveservices.vision.compu
tervision.models._models_py3.ObjectHierarchy object at 0x0000028BE0E2B280>}
{'additional_properties': {}, 'rectangle': <azure.cognitiveservices.vision.computervision.models._models_py3.BoundingRect objec
t at 0x0000028BE0E2B970>, 'object_property': 'Wall clock', 'confidence': 0.701, 'parent': <azure.cognitiveservices.vision.compu
tervision.models._models_py3.ObjectHierarchy object at 0x0000028BE0E2B640>}
{'additional_properties': {}, 'rectangle': <azure.cognitiveservices.vision.computervision.models._models_py3.BoundingRect objec
t at 0x0000028BE0E2BA90>, 'object_property': 'cup', 'confidence': 0.696, 'parent': <azure.cognitiveservices.vision.computervisi
on.models._models_py3.ObjectHierarchy object at 0x0000028BE0E2B9D0>}
{'additional_properties': {}, 'rectangle': <azure.cognitiveservices.vision.computervision.models._models_py3.BoundingRect objec
t at 0x0000028BE0E2BBB0>, 'object_property': 'kitchen appliance', 'confidence': 0.594, 'parent': None}
{'additional_properties': {}, 'rectangle': <azure.cognitiveservices.vision.computervision.models._models_py3.BoundingRect objec
t at 0x0000028BE0E2BC10>, 'object_property': 'cup', 'confidence': 0.544, 'parent': <azure.cognitiveservices.vision.computervisi
on.models._models_py3.ObjectHierarchy object at 0x0000028BE0E2BC70>}
{'additional_properties': {}, 'rectangle': <azure.cognitiveservices.vision.computervision.models._models_py3.BoundingRect objec
t at 0x0000028BE0E2BD90>, 'object_property': 'cup', 'confidence': 0.538, 'parent': <azure.cognitiveservices.vision.computervisi
on.models._models_py3.ObjectHierarchy object at 0x0000028BE0E2BCD0>}
{'additional_properties': {}, 'rectangle': <azure.cognitiveservices.vision.computervision.models._models_py3.BoundingRect objec
t at 0x0000028BE0E2BEB0>, 'object_property': 'Luggage and bags', 'confidence': 0.676, 'parent': None}
{'additional_properties': {}, 'rectangle': <azure.cognitiveservices.vision.computervision.models._models_py3.BoundingRect objec
t at 0x0000028BE0E2BF10>, 'object_property': 'potted plant', 'confidence': 0.644, 'parent': <azure.cognitiveservices.vision.com
putervision.models._models_py3.ObjectHierarchy object at 0x0000028BE0E2BFD0>}
```

Here some objects like Camera, Table tennis racket, luggage's, etc are incorrectly detected.

Works Cited

1. Anon., 2023. *ASPOSE.* [Online]
   Available at: https://products.aspose.app/imaging/object-detection
   [Accessed 7 April 2023].
2. Anon., 2023. *Google Images.* [Online]
   Available at:
   https://www.bing.com/images/search?view=detailV2&ccid=lHJQJ5Nx&id=31CD8A05337256A81543D61117E26F35DE816558&thid=OIP.lHJQJ5NxzfE8fGgTlmGQiAHaH6&mediaurl=https%3a%2f%2fthumbs.dreamstime.com%2fz%2fset-different-objects-20188845.jpg&exph=1390&expw=1300&q=diff
   [Accessed 7 April 2023].
3. Microsoft, 2023. *Microsoft - Azure - learning.* [Online]
   Available at: https://learn.microsoft.com/en-us/python/api/overview/azure/cognitiveservices-vision-computervision-readme?view=azure-python
   [Accessed April 2023].
4. Microsoft, n.d. *Microsoft Azure.* [Online]
   Available at: https://learn.microsoft.com/en-us/azure/cognitive-services/computer-vision/overview
5. Nakisa, Dr Bahareh, 2023. *Great Learning.* [Online]
   Available at: https://olympus.mygreatlearning.com/courses/91262/pages/4-dot-2-what-is-computer-vision-2?module_item_id=3833099
6. Sidhardhan, N., 2023. *Github.* [Online]
   Available at: https://github.com/Neethusidh30/Computer_Vision1
   [Accessed 1 April 2023].