

# main

May 1, 2022

## 1 Keep track of your pictures: how to automatically add keywords + how to build and maintain an image database

Author: Federica Lionetto  
Email: federica.lionetto@gmail.com  
Date: 30 April 2022

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

### 1.1 Objective

Let's consider the following scenario. You love taking pictures and always have your camera with you to be ready to capture the best moments of life. However, most of your pictures simply go from your memory card to your external hard drive and you would have no idea where to find what. You considered several times the possibility to add keywords to your pictures to be able to easily find them later on. Maybe you even started with that, but gave up after a few attempts. Does this sound familiar to you?

Generating keywords is a time-consuming task, and is likely to be very boring as well. Instead of generating keywords manually, in this short tutorial we will draw on the power of the cloud and let the **Vision AI API** on Google Cloud solve this repetitive task for us. We will also go through a way to keep our pictures organized into a database on Google Cloud.

For each image in our collection, we will store the following information in the database: - **file\_name**, that is, the name of the image - **creation\_date\_time**, that is, the date and time of the creation of the image - **keywords**, that is, the keywords associated with the image

It's time to get started!

### 1.2 Cloud settings

To run the tutorial, we need the following resources on Google Cloud: - a **Cloud Storage** bucket, where we will upload the pictures - a **BigQuery** dataset, where we will store the information about the pictures - a **Vertex AI** notebook, where we will write and run our Python code

You can create the **Cloud Storage** bucket, the **BigQuery** dataset, and the **Vertex AI** notebook from the Console.

### 1.3 Step 1: Import modules

```
[1]: import io
import os
import shutil

import datetime

import matplotlib.pyplot as plt
import seaborn as sns

from IPython.display import Image, display

# Imports the Google Cloud client library
from google.cloud import vision
from google.cloud import storage
from google.cloud import bigquery

# %load_ext google.cloud.bigquery
```

### 1.4 Step 2: Configuration

Here we can configure some of the variables that are used throughout the tutorial.

```
[2]: verbose = True

region = "[REGION GOES HERE]" # Where to create the resources on Google Cloud
gcp_project_name = "[GCP PROJECT NAME GOES HERE]" # The name of the GCP project

bucket_name = "[BUCKET NAME GOES HERE]" # The name of the Cloud Storage bucket
folder_name_landing = "image/landing" # The name of the folder in the Cloud
    ↪Storage bucket where new pictures are uploaded
folder_name_archive = "image/archive" # The name of the folder in the Cloud
    ↪Storage bucket where processed pictures are archived

scale = 0.2 # Used to scale images for easier visualization

bq_dataset_name = '[DATASET NAME GOES HERE]' # The name of the BigQuery dataset
bq_table_name = 'image' # The name of the BigQuery table

sample_image_name = '[SAMPLE IMAGE NAME GOES HERE]' # A sample image to use as
    ↪an example
```

```
[3]: verbose = True

region = "europe-west6" # Where to create the resources on Google Cloud
gcp_project_name = "personalproject-348318" # The name of the GCP project
```

```

bucket_name = "whitebloomingtulip-input-164" # The name of the Cloud Storage
    ↵bucket
folder_name_landing = "image/landing" # The name of the folder in the Cloud
    ↵Storage bucket where new pictures are uploaded
folder_name_archive = "image/archive" # The name of the folder in the Cloud
    ↵Storage bucket where processed pictures are archived

scale = 0.2 # Scale images for easier visualization

bq_dataset_name = 'whitebloomingtulip_db' # The name of the BigQuery dataset
bq_table_name = 'image' # The name of the BigQuery table

sample_image_name = 'IMG_3579.jpeg' # A sample image to use as an example

```

## 1.5 3: Instantiate the clients

We need to instantiate three clients: one for Cloud Storage, one for BigQuery, and one for the Vision AI API.

```
[4]: # Instantiate the clients
storage_client = storage.Client()
vision_client = vision.ImageAnnotatorClient()
bq_client = bigquery.Client()
```

## 1.6 4: Copy images from GCS to the machine

```
[5]: # Create 2 folders (if they do not exist), one for images and one for keywords
if not os.path.exists('images'):
    os.mkdir('images')
if not os.path.exists('keywords'):
    os.mkdir('keywords')

# Create an empty list for the images to be copied
file_names = []

# Access the GCS bucket containing the images to be copied
bucket = storage_client.get_bucket(bucket_name)

# If the images are in a subfolder of the GCS bucket, specify the subfolder
    ↵structure
prefix_landing = f"{folder_name_landing}/"
blobs_landing = bucket.list_blobs(prefix = prefix_landing, delimiter = '/')

for blob in blobs_landing:
    if(blob.name != prefix_landing): # Ignore the subfolder itself
        file_name = blob.name.replace(prefix_landing, "")
```

```

blob.download_to_filename('images/' + file_name) # Download the file to
→the machine
file_names.append('images/' + file_name)

print("Images:")
print(file_names)
print('')

```

Images:

- 'images/IMG\_3579.jpeg', 'images/IMG\_3586.jpeg', 'images/IMG\_3609.jpeg',
- 'images/IMG\_3623.jpeg', 'images/IMG\_3638.jpeg', 'images/IMG\_3757.jpeg',
- 'images/IMG\_3793.jpeg', 'images/IMG\_3805.jpeg', 'images/IMG\_3841.jpeg',
- 'images/IMG\_3850.jpeg'

## 1.7 5: Generate keywords and display their score

The Vision AI API allows to annotate an image with keywords that describe the contents of that image. Each keyword has an associated score, where a higher score means that the algorithm has a higher confidence that the keyword describes something in the image.

You can try out the Vision AI API interactively here: <https://cloud.google.com/vision>

```
[6]: label_annotation_desc_dict = {}
label_annotation_score_dict = {}
# file_names_out = []

for file_name in file_names:
    # Get the two parts of the file name
    file_name_without_extension = file_name.rsplit('.', 1)[0]
    file_name_extension = file_name.rsplit('.', 1)[1]
    if verbose:
        print('File name without extension:', file_name_without_extension)
        print('File name extension:', file_name_extension)
        print('')

    # Display selected image
    display(Image(filename=file_name, width=500))

    # Annotate selected image
    file_name = os.path.abspath(file_name)

    # Load the image into memory
    with io.open(file_name, 'rb') as image_file:
        content = image_file.read()

    image = vision.Image(content=content)
```

```

# Perform label detection on the image
response = vision_client.label_detection(image=image)
if verbose:
    print('Response:')
    print(response)
    print('')
    print('Label annotations:')
    print(response.label_annotations)
    print('')
    print('First element of label annotations:')
    print(response.label_annotations[0])
    print('')
    print('Description of first element of label annotations:')
    print(response.label_annotations[0].description)
    print('')

labels = response.label_annotations

print('Keywords:')
for label in labels:
    print(label.description)
print('')

# Create lists of description and score for selected image
n_label_annotations = len(response.label_annotations)

label_annotation_desc = []
label_annotation_score = []

for i in range(n_label_annotations):
    label_annotation_desc.append(response.label_annotations[i].description)
    label_annotation_score.append(response.label_annotations[i].score)
print('List of keywords:')
print(label_annotation_desc)
print('')
print('List of scores:')
print(label_annotation_score)
print('')

label_annotation_desc_dict[file_name] = label_annotation_desc
label_annotation_score_dict[file_name] = label_annotation_score

# Display label annotations (description and score) for selected image
plt.figure()
sns.barplot(x=label_annotation_score, y=label_annotation_desc, color='red')
plt.savefig(file_name_without_extension.replace('images/', 'keywords/
˓→')+'_keywords', format='png')

```

```

plt.show()

# Create filename based on label annotations
# file_name_out = "_".join(label_annotation_desc)
# file_name_out = file_name_out.replace(" ", "-")
# file_name_out = file_name_out+".jpeg"
# print("Input file name:", file_name)
# print('')
# print("Output file name:", file_name_out)
# print('')
# file_names_out.append(file_name_out)

```

File name without extension: images/IMG\_3579  
 File name extension: jpeg



Response:

```

label_annotations {
  mid: "/m/0csby"
  description: "Cloud"
  score: 0.9827239513397217
  topicality: 0.9827239513397217
}
label_annotations {

```

```
    mid: "/m/09d_r"
    description: "Mountain"
    score: 0.9588522911071777
    topicality: 0.9588522911071777
}
label_annotations {
    mid: "/m/01bqvp"
    description: "Sky"
    score: 0.9572277665138245
    topicality: 0.9572277665138245
}
label_annotations {
    mid: "/m/078hm"
    description: "Slope"
    score: 0.8840519189834595
    topicality: 0.8840519189834595
}
label_annotations {
    mid: "/m/06_dn"
    description: "Snow"
    score: 0.8719691634178162
    topicality: 0.8719691634178162
}
label_annotations {
    mid: "/m/04b966"
    description: "Ice cap"
    score: 0.8418418169021606
    topicality: 0.8418418169021606
}
label_annotations {
    mid: "/m/025tn5c"
    description: "Terrain"
    score: 0.8227407932281494
    topicality: 0.8227407932281494
}
label_annotations {
    mid: "/m/03d28y3"
    description: "Natural landscape"
    score: 0.7953419089317322
    topicality: 0.7953419089317322
}
label_annotations {
    mid: "/m/0csh5"
    description: "Cumulus"
    score: 0.784989058971405
    topicality: 0.784989058971405
}
label_annotations {
```

```
    mid: "/m/025s3q0"
    description: "Landscape"
    score: 0.7753972411155701
    topicality: 0.7753972411155701
}
```

Label annotations:

```
[mid: "/m/0csby"
description: "Cloud"
score: 0.9827239513397217
topicality: 0.9827239513397217
, mid: "/m/09d_r"
description: "Mountain"
score: 0.9588522911071777
topicality: 0.9588522911071777
, mid: "/m/01bqvp"
description: "Sky"
score: 0.9572277665138245
topicality: 0.9572277665138245
, mid: "/m/078hm"
description: "Slope"
score: 0.8840519189834595
topicality: 0.8840519189834595
, mid: "/m/06_dn"
description: "Snow"
score: 0.8719691634178162
topicality: 0.8719691634178162
, mid: "/m/04b966"
description: "Ice cap"
score: 0.8418418169021606
topicality: 0.8418418169021606
, mid: "/m/025tn5c"
description: "Terrain"
score: 0.8227407932281494
topicality: 0.8227407932281494
, mid: "/m/03d28y3"
description: "Natural landscape"
score: 0.7953419089317322
topicality: 0.7953419089317322
, mid: "/m/0csh5"
description: "Cumulus"
score: 0.784989058971405
topicality: 0.784989058971405
, mid: "/m/025s3q0"
description: "Landscape"
score: 0.7753972411155701
topicality: 0.7753972411155701
```

]

First element of label annotations:

```
mid: "/m/0csby"
description: "Cloud"
score: 0.9827239513397217
topicality: 0.9827239513397217
```

Description of first element of label annotations:

Cloud

Keywords:

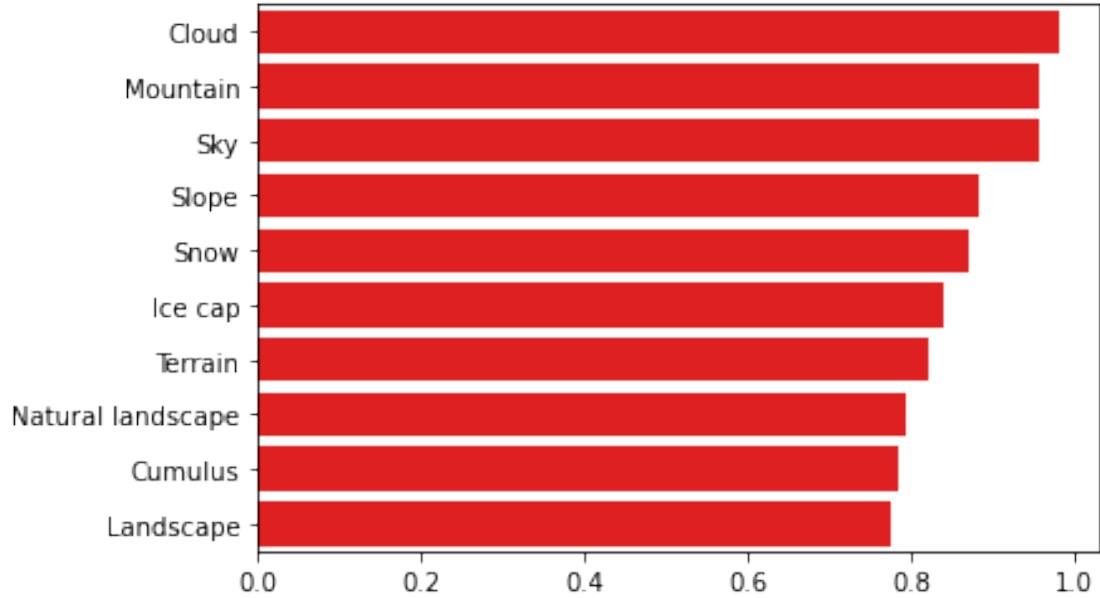
```
Cloud
Mountain
Sky
Slope
Snow
Ice cap
Terrain
Natural landscape
Cumulus
Landscape
```

List of keywords:

```
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural landscape', 'Cumulus', 'Landscape']
```

List of scores:

```
[0.9827239513397217, 0.9588522911071777, 0.9572277665138245, 0.8840519189834595,
0.8719691634178162, 0.8418418169021606, 0.8227407932281494, 0.7953419089317322,
0.784989058971405, 0.7753972411155701]
```



File name without extension: images/IMG\_3586

File name extension: jpeg



```
Response:  
label_annotations {  
    mid: "/m/01bqvp"  
    description: "Sky"  
    score: 0.9577324390411377  
    topicality: 0.9577324390411377  
}  
label_annotations {  
    mid: "/m/09d_r"  
    description: "Mountain"  
    score: 0.9488414525985718  
    topicality: 0.9488414525985718  
}  
label_annotations {  
    mid: "/m/06_dn"  
    description: "Snow"  
    score: 0.9056057929992676  
    topicality: 0.9056057929992676  
}  
label_annotations {  
    mid: "/m/05s2s"  
    description: "Plant"  
    score: 0.9005176424980164  
    topicality: 0.9005176424980164  
}  
label_annotations {  
    mid: "/m/03d28y3"  
    description: "Natural landscape"  
    score: 0.8744946122169495  
    topicality: 0.8744946122169495  
}  
label_annotations {  
    mid: "/m/078hm"  
    description: "Slope"  
    score: 0.8737100958824158  
    topicality: 0.8737100958824158  
}  
label_annotations {  
    mid: "/m/03cjrt"  
    description: "Highland"  
    score: 0.8499352335929871  
    topicality: 0.8499352335929871  
}  
label_annotations {  
    mid: "/m/07pw27b"  
    description: "Atmospheric phenomenon"  
    score: 0.8342433571815491  
    topicality: 0.8342433571815491
```

```

}

label_annotations {
    mid: "/m/025tn5c"
    description: "Terrain"
    score: 0.8180684447288513
    topicality: 0.8180684447288513
}
label_annotations {
    mid: "/m/07j7r"
    description: "Tree"
    score: 0.8042957186698914
    topicality: 0.8042957186698914
}

Label annotations:
[mid: "/m/01bqvp"
description: "Sky"
score: 0.9577324390411377
topicality: 0.9577324390411377
, mid: "/m/09d_r"
description: "Mountain"
score: 0.9488414525985718
topicality: 0.9488414525985718
, mid: "/m/06_dn"
description: "Snow"
score: 0.9056057929992676
topicality: 0.9056057929992676
, mid: "/m/05s2s"
description: "Plant"
score: 0.9005176424980164
topicality: 0.9005176424980164
, mid: "/m/03d28y3"
description: "Natural landscape"
score: 0.8744946122169495
topicality: 0.8744946122169495
, mid: "/m/078hm"
description: "Slope"
score: 0.8737100958824158
topicality: 0.8737100958824158
, mid: "/m/03cjrt"
description: "Highland"
score: 0.8499352335929871
topicality: 0.8499352335929871
, mid: "/m/07pw27b"
description: "Atmospheric phenomenon"
score: 0.8342433571815491
topicality: 0.8342433571815491

```

```
, mid: "/m/025tn5c"
description: "Terrain"
score: 0.8180684447288513
topicality: 0.8180684447288513
, mid: "/m/07j7r"
description: "Tree"
score: 0.8042957186698914
topicality: 0.8042957186698914
]
```

First element of label annotations:

```
mid: "/m/01bqvp"
description: "Sky"
score: 0.9577324390411377
topicality: 0.9577324390411377
```

Description of first element of label annotations:

Sky

Keywords:

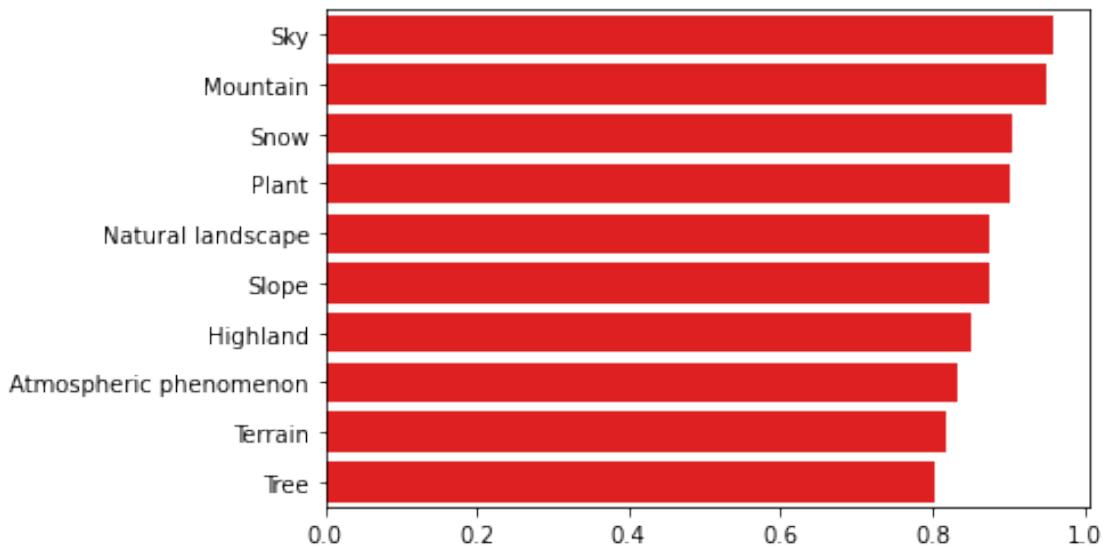
```
Sky
Mountain
Snow
Plant
Natural landscape
Slope
Highland
Atmospheric phenomenon
Terrain
Tree
```

List of keywords:

```
['Sky', 'Mountain', 'Snow', 'Plant', 'Natural landscape', 'Slope', 'Highland',
'Atmospheric phenomenon', 'Terrain', 'Tree']
```

List of scores:

```
[0.9577324390411377, 0.9488414525985718, 0.9056057929992676, 0.9005176424980164,
0.8744946122169495, 0.8737100958824158, 0.8499352335929871, 0.8342433571815491,
0.8180684447288513, 0.8042957186698914]
```



File name without extension: images/IMG\_3609

File name extension: jpeg



Response:

```
label_annotations {
    mid: "/m/0csby"
    description: "Cloud"
    score: 0.9777280688285828
    topicality: 0.9777280688285828
}
label_annotations {
    mid: "/m/01bqvp"
    description: "Sky"
    score: 0.9730981588363647
    topicality: 0.9730981588363647
}
label_annotations {
    mid: "/m/09d_r"
    description: "Mountain"
    score: 0.9400748014450073
    topicality: 0.9400748014450073
}
label_annotations {
    mid: "/m/06_dn"
    description: "Snow"
    score: 0.8958427906036377
    topicality: 0.8958427906036377
}
label_annotations {
    mid: "/m/078hm"
    description: "Slope"
    score: 0.8739033341407776
    topicality: 0.8739033341407776
}
label_annotations {
    mid: "/m/03d28y3"
    description: "Natural landscape"
    score: 0.8660182952880859
    topicality: 0.8660182952880859
}
label_annotations {
    mid: "/m/03cjrt"
    description: "Highland"
    score: 0.8601166009902954
    topicality: 0.8601166009902954
}
label_annotations {
    mid: "/m/09nm_"
    description: "World"
    score: 0.85157310962677
    topicality: 0.85157310962677
}
```

```
label_annotations {
    mid: "/m/07j7r"
    description: "Tree"
    score: 0.8151430487632751
    topicality: 0.8151430487632751
}
label_annotations {
    mid: "/g/11jxkqbpp"
    description: "Mountainous landforms"
    score: 0.8111746907234192
    topicality: 0.8111746907234192
}
```

```
Label annotations:
[mid: "/m/0csby"
description: "Cloud"
score: 0.9777280688285828
topicality: 0.9777280688285828
, mid: "/m/01bqvp"
description: "Sky"
score: 0.9730981588363647
topicality: 0.9730981588363647
, mid: "/m/09d_r"
description: "Mountain"
score: 0.9400748014450073
topicality: 0.9400748014450073
, mid: "/m/06_dn"
description: "Snow"
score: 0.8958427906036377
topicality: 0.8958427906036377
, mid: "/m/078hm"
description: "Slope"
score: 0.8739033341407776
topicality: 0.8739033341407776
, mid: "/m/03d28y3"
description: "Natural landscape"
score: 0.8660182952880859
topicality: 0.8660182952880859
, mid: "/m/03cjrt"
description: "Highland"
score: 0.8601166009902954
topicality: 0.8601166009902954
, mid: "/m/09nm_"
description: "World"
score: 0.85157310962677
topicality: 0.85157310962677
, mid: "/m/07j7r"
```

```
description: "Tree"
score: 0.8151430487632751
topicality: 0.8151430487632751
, mid: "/g/11jxkqbpp"
description: "Mountainous landforms"
score: 0.8111746907234192
topicality: 0.8111746907234192
]
```

First element of label annotations:

```
mid: "/m/0csby"
description: "Cloud"
score: 0.9777280688285828
topicality: 0.9777280688285828
```

Description of first element of label annotations:

Cloud

Keywords:

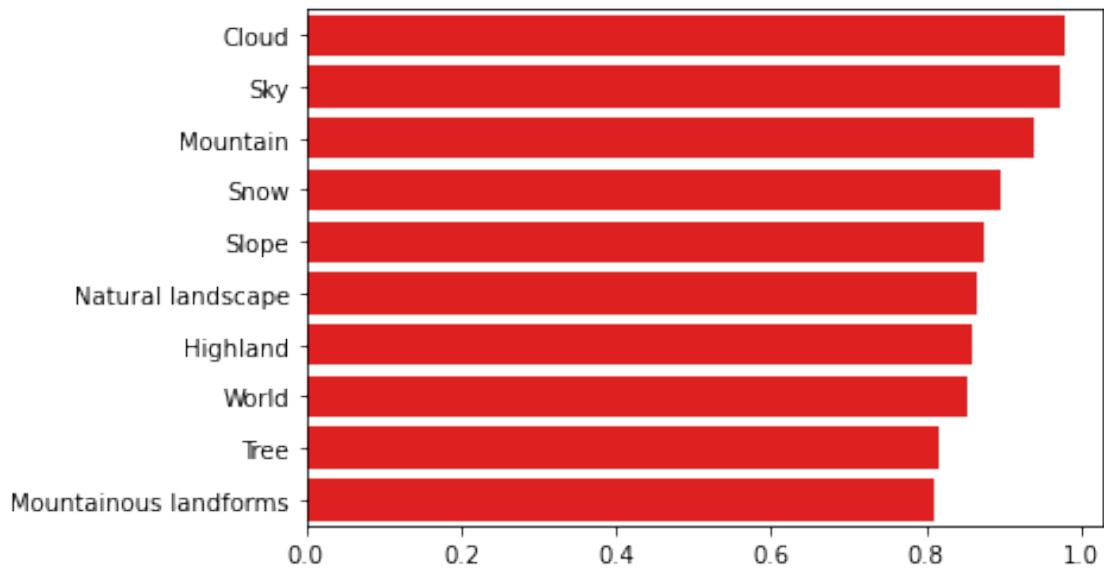
```
Cloud
Sky
Mountain
Snow
Slope
Natural landscape
Highland
World
Tree
Mountainous landforms
```

List of keywords:

```
['Cloud', 'Sky', 'Mountain', 'Snow', 'Slope', 'Natural landscape', 'Highland',
'World', 'Tree', 'Mountainous landforms']
```

List of scores:

```
[0.9777280688285828, 0.9730981588363647, 0.9400748014450073, 0.8958427906036377,
0.8739033341407776, 0.8660182952880859, 0.8601166009902954, 0.85157310962677,
0.8151430487632751, 0.8111746907234192]
```



File name without extension: images/IMG\_3623

File name extension: jpeg



```
Response:  
label_annotations {  
    mid: "/m/01bqvp"  
    description: "Sky"  
    score: 0.9697284698486328  
    topicality: 0.9697284698486328  
}  
label_annotations {  
    mid: "/m/0cgh4"  
    description: "Building"  
    score: 0.9520781636238098  
    topicality: 0.9520781636238098  
}  
label_annotations {  
    mid: "/m/0d4v4"  
    description: "Window"  
    score: 0.9380078911781311  
    topicality: 0.9380078911781311  
}  
label_annotations {  
    mid: "/m/01g5v"  
    description: "Blue"  
    score: 0.8947259783744812  
    topicality: 0.8947259783744812  
}  
label_annotations {  
    mid: "/m/03jm5"  
    description: "House"  
    score: 0.8736454248428345  
    topicality: 0.8736454248428345  
}  
label_annotations {  
    mid: "/m/01fdzj"  
    description: "Tower"  
    score: 0.8473043441772461  
    topicality: 0.8473043441772461  
}  
label_annotations {  
    mid: "/m/01g0g"  
    description: "Brick"  
    score: 0.7888298630714417  
    topicality: 0.7888298630714417  
}  
label_annotations {  
    mid: "/m/04wnmd"
```

```

    description: "Fixture"
    score: 0.7770969271659851
    topicality: 0.7770969271659851
}
label_annotations {
    mid: "/m/05s2s"
    description: "Plant"
    score: 0.7690051198005676
    topicality: 0.7690051198005676
}
label_annotations {
    mid: "/m/01x314"
    description: "Facade"
    score: 0.7533137202262878
    topicality: 0.7533137202262878
}

```

```

Label annotations:
[mid: "/m/01bqvp"
description: "Sky"
score: 0.9697284698486328
topicality: 0.9697284698486328
, mid: "/m/0cgh4"
description: "Building"
score: 0.9520781636238098
topicality: 0.9520781636238098
, mid: "/m/0d4v4"
description: "Window"
score: 0.9380078911781311
topicality: 0.9380078911781311
, mid: "/m/01g5v"
description: "Blue"
score: 0.8947259783744812
topicality: 0.8947259783744812
, mid: "/m/03jm5"
description: "House"
score: 0.8736454248428345
topicality: 0.8736454248428345
, mid: "/m/01fdzj"
description: "Tower"
score: 0.8473043441772461
topicality: 0.8473043441772461
, mid: "/m/01g0g"
description: "Brick"
score: 0.7888298630714417
topicality: 0.7888298630714417
, mid: "/m/04wnmd"

```

```
description: "Fixture"
score: 0.7770969271659851
topicality: 0.7770969271659851
, mid: "/m/05s2s"
description: "Plant"
score: 0.7690051198005676
topicality: 0.7690051198005676
, mid: "/m/01x314"
description: "Facade"
score: 0.7533137202262878
topicality: 0.7533137202262878
]
```

First element of label annotations:

```
mid: "/m/01bqvp"
description: "Sky"
score: 0.9697284698486328
topicality: 0.9697284698486328
```

Description of first element of label annotations:

Sky

Keywords:

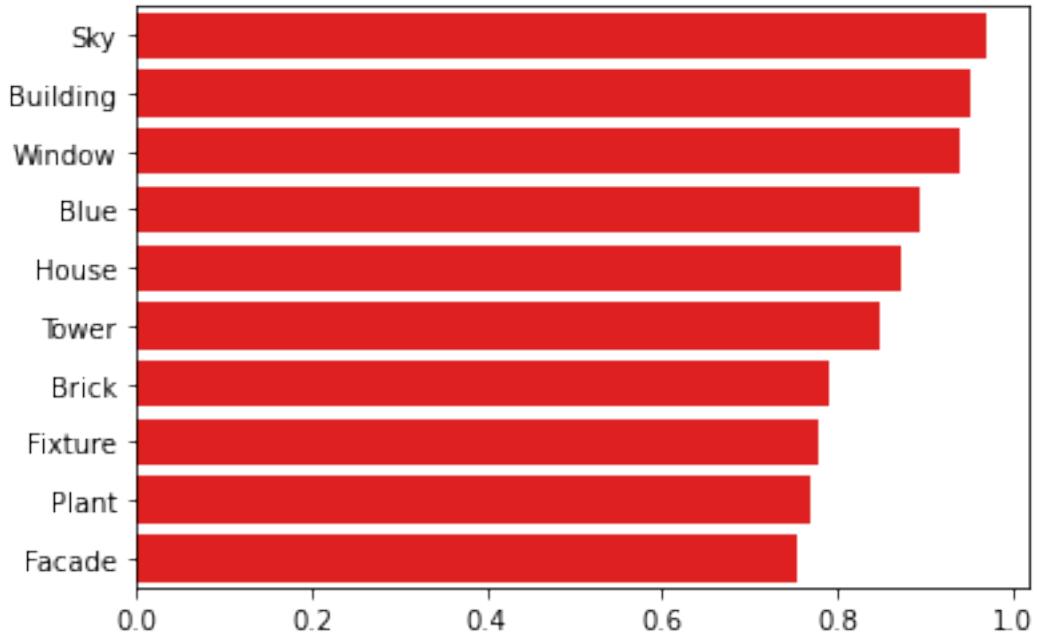
```
Sky
Building
Window
Blue
House
Tower
Brick
Fixture
Plant
Facade
```

List of keywords:

```
['Sky', 'Building', 'Window', 'Blue', 'House', 'Tower', 'Brick', 'Fixture',
'Plant', 'Facade']
```

List of scores:

```
[0.9697284698486328, 0.9520781636238098, 0.9380078911781311, 0.8947259783744812,
0.8736454248428345, 0.8473043441772461, 0.7888298630714417, 0.7770969271659851,
0.7690051198005676, 0.7533137202262878]
```



File name without extension: images/IMG\_3638

File name extension: jpeg



```
Response:  
label_annotations {  
    mid: "/m/01bqvp"  
    description: "Sky"  
    score: 0.9698122143745422  
    topicality: 0.9698122143745422  
}  
label_annotations {  
    mid: "/m/09d_r"  
    description: "Mountain"  
    score: 0.9461285471916199  
    topicality: 0.9461285471916199  
}  
label_annotations {  
    mid: "/m/06_dn"  
    description: "Snow"  
    score: 0.9449636936187744  
    topicality: 0.9449636936187744  
}  
label_annotations {  
    mid: "/m/0cblv"  
    description: "Ecoregion"  
    score: 0.9219450354576111  
    topicality: 0.9219450354576111  
}  
label_annotations {  
    mid: "/m/03d28y3"  
    description: "Natural landscape"  
    score: 0.8867104053497314  
    topicality: 0.8867104053497314  
}  
label_annotations {  
    mid: "/m/03cjrt"  
    description: "Highland"  
    score: 0.8795045614242554  
    topicality: 0.8795045614242554  
}  
label_annotations {  
    mid: "/m/078hm"  
    description: "Slope"  
    score: 0.8695122003555298  
    topicality: 0.8695122003555298  
}  
label_annotations {  
    mid: "/m/0pcm_"  
    description: "Larch"
```

```

    score: 0.8389416933059692
    topicality: 0.8389416933059692
}
label_annotations {
    mid: "/m/01c791"
    description: "Freezing"
    score: 0.828321099281311
    topicality: 0.828321099281311
}
label_annotations {
    mid: "/m/025tn5c"
    description: "Terrain"
    score: 0.818787693977356
    topicality: 0.818787693977356
}

```

Label annotations:

```

[{"mid": "/m/01bqvp",
"description": "Sky",
"score": 0.9698122143745422,
"topicality": 0.9698122143745422
, {"mid": "/m/09d_r",
"description": "Mountain",
"score": 0.9461285471916199,
"topicality": 0.9461285471916199
, {"mid": "/m/06_dn",
"description": "Snow",
"score": 0.9449636936187744,
"topicality": 0.9449636936187744
, {"mid": "/m/0cblv",
"description": "Ecoregion",
"score": 0.9219450354576111,
"topicality": 0.9219450354576111
, {"mid": "/m/03d28y3",
"description": "Natural landscape",
"score": 0.8867104053497314,
"topicality": 0.8867104053497314
, {"mid": "/m/03cjrt",
"description": "Highland",
"score": 0.8795045614242554,
"topicality": 0.8795045614242554
, {"mid": "/m/078hm",
"description": "Slope",
"score": 0.8695122003555298,
"topicality": 0.8695122003555298
, {"mid": "/m/0pcm_",
"description": "Larch"
}
```

```
score: 0.8389416933059692
topicality: 0.8389416933059692
, mid: "/m/01c791"
description: "Freezing"
score: 0.828321099281311
topicality: 0.828321099281311
, mid: "/m/025tn5c"
description: "Terrain"
score: 0.818787693977356
topicality: 0.818787693977356
]
```

First element of label annotations:

```
mid: "/m/01bqvp"
description: "Sky"
score: 0.9698122143745422
topicality: 0.9698122143745422
```

Description of first element of label annotations:

```
Sky
```

Keywords:

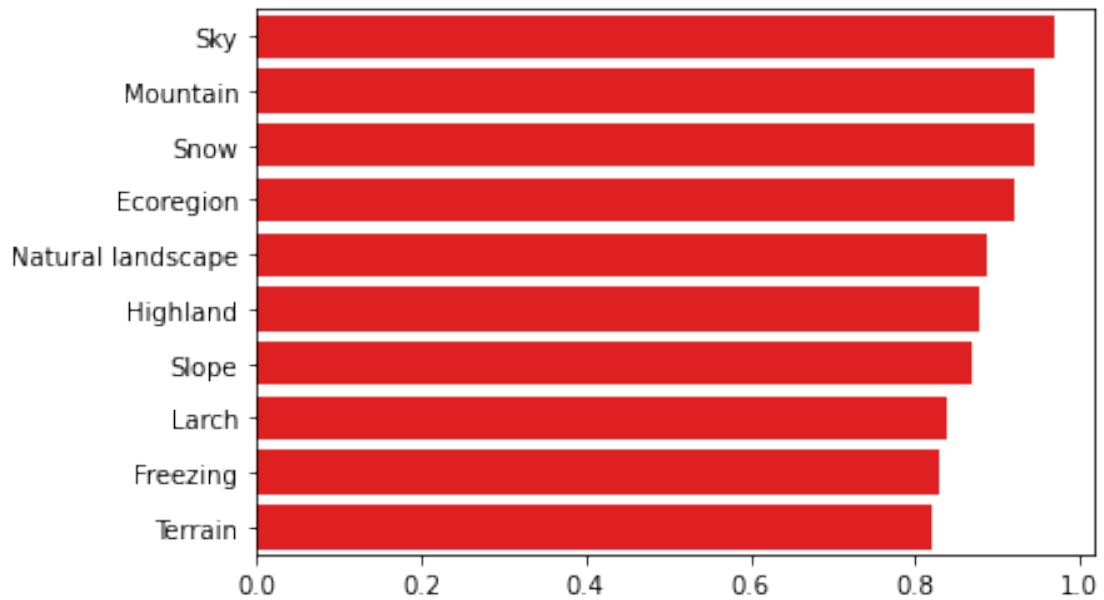
```
Sky
Mountain
Snow
Ecoregion
Natural landscape
Highland
Slope
Larch
Freezing
Terrain
```

List of keywords:

```
['Sky', 'Mountain', 'Snow', 'Ecoregion', 'Natural landscape', 'Highland',
'Slope', 'Larch', 'Freezing', 'Terrain']
```

List of scores:

```
[0.9698122143745422, 0.9461285471916199, 0.9449636936187744, 0.9219450354576111,
0.8867104053497314, 0.8795045614242554, 0.8695122003555298, 0.8389416933059692,
0.828321099281311, 0.818787693977356]
```



File name without extension: images/IMG\_3757

File name extension: jpeg



```
Response:  
label_annotations {  
    mid: "/m/09q2t"  
    description: "Brown"  
    score: 0.9804678559303284  
    topicality: 0.9804678559303284  
}  
label_annotations {  
    mid: "/m/01c4rd"  
    description: "Beak"  
    score: 0.8813260793685913  
    topicality: 0.8813260793685913  
}  
label_annotations {  
    mid: "/m/083vt"  
    description: "Wood"  
    score: 0.8780773878097534  
    topicality: 0.8780773878097534  
}  
label_annotations {  
    mid: "/m/05s2s"  
    description: "Plant"  
    score: 0.861660361289978  
    topicality: 0.861660361289978  
}  
label_annotations {  
    mid: "/m/016nqt"  
    description: "Twig"  
    score: 0.8608095049858093  
    topicality: 0.8608095049858093  
}  
label_annotations {  
    mid: "/m/02tcwp"  
    description: "Trunk"  
    score: 0.8503636717796326  
    topicality: 0.8503636717796326  
}  
label_annotations {  
    mid: "/m/071qp"  
    description: "Squirrel"  
    score: 0.8276546597480774  
    topicality: 0.8276546597480774  
}  
label_annotations {  
    mid: "/m/06hps"  
    description: "Rodent"  
    score: 0.8099303841590881  
    topicality: 0.8099303841590881
```

```

}

label_annotations {
    mid: "/m/02q_bfg"
    description: "Tints and shades"
    score: 0.7666783332824707
    topicality: 0.7666783332824707
}
label_annotations {
    mid: "/m/06z_nw"
    description: "Tail"
    score: 0.7540639042854309
    topicality: 0.7540639042854309
}

```

Label annotations:

```

[ mid: "/m/09q2t"
description: "Brown"
score: 0.9804678559303284
topicality: 0.9804678559303284
, mid: "/m/01c4rd"
description: "Beak"
score: 0.8813260793685913
topicality: 0.8813260793685913
, mid: "/m/083vt"
description: "Wood"
score: 0.8780773878097534
topicality: 0.8780773878097534
, mid: "/m/05s2s"
description: "Plant"
score: 0.861660361289978
topicality: 0.861660361289978
, mid: "/m/016nqt"
description: "Twig"
score: 0.8608095049858093
topicality: 0.8608095049858093
, mid: "/m/02tcwp"
description: "Trunk"
score: 0.8503636717796326
topicality: 0.8503636717796326
, mid: "/m/071qp"
description: "Squirrel"
score: 0.8276546597480774
topicality: 0.8276546597480774
, mid: "/m/06hps"
description: "Rodent"
score: 0.8099303841590881
topicality: 0.8099303841590881

```

```
, mid: "/m/02q_bfg"
description: "Tints and shades"
score: 0.7666783332824707
topicality: 0.7666783332824707
, mid: "/m/06z_nw"
description: "Tail"
score: 0.7540639042854309
topicality: 0.7540639042854309
]
```

First element of label annotations:

```
mid: "/m/09q2t"
description: "Brown"
score: 0.9804678559303284
topicality: 0.9804678559303284
```

Description of first element of label annotations:

Brown

Keywords:

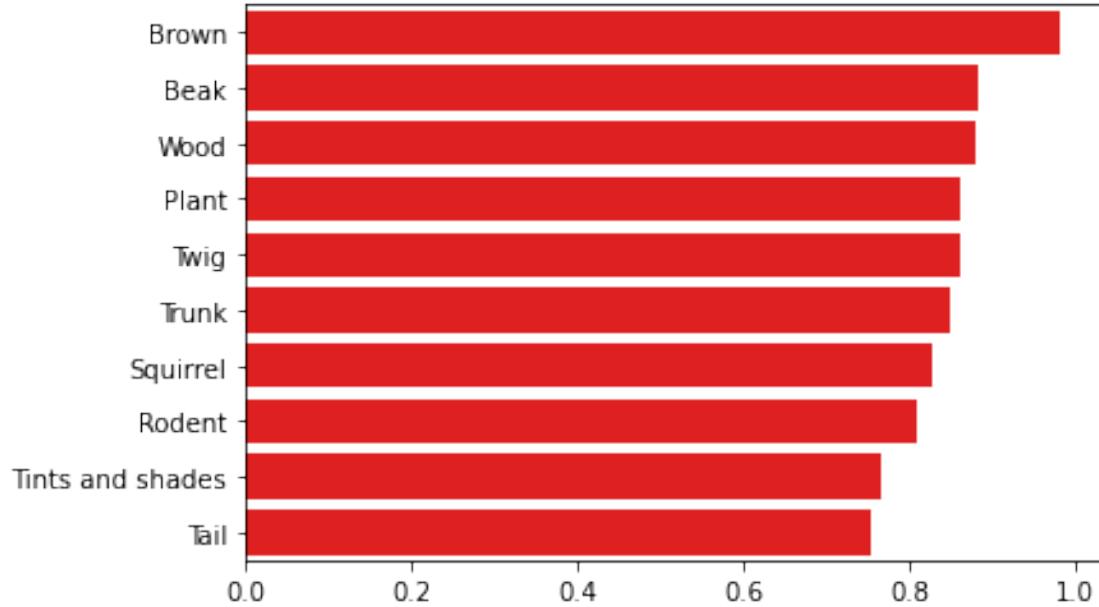
```
Brown
Beak
Wood
Plant
Twig
Trunk
Squirrel
Rodent
Tints and shades
Tail
```

List of keywords:

```
['Brown', 'Beak', 'Wood', 'Plant', 'Twig', 'Trunk', 'Squirrel', 'Rodent', 'Tints and shades', 'Tail']
```

List of scores:

```
[0.9804678559303284, 0.8813260793685913, 0.8780773878097534, 0.861660361289978, 0.8608095049858093, 0.8503636717796326, 0.8276546597480774, 0.8099303841590881, 0.7666783332824707, 0.7540639042854309]
```



File name without extension: images/IMG\_3793

File name extension: jpeg



```

Response:
label_annotations {
    mid: "/m/01bqvp"
    description: "Sky"
    score: 0.9541574716567993
    topicality: 0.9541574716567993
}
label_annotations {
    mid: "/m/06_dn"
    description: "Snow"
    score: 0.9524088501930237
    topicality: 0.9524088501930237
}
label_annotations {
    mid: "/m/05s2s"
    description: "Plant"
    score: 0.9393146634101868
    topicality: 0.9393146634101868
}
label_annotations {
    mid: "/m/09d_r"
    description: "Mountain"
    score: 0.9078800678253174
    topicality: 0.9078800678253174
}
label_annotations {
    mid: "/m/03d28y3"
    description: "Natural landscape"
    score: 0.864658772945404
    topicality: 0.864658772945404
}
label_annotations {
    mid: "/m/07j7r"
    description: "Tree"
    score: 0.8577496409416199
    topicality: 0.8577496409416199
}
label_annotations {
    mid: "/m/0pcm_"
    description: "Larch"
    score: 0.8551145195960999
    topicality: 0.8551145195960999
}
label_annotations {
    mid: "/m/019gvf"
    description: "Fluvial landforms of streams"
    score: 0.8429534435272217
    topicality: 0.8429534435272217
}

```

```

}

label_annotations {
    mid: "/m/03kj4q"
    description: "Watercourse"
    score: 0.8395840525627136
    topicality: 0.8395840525627136
}
label_annotations {
    mid: "/m/03ktm1"
    description: "Body of water"
    score: 0.8368027806282043
    topicality: 0.8368027806282043
}

Label annotations:
[mid: "/m/01bqv"
description: "Sky"
score: 0.9541574716567993
topicality: 0.9541574716567993
, mid: "/m/06_dn"
description: "Snow"
score: 0.9524088501930237
topicality: 0.9524088501930237
, mid: "/m/05s2s"
description: "Plant"
score: 0.9393146634101868
topicality: 0.9393146634101868
, mid: "/m/09d_r"
description: "Mountain"
score: 0.9078800678253174
topicality: 0.9078800678253174
, mid: "/m/03d28y3"
description: "Natural landscape"
score: 0.864658772945404
topicality: 0.864658772945404
, mid: "/m/07j7r"
description: "Tree"
score: 0.8577496409416199
topicality: 0.8577496409416199
, mid: "/m/0pcm_"
description: "Larch"
score: 0.8551145195960999
topicality: 0.8551145195960999
, mid: "/m/019gvf"
description: "Fluvial landforms of streams"
score: 0.8429534435272217
topicality: 0.8429534435272217

```

```
, mid: "/m/03kj4q"
description: "Watercourse"
score: 0.8395840525627136
topicality: 0.8395840525627136
, mid: "/m/03ktm1"
description: "Body of water"
score: 0.8368027806282043
topicality: 0.8368027806282043
]
```

First element of label annotations:

```
mid: "/m/01bqv"
description: "Sky"
score: 0.9541574716567993
topicality: 0.9541574716567993
```

Description of first element of label annotations:

Sky

Keywords:

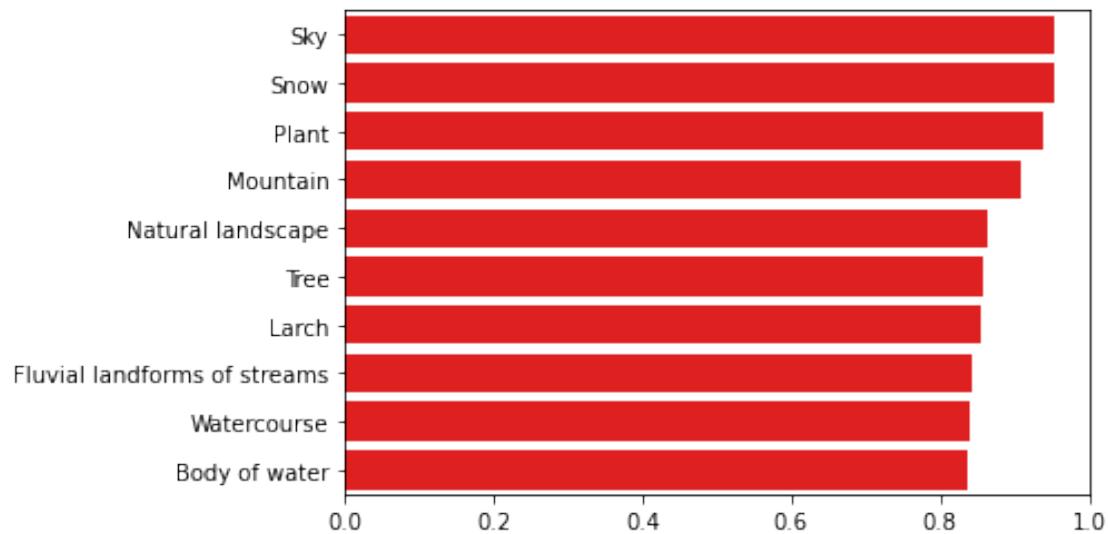
```
Sky
Snow
Plant
Mountain
Natural landscape
Tree
Larch
Fluvial landforms of streams
Watercourse
Body of water
```

List of keywords:

```
['Sky', 'Snow', 'Plant', 'Mountain', 'Natural landscape', 'Tree', 'Larch',
'Fluvial landforms of streams', 'Watercourse', 'Body of water']
```

List of scores:

```
[0.9541574716567993, 0.9524088501930237, 0.9393146634101868, 0.9078800678253174,
0.864658772945404, 0.8577496409416199, 0.8551145195960999, 0.8429534435272217,
0.8395840525627136, 0.8368027806282043]
```



File name without extension: images/IMG\_3805

File name extension: jpeg



Response:

```
label_annotations {
    mid: "/m/01ctsf"
    description: "Atmosphere"
    score: 0.9481807947158813
    topicality: 0.9481807947158813
}
label_annotations {
    mid: "/m/0h8pb3l"
    description: "Automotive tire"
    score: 0.9074817299842834
    topicality: 0.9074817299842834
}
label_annotations {
    mid: "/m/01k0mv"
    description: "Road surface"
    score: 0.8826594352722168
    topicality: 0.8826594352722168
}
label_annotations {
    mid: "/m/0hr8"
    description: "Asphalt"
    score: 0.8705956339836121
    topicality: 0.8705956339836121
}
label_annotations {
    mid: "/m/036k5h"
    description: "Grey"
    score: 0.8440105319023132
    topicality: 0.8440105319023132
}
label_annotations {
    mid: "/m/03ktm1"
    description: "Body of water"
    score: 0.8399625420570374
    topicality: 0.8399625420570374
}
label_annotations {
    mid: "/m/01g6gs"
    description: "Black-and-white"
    score: 0.8282089233398438
    topicality: 0.8282089233398438
}
label_annotations {
    mid: "/m/083vt"
    description: "Wood"
    score: 0.8260961174964905
    topicality: 0.8260961174964905
}
```

```

label_annotations {
    mid: "/m/07pw27b"
    description: "Atmospheric phenomenon"
    score: 0.8176640272140503
    topicality: 0.8176640272140503
}
label_annotations {
    mid: "/m/02q_bfg"
    description: "Tints and shades"
    score: 0.7737652063369751
    topicality: 0.7737652063369751
}

```

Label annotations:

```

[{"mid": "/m/01ctsf",
"description": "Atmosphere",
"score": 0.9481807947158813,
"topicality": 0.9481807947158813,
, "mid": "/m/0h8pb3l",
"description": "Automotive tire",
"score": 0.9074817299842834,
"topicality": 0.9074817299842834,
, "mid": "/m/01k0mv",
"description": "Road surface",
"score": 0.8826594352722168,
"topicality": 0.8826594352722168,
, "mid": "/m/0hr8",
"description": "Asphalt",
"score": 0.8705956339836121,
"topicality": 0.8705956339836121,
, "mid": "/m/036k5h",
"description": "Grey",
"score": 0.8440105319023132,
"topicality": 0.8440105319023132,
, "mid": "/m/03ktm1",
"description": "Body of water",
"score": 0.8399625420570374,
"topicality": 0.8399625420570374,
, "mid": "/m/01g6gs",
"description": "Black-and-white",
"score": 0.8282089233398438,
"topicality": 0.8282089233398438,
, "mid": "/m/083vt",
"description": "Wood",
"score": 0.8260961174964905,
"topicality": 0.8260961174964905,
, "mid": "/m/07pw27b"
}
```

```
description: "Atmospheric phenomenon"
score: 0.8176640272140503
topicality: 0.8176640272140503
, mid: "/m/02q_bfg"
description: "Tints and shades"
score: 0.7737652063369751
topicality: 0.7737652063369751
]
```

First element of label annotations:

```
mid: "/m/01ctsf"
description: "Atmosphere"
score: 0.9481807947158813
topicality: 0.9481807947158813
```

Description of first element of label annotations:

Atmosphere

Keywords:

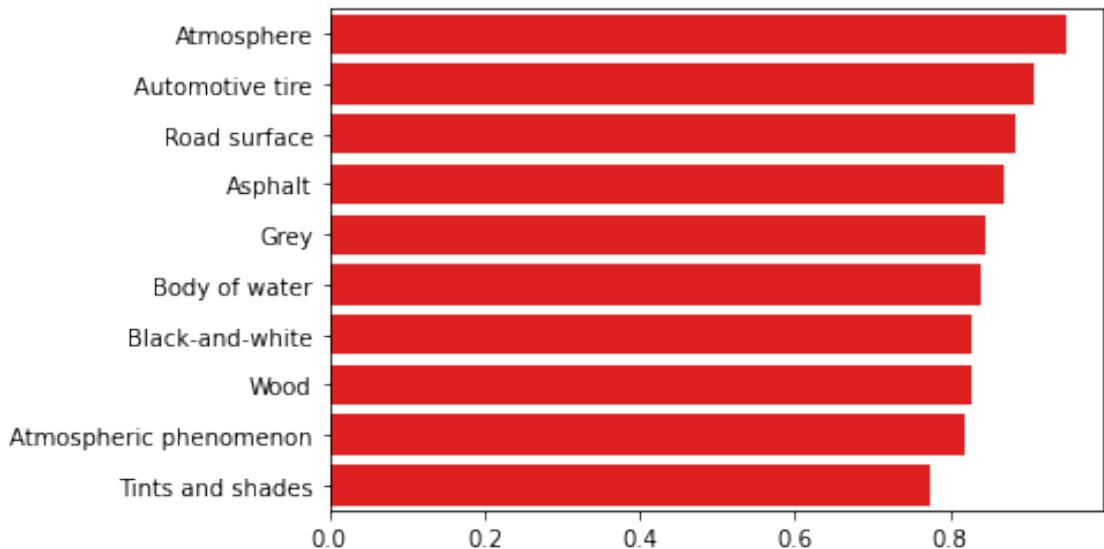
```
Atmosphere
Automotive tire
Road surface
Asphalt
Grey
Body of water
Black-and-white
Wood
Atmospheric phenomenon
Tints and shades
```

List of keywords:

```
['Atmosphere', 'Automotive tire', 'Road surface', 'Asphalt', 'Grey', 'Body of
water', 'Black-and-white', 'Wood', 'Atmospheric phenomenon', 'Tints and shades']
```

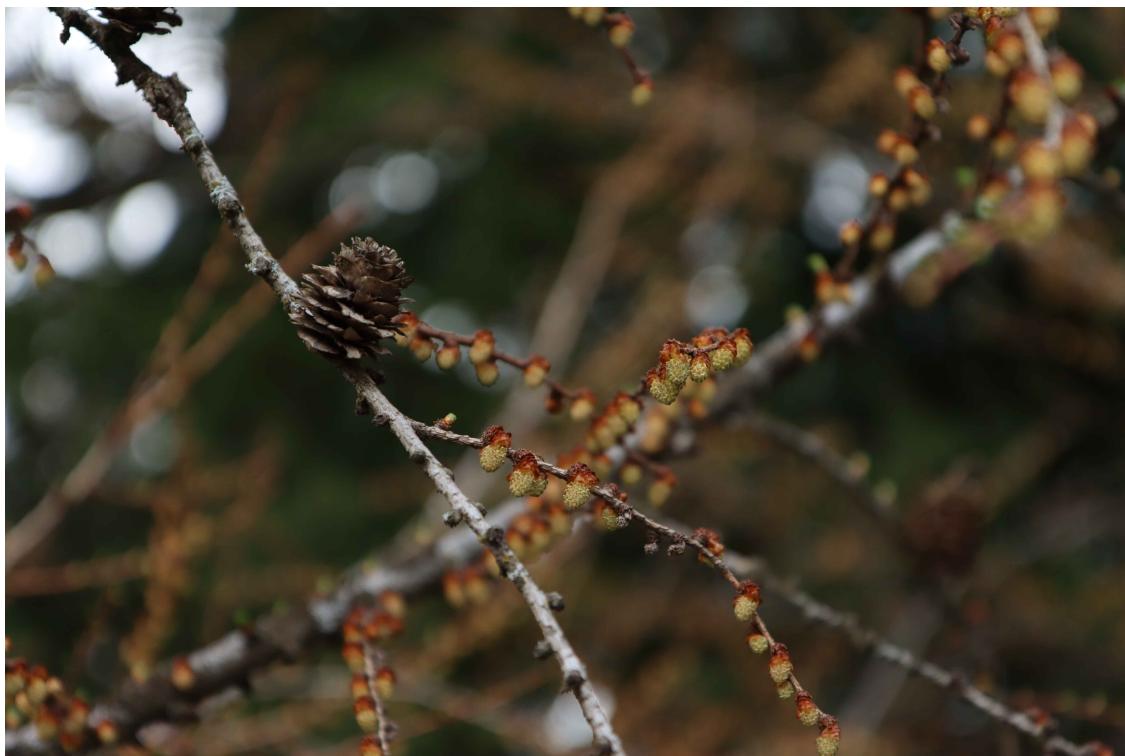
List of scores:

```
[0.9481807947158813, 0.9074817299842834, 0.8826594352722168, 0.8705956339836121,
0.8440105319023132, 0.8399625420570374, 0.8282089233398438, 0.8260961174964905,
0.8176640272140503, 0.7737652063369751]
```



File name without extension: images/IMG\_3841

File name extension: jpeg



Response:

```
label_annotations {
    mid: "/m/05s2s"
    description: "Plant"
    score: 0.8909375667572021
    topicality: 0.8909375667572021
}
label_annotations {
    mid: "/m/016nqt"
    description: "Twig"
    score: 0.8649297952651978
    topicality: 0.8649297952651978
}
label_annotations {
    mid: "/m/0fbflw"
    description: "Terrestrial plant"
    score: 0.850361704826355
    topicality: 0.850361704826355
}
label_annotations {
    mid: "/m/07sx2n"
    description: "Natural material"
    score: 0.8209806084632874
    topicality: 0.8209806084632874
}
label_annotations {
    mid: "/m/02cqfm"
    description: "Close-up"
    score: 0.7083485722541809
    topicality: 0.7083485722541809
}
label_annotations {
    mid: "/m/04sjm"
    description: "Flowering plant"
    score: 0.7076372504234314
    topicality: 0.7076372504234314
}
label_annotations {
    mid: "/m/01tksn"
    description: "Spider web"
    score: 0.699665904045105
    topicality: 0.699665904045105
}
label_annotations {
    mid: "/m/07j7r"
    description: "Tree"
    score: 0.6758040189743042
    topicality: 0.6758040189743042
}
```

```

label_annotations {
  mid: "/m/08t9c_"
  description: "Grass"
  score: 0.6711256504058838
  topicality: 0.6711256504058838
}
label_annotations {
  mid: "/m/03rbf6"
  description: "Macro photography"
  score: 0.6426244974136353
  topicality: 0.6426244974136353
}

```

```

Label annotations:
[mid: "/m/05s2s"
description: "Plant"
score: 0.8909375667572021
topicality: 0.8909375667572021
, mid: "/m/016nqt"
description: "Twig"
score: 0.8649297952651978
topicality: 0.8649297952651978
, mid: "/m/0fbflw"
description: "Terrestrial plant"
score: 0.850361704826355
topicality: 0.850361704826355
, mid: "/m/07sx2n"
description: "Natural material"
score: 0.8209806084632874
topicality: 0.8209806084632874
, mid: "/m/02cqfm"
description: "Close-up"
score: 0.7083485722541809
topicality: 0.7083485722541809
, mid: "/m/04sjm"
description: "Flowering plant"
score: 0.7076372504234314
topicality: 0.7076372504234314
, mid: "/m/01tksn"
description: "Spider web"
score: 0.699665904045105
topicality: 0.699665904045105
, mid: "/m/07j7r"
description: "Tree"
score: 0.6758040189743042
topicality: 0.6758040189743042
, mid: "/m/08t9c_"

```

```
description: "Grass"
score: 0.6711256504058838
topicality: 0.6711256504058838
, mid: "/m/03rbf6"
description: "Macro photography"
score: 0.6426244974136353
topicality: 0.6426244974136353
]
```

First element of label annotations:

```
mid: "/m/05s2s"
description: "Plant"
score: 0.8909375667572021
topicality: 0.8909375667572021
```

Description of first element of label annotations:

Plant

Keywords:

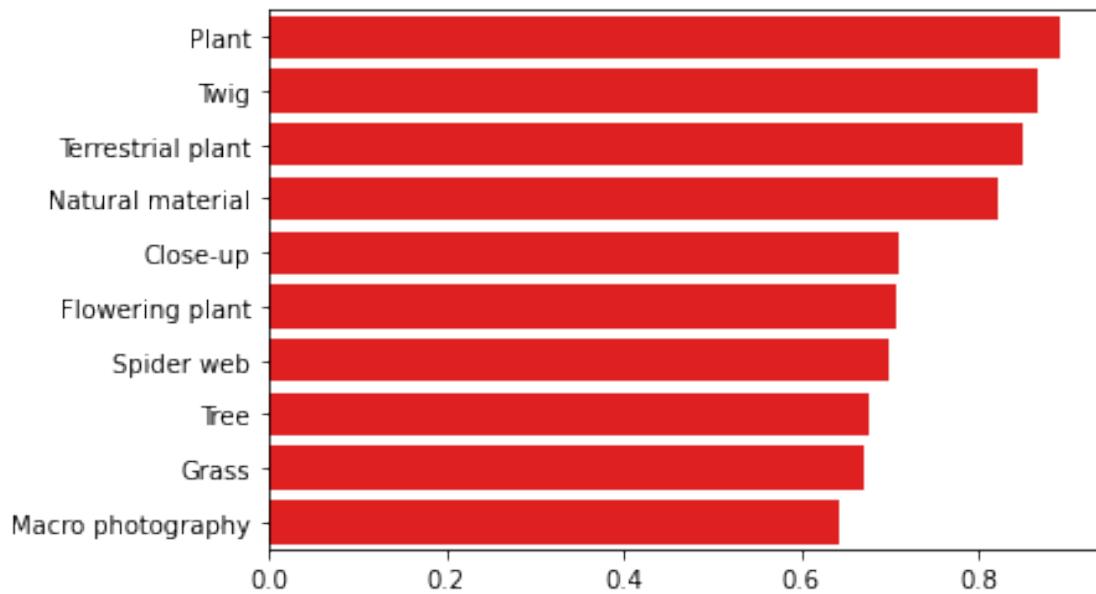
```
Plant
Twig
Terrestrial plant
Natural material
Close-up
Flowering plant
Spider web
Tree
Grass
Macro photography
```

List of keywords:

```
['Plant', 'Twig', 'Terrestrial plant', 'Natural material', 'Close-up',
'Flowering plant', 'Spider web', 'Tree', 'Grass', 'Macro photography']
```

List of scores:

```
[0.8909375667572021, 0.8649297952651978, 0.850361704826355, 0.8209806084632874,
0.7083485722541809, 0.7076372504234314, 0.699665904045105, 0.6758040189743042,
0.6711256504058838, 0.6426244974136353]
```



File name without extension: images/IMG\_3850

File name extension: jpeg



```

Response:
label_annotations {
    mid: "/m/0838f"
    description: "Water"
    score: 0.9749771952629089
    topicality: 0.9749771952629089
}
label_annotations {
    mid: "/m/05s2s"
    description: "Plant"
    score: 0.9485065937042236
    topicality: 0.9485065937042236
}
label_annotations {
    mid: "/m/01fnns"
    description: "Vegetation"
    score: 0.8511428236961365
    topicality: 0.8511428236961365
}
label_annotations {
    mid: "/m/03d28y3"
    description: "Natural landscape"
    score: 0.84632408618927
    topicality: 0.84632408618927
}
label_annotations {
    mid: "/m/019gvf"
    description: "Fluvial landforms of streams"
    score: 0.8307594656944275
    topicality: 0.8307594656944275
}
label_annotations {
    mid: "/m/096xlh"
    description: "Riparian zone"
    score: 0.7959167957305908
    topicality: 0.7959167957305908
}
label_annotations {
    mid: "/m/08t9c_"
    description: "Grass"
    score: 0.7904033064842224
    topicality: 0.7904033064842224
}
label_annotations {
    mid: "/m/066xq"
    description: "Pollution"
    score: 0.7885063886642456
    topicality: 0.7885063886642456
}

```

```

}
label_annotations {
  mid: "/m/018ssc"
  description: "Groundcover"
  score: 0.7780022025108337
  topicality: 0.7780022025108337
}
label_annotations {
  mid: "/m/0308t8"
  description: "Bedrock"
  score: 0.7514669299125671
  topicality: 0.7514669299125671
}

```

Label annotations:

```

[ mid: "/m/0838f"
description: "Water"
score: 0.9749771952629089
topicality: 0.9749771952629089
, mid: "/m/05s2s"
description: "Plant"
score: 0.9485065937042236
topicality: 0.9485065937042236
, mid: "/m/01fnns"
description: "Vegetation"
score: 0.8511428236961365
topicality: 0.8511428236961365
, mid: "/m/03d28y3"
description: "Natural landscape"
score: 0.84632408618927
topicality: 0.84632408618927
, mid: "/m/019gvf"
description: "Fluvial landforms of streams"
score: 0.8307594656944275
topicality: 0.8307594656944275
, mid: "/m/096x1h"
description: "Riparian zone"
score: 0.7959167957305908
topicality: 0.7959167957305908
, mid: "/m/08t9c_"
description: "Grass"
score: 0.7904033064842224
topicality: 0.7904033064842224
, mid: "/m/066xq"
description: "Pollution"
score: 0.7885063886642456
topicality: 0.7885063886642456

```

```
, mid: "/m/018ssc"
description: "Groundcover"
score: 0.7780022025108337
topicality: 0.7780022025108337
, mid: "/m/0308t8"
description: "Bedrock"
score: 0.7514669299125671
topicality: 0.7514669299125671
]
```

First element of label annotations:

```
mid: "/m/0838f"
description: "Water"
score: 0.9749771952629089
topicality: 0.9749771952629089
```

Description of first element of label annotations:

Water

Keywords:

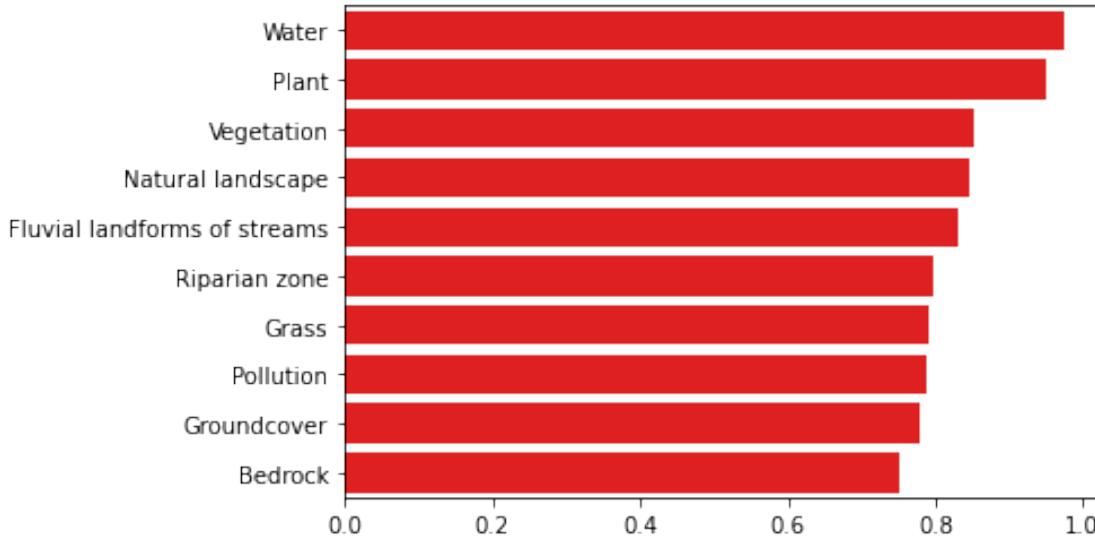
```
Water
Plant
Vegetation
Natural landscape
Fluvial landforms of streams
Riparian zone
Grass
Pollution
Groundcover
Bedrock
```

List of keywords:

```
['Water', 'Plant', 'Vegetation', 'Natural landscape', 'Fluvial landforms of streams', 'Riparian zone', 'Grass', 'Pollution', 'Groundcover', 'Bedrock']
```

List of scores:

```
[0.9749771952629089, 0.9485065937042236, 0.8511428236961365, 0.84632408618927,
0.8307594656944275, 0.7959167957305908, 0.7904033064842224, 0.7885063886642456,
0.7780022025108337, 0.7514669299125671]
```



```
[7]: print('Dictionary of list of keywords:')
print(label_annotation_desc_dict)
print('')
print('Dictionary of list of scores:')
print(label_annotation_score_dict)
print('')
```

Dictionary of list of keywords:

```
{'/home/jupyter/images/IMG_3579.jpeg': ['Cloud', 'Mountain', 'Sky', 'Slope',
'Snow', 'Ice cap', 'Terrain', 'Natural landscape', 'Cumulus', 'Landscape'],
'/home/jupyter/images/IMG_3586.jpeg': ['Sky', 'Mountain', 'Snow', 'Plant',
'Natural landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain',
'Tree'],
'/home/jupyter/images/IMG_3609.jpeg': ['Cloud', 'Sky', 'Mountain',
'Snow', 'Slope', 'Natural landscape', 'Highland', 'World', 'Tree', 'Mountainous
landforms'],
'/home/jupyter/images/IMG_3623.jpeg': ['Sky', 'Building', 'Window',
'Blue', 'House', 'Tower', 'Brick', 'Fixture', 'Plant', 'Facade'],
'/home/jupyter/images/IMG_3638.jpeg': ['Sky', 'Mountain', 'Snow', 'Ecoregion',
'Natural landscape', 'Highland', 'Slope', 'Larch', 'Freezing', 'Terrain'],
'/home/jupyter/images/IMG_3757.jpeg': ['Brown', 'Beak', 'Wood', 'Plant', 'Twig',
'Trunk', 'Squirrel', 'Rodent', 'Tints and shades', 'Tail'],
'/home/jupyter/images/IMG_3793.jpeg': ['Sky', 'Snow', 'Plant', 'Mountain',
'Natural landscape', 'Tree', 'Larch', 'Fluvial landforms of streams',
'Watercourse', 'Body of water'],
'/home/jupyter/images/IMG_3805.jpeg': ['Atmosphere', 'Automotive tire', 'Road surface',
'Asphalt', 'Grey', 'Body of water', 'Black-and-white', 'Wood', 'Atmospheric phenomenon',
'Tints and shades'],
'/home/jupyter/images/IMG_3841.jpeg': ['Plant', 'Twig', 'Terrestrial
plant', 'Natural material', 'Close-up', 'Flowering plant', 'Spider web', 'Tree',
'Grass', 'Macro photography'],
'/home/jupyter/images/IMG_3850.jpeg': ['Water', 'Plant', 'Vegetation', 'Natural landscape',
'Fluvial landforms of streams']}
```

```
'Riparian zone', 'Grass', 'Pollution', 'Groundcover', 'Bedrock']]}

Dictionary of list of scores:
{'/home/jupyter/images/IMG_3579.jpeg': [0.9827239513397217, 0.9588522911071777,
0.9572277665138245, 0.8840519189834595, 0.8719691634178162, 0.8418418169021606,
0.8227407932281494, 0.7953419089317322, 0.784989058971405, 0.7753972411155701],
'/home/jupyter/images/IMG_3586.jpeg': [0.9577324390411377, 0.9488414525985718,
0.9056057929992676, 0.9005176424980164, 0.8744946122169495, 0.8737100958824158,
0.8499352335929871, 0.8342433571815491, 0.8180684447288513, 0.8042957186698914],
'/home/jupyter/images/IMG_3609.jpeg': [0.9777280688285828, 0.9730981588363647,
0.9400748014450073, 0.8958427906036377, 0.8739033341407776, 0.8660182952880859,
0.8601166009902954, 0.85157310962677, 0.8151430487632751, 0.8111746907234192],
'/home/jupyter/images/IMG_3623.jpeg': [0.9697284698486328, 0.9520781636238098,
0.9380078911781311, 0.8947259783744812, 0.8736454248428345, 0.8473043441772461,
0.7888298630714417, 0.7770969271659851, 0.7690051198005676, 0.7533137202262878],
'/home/jupyter/images/IMG_3638.jpeg': [0.9698122143745422, 0.9461285471916199,
0.9449636936187744, 0.9219450354576111, 0.8867104053497314, 0.8795045614242554,
0.8695122003555298, 0.8389416933059692, 0.828321099281311, 0.818787693977356],
'/home/jupyter/images/IMG_3757.jpeg': [0.9804678559303284, 0.8813260793685913,
0.8780773878097534, 0.861660361289978, 0.8608095049858093, 0.8503636717796326,
0.8276546597480774, 0.8099303841590881, 0.7666783332824707, 0.7540639042854309],
'/home/jupyter/images/IMG_3793.jpeg': [0.9541574716567993, 0.9524088501930237,
0.9393146634101868, 0.9078800678253174, 0.864658772945404, 0.8577496409416199,
0.8551145195960999, 0.8429534435272217, 0.8395840525627136, 0.8368027806282043],
'/home/jupyter/images/IMG_3805.jpeg': [0.9481807947158813, 0.9074817299842834,
0.8826594352722168, 0.8705956339836121, 0.8440105319023132, 0.8399625420570374,
0.8282089233398438, 0.8260961174964905, 0.8176640272140503, 0.7737652063369751],
'/home/jupyter/images/IMG_3841.jpeg': [0.8909375667572021, 0.8649297952651978,
0.850361704826355, 0.8209806084632874, 0.7083485722541809, 0.7076372504234314,
0.699665904045105, 0.6758040189743042, 0.6711256504058838, 0.6426244974136353],
'/home/jupyter/images/IMG_3850.jpeg': [0.9749771952629089, 0.9485065937042236,
0.8511428236961365, 0.84632408618927, 0.8307594656944275, 0.7959167957305908,
0.7904033064842224, 0.7885063886642456, 0.7780022025108337, 0.7514669299125671]}
```

## 1.8 6: Create database with keywords

```
[8]: # bq_dataset = bq_client.dataset(bq_dataset_name)
```

### 1.8.1 Create empty table

```
[9]: query_create = f"""
CREATE TABLE IF NOT EXISTS `gcp_project_name`.`bq_dataset_name`.
`{bq_table_name}`
(
    file_name STRING OPTIONS(description="The name of the file"),
```

```

creation_date_time DATETIME OPTIONS(description="The date and time when the file was uploaded to the cloud"),
keywords ARRAY<STRING> OPTIONS(description="The keywords associated to the image")
)
PARTITION BY DATETIME_TRUNC(creation_date_time, DAY)
OPTIONS(
description="Image database"
)
"""

print('Query for creating empty table:')
print(query_create)
print('')

```

Query for creating empty table:

```

CREATE TABLE IF NOT EXISTS `personalproject-348318.whitebloomingtulip_db.image` (
file_name STRING OPTIONS(description="The name of the file"),
creation_date_time DATETIME OPTIONS(description="The date and time when the file was uploaded to the cloud"),
keywords ARRAY<STRING> OPTIONS(description="The keywords associated to the image")
)
PARTITION BY DATETIME_TRUNC(creation_date_time, DAY)
OPTIONS(
description="Image database"
)

```

[10]: # Execute the query  
query\_job\_create = bq\_client.query(query\_create, location=region)

### 1.8.2 Insert one row in the table

[11]: creation\_date\_time = datetime.datetime.fromtimestamp(os.path.getmtime('/home/jupyter/images/'+sample\_image\_name))
print('Creation date time:')
print(creation\_date\_time)
print('')

# Insert one row
query\_insert\_one = f"""
INSERT `{{gcp\_project\_name}}.{{bq\_dataset\_name}}.{{bq\_table\_name}}` (file\_name, creation\_date\_time, keywords)

```

VALUES('/home/jupyter/images/{sample_image_name}', '{creation_date_time}', {
    ↪label_annotation_desc_dict[f'/home/jupyter/images/{sample_image_name}'])
"""

print('Query for inserting one row:')
print(query_insert_one)
print('')

```

Creation date time:  
2022-05-01 20:23:51.690000

Query for inserting one row:

```

INSERT `personalproject-348318.whitebloomingtulip_db.image` (file_name,
creation_date_time, keywords)
VALUES('/home/jupyter/images/IMG_3579.jpeg', '2022-05-01 20:23:51.690000',
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural
landscape', 'Cumulus', 'Landscape'])

```

[12]: # Execute the query  
query\_job\_insert\_one = bq\_client.query(query\_insert\_one, location=region)

### 1.8.3 Insert multiple rows in the table

[13]: values\_str = ''  
for key in label\_annotation\_desc\_dict.keys():
 creation\_date\_time\_temp = datetime.datetime.fromtimestamp(os.path.
 ↪getmtime(key))
 keywords\_temp = label\_annotation\_desc\_dict[key]
 if values\_str == '':
 values\_str = values\_str+f"VALUES('{key}', '{creation\_date\_time\_temp}', {
 ↪keywords\_temp})"
 else:
 values\_str = values\_str+f", ('{key}', '{creation\_date\_time\_temp}', {
 ↪keywords\_temp})"
 if verbose:
 print(values\_str)
 print('')

print('String for inserting multiple rows:')
print(values\_str)
print('')

# Insert multiple rows
query\_insert\_many = f"""

```

INSERT `{{gcp_project_name}}.{{bq_dataset_name}}.{{bq_table_name}}` (file_name, {{creation_date_time}}, keywords)
{{values_str}}
"""

print('Query for inserting multiple rows:')
print(query_insert_many)
print('')

```

VALUES('/home/jupyter/images/IMG\_3579.jpeg', '2022-05-01 20:23:51.690000',  
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural  
landscape', 'Cumulus', 'Landscape'])

VALUES('/home/jupyter/images/IMG\_3579.jpeg', '2022-05-01 20:23:51.690000',  
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural  
landscape', 'Cumulus', 'Landscape']), ('/home/jupyter/images/IMG\_3586.jpeg',  
'2022-05-01 20:23:51.746000', ['Sky', 'Mountain', 'Snow', 'Plant', 'Natural  
landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain', 'Tree'])

VALUES('/home/jupyter/images/IMG\_3579.jpeg', '2022-05-01 20:23:51.690000',  
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural  
landscape', 'Cumulus', 'Landscape']), ('/home/jupyter/images/IMG\_3586.jpeg',  
'2022-05-01 20:23:51.746000', ['Sky', 'Mountain', 'Snow', 'Plant', 'Natural  
landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain',  
'Tree']), ('/home/jupyter/images/IMG\_3609.jpeg', '2022-05-01 20:23:51.800000',  
['Cloud', 'Sky', 'Mountain', 'Snow', 'Slope', 'Natural landscape', 'Highland',  
'World', 'Tree', 'Mountainous landforms'])

VALUES('/home/jupyter/images/IMG\_3579.jpeg', '2022-05-01 20:23:51.690000',  
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural  
landscape', 'Cumulus', 'Landscape']), ('/home/jupyter/images/IMG\_3586.jpeg',  
'2022-05-01 20:23:51.746000', ['Sky', 'Mountain', 'Snow', 'Plant', 'Natural  
landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain',  
'Tree']), ('/home/jupyter/images/IMG\_3609.jpeg', '2022-05-01 20:23:51.800000',  
['Cloud', 'Sky', 'Mountain', 'Snow', 'Slope', 'Natural landscape', 'Highland',  
'World', 'Tree', 'Mountainous  
landforms']), ('/home/jupyter/images/IMG\_3623.jpeg', '2022-05-01  
20:23:51.855000', ['Sky', 'Building', 'Window', 'Blue', 'House', 'Tower',  
'Brick', 'Fixture', 'Plant', 'Facade'])

VALUES('/home/jupyter/images/IMG\_3579.jpeg', '2022-05-01 20:23:51.690000',  
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural  
landscape', 'Cumulus', 'Landscape']), ('/home/jupyter/images/IMG\_3586.jpeg',  
'2022-05-01 20:23:51.746000', ['Sky', 'Mountain', 'Snow', 'Plant', 'Natural  
landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain',  
'Tree']), ('/home/jupyter/images/IMG\_3609.jpeg', '2022-05-01 20:23:51.800000',  
['Cloud', 'Sky', 'Mountain', 'Snow', 'Slope', 'Natural landscape', 'Highland',  
'World', 'Tree', 'Mountainous  
landforms'])

```

'World', 'Tree', 'Mountainous
landforms']), ('/home/jupyter/images/IMG_3623.jpeg', '2022-05-01
20:23:51.855000', ['Sky', 'Building', 'Window', 'Blue', 'House', 'Tower',
'Brick', 'Fixture', 'Plant', 'Facade']), ('/home/jupyter/images/IMG_3638.jpeg',
'2022-05-01 20:23:51.911000', ['Sky', 'Mountain', 'Snow', 'Ecoregion', 'Natural
landscape', 'Highland', 'Slope', 'Larch', 'Freezing', 'Terrain'])

VALUES('/home/jupyter/images/IMG_3579.jpeg', '2022-05-01 20:23:51.690000',
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural
landscape', 'Cumulus', 'Landscape']), ('/home/jupyter/images/IMG_3586.jpeg',
'2022-05-01 20:23:51.746000', ['Sky', 'Mountain', 'Snow', 'Plant', 'Natural
landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain',
'Tree']), ('/home/jupyter/images/IMG_3609.jpeg', '2022-05-01 20:23:51.800000',
['Cloud', 'Sky', 'Mountain', 'Snow', 'Slope', 'Natural landscape', 'Highland',
'World', 'Tree', 'Mountainous
landforms']), ('/home/jupyter/images/IMG_3623.jpeg', '2022-05-01
20:23:51.855000', ['Sky', 'Building', 'Window', 'Blue', 'House', 'Tower',
'Brick', 'Fixture', 'Plant', 'Facade']), ('/home/jupyter/images/IMG_3638.jpeg',
'2022-05-01 20:23:51.911000', ['Sky', 'Mountain', 'Snow', 'Ecoregion', 'Natural
landscape', 'Highland', 'Slope', 'Larch', 'Freezing',
'Terrain']), ('/home/jupyter/images/IMG_3757.jpeg', '2022-05-01 20:23:51.970000',
['Brown', 'Beak', 'Wood', 'Plant', 'Twig', 'Trunk', 'Squirrel', 'Rodent', 'Tints
and shades', 'Tail'])

VALUES('/home/jupyter/images/IMG_3579.jpeg', '2022-05-01 20:23:51.690000',
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural
landscape', 'Cumulus', 'Landscape']), ('/home/jupyter/images/IMG_3586.jpeg',
'2022-05-01 20:23:51.746000', ['Sky', 'Mountain', 'Snow', 'Plant', 'Natural
landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain',
'Tree']), ('/home/jupyter/images/IMG_3609.jpeg', '2022-05-01 20:23:51.800000',
['Cloud', 'Sky', 'Mountain', 'Snow', 'Slope', 'Natural landscape', 'Highland',
'World', 'Tree', 'Mountainous
landforms']), ('/home/jupyter/images/IMG_3623.jpeg', '2022-05-01
20:23:51.855000', ['Sky', 'Building', 'Window', 'Blue', 'House', 'Tower',
'Brick', 'Fixture', 'Plant', 'Facade']), ('/home/jupyter/images/IMG_3638.jpeg',
'2022-05-01 20:23:51.911000', ['Sky', 'Mountain', 'Snow', 'Ecoregion', 'Natural
landscape', 'Highland', 'Slope', 'Larch', 'Freezing',
'Terrain']), ('/home/jupyter/images/IMG_3757.jpeg', '2022-05-01 20:23:51.970000',
['Brown', 'Beak', 'Wood', 'Plant', 'Twig', 'Trunk', 'Squirrel', 'Rodent', 'Tints
and shades', 'Tail']), ('/home/jupyter/images/IMG_3793.jpeg', '2022-05-01
20:23:52.030000', ['Sky', 'Snow', 'Plant', 'Mountain', 'Natural landscape',
'Tree', 'Larch', 'Fluvial landforms of streams', 'Watercourse', 'Body of
water'])

VALUES('/home/jupyter/images/IMG_3579.jpeg', '2022-05-01 20:23:51.690000',
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural
landscape', 'Cumulus', 'Landscape']), ('/home/jupyter/images/IMG_3586.jpeg',
'2022-05-01 20:23:51.746000', ['Sky', 'Mountain', 'Snow', 'Plant', 'Natural
landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain',
'Tree'])

```

```

landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain',
'Tree']), ('/home/jupyter/images/IMG_3609.jpeg', '2022-05-01 20:23:51.800000',
['Cloud', 'Sky', 'Mountain', 'Snow', 'Slope', 'Natural landscape', 'Highland',
'World', 'Tree', 'Mountainous
landforms']), ('/home/jupyter/images/IMG_3623.jpeg', '2022-05-01
20:23:51.855000', ['Sky', 'Building', 'Window', 'Blue', 'House', 'Tower',
'Brick', 'Fixture', 'Plant', 'Facade']), ('/home/jupyter/images/IMG_3638.jpeg',
'2022-05-01 20:23:51.911000', ['Sky', 'Mountain', 'Snow', 'Ecoregion', 'Natural
landscape', 'Highland', 'Slope', 'Larch', 'Freezing',
'Terrain']), ('/home/jupyter/images/IMG_3757.jpeg', '2022-05-01 20:23:51.970000',
['Brown', 'Beak', 'Wood', 'Plant', 'Twig', 'Trunk', 'Squirrel', 'Rodent', 'Tints
and shades', 'Tail']), ('/home/jupyter/images/IMG_3793.jpeg', '2022-05-01
20:23:52.030000', ['Sky', 'Snow', 'Plant', 'Mountain', 'Natural landscape',
'Tree', 'Larch', 'Fluvial landforms of streams', 'Watercourse', 'Body of
water']), ('/home/jupyter/images/IMG_3805.jpeg', '2022-05-01 20:23:52.084000',
['Atmosphere', 'Automotive tire', 'Road surface', 'Asphalt', 'Grey', 'Body of
water', 'Black-and-white', 'Wood', 'Atmospheric phenomenon', 'Tints and
shades'])

```

```

VALUES('/home/jupyter/images/IMG_3579.jpeg', '2022-05-01 20:23:51.690000',
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural
landscape', 'Cumulus', 'Landscape']), ('/home/jupyter/images/IMG_3586.jpeg',
'2022-05-01 20:23:51.746000', ['Sky', 'Mountain', 'Snow', 'Plant', 'Natural
landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain',
'Tree']), ('/home/jupyter/images/IMG_3609.jpeg', '2022-05-01 20:23:51.800000',
['Cloud', 'Sky', 'Mountain', 'Snow', 'Slope', 'Natural landscape', 'Highland',
'World', 'Tree', 'Mountainous
landforms']), ('/home/jupyter/images/IMG_3623.jpeg', '2022-05-01
20:23:51.855000', ['Sky', 'Building', 'Window', 'Blue', 'House', 'Tower',
'Brick', 'Fixture', 'Plant', 'Facade']), ('/home/jupyter/images/IMG_3638.jpeg',
'2022-05-01 20:23:51.911000', ['Sky', 'Mountain', 'Snow', 'Ecoregion', 'Natural
landscape', 'Highland', 'Slope', 'Larch', 'Freezing',
'Terrain']), ('/home/jupyter/images/IMG_3757.jpeg', '2022-05-01 20:23:51.970000',
['Brown', 'Beak', 'Wood', 'Plant', 'Twig', 'Trunk', 'Squirrel', 'Rodent', 'Tints
and shades', 'Tail']), ('/home/jupyter/images/IMG_3793.jpeg', '2022-05-01
20:23:52.030000', ['Sky', 'Snow', 'Plant', 'Mountain', 'Natural landscape',
'Tree', 'Larch', 'Fluvial landforms of streams', 'Watercourse', 'Body of
water']), ('/home/jupyter/images/IMG_3805.jpeg', '2022-05-01 20:23:52.084000',
['Atmosphere', 'Automotive tire', 'Road surface', 'Asphalt', 'Grey', 'Body of
water', 'Black-and-white', 'Wood', 'Atmospheric phenomenon', 'Tints and
shades']), ('/home/jupyter/images/IMG_3841.jpeg', '2022-05-01 20:23:52.135000',
['Plant', 'Twig', 'Terrestrial plant', 'Natural material', 'Close-up',
'Flowering plant', 'Spider web', 'Tree', 'Grass', 'Macro photography'])

```

```

VALUES('/home/jupyter/images/IMG_3579.jpeg', '2022-05-01 20:23:51.690000',
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural
landscape', 'Cumulus', 'Landscape']), ('/home/jupyter/images/IMG_3586.jpeg',
'2022-05-01 20:23:51.746000', ['Sky', 'Mountain', 'Snow', 'Plant', 'Natural
landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain',
'Tree'])

```

```

landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain',
'Tree']), ('/home/jupyter/images/IMG_3609.jpeg', '2022-05-01 20:23:51.800000',
['Cloud', 'Sky', 'Mountain', 'Snow', 'Slope', 'Natural landscape', 'Highland',
'World', 'Tree', 'Mountainous
landforms']), ('/home/jupyter/images/IMG_3623.jpeg', '2022-05-01
20:23:51.855000', ['Sky', 'Building', 'Window', 'Blue', 'House', 'Tower',
'Brick', 'Fixture', 'Plant', 'Facade']), ('/home/jupyter/images/IMG_3638.jpeg',
'2022-05-01 20:23:51.911000', ['Sky', 'Mountain', 'Snow', 'Ecoregion', 'Natural
landscape', 'Highland', 'Slope', 'Larch', 'Freezing',
'Terrain']), ('/home/jupyter/images/IMG_3757.jpeg', '2022-05-01 20:23:51.970000',
['Brown', 'Beak', 'Wood', 'Plant', 'Twig', 'Trunk', 'Squirrel', 'Rodent', 'Tints
and shades', 'Tail']), ('/home/jupyter/images/IMG_3793.jpeg', '2022-05-01
20:23:52.030000', ['Sky', 'Snow', 'Plant', 'Mountain', 'Natural landscape',
'Tree', 'Larch', 'Fluvial landforms of streams', 'Watercourse', 'Body of
water']), ('/home/jupyter/images/IMG_3805.jpeg', '2022-05-01 20:23:52.084000',
['Atmosphere', 'Automotive tire', 'Road surface', 'Asphalt', 'Grey', 'Body of
water', 'Black-and-white', 'Wood', 'Atmospheric phenomenon', 'Tints and
shades']), ('/home/jupyter/images/IMG_3841.jpeg', '2022-05-01 20:23:52.135000',
['Plant', 'Twig', 'Terrestrial plant', 'Natural material', 'Close-up',
'Flowering plant', 'Spider web', 'Tree', 'Grass', 'Macro
photography']), ('/home/jupyter/images/IMG_3850.jpeg', '2022-05-01
20:23:52.188000', ['Water', 'Plant', 'Vegetation', 'Natural landscape', 'Fluvial
landforms of streams', 'Riparian zone', 'Grass', 'Pollution', 'Groundcover',
'Bedrock'])

```

String for inserting multiple rows:

```

VALUES('/home/jupyter/images/IMG_3579.jpeg', '2022-05-01 20:23:51.690000',
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural
landscape', 'Cumulus', 'Landscape']), ('/home/jupyter/images/IMG_3586.jpeg',
'2022-05-01 20:23:51.746000', ['Sky', 'Mountain', 'Snow', 'Plant', 'Natural
landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain',
'Tree']), ('/home/jupyter/images/IMG_3609.jpeg', '2022-05-01 20:23:51.800000',
['Cloud', 'Sky', 'Mountain', 'Snow', 'Slope', 'Natural landscape', 'Highland',
'World', 'Tree', 'Mountainous
landforms']), ('/home/jupyter/images/IMG_3623.jpeg', '2022-05-01
20:23:51.855000', ['Sky', 'Building', 'Window', 'Blue', 'House', 'Tower',
'Brick', 'Fixture', 'Plant', 'Facade']), ('/home/jupyter/images/IMG_3638.jpeg',
'2022-05-01 20:23:51.911000', ['Sky', 'Mountain', 'Snow', 'Ecoregion', 'Natural
landscape', 'Highland', 'Slope', 'Larch', 'Freezing',
'Terrain']), ('/home/jupyter/images/IMG_3757.jpeg', '2022-05-01 20:23:51.970000',
['Brown', 'Beak', 'Wood', 'Plant', 'Twig', 'Trunk', 'Squirrel', 'Rodent', 'Tints
and shades', 'Tail']), ('/home/jupyter/images/IMG_3793.jpeg', '2022-05-01
20:23:52.030000', ['Sky', 'Snow', 'Plant', 'Mountain', 'Natural landscape',
'Tree', 'Larch', 'Fluvial landforms of streams', 'Watercourse', 'Body of
water']), ('/home/jupyter/images/IMG_3805.jpeg', '2022-05-01 20:23:52.084000',
['Atmosphere', 'Automotive tire', 'Road surface', 'Asphalt', 'Grey', 'Body of
water', 'Black-and-white', 'Wood', 'Atmospheric phenomenon', 'Tints and
shades']), ('/home/jupyter/images/IMG_3841.jpeg', '2022-05-01 20:23:52.135000',
['Plant', 'Twig', 'Terrestrial plant', 'Natural material', 'Close-up',
'Flowering plant', 'Spider web', 'Tree', 'Grass', 'Macro
photography']), ('/home/jupyter/images/IMG_3850.jpeg', '2022-05-01
20:23:52.188000', ['Water', 'Plant', 'Vegetation', 'Natural landscape', 'Fluvial
landforms of streams', 'Riparian zone', 'Grass', 'Pollution', 'Groundcover',
'Bedrock'])

```

```
['Plant', 'Twig', 'Terrestrial plant', 'Natural material', 'Close-up',
'Flowering plant', 'Spider web', 'Tree', 'Grass', 'Macro
photography']),('/home/jupyter/images/IMG_3850.jpeg', '2022-05-01
20:23:52.188000', ['Water', 'Plant', 'Vegetation', 'Natural landscape', 'Fluvial
landforms of streams', 'Riparian zone', 'Grass', 'Pollution', 'Groundcover',
'Bedrock'])
```

Query for inserting multiple rows:

```
INSERT `personalproject-348318.whitebloomingtulip_db.image` (file_name,
creation_date_time, keywords)
VALUES('/home/jupyter/images/IMG_3579.jpeg', '2022-05-01 20:23:51.690000',
['Cloud', 'Mountain', 'Sky', 'Slope', 'Snow', 'Ice cap', 'Terrain', 'Natural
landscape', 'Cumulus', 'Landscape']),('/home/jupyter/images/IMG_3586.jpeg',
'2022-05-01 20:23:51.746000', ['Sky', 'Mountain', 'Snow', 'Plant', 'Natural
landscape', 'Slope', 'Highland', 'Atmospheric phenomenon', 'Terrain',
'Tree']),('/home/jupyter/images/IMG_3609.jpeg', '2022-05-01 20:23:51.800000',
['Cloud', 'Sky', 'Mountain', 'Snow', 'Slope', 'Natural landscape', 'Highland',
'World', 'Tree', 'Mountainous
landforms']),('/home/jupyter/images/IMG_3623.jpeg', '2022-05-01
20:23:51.855000', ['Sky', 'Building', 'Window', 'Blue', 'House', 'Tower',
'Brick', 'Fixture', 'Plant', 'Facade']),('/home/jupyter/images/IMG_3638.jpeg',
'2022-05-01 20:23:51.911000', ['Sky', 'Mountain', 'Snow', 'Ecoregion', 'Natural
landscape', 'Highland', 'Slope', 'Larch', 'Freezing',
'Terrain']),('/home/jupyter/images/IMG_3757.jpeg', '2022-05-01 20:23:51.970000',
['Brown', 'Beak', 'Wood', 'Plant', 'Twig', 'Trunk', 'Squirrel', 'Rodent', 'Tints
and shades', 'Tail']),('/home/jupyter/images/IMG_3793.jpeg', '2022-05-01
20:23:52.030000', ['Sky', 'Snow', 'Plant', 'Mountain', 'Natural landscape',
'Tree', 'Larch', 'Fluvial landforms of streams', 'Watercourse', 'Body of
water']),('/home/jupyter/images/IMG_3805.jpeg', '2022-05-01 20:23:52.084000',
['Atmosphere', 'Automotive tire', 'Road surface', 'Asphalt', 'Grey', 'Body of
water', 'Black-and-white', 'Wood', 'Atmospheric phenomenon', 'Tints and
shades']),('/home/jupyter/images/IMG_3841.jpeg', '2022-05-01 20:23:52.135000',
['Plant', 'Twig', 'Terrestrial plant', 'Natural material', 'Close-up',
'Flowering plant', 'Spider web', 'Tree', 'Grass', 'Macro
photography']),('/home/jupyter/images/IMG_3850.jpeg', '2022-05-01
20:23:52.188000', ['Water', 'Plant', 'Vegetation', 'Natural landscape', 'Fluvial
landforms of streams', 'Riparian zone', 'Grass', 'Pollution', 'Groundcover',
'Bedrock'])
```

[14]: # Execute the query  
query\_job\_insert\_many = bq\_client.query(query\_insert\_many, location=region)

#### 1.8.4 Drop duplicates

```
[15]: # Drop duplicated rows
query_drop_duplicates = f"""
CREATE OR REPLACE TABLE `{{gcp_project_name}}.{{bq_dataset_name}}.{{bq_table_name}}` 
PARTITION BY DATETIME_TRUNC(creation_date_time, DAY) AS (
    SELECT
        * EXCEPT(row_number)
    FROM (
        SELECT
            *,
            ROW_NUMBER() OVER (PARTITION BY file_name) row_number
        FROM
            `{{gcp_project_name}}.{{bq_dataset_name}}.{{bq_table_name}}`)
    WHERE row_number = 1
)
"""

print('Query for dropping duplicates:')
print(query_drop_duplicates)
print('')
```

Query for dropping duplicates:

```
CREATE OR REPLACE TABLE `personalproject-348318.whitebloomingtulip_db.image` 
PARTITION BY DATETIME_TRUNC(creation_date_time, DAY) AS (
    SELECT
        * EXCEPT(row_number)
    FROM (
        SELECT
            *,
            ROW_NUMBER() OVER (PARTITION BY file_name) row_number
        FROM
            `personalproject-348318.whitebloomingtulip_db.image`)
    WHERE row_number = 1
)
```

```
[16]: # Execute the query
query_job_drop_duplicates = bq_client.query(query_drop_duplicates, 
                                         location=region)
```

#### 1.8.5 Read the contents of the table

This is used to cross-check the contents of the table by importing the data in a **Pandas** dataframe.

```
[17]: # Retrieve rows from table
query_read = f"""
SELECT *
FROM `gcp_project_name.{bq_dataset_name}.{bq_table_name}`
"""

print('Query for reading the contents of the table:')
print(query_read)
print('')

# Execute the query
query_job_read = bq_client.query(query_read, location=region)

df = query_job_read.result().to_dataframe()
```

Query for reading the contents of the table:

```
SELECT *
FROM `personalproject-348318.whitebloomingtulip_db.image`
```

```
[18]: df.head(50)
```

	file_name	creation_date_time	keywords
0	/home/jupyter/images/IMG_3850.jpeg	2022-05-01 20:23:52.188	[Water, Plant, Vegetation, Natural landscape, ...]
1	/home/jupyter/images/IMG_3638.jpeg	2022-05-01 20:23:51.911	[Sky, Mountain, Snow, Ecoregion, Natural lands...]
2	/home/jupyter/images/IMG_3579.jpeg	2022-05-01 20:23:51.690	[Cloud, Mountain, Sky, Slope, Snow, Ice cap, T...]
3	/home/jupyter/images/IMG_3623.jpeg	2022-05-01 20:23:51.855	[Sky, Building, Window, Blue, House, Tower, Br...]
4	/home/jupyter/images/IMG_3805.jpeg	2022-05-01 20:23:52.084	[Atmosphere, Automotive tire, Road surface, As...]
5	/home/jupyter/images/IMG_3841.jpeg	2022-05-01 20:23:52.135	[Plant, Twig, Terrestrial plant, Natural mater...]
6	/home/jupyter/images/IMG_3793.jpeg	2022-05-01 20:23:52.030	[Sky, Snow, Plant, Mountain, Natural landscape...]
7	/home/jupyter/images/IMG_3586.jpeg	2022-05-01 20:23:51.746	[Sky, Mountain, Snow, Plant, Natural landscape...]
8	/home/jupyter/images/IMG_3609.jpeg	2022-05-01 20:23:51.800	[Cloud, Sky, Mountain, Snow, Slope, Natural la...]
9	/home/jupyter/images/IMG_3757.jpeg	2022-05-01 20:23:51.970	[Brown, Beak, Wood, Plant, Twig, Trunk, Squirr...]

## 1.9 7: Archive processed images

```
[19]: # If the images must be archived in a subfolder of the GCS bucket, specify the ↴subfolder structure
prefix_archive = f'{folder_name_archive}/'
blobs_landing = bucket.list_blobs(prefix = prefix_landing, delimiter = '/')

for blob in blobs_landing:
    if(blob.name != prefix_landing): # Ignore the subfolder itself
        bucket.rename_blob(blob, new_name=blob.name.replace(prefix_landing, ↴prefix_archive))
        print(f'{blob.name} renamed to {blob.name.replace(prefix_landing, ↴prefix_archive)}')
```

```
image/landing/IMG_3579.jpeg renamed to image/archive/IMG_3579.jpeg
image/landing/IMG_3586.jpeg renamed to image/archive/IMG_3586.jpeg
image/landing/IMG_3609.jpeg renamed to image/archive/IMG_3609.jpeg
image/landing/IMG_3623.jpeg renamed to image/archive/IMG_3623.jpeg
image/landing/IMG_3638.jpeg renamed to image/archive/IMG_3638.jpeg
image/landing/IMG_3757.jpeg renamed to image/archive/IMG_3757.jpeg
image/landing/IMG_3793.jpeg renamed to image/archive/IMG_3793.jpeg
image/landing/IMG_3805.jpeg renamed to image/archive/IMG_3805.jpeg
image/landing/IMG_3841.jpeg renamed to image/archive/IMG_3841.jpeg
image/landing/IMG_3850.jpeg renamed to image/archive/IMG_3850.jpeg
```

## 1.10 8: Clean up

### 1.10.1 Clean up the machine

```
[20]: # Delete all files in the machine
shutil.rmtree('images')
shutil.rmtree('keywords')
```

### 1.10.2 Clean up

```
[21]: # Move the images back from archive to landing
blobs_archive = bucket.list_blobs(prefix = prefix_archive, delimiter = '/')

for blob in blobs_archive:
    if(blob.name != prefix_archive): # Ignore the subfolder itself
        bucket.rename_blob(blob, new_name=blob.name.replace(prefix_archive, ↴prefix_landing))
        print(f'{blob.name} renamed to {blob.name.replace(prefix_archive, ↴prefix_landing)}')
```

```
image/archive/IMG_3579.jpeg renamed to image/landing/IMG_3579.jpeg
image/archive/IMG_3586.jpeg renamed to image/landing/IMG_3586.jpeg
image/archive/IMG_3609.jpeg renamed to image/landing/IMG_3609.jpeg
```

```
image/archive/IMG_3623.jpeg renamed to image/landing/IMG_3623.jpeg
image/archive/IMG_3638.jpeg renamed to image/landing/IMG_3638.jpeg
image/archive/IMG_3757.jpeg renamed to image/landing/IMG_3757.jpeg
image/archive/IMG_3793.jpeg renamed to image/landing/IMG_3793.jpeg
image/archive/IMG_3805.jpeg renamed to image/landing/IMG_3805.jpeg
image/archive/IMG_3841.jpeg renamed to image/landing/IMG_3841.jpeg
image/archive/IMG_3850.jpeg renamed to image/landing/IMG_3850.jpeg
```

### 1.10.3 Clean up BigQuery

Run this only if you want to delete the database.

```
[22]: # Delete all rows in the table
query_del = f"""
DROP TABLE `gcp_project_name.{bq_dataset_name}.{bq_table_name}`
"""

print('Query for deleting all rows in the table:')
print(query_del)
print('')

# Execute the query
query_job_del = bq_client.query(query_del, location=region)
```

Query for deleting all rows in the table:

```
DROP TABLE `personalproject-348318.whitebloomingtulip_db.image`
```

## 1.11 Possible extensions

So far we assumed that, after uploading new pictures to Google Cloud, we manually execute the code in the `Vertex AI` notebook to generate the keywords and update the database. One step further could be to trigger the execution of the code when one or more files are uploaded to the relevant `Cloud Storage` bucket in Google Cloud. We can do that in a couple of different ways. One way is to use a `Cloud Function`. `Cloud Functions` are a function-as-a-service (FaaS) product that allows you to execute your code without having to worry about infrastructure (no servers and no containers to manage) and to pay only for the execution time of the code. `Cloud Functions` are event-driven and can be triggered by several events related to `Cloud Storage`, in particular: - object creation - object deletion - object archiving - metadata updates

Another way to achieve the same result is to use `Dataflow`, a product that allows to deal with batch and stream data processing in a serverless way. `Dataflow` is built around `Apache Beam`, an open-source model for defining batch and stream data processing pipelines.

Instead of having our code triggered by an event, we could decide to execute it according to a certain schedule, for example once a day or once a week. This can be done using `Composer`, a workflow orchestration service built on `Apache Airflow`.

## 1.12 References

- Cloud Storage on Google Cloud: <https://cloud.google.com/storage>
- BigQuery on Google Cloud: <https://cloud.google.com/bigquery>
- Vision AI API on Google Cloud: <https://cloud.google.com/vision>
- Cloud Functions on Google Cloud: <https://cloud.google.com/functions>
- Dataflow on Google Cloud: <https://cloud.google.com/dataflow>
- Composer on Google Cloud: <https://cloud.google.com/composer>
- How to use the BigQuery API: <https://cloud.google.com/bigquery/docs/quickstarts/quickstart-client-libraries>
- How to specify a schema in BigQuery: <https://cloud.google.com/bigquery/docs/schemas>
- How to create a partitioned table in BigQuery: <https://cloud.google.com/bigquery/docs/creating-partitioned-tables>
- How to write the results of a query in BigQuery: <https://cloud.google.com/bigquery/docs/writing-results>
- How to download BigQuery data to a Pandas dataframe: <https://cloud.google.com/bigquery/docs/bigquery-storage-python-pandas>
- How to trigger a Cloud Function from a Cloud Storage event: <https://cloud.google.com/functions/docs/calling/storage>
- Apache Beam programming guide, including triggers: <https://beam.apache.org/documentation/programming-guide/>
- Apache Airflow: <https://airflow.apache.org/>

[ ]: