

ITIS 6177 System Integration Final Project
Snehal Manish Thakkar
801255075

In this project, I am using the Microsoft Azure [Cognitive Services - Computer Vision](#) to analyze an image, describe an image, detect the objects in the image, get areas of the main object of an image and generate the thumbnail of an image.

In this project, node js framework is used, which embeds the microsoft API to create our own custom API. The method POST, that has the imageUrl parameter as an input from the user. Once we give a URL of the image, we will be able to get the following JSON data back to the developer to further enhance their personal projects. We are documenting the APIs using swagger and also defines whether the request returns with following code:

200: Successfully analyzes the image and provides json output containing the analyzed data of image

400: Bad Request or invalid request

500: Internal Server Error.

1. Analyze an image:

Based on the image content, this process extracts a diverse range of visual attributes. . By default, picture categories are returned in the response. A successful response will be provided in JSON. If the request failed, the response will include an error code and a message to assist you figure out what went wrong.

Visual features in JSON data is a string describing the sorts of visual features to return. Multiple values should be separated by commas. The following are examples of valid visual feature types: Categories - classifies picture material based on a taxonomy given in the documentation. Tags - a thorough list of terms relating to the picture's content is used to tag the image. Description - a complete English statement that explains the image's content. Faces – identifies the presence of faces. If coordinates, gender, and age are present, produce them. ImageType - determines if an image is clipart or a line drawing. Color - defines the accent color, dominant color, and whether or not a picture is black and white. Adult - determines if the image is pornographic (depicting nudity or a sex act) or gruesome (depicts extreme violence or blood). Sexually provocative (or racy) information is also recognized. Objects - detects

Using postman or swagger documentation, we could give out the imageUrl parameter.

Using postman:

<http://159.223.178.248:3000/analyse>

Home Workspaces Explore

POST http://159.223.178.248:3000/analyse

http://159.223.178.248:3000/analyse

POST http://159.223.178.248:3000/analyse

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 ... "imageUrl": "https://upload.wikimedia.org/wikipedia/commons/thumb/d/dc/Steve_Jobs_Headshot_2010-CROP_%28cropped_2%29.jpg/800px-Steve_Jobs_Headshot_2010-CROP_%28cropped_2%29.jpg"

2

3

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "categories": [
3     {
4       "name": "others",
5       "score": 0.00390625
6     },
7     {
8       "name": "people_portrait",
9       "score": 0.609375
10    }
11  ],
12  "adult": {
13    "isAdultContent": false,
14    "isRacyContent": false,
15    "isGoryContent": false,
16    "adultScore": 0.002573683392564983,
17    "racyScore": 0.004212106990116596,
18    "goryScore": 0.02768128365278244
19  },
20  "color": {
21    "dominantColorForeground": "White",
22    "dominantColorBackground": "Black",
23    "dominantColors": [
24      "Black"
25    ]
26  }
27}
```

Status: 200 OK Time: 1011 ms Size: 2.07 KB Save Response

Find and Replace Console

Runner Trash

ImageAnalysis

Result of AnalyzeImage operation.

Name	Type	Description
adult	AdultInfo	An object describing whether the image contains adult-oriented content and/or is racy.
brands	DetectedBrand[]	Array of brands detected in the image.
categories	Category[]	An array indicating identified categories.
color	ColorInfo	An object providing additional metadata describing color attributes.
description	Image Description Details	A collection of content tags, along with a list of captions sorted by confidence level, and image metadata.
faces	Face Description[]	An array of possible faces within the image.
imageType	ImageType	An object providing possible image types and matching confidence levels.
metadata	Image Metadata	Image metadata.
objects	DetectedObject[]	Array of objects describing what was detected in the image.
requestId	string	Id of the REST API request.
tags	ImageTag[]	A list of tags with confidence level.

ImageCaption

An image caption, i.e. a brief description of what the image depicts.

Name	Type	Description
confidence	number	The level of confidence the service has in the caption.
text	string	The text of the caption.

ImageDescriptionDetails

A collection of content tags, along with a list of captions sorted by confidence level, and image metadata.

Name	Type	Description
captions	Image Caption[]	A list of captions, sorted by confidence level.
tags	string[]	A collection of image tags.

Using web browser:

<http://159.223.178.248:3000/docs/>

Swagger UI x low resolution image - Google x 62863642c81b7f77871a342a... x +

Dangerous | 159.223.178.248:3000/docs/#/default/post_analyse

POST /analyse

Gives out image tags

Parameters

Name Description

imageUrl * required the image url for image detection
(body) Edit Value | Model

```
{ "imageUrl": "https://assets-global.website-files.com/6005fac27a49a9cd477afb63/62863642c81b7f77871a342a_topaz-labs-gigapixel-ai-face-recovery-1-before.jpg" }
```

Cancel

Parameter content type application/json

Execute Clear

Swagger UI x low resolution image - Google x 62863642c81b7f77871a342a... x +

Dangerous | 159.223.178.248:3000/docs/#/default/post_analyse

Responses

Response content type application/json

Curl

```
curl -X 'POST' \
'http://159.223.178.248:3000/analyse' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{ "imageUrl": "https://assets-global.website-files.com/6005fac27a49a9cd477afb63/62863642c81b7f77871a342a_topaz-labs-gigapixel-ai-face-recovery-1-before.jpg" }'
```

Request URL

```
http://159.223.178.248:3000/analyse
```

Server response

Code Details

200 Response body

```
{
  "categories": [
    {
      "name": "people",
      "score": 0.34765625
    },
    {
      "name": "people_portrait",
      "score": 0.6484375
    }
  ],
  "adult": {
    "isAdultContent": false,
    "isRacyContent": false,
    "isGoryContent": false,
    "adultScore": 0.0034034138079732656,
    "racyScore": 0.00429260544788933,
    "goreScore": 0.07217106223106384
  },
  "color": {
    "dominantColorForeground": "Black",
    "dominantColorBackground": "Grey",
    "dominantColors": [
      "Grey",
      "Brown"
    ]
  }
}
```

Download

Response headers

```
        },
    ],
    "description": {
        "tags": [
            "text",
            "person"
        ],
        "captions": [
            {
                "text": "a woman smiling for the camera",
                "confidence": 0.5344734787940979
            }
        ]
    },
    "faces": [
        {
            "faceRectangle": {
                "left": 124,
                "top": 96,
                "width": 198,
                "height": 198
            }
        }
    ],
    "objects": [
```

2. Describe an image

This action provides a human-readable visual description with entire phrases. The description is based on a set of content tags that the operation also returns. More than one description can be created for each image. Descriptions are sorted by their level of certainty. Descriptions may incorporate findings from celebrity and landmark domain models, if relevant. There are two input options available: (1) uploading an image and (2) supplying an image URL. A successful answer will be in JSON format. If the request failed, the response will include an error code and a message that will help you figure out what went wrong.

Using Postman:

<http://159.223.178.248:3000/describe>

POST http://159.223.178.248:3000/describe

Body (JSON)

```

1
2   ...
3     "imageUrl": "https://upload.wikimedia.org/wikipedia/commons/thumb/d/dc/Steve_Jobs_Headshot_2010-CROP_%28cropped_2%29.jpg"
4       800px-Steve_Jobs_Headshot_2010-CROP_%28cropped_2%29.jpg"
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
  
```

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 912 ms Size: 550 B Save Response

Using endpoint:
<http://159.223.178.248:3000/docs/>

POST /describe

Gives out image tags

Parameters

Name	Description
imageUrl * required	the image url for image detection (body)

```
{
  "imageUrl": "https://assets-global.website-files.com/6005fac27a49a9cd477afb63/62863642c81b7f77871a342a_topaz-labs-gigapixel-ai-face-recovery-1-before.jpg"
}
```

Cancel

Parameter content type application/json

Execute Clear

Request URL
<http://159.223.178.248:3000/describe>

Server response

Code	Details
200	<p>Response body</p> <pre>{ "description": { "tags": ["text", "person"], "captions": [{ "text": "a woman smiling for the camera", "confidence": 0.5344730615615845 }] }, "requestId": "7342b33b-d883-4c10-8460-c682b9651813", "metadata": { "height": 406, "width": 406, "format": "Jpeg" }, "modelVersion": "2021-05-01" }</pre> <p>Response headers</p> <pre>access-control-allow-origin: * connection: keep-alive content-length: 262 content-type: application/json; charset=utf-8 date: Thu, 15 Dec 2022 15:36:20 GMT etag: W/"106-VHR165jYM5/hv+CmLaRcJ/5AeNY" keep-alive: timeout=5 x-powered-by: express</pre>

Responses

Code	Description
200	Successfully describes the image and provide a json data of the description of image.

ImageDescription

A collection of content tags, along with a list of captions sorted by confidence level, and image metadata.

Name	Type	Description
description.captions	Image Caption[]	A list of captions, sorted by confidence level.
description.tags	string[]	A collection of image tags.
metadata	Image Metadata	Image metadata.
requestId	string	Id of the REST API request.

ImageMetadata

Image metadata.

Name	Type	Description
format	string	Image format.
height	integer	Image height, in pixels.
width	integer	Image width, in pixels.

3. Detect an image

Detects objects in the image given to it. A successful answer will be in JSON format. If the request failed, the response will include an error code and a message that will help you figure out what went wrong.

Using Postman:

POST http://159.223.178.248:3000/detect

Body (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1
2   "imageUrl": "https://upload.wikimedia.org/wikipedia/commons/thumb/d/dc/Steve_Jobs_Headshot_2010-CROP_%28cropped_2%29.jpg/
3   800px-Steve_Jobs_Headshot_2010-CROP_%28cropped_2%29.jpg"
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
  
```

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 1098 ms Size: 778 B Save Response

Using endpoint:

<http://159.223.178.248:3000/docs/>

POST /detect

Gives out image tags

Parameters

Name	Description
imageUrl * required	the image url for image detection (body)

```
{
  "imageUrl": "https://assets-global.website-files.com/6005fac27a49a9cd477afb63/62863642c81b7f77871a342a_topaz-labs-gigapixel-ai-face-recovery-1-before.jpg"
}
```

Cancel

Parameter content type application/json

Execute Clear

```

Request URL
http://159.223.178.248:3000/detect

Server response
Code Details
200 Response body
{
  "objects": [
    {
      "rectangle": {
        "x": 6,
        "y": 3,
        "w": 397,
        "h": 401
      },
      "object": "person",
      "confidence": 0.687
    }
  ],
  "requestId": "17d1a70b-2fec-4a8b-b8bf-cf2e788fd663",
  "metadata": {
    "height": 406,
    "width": 406,
    "format": "Jpeg"
  },
  "modelVersion": "2021-04-01"
}

Response headers
access-control-allow-origin: *
connection: keep-alive
content-length: 227
content-type: application/json; charset=utf-8
date: Thu, 15 Dec 2022 15:37:09 GMT
etag: W/e3-qMU0zLq0bz43TY27eoW0a/TXls'
keep-alive: timeout=5
x-powered-by: Express

Responses
Code Description
200 Successfully detects the image and provides json data of the details of image.

```

DetectedObject

An object detected in an image.

Name	Type	Description
confidence	number	Confidence score of having observed the object in the image, as a value ranging from 0 to 1.
object	string	Label for the object.
parent	Object Hierarchy	The parent object, from a taxonomy perspective. The parent object is a more generic form of this object. For example, a 'bulldog' would have a parent of 'dog'.
rectangle	Bounding Rect	Approximate location of the detected object.

4. Define areas of interest of the image

This procedure returns a bounding box containing the image's most essential area. A successful answer will be in JSON format. If the request failed, the response provides an error code and a message to assist discover what went wrong. In the event of a failure, the error code and a message are returned. The error code might be one of InvalidImageUrl,

InvalidImageFormat, InvalidImageSize, NotSupportedImage, FailedToProcess, Timeout, or InternalServerError.

The screenshot shows the Postman application interface. At the top, there are tabs for Home, Workspaces, and Explore, along with a search bar and user authentication buttons. The main workspace shows a list of recent requests. A specific POST request to `http://159.223.178.248:3000/areaOfInterest` is selected. The request details panel shows the method (POST), URL, and various configuration tabs like Params, Authorization, Headers, Body, Pre-request Script, Tests, and Settings. The Body tab is active, displaying a JSON payload:

```
1  ...
2  ...
3  ...
4  ...
5  ...
6  ...
7  ...
8  ...
9  ...
10  ...
11  ...
12  ...
13  ...
14  ...
15  ...
```

The JSON payload is as follows:

```
1  {
2      "areaOfInterest": {
3          "x": 0,
4          "y": 0,
5          "w": 773,
6          "h": 773
7      },
8      "requestId": "ac95b22f-d710-4527-b0ac-1a7d632d981e",
9      "metadata": {
10          "height": 774,
11          "width": 800,
12          "format": "Jpeg"
13      },
14      "modelVersion": "2021-04-01"
15 }
```

Below the body, the response status is shown as 200 OK with a time of 714 ms and a size of 449 B. The response content is also displayed in JSON format. At the bottom, there are navigation buttons for Find and Replace, Console, Runner, Trash, and other application controls.

Using endpoint:

<http://159.223.178.248:3000/docs/>

POST /areaOfInterest

Gives out image tags

Parameters

Name	Description
imageUrl * required	the image url for image detection (body)

```
{
  "imageUrl": "https://assets-global.website-files.com/6005fac27a49a9cd477afb63/62863642c81b7f77871a342a_topaz-labs-gigapixel-ai-face-recovery-1-before.jpg"
}
```

Cancel

Parameter content type
application/json

Execute **Clear**

'http://159.223.178.248:3000/areaOfInterest' \
-H 'accept: application/json' \
-H 'Content-type: application/json' \
-d '{
 "imageUrl": "https://assets-global.website-files.com/6005fac27a49a9cd477afb63/62863642c81b7f77871a342a_topaz-labs-gigapixel-ai-face-recovery-1-before.jpg"
}'

Request URL
http://159.223.178.248:3000/areaOfInterest

Server response

Code	Details
200	Response body <pre>{ "areaOfInterest": { "x": 0, "y": 0, "w": 406, "h": 406 }, "requestId": "0be2b6f4-771d-49a5-b2ad-e7b316fad00", "metadata": { "height": 406, "width": 406, "format": "Jpeg" }, "modelVersion": "2021-04-01" }</pre> <p>Download</p> Response headers <pre>access-control-allow-origin: * connection: keep-alive content-length: 181 content-type: application/json; charset=utf-8 date: Thu, 15 Dec 2022 15:37:47 GMT etag: W/b5-aH0QaqU7AjDtNm0Oyvi6vX4pXg' keep-alive: timeout=5 x-powered-by: Express</pre>

Responses

Code	Description
200	Successfully detects the image and provide a array of object containing the details of image.
400	Bad Request or invalid request

AreaOfInterestResult

Result of AreaOfInterest operation.

Name	Type	Description
areaOfInterest	Bounding Rect	A bounding box for an area of interest inside an image.
metadata	Image Metadata	Image metadata.
requestId	string	Id of the REST API request.

BoundingRect

A bounding box for an area inside an image.

Name	Type	Description
h	integer	Height measured from the top-left point of the area, in pixels.
w	integer	Width measured from the top-left point of the area, in pixels.
x	integer	X-coordinate of the top left point of the area, in pixels.
y	integer	Y-coordinate of the top left point of the area, in pixels.

5. Generate thumbnail of the image

This action creates a thumbnail image with the user-specified length and width. By default, the service analyzes the image, finds the region of interest (ROI), and provides smart cropping coordinates based on the ROI. Smart cropping helps when you provide an aspect ratio that differs from that of the input image. A successful answer includes the thumbnail picture binary. If the request failed, the response provides an error code and a message to assist discover what went wrong. In the event of a failure, the error code and message are returned. The error code might be InvalidImageUrl, InvalidImageFormat, InvalidImageSize, InvalidThumbnailSize, NotSupportedImage, FailedToProcess, Timeout, or InternalServerError.

The JSON output is a binary file which needs to be downloaded and converted into the thumbnail.

http://159.223.178.248:3000/generateThumbnail

POST http://159.223.178.248:3000/generateThumbnail

Body (8)

```
{ "imageUrl": "https://upload.wikimedia.org/wikipedia/commons/thumb/d/dc/Steve_Jobs_Headshot_2010-CROP_%28cropped_2%29.jpg" }
```

Status: 200 OK Time: 965 ms Size: 2.76 KB

Using endpoint:
<http://159.223.178.248:3000/docs/>

```
-H 'Content-Type: application/json' \
-d '{
  "imageUrl": "https://assets-global.website-files.com/6005fac27a49bd4d77fb63/62863642c81b7f77871a342a_topaz-labs-gigapixel-ai-face-recovery-1-before.jpg"
}'
```

Request URL
http://159.223.178.248:3000/generateThumbnail

Server response

Code Details

200 Response body

```
data: <base64 encoded binary image>
```

Response headers

```
access-control-allow-origin: *
connection: keep-alive
content-length: 2524
content-type: application/json; charset=utf-8
date: Thu, 15 Dec 2022 15:38:19 GMT
etag: W/"9dc-XQ8BDOb6fskURB6/gMhDksek0"
keep-alive: timeout=5
x-powered-by: Express
```

Responses

Code	Description
200	Successfully detects the image and provide a array of object containing the details of image.
400	Bad Request or invalid request

! response_16711...json may be dangerous, so Chrome has blocked it. Discard

Following is the image url when I used Postman:

https://upload.wikimedia.org/wikipedia/commons/thumb/d/dc/Steve_Jobs_Headshot_2010-CRO

[P %28cropped 2%29.jpg/800px-Steve Jobs Headshot 2010-CROP %28cropped 2%29.jpg](#)



Following is the image url when I used the endpoint. This is a low resolution image, just to check whether it detects the objects even in low resolution quality.

https://assets-global.website-files.com/6005fac27a49a9cd477afb63/62863642c81b7f77871a342a_topaz-labs-gigapixel-ai-face-recovery-1-before.jpg



I have provided the JSON data in image1.json and image2.json files.