AI & Machine Learning Products (https://cloud.google.com/products/machine-learning/)
Cloud Vision API (https://cloud.google.com/vision/)
Documentation (https://cloud.google.com/vision/docs/) Guides

Quickstart: Using the command line

This page shows you how to send three feature detection and annotation requests to Cloud Vision using the REST interface (https://cloud.google.com/vision/docs//reference/rest) and the `curl` command.

Cloud Vision API enables easy integration of Google vision recognition technologies into developer applications. You can send image data and desired feature types to the Vision API, which then returns a corresponding response based on the image attributes you are interested in. For more information about the feature types offered, see the List of all Vision API features (https://cloud.google.com/vision/docs//features-list).

Before you begin

1. Sign in (https://accounts.google.com/Login) to your Google Account.

   If you don't already have one, sign up for a new account (https://accounts.google.com/SignUp).

2. In the Cloud Console, on the project selector page, select or create a Google Cloud project.

★ **Note**: If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

GO TO THE PROJECT SELECTOR PAGE (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/PROJECTSELECT

3. Make sure that billing is enabled for your Google Cloud project. Learn how to confirm billing is enabled for your project (https://cloud.google.com/billing/docs/how-to/modify-project).
4. Enable the Vision API.

   ENABLE THE API (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/FLOWS/ENABLEAPI?APIID=VISION.GO

5. Set up authentication:

   a. In the Cloud Console, go to the **Create service account key** page.

GO TO THE CREATE SERVICE ACCOUNT KEY PAGE (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/

b. From the **Service account** list, select **New service account**.

c. In the **Service account name** field, enter a name.

d. From the **Role** list, select **Project** > **Owner**.

★ **Note**: The **Role** field authorizes your service account to access resources. You can view and change this field later by using the Cloud Console (https://console.cloud.google.com/). If you are developing a production app, specify more granular permissions than **Project > Owner**. For more information, see granting roles to service accounts (https://cloud.google.com/iam/docs/granting-roles-to-service-accounts).

e. Click **Create**. A JSON file that contains your key downloads to your computer.

6. Set the environment variable ***GOOGLE_APPLICATION_CREDENTIALS*** to the file path of the JSON file that contains your service account key. This variable only applies to your current shell session, so if you open a new session, set the variable again.

Example: Linux or macOS

Replace ***[PATH]*** with the file path of the JSON file that contains your service account key.

```
export GOOGLE_APPLICATION_CREDENTIALS="[PATH]"
```

For example:

```
export GOOGLE_APPLICATION_CREDENTIALS="/home/user/Downloads/service-account-fil
```

Example: Windows

Replace ***[PATH]*** with the file path of the JSON file that contains your service account key, and ***[FILE_NAME]*** with the filename.

With PowerShell:

```
$env:GOOGLE_APPLICATION_CREDENTIALS="[PATH]"
```

For example:

```
$env:GOOGLE_APPLICATION_CREDENTIALS="C:\Users\username\Downloads\[FILE_NAME].js
```

With command prompt:

```
set GOOGLE_APPLICATION_CREDENTIALS=[PATH]
```

7. Install and initialize the Cloud SDK (https://cloud.google.com/sdk/docs/).

Make an image annotation request

After completing the Before you begin (#before-you-begin) steps you can use Vision API to annotate an image file.

In this example you use curl to send a request to the Vision API using the following image:

Cloud Storage URI:

```
gs://cloud-samples-data/vision/using_curl/shanghai.jpeg
```

HTTPS URL:

```
https://console.cloud.google.com/storage/browser/cloud-samples-data/vision/using_cur
```

Image credit: Steve Long (https://unsplash.com/photos/jNGs6ys7MKk) on Unsplash (https://unsplash.com/).

Create the request JSON

The following `request.json` file demonstrates how to request three `images:annotate` features (https://cloud.google.com/vision/docs/features-list) and limit the results in the response.

Create the JSON request file with the following text, and save it as a `request.json` plain text file in your working directory:

`request.json`

```
{
  "requests": [
    {
      "image": {
        "source": {
          "imageUri": "gs://cloud-samples-data/vision/using_curl/shanghai.jpeg"
```

```
        }
      },
      "features": [
        {
          "type": "LABEL_DETECTION",
          "maxResults": 3
        },
        {
          "type": "OBJECT_LOCALIZATION",
          "maxResults": 1
        },
        {
          "type": "TEXT_DETECTION",
          "maxResults": 1,
          "model": "builtin/latest"
        }
      ]
    }
  ]
}
```

Field value details

- `image.source.gcsImageUri` - Indicates the image stored in a Google Cloud Storage bucket. You change this request to `image.source.imageUri` for a publicly available URI, or `image.content` to pass a base64 encoded string representation of an image.
- `features` - An object representing a specific feature type. You can request multiple feature types for a single image.
  - `type` - The enum value (https://cloud.google.com/vision/docs/reference/rest/v1/Feature#type) specifying a feature.
  - `maxResults` (optional) - A limiting value on the results returned.
  - `model` (optional) - If applicable you can specify either `builtin/stable` (the default if unset) or `builtin/latest` to choose your model. Refer to the Release notes (https://cloud.google.com/vision/docs/release-notes) topic for a list of recently updated models.

Send the request

You use curl and the body content from `request.json` to send the request to the Cloud Vision API. Enter the following on your command line:

```
curl -X POST \
-H "Authorization: Bearer "$(gcloud auth application-default print-access-token) \
-H "Content-Type: application/json; charset=utf-8" \
https://vision.googleapis.com/v1/images:annotate -d @request.json
```

- The sample `curl` command uses the `gcloud auth application-default print-access-token` command to get an authentication token.
- Note that to pass a filename to `curl` you use the `-d` option (for "data") and precede the filename with an `@` sign. This file should be in the same directory in which you execute the `curl` command.

Interpret the response

You should see a JSON response similar to the one below.

The request JSON body specified `maxResults` for each annotation type. Consequently, you will see the following in the response JSON:

- three `labelAnnotations` results (#label-results)
- one `textAnnotations` result (#text-results) (shortened for clarity)
- one `localizedObjectAnnotations` result (#object-results)

Response

Note: Zero coordinate values omitted. When the API detects a coordinate ("x" or "y") value of 0, ***that coordinate is omitted in the JSON response***. Thus, a response with a bounding poly around the entire image would be
[{},{"x": 100},{"x": 100,"y": 100},{"y": 100}] for an image that is 100px by 100px. For more information, see the API Reference documentation (https://cloud.google.com/vision/docs/reference/rest/v1/images/annotate#boundingpoly).

```
{
  "responses": [
```

```
{
  "labelAnnotations": [
    {
      "mid": "/m/09g5pq",
      "description": "People",
      "score": 0.9504782,
      "topicality": 0.9504782
    },
    {
      "mid": "/m/01c8br",
      "description": "Street",
      "score": 0.8911568,
      "topicality": 0.8911568
    },
    {
      "mid": "/m/079bkr",
      "description": "Mode of transport",
      "score": 0.89089024,
      "topicality": 0.89089024
    }
  ],
  "textAnnotations": [
    {
      "locale": "zh",
      "description": "□□□□\n",
      "boundingPoly": {
        "vertices": [
          {
            "x": 159,
            "y": 212
          },
          {
            "x": 947,
            "y": 212
          },
          {
            "x": 947,
            "y": 354
          },
          {
            "x": 159,
            "y": 354
          }
        ]
      }
```

```
    },
    ...
  ],
  "fullTextAnnotation": {
    "pages": [
      {
        ...
            "paragraphs": [
              {
                ...
                "words": [
                  {
                    ...
                    "symbols": [
                      {
                        ...
            ],
            "blockType": "TEXT"
          }
        ]
      }
    ],
    "text": "□□□□\n"
  },
  "localizedObjectAnnotations": [
    {
      "mid": "/m/01g317",
      "name": "Person",
      "score": 0.94413143,
      "boundingPoly": {
        "normalizedVertices": [
          {
            "x": 0.26063988,
            "y": 0.46869153
          },
          {
            "x": 0.40736017,
            "y": 0.46869153
          },
          {
            "x": 0.40736017,
            "y": 0.8957791
          },
          {
            "x": 0.26063988,
```

```
                "y": 0.8957791
              }
            ]
          }
        }
      ]
    }
  ]
}
```

Label detection results

1. description: "People", score: 0.950
2. description: "Street", score: 0.891
3. description: "Mode of transport", score: 0.890

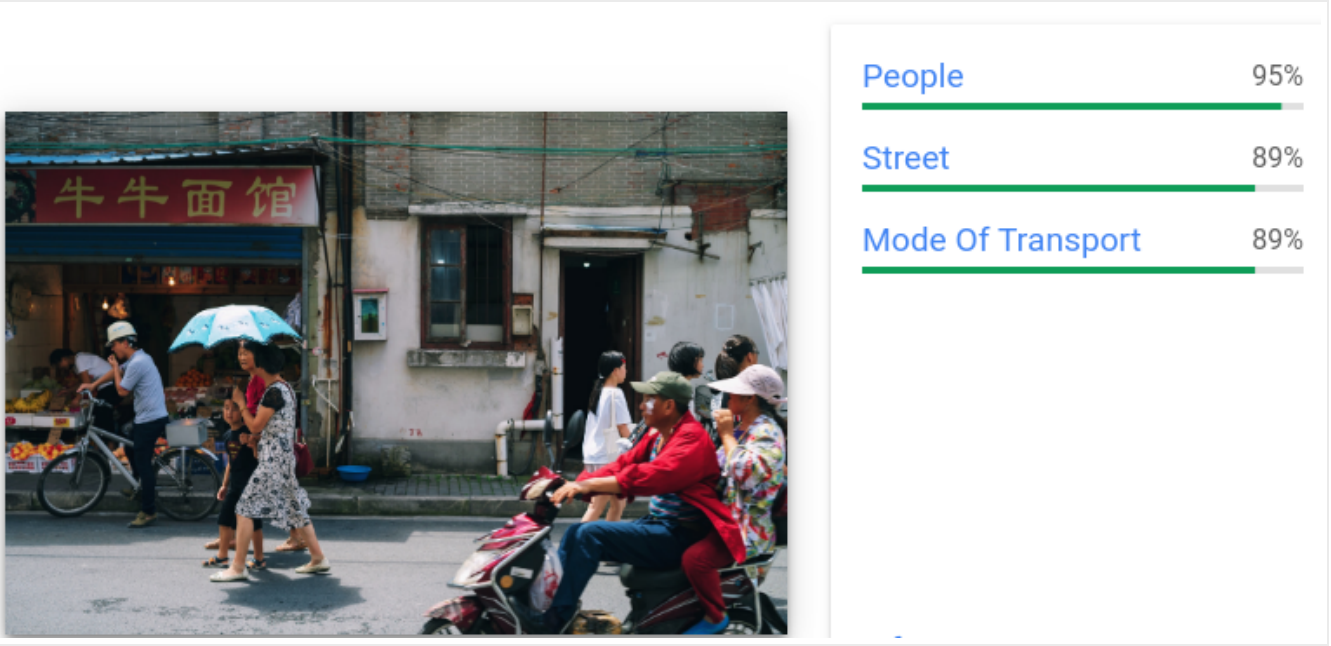

Image credit: Steve Long (https://unsplash.com/photos/jNGs6ys7MKk) on Unsplash (https://unsplash.com/) (*annotations added*).

Text detection results

- text: □□□□\n
- vertices: (x: 159, y: 212), (x: 947, y: 212), (x: 947, y: 354), (x: 159, y: 354)

Image credit: Steve Long (https://unsplash.com/photos/jNGs6ys7MKk) on Unsplash (https://unsplash.com/) (*annotations added*).

Object detection results

- name: "Person", score: 0.944
- normalized vertices: (x: 0.260, y: 0.468), (x: 0.407, y: 0.468), (x: 0.407, y: 0.895), (x: 0.260, y: 0.895)

Image credit: Steve Long (https://unsplash.com/photos/jNGs6ys7MKk) on Unsplash (https://unsplash.com/) (*annotations added*).

Congratulations! You've sent your first request to Vision API.

What's next

- See a list of all feature types (https://cloud.google.com/vision/docs/features-list) and their uses.
- Get started with the Vision API in your language of choice by using a Vision API Client Library (https://cloud.google.com/vision/docs/quickstart-client-libraries).
- Use the How-to guides (https://cloud.google.com/vision/docs/how-to) to learn more about specific features, see example annotations, and get annotations for an individual file or image.
- Learn about batch image (https://cloud.google.com/vision/docs/batch) and file (https://cloud.google.com/vision/docs/file-small-batch) (PDF/TIFF/GIF) annotation.
- Work through the sample applications (https://cloud.google.com/vision/docs/samples).

- Browse more specific use cases on the community tutorials (https://cloud.google.com/community/tutorials?q=vision) page.

Last updated November 6, 2019.