

```
# Install the Google Cloud Vision library
!pip install --upgrade google-cloud-vision
```

```
from google.cloud import vision
from google.colab import files
import os
import io
from PIL import Image, ImageDraw
import re
```

```
Requirement already satisfied: google-cloud-vision in /usr/local/lib/python3.10/dist-packages (3.7.4)
Requirement already satisfied: google-api-core!=2.0.*,!=2.1.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.* in /usr/local/lib/python3.10/dist-packages (from google-cloud-vision) (2.14.1)
Requirement already satisfied: google-auth!=2.24.0,!=2.25.0,<3.0.0dev,=>2.14.1 in /usr/local/lib/python3.10/dist-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*->google-api-core) (2.24.0)
Requirement already satisfied: proto-plus<2.0.0dev,=>1.22.3 in /usr/local/lib/python3.10/dist-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*->google-api-core) (1.22.3)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,=>3.20.2 in /usr/local/lib/python3.10/dist-packages (from proto-plus<2.0.0dev,=>1.22.3->google-api-core) (3.20.2)
Requirement already satisfied: googleapis-common-protos<2.0.dev0,=>1.56.2 in /usr/local/lib/python3.10/dist-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*->google-api-core) (1.56.2)
Requirement already satisfied: requests<3.0.0.dev0,=>2.18.0 in /usr/local/lib/python3.10/dist-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*->google-api-core) (2.18.0)
Requirement already satisfied: grpcio<2.0dev,=>1.33.2 in /usr/local/lib/python3.10/dist-packages (from google-api-core[grpc]!=2.0.*,!=2.1.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*->google-api-core) (1.33.2)
Requirement already satisfied: grpcio-status<2.0.dev0,=>1.33.2 in /usr/local/lib/python3.10/dist-packages (from google-api-core[grpc]!=2.0.*,!=2.1.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*->google-api-core) (1.33.2)
Requirement already satisfied: cachetools<6.0,=>2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth!=2.24.0,!=2.25.0,<3.0.0dev,=>2.14.1) (2.0.0)
Requirement already satisfied: pyasn1-modules<0.4.0,=>0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth!=2.24.0,!=2.25.0,<3.0.0dev,=>2.14.1) (0.2.1)
Requirement already satisfied: rsa<5,=>3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth!=2.24.0,!=2.25.0,<3.0.0dev,=>2.14.1) (3.1.4)
Requirement already satisfied: pyasn1<0.7.0,=>0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules<0.4.0,=>0.2.1->google-auth) (0.4.6)
Requirement already satisfied: charset-normalizer<4,=>2 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0.dev0,=>2.18.0) (2.0.0)
Requirement already satisfied: idna<4,=>2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0.dev0,=>2.18.0) (2.5)
Requirement already satisfied: urllib3<3,=>1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0.dev0,=>2.18.0) (1.21.1)
Requirement already satisfied: certifi<2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0.dev0,=>2.18.0) (2017.4.17)
```

```
# Upload the JSON key for Google Cloud Vision API
uploaded = files.upload()
key_file = next(iter(uploaded))
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = key_file
```

 No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to

```
# Create a client for the Vision API
client = vision.ImageAnnotatorClient()

def detect_text(path):
    """Detects text in the file."""
    with io.open(path, 'rb') as image_file:
        content = image_file.read()

    image = vision.Image(content=content)

    # You can switch between text_detection (for sparse text) and document_text_detection (for dense text)
    response = client.document_text_detection(image=image)
    texts = response.text_annotations

    if response.error.message:
        raise Exception(f'{response.error.message}\nFor more info on error messages, check: https://cloud.google.com/apis/design/errors')

    return texts
```

```
# Upload an image file
uploaded = files.upload()
image_path = next(iter(uploaded))
```

```
# Display the uploaded image
image = Image.open(image_path)
display(image)
```



```
# Use the detect_text function
texts = detect_text(image_path)
```

```
# Print detected text
for text in texts:
    print(text.description)
```

```
AX
ARMANI EXCHANGE
shutterstock.com 2014520585
AX
ARMANI
EXCHANGE
shutterstock.com
2014520585
```

```
# Optionally, draw bounding boxes around detected texts
```

```
def draw_boxes(image, bounds, color='yellow', width=2):
    draw = ImageDraw.Draw(image)
    for bound in bounds:
        p0, p1, p2, p3 = bound.vertices
        draw.line([
            (p0.x, p0.y), (p1.x, p1.y),
            (p1.x, p1.y), (p2.x, p2.y),
            (p2.x, p2.y), (p3.x, p3.y),
            (p3.x, p3.y), (p0.x, p0.y)], fill=color, width=width)
    return image
```

```
# Extract bounding box information (only if necessary)
```

```
bounds = [text.bounding_poly for text in texts[1:]] # Skip the first element, which contains all text
drawn_image = draw_boxes(image, bounds)
display(drawn_image)
```