ITIS 6177 System Integration Final Project Documentation

PROJECT DESCRIPTION:

With Azure's Computer Vision service, you have access to algorithms that analyze photos and provide data depending on the visual characteristics you're interested in. With only one Analyze Picture operation, this API offers image captioning, image tagging, object detection, people detection, and Read OCR features.

The services that are used for this project:

1. OCR (Optical Character Recognition)
2. Analyze Image
3. Describe Image
4. Detect Objects

1. **OCR (Optical Character Recognition) -** Optical character recognition, usually known as OCR, is also known as text extraction or text recognition. You may extract printed or handwritten text from pictures like posters, street signs, and product labels as well as from documents like articles, reports, forms, and invoices using OCR algorithms based on machine learning. To access the digital form of the scanned text, the text is often retrieved as words, text lines, paragraphs, or text blocks. As a result, there is no longer any requirement for manual data entry or very little of it.

INPUT	EXAMPLES
Images: General, in-the-wild images	Labels, street signs and posters

API ENDPOINT- The API provides you with AI algorithms for extracting text from images and returning it as structured strings.

http://165.22.176.154:3000/ocr

Tools required to check this API: Postman

Steps to run in Postman:

- Open POSTMAN
- Method Type- POST

- Copy and paste the link- http://165.22.176.154:3000/ocr
- Image URL- https://imgv3.fotor.com/images/blog-cover-image/How-to-Make-Text-Stand-Out-And-More-Readable.jpg

- In Body choose Raw and JSON format.

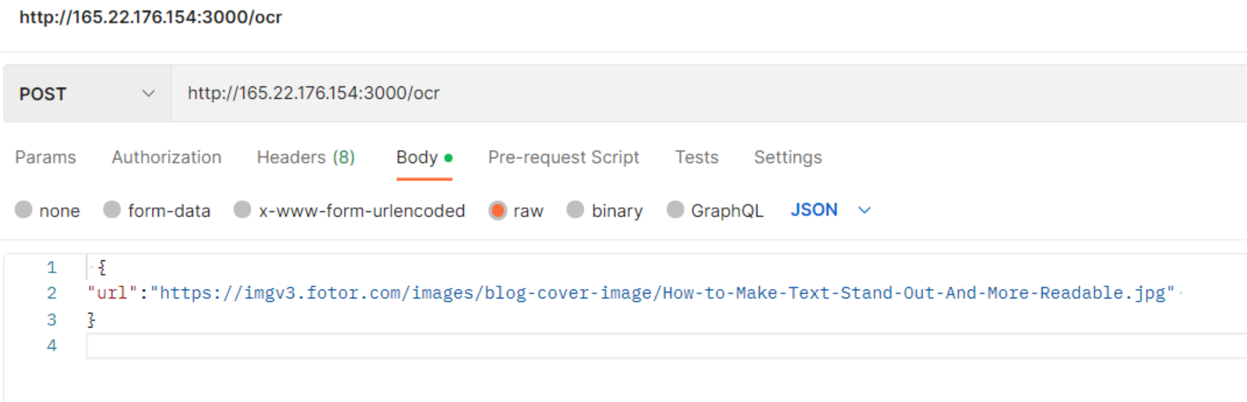
Paste following:

{

"url":"https://imgv3.fotor.com/images/blog-cover-image/How-to-Make-Text-Stand-Out-And-More-Readable.jpg"

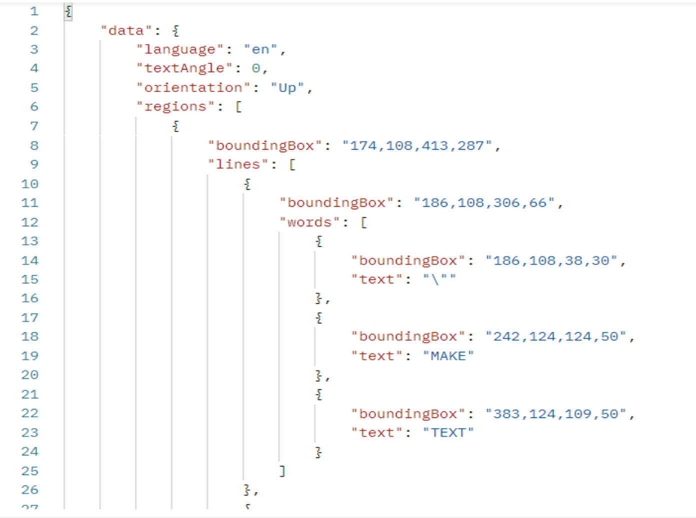
}

Click Send.



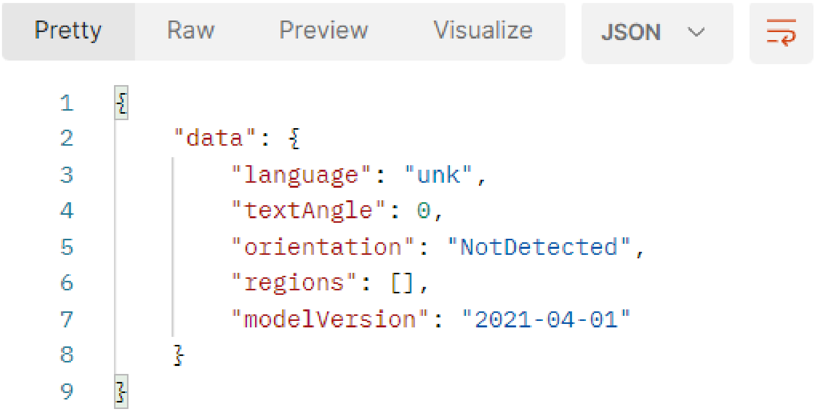
```
{
"url":"https://imgv3.fotor.com/images/blog-cover-image/How-to-Make-Text-Stand-Out-And-More-Readable.jpg"
}
```

- You will get the following output that will read the text from Image.



```
{
    "data": {
        "language": "en",
        "textAngle": 0,
        "orientation": "Up",
        "regions": [
            {
                "boundingBox": "174,108,413,287",
                "lines": [
                    {
                        "boundingBox": "186,108,306,66",
                        "words": [
                            {
                                "boundingBox": "186,108,38,30",
                                "text": "\""
                            },
                            {
                                "boundingBox": "242,124,124,50",
                                "text": "MAKE"
                            },
                            {
                                "boundingBox": "383,124,109,50",
                                "text": "TEXT"
                            }
                        ]
                    },
```

- If there is any image that has no text, it will send the following response-



```
{
    "data": {
        "language": "unk",
        "textAngle": 0,
        "orientation": "NotDetected",
        "regions": [],
        "modelVersion": "2021-04-01"
    }
}
```

- Similarly, you can try different Image URL to check this Endpoint.

2. **Analyze Image:** A range of visual elements can be extracted from your photographs using the Computer Vision Image Analysis service. It may be used to detect specific brands or things, ascertain whether an image contains adult content, or even locate captions for the pictures.

   This operation extracts a rich set of visual features based on the image content.

   API ENDPOINT- The API analyzes images with AI algorithms.

   http://165.22.176.154:3000/analyze

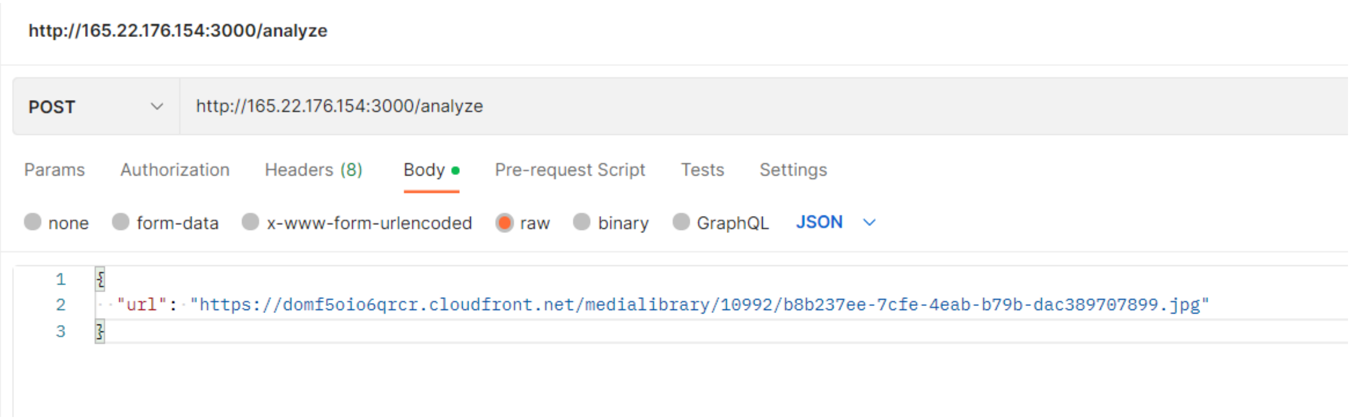
   Tools required to check this API: Postman

   Steps to run in Postman:

   - Open POSTMAN
   - Method Type- POST
   - Image URL- https://domf5oio6qrcr.cloudfront.net/medialibrary/10992/b8b237ee-7cfe-4eab-b79b-dac389707899.jpg
   - Copy and paste the link-http://165.22.176.154:3000/analyze
   - In Body choose Raw and JSON format.

   Paste following: {

   "url": "https://domf5oio6qrcr.cloudfront.net/medialibrary/10992/b8b237ee-7cfe-4eab-b79b-dac389707899.jpg" }



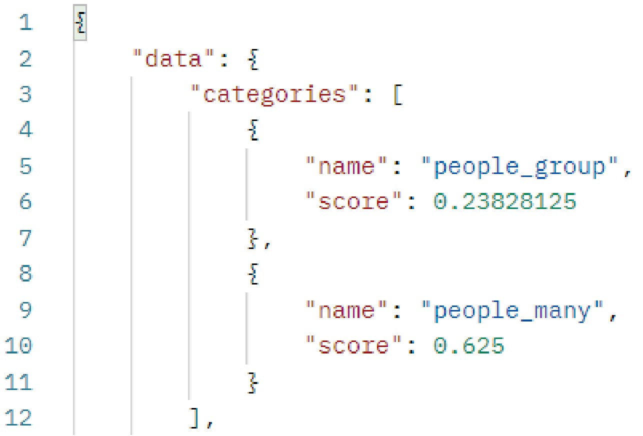
http://165.22.176.154:3000/analyze

http://165.22.176.154:3000/analyze

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

```
{
  "url": "https://domf5oio6qrcr.cloudfront.net/medialibrary/10992/b8b237ee-7cfe-4eab-b79b-dac389707899.jpg"
}
```

- You will get the following output that will analyze the image.



```
{
    "data": {
        "categories": [
            {
                "name": "people_group",
                "score": 0.23828125
            },
            {
                "name": "people_many",
                "score": 0.625
            }
        ],
```

- Similarly, you can try different Image URL to check this Endpoint.

3. **Describe Image:** A range of visual elements can be extracted from your photographs using the Computer Vision Image Analysis service. With the use of entire phrases, this process creates a human readable description of an image. A set of content tags that are also returned by the operation serve as the foundation for the description. For each image, multiple descriptions may be generated. The confidence score of each description determines its order. Descriptions are all written in English.

   API ENDPOINT- The API analyzes images with AI algorithms.

   http://165.22.176.154:3000/describe

   Tools required to check this API: Postman

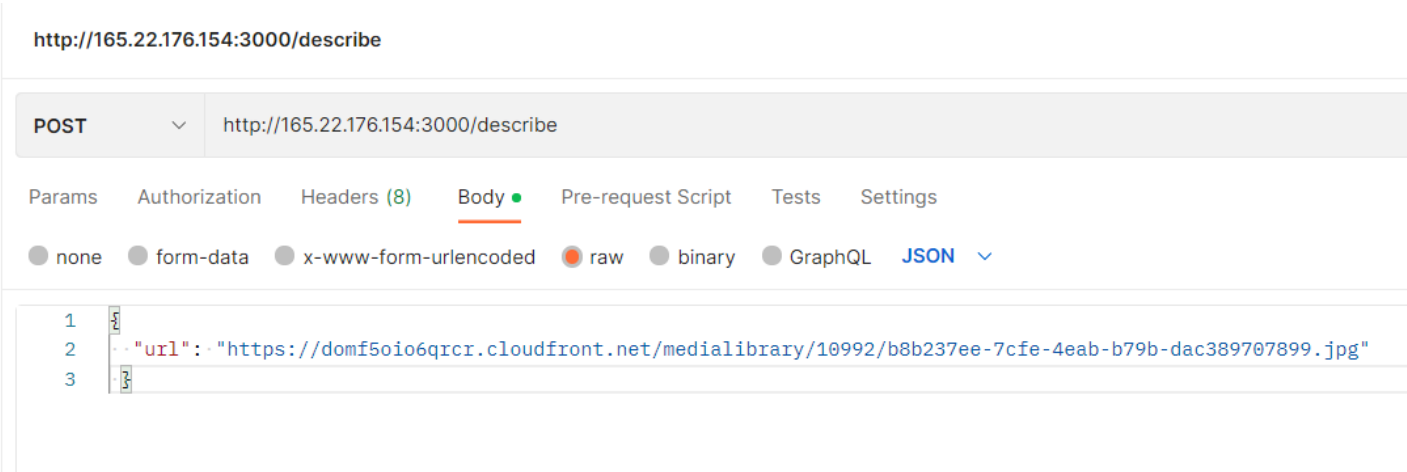
   Steps to run in Postman:

   - Open POSTMAN
   - Method Type- POST

- Copy and paste the link-http://165.22.176.154:3000/describe
- Image URL- https://domf5oio6qrcr.cloudfront.net/medialibrary/10992/b8b237ee-7cfe-4eab-b79b-dac389707899.jpg
- In Body choose Raw and JSON format.

Paste following: {

"url": “https://domf5oio6qrcr.cloudfront.net/medialibrary/10992/b8b237ee-7cfe-4eab-b79b-dac389707899.jpg” }

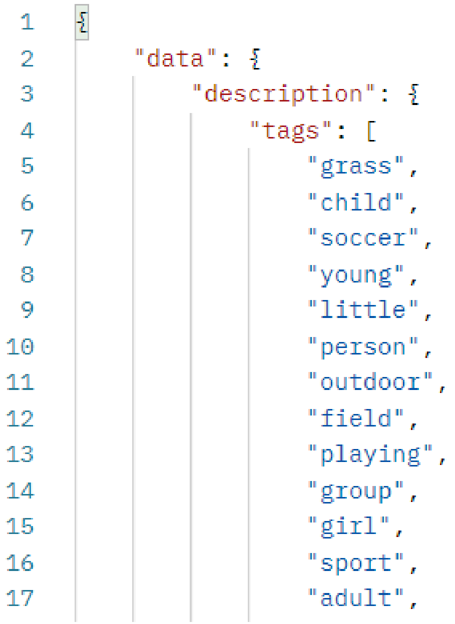


http://165.22.176.154:3000/describe

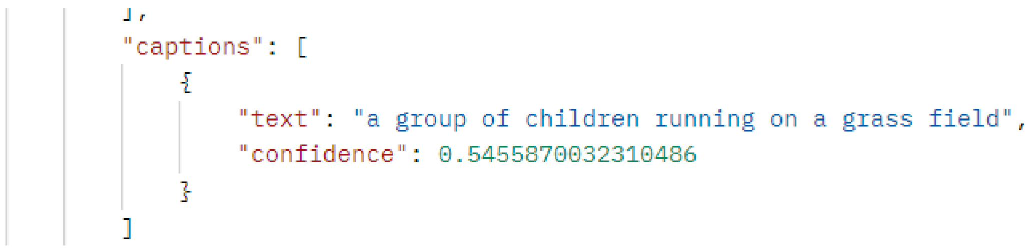
POST http://165.22.176.154:3000/describe

```
{
  "url": "https://domf5oio6qrcr.cloudfront.net/medialibrary/10992/b8b237ee-7cfe-4eab-b79b-dac389707899.jpg"
}
```

- You will get the following output that will analyze the image.



```
{
    "data": {
        "description": {
            "tags": [
                "grass",
                "child",
                "soccer",
                "young",
                "little",
                "person",
                "outdoor",
                "field",
                "playing",
                "group",
                "girl",
                "sport",
                "adult",
```



```
        ],
        "captions": [
            {
                "text": "a group of children running on a grass field",
                "confidence": 0.5455870032310486
            }
        ]
```

- Similarly, you can try different Image URL to check this Endpoint.

4. **Detect Objects:** A range of visual elements can be extracted from your photographs using the Computer Vision Image Analysis service. This operation Performs object detection on the specified image.

   A successful response will be returned in JSON

   API ENDPOINT- The API analyzes images with AI algorithms.

   http://165.22.176.154:3000/detect

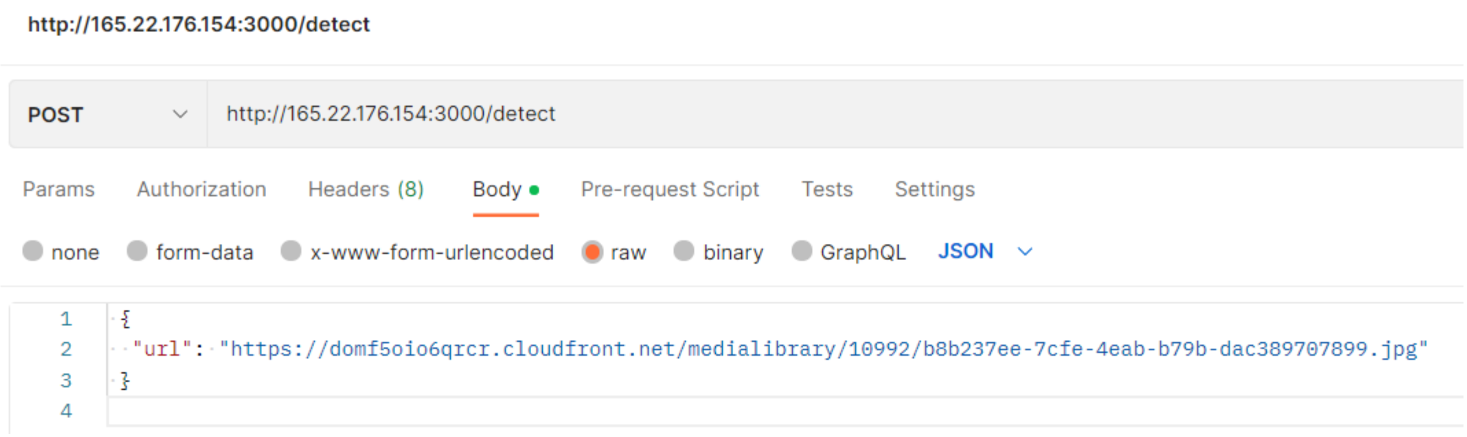
   Tools required to check this API: Postman

   Steps to run in Postman:

   - Open POSTMAN
   - Method Type- POST
   - Copy and paste the link- http://165.22.176.154:3000/detect
   - Image URL- https://domf5oio6qrcr.cloudfront.net/medialibrary/10992/b8b237ee-7cfe-4eab-b79b-dac389707899.jpg
   - In Body choose Raw and JSON format.

   Paste following: {

   "url": “https://domf5oio6qrcr.cloudfront.net/medialibrary/10992/b8b237ee-7cfe-4eab-b79b-dac389707899.jpg” }



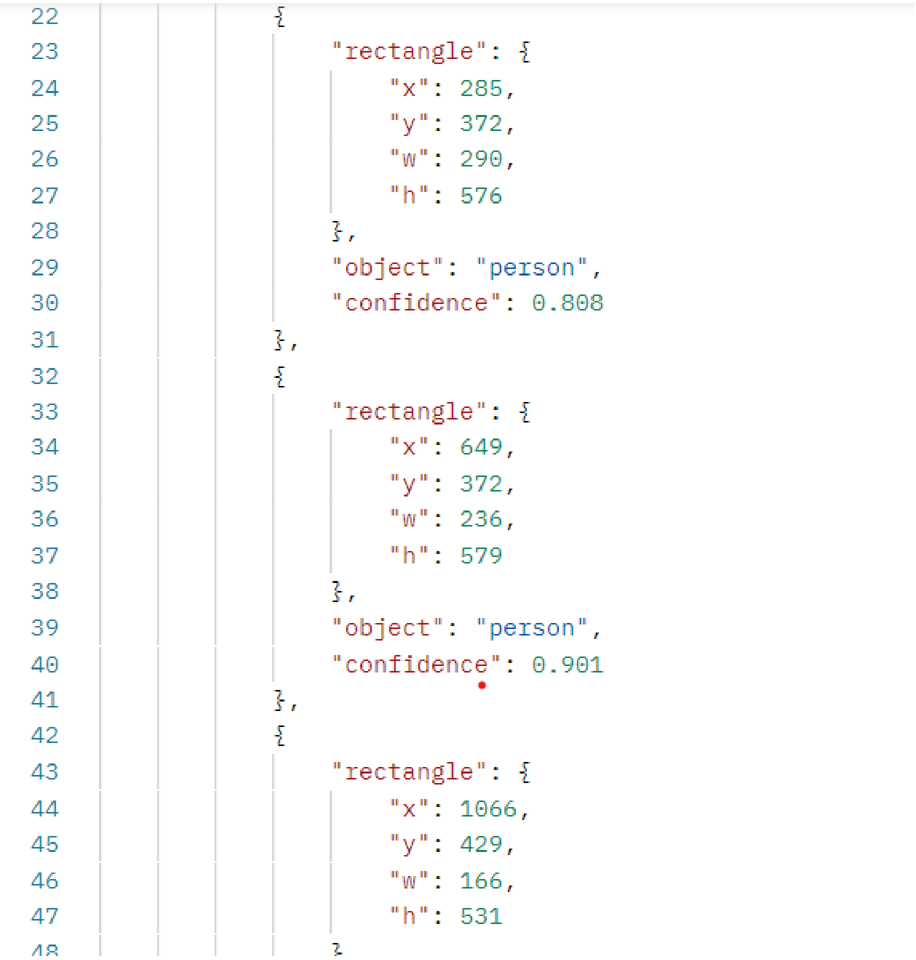
http://165.22.176.154:3000/detect

POST http://165.22.176.154:3000/detect

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

```
{
  "url": "https://domf5oio6qrcr.cloudfront.net/medialibrary/10992/b8b237ee-7cfe-4eab-b79b-dac389707899.jpg"
}
```

- You will get the following output that will detect the objects in the Image.



```
                {
                    "rectangle": {
                        "x": 285,
                        "y": 372,
                        "w": 290,
                        "h": 576
                    },
                    "object": "person",
                    "confidence": 0.808
                },
                {
                    "rectangle": {
                        "x": 649,
                        "y": 372,
                        "w": 236,
                        "h": 579
                    },
                    "object": "person",
                    "confidence": 0.901
                },
                {
                    "rectangle": {
                        "x": 1066,
                        "y": 429,
                        "w": 166,
                        "h": 531
                    }
```

- Similarly, you can try different Image URL to check this Endpoint.

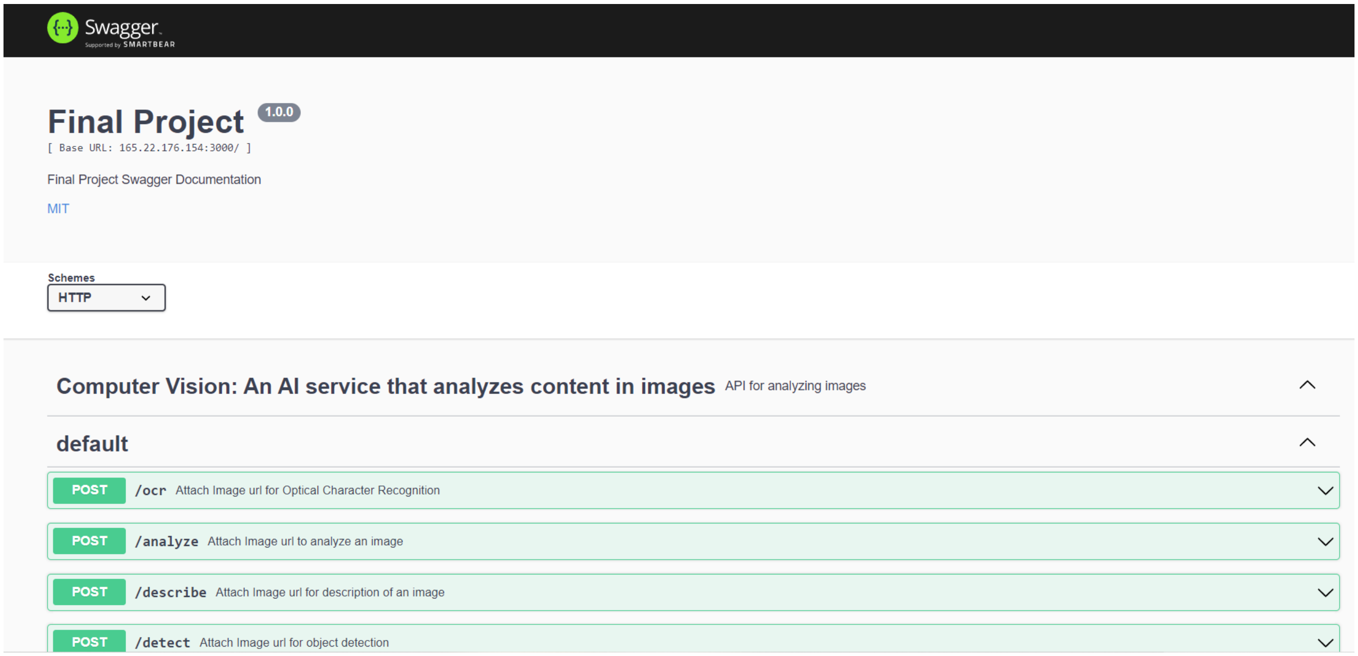
We have used same image for Detect, Describe, and analyze endpoints but the output is different for each API.

Swagger API Documentation for all responses of endpoints:

http://165.22.176.154:3000/api-docs/

Steps to check the API endpoints in Swagger:

- Click on above link mentioned.



Final Project 1.0.0

[Base URL: 165.22.176.154:3000/]

Final Project Swagger Documentation

MIT

Schemes: HTTP

Computer Vision: An AI service that analyzes content in images API for analyzing images

default

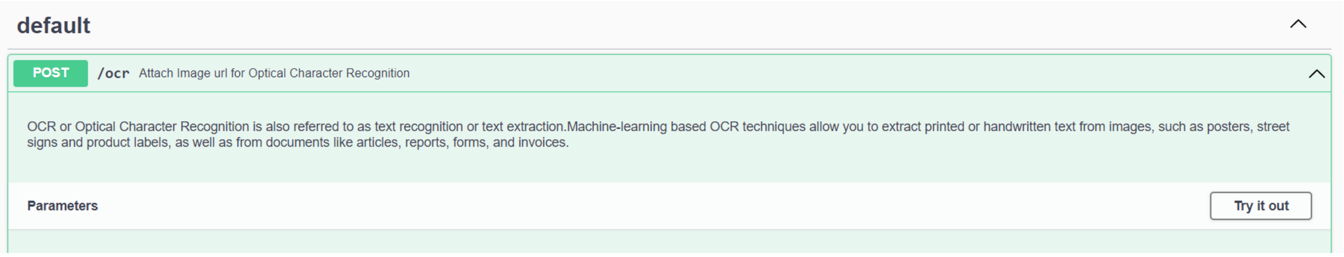
POST /ocr Attach Image url for Optical Character Recognition

POST /analyze Attach Image url to analyze an image

POST /describe Attach Image url for description of an image

POST /detect Attach Image url for object detection

- Click on POST /ocr and then click “Try it out”



default

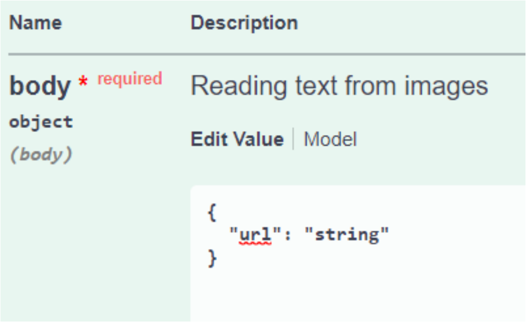
POST /ocr Attach Image url for Optical Character Recognition

OCR or Optical Character Recognition is also referred to as text recognition or text extraction.Machine-learning based OCR techniques allow you to extract printed or handwritten text from images, such as posters, street signs and product labels, as well as from documents like articles, reports, forms, and invoices.

Parameters

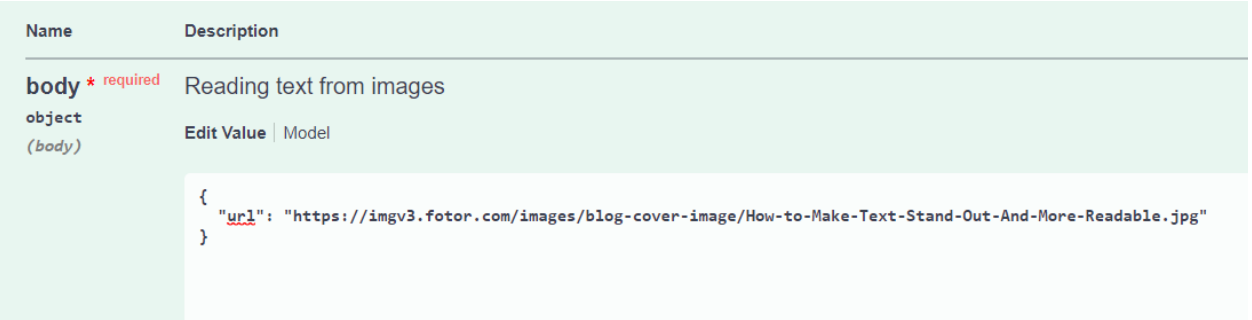
- In the body, in place of string, copy following image URL-

https://imgv3.fotor.com/images/blog-cover-image/How-to-Make-Text-Stand-Out-And-More-Readable.jpg



Name	Description
body * required object *(body)*	Reading text from images **Edit Value** | Model

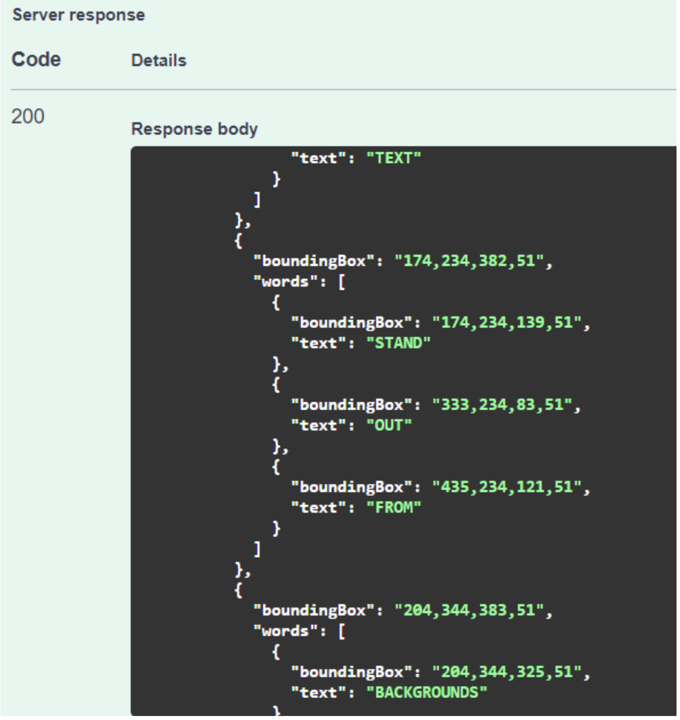
```
{
  "url": "string"
}
```



Name	Description
body * required object *(body)*	Reading text from images **Edit Value** | Model

```
{
  "url": "https://imgv3.fotor.com/images/blog-cover-image/How-to-Make-Text-Stand-Out-And-More-Readable.jpg"
}
```

- Click on Execute.
- You will see following Output.



Server response

Code	Details
200	Response body

```
          "text": "TEXT"
        }
      ]
    },
    {
      "boundingBox": "174,234,382,51",
      "words": [
        {
          "boundingBox": "174,234,139,51",
          "text": "STAND"
        },
        {
          "boundingBox": "333,234,83,51",
          "text": "OUT"
        },
        {
          "boundingBox": "435,234,121,51",
          "text": "FROM"
        }
      ]
    },
    {
      "boundingBox": "204,344,383,51",
      "words": [
        {
          "boundingBox": "204,344,325,51",
          "text": "BACKGROUNDS"
```

Similarly, you can try all the other endpoints mentioned in the Swagger documentation.