In [ ]:

```
! pip install -r requirements.txt
```

```
Collecting pymilvus==2.0.0
  Using cached pymilvus-2.0.0-py3-none-any.whl (119 kB)
Collecting towhee==0.6.0
  Using cached towhee-0.6.0-py3-none-any.whl (322 kB)
Requirement already satisfied: redis in /home/sumit/.local/lib/python3.8/site-packages (from -r requirements.txt
(line 3)) (4.3.4)
Requirement already satisfied: torch in /home/sumit/.local/lib/python3.8/site-packages (from -r requirements.txt
(line 4)) (1.12.0)
Requirement already satisfied: torchvision in /home/sumit/.local/lib/python3.8/site-packages (from -r requirement
s.txt (line 5)) (0.13.0)
Requirement already satisfied: opencv-python in /home/sumit/.local/lib/python3.8/site-packages (from -r requiremen
ts.txt (line 6)) (4.6.0.66)
Requirement already satisfied: matplotlib in /home/sumit/.local/lib/python3.8/site-packages (from -r requirements.
txt (line 7)) (3.5.2)
Requirement already satisfied: pillow in /usr/lib/python3/dist-packages (from -r requirements.txt (line 8)) (7.0.
0)
Requirement already satisfied: numpy in /home/sumit/.local/lib/python3.8/site-packages (from -r requirements.txt
(line 9)) (1.22.4)
Requirement already satisfied: gdown in /home/sumit/.local/lib/python3.8/site-packages (from -r requirements.txt
(line 10)) (4.5.1)
Requirement already satisfied: mmh3<=3.0.0,>=2.0 in /home/sumit/.local/lib/python3.8/site-packages (from pymilvus=
=2.0.0->-r requirements.txt (line 1)) (3.0.0)
Requirement already satisfied: pandas<=1.3.5,>=1.2.4; python_version > "3.6" in /home/sumit/.local/lib/python3.8/s
ite-packages (from pymilvus==2.0.0->-r requirements.txt (line 1)) (1.3.5)
Requirement already satisfied: grpcio-tools==1.37.1 in /home/sumit/.local/lib/python3.8/site-packages (from pymilv
us==2.0.0->-r requirements.txt (line 1)) (1.37.1)
Requirement already satisfied: ujson<=5.1.0,>=2.0.0 in /home/sumit/.local/lib/python3.8/site-packages (from pymilv
us==2.0.0->-r requirements.txt (line 1)) (5.1.0)
Requirement already satisfied: grpcio==1.37.1 in /home/sumit/.local/lib/python3.8/site-packages (from pymilvus==2.
0.0->-r requirements.txt (line 1)) (1.37.1)
Requirement already satisfied: ruamel.yaml<=0.16.6 in /home/sumit/.local/lib/python3.8/site-packages (from towhee=
=0.6.0->-r requirements.txt (line 2)) (0.16.6)
Requirement already satisfied: tqdm>=4.59.0 in /home/sumit/.local/lib/python3.8/site-packages (from towhee==0.6.0-
>-r requirements.txt (line 2)) (4.64.0)
Requirement already satisfied: tabulate in /home/sumit/.local/lib/python3.8/site-packages (from towhee==0.6.0->-r
requirements.txt (line 2)) (0.8.10)
Requirement already satisfied: requests>=2.12.5 in /usr/lib/python3/dist-packages (from towhee==0.6.0->-r requirem
ents.txt (line 2)) (2.22.0)
Requirement already satisfied: deprecated>=1.2.3 in /home/sumit/.local/lib/python3.8/site-packages (from redis->-r
requirements.txt (line 3)) (1.2.13)
Requirement already satisfied: async-timeout>=4.0.2 in /home/sumit/.local/lib/python3.8/site-packages (from redis-
>-r requirements.txt (line 3)) (4.0.2)
```

```
Requirement already satisfied: packaging>=20.4 in /home/sumit/.local/lib/python3.8/site-packages (from redis->-r r
equirements.txt (line 3)) (21.3)
Requirement already satisfied: typing-extensions in /home/sumit/.local/lib/python3.8/site-packages (from torch->-r
requirements.txt (line 4)) (4.3.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /home/sumit/.local/lib/python3.8/site-packages (from matplotli
b->-r requirements.txt (line 7)) (1.4.3)
Requirement already satisfied: python-dateutil>=2.7 in /home/sumit/.local/lib/python3.8/site-packages (from matplo
tlib->-r requirements.txt (line 7)) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in /home/sumit/.local/lib/python3.8/site-packages (from matplotlib
->-r requirements.txt (line 7)) (3.0.9)
Requirement already satisfied: cycler>=0.10 in /home/sumit/.local/lib/python3.8/site-packages (from matplotlib->-r
requirements.txt (line 7)) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /home/sumit/.local/lib/python3.8/site-packages (from matplotli
b->-r requirements.txt (line 7)) (4.33.3)
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from gdown->-r requirements.txt (line 10))
(1.14.0)
Requirement already satisfied: beautifulsoup4 in /home/sumit/.local/lib/python3.8/site-packages (from gdown->-r re
quirements.txt (line 10)) (4.11.1)
Requirement already satisfied: filelock in /home/sumit/.local/lib/python3.8/site-packages (from gdown->-r requirem
ents.txt (line 10)) (3.7.1)
Requirement already satisfied: pytz>=2017.3 in /home/sumit/.local/lib/python3.8/site-packages (from pandas<=1.3.5,
>=1.2.4; python_version > "3.6"->pymilvus==2.0.0->-r requirements.txt (line 1)) (2022.1)
Requirement already satisfied: protobuf<4.0dev,>=3.5.0.post1 in /home/sumit/.local/lib/python3.8/site-packages (fr
om grpcio-tools==1.37.1->pymilvus==2.0.0->-r requirements.txt (line 1)) (3.20.1)
Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages (from grpcio-tools==1.37.1->pymilvus==
2.0.0->-r requirements.txt (line 1)) (45.2.0)
Requirement already satisfied: wrapt<2,>=1.10 in /home/sumit/.local/lib/python3.8/site-packages (from deprecated>=
1.2.3->redis->-r requirements.txt (line 3)) (1.14.1)
Requirement already satisfied: soupsieve>1.2 in /home/sumit/.local/lib/python3.8/site-packages (from beautifulsoup
4->gdown->-r requirements.txt (line 10)) (2.3.2.post1)
Installing collected packages: pymilvus, towhee
  Attempting uninstall: pymilvus
    Found existing installation: pymilvus 2.0.1
    Uninstalling pymilvus-2.0.1:
      Successfully uninstalled pymilvus-2.0.1
  Attempting uninstall: towhee
    Found existing installation: towhee 0.7.1
    Uninstalling towhee-0.7.1:
      Successfully uninstalled towhee-0.7.1
Successfully installed pymilvus-2.0.0 towhee-0.6.0
```

In [ ]:

```
! docker run --name redis -d -p 6379:6379 redis
```

In [ ]:

```
#Connectings to Milvus and Redis

from pymilvus import connections
import redis

connections.connect(host="127.0.0.1", port=19530)
red = redis.Redis(host = '127.0.0.1', port=6379, db=0)
red.flushdb()
```

Out[ ]:

```
True
```

In [ ]:

```
#Creat a collection
from pymilvus import CollectionSchema, FieldSchema, DataType, Collection

collection_name = "image_search_engine"
dim = 2048
default_fields = [
    FieldSchema(name="id", dtype=DataType.INT64, is_primary=True, auto_id=True),
    FieldSchema(name="vector", dtype=DataType.FLOAT_VECTOR, dim=dim)
]
default_schema = CollectionSchema(fields=default_fields, description="Search Image collection")

collection = Collection(name=collection_name, schema=default_schema)
```

In [ ]:

```
# Create IVF_SQ8 index to the  collection
default_index = {"index_type": "IVF_SQ8", "params": {"nlist": 2048}, "metric_type": "L2"}
collection.create_index(field_name="vector", index_params=default_index)
collection.load()
```

In [ ]:

```
! pip install scipy
```

Download Training Image data from Cortx S3

In [ ]:

```
import boto3

# create bucket named the current date
ACCESS_KEY = 'sgiamadmin'
SECRET_ACCESS_KEY = 'ldapadmin'
END_POINT_URL = 'http://192.168.1.14:31949'

s3_client = boto3.client('s3', endpoint_url=END_POINT_URL,
                         aws_access_key_id=ACCESS_KEY,
                         aws_secret_access_key=SECRET_ACCESS_KEY,
                         verify=False)

s3_resource = boto3.resource('s3', endpoint_url=END_POINT_URL,
                             aws_access_key_id=ACCESS_KEY,
                             aws_secret_access_key=SECRET_ACCESS_KEY,
                             region_name='None',
                             verify=False)
```

In [ ]:

```
buckets = s3_client.list_buckets()

if buckets['Buckets']:
    for bucket in buckets['Buckets']:
        print(bucket)
```

```
{'Name': 'datasets', 'CreationDate': datetime.datetime(2022, 7, 4, 3, 7, 36, 664000, tzinfo=tzutc())}
{'Name': 'milvus', 'CreationDate': datetime.datetime(2022, 7, 3, 12, 14, 21, 955000, tzinfo=tzutc())}
{'Name': 'mybucket', 'CreationDate': datetime.datetime(2022, 6, 30, 13, 1, 26, 77000, tzinfo=tzutc())}
```

In [ ]:

```
s3_resource.Bucket('datasets').download_file('VOCtrainval_11-May-2012.zip', 'VOCtrainval_11-May-2012.zip')
```

Extract the image files to decode them and translate them to vectors

In [ ]:

```
import zipfile

with zipfile.ZipFile("VOCtrainval_11-May-2012.zip","r") as zip_ref:
    zip_ref.extractall("./VOCtrainval_11-May-2012")
```

In [ ]:

```
! rm -rf ./VOCtrainval_11-May-2012/VOCtrainval_11-May-2012/VOCdevkit/VOC2012/Annotations
! rm -rf ./VOCtrainval_11-May-2012/VOCtrainval_11-May-2012/VOCdevkit/VOC2012/ImageSets
! rm -rf ./VOCtrainval_11-May-2012/VOCtrainval_11-May-2012/VOCdevkit/VOC2012/Annotations
! rm -rf ./VOCtrainval_11-May-2012/VOCtrainval_11-May-2012/VOCdevkit/VOC2012/SegmentationClass
! rm -rf ./VOCtrainval_11-May-2012/VOCtrainval_11-May-2012/VOCdevkit/VOC2012/SegmentationObject
```

Insert the data into milvus tool

In [ ]:

```
import os
import towhee

data_dir = "./VOCtrainval_11-May-2012/VOCtrainval_11-May-2012/VOCdevkit/VOC2012/" # You can replace this to your lo
pattern = "*.jpg"

subfolders = [os.path.join(data_dir, x) for x in os.listdir(data_dir) if os.path.isdir(os.path.join(data_dir, x))]
print(subfolders)
steps = len(subfolders)
step = 1
for sub_dir in subfolders:
    img_pattern = os.path.join(sub_dir, pattern)
    paths = towhee.glob(img_pattern).to_list()
    vectors = towhee.glob(img_pattern).exception_safe() \
                    .image_decode() \
                    .image_embedding.timm(model_name="resnet50") \
                    .drop_empty() \
                    .tensor_normalize() \
                    .to_list()

    mr = collection.insert([vectors])
    ids = mr.primary_keys

    for x in range(len(ids)):
        red.set(str(ids[x]), paths[x])


    print("Inserting progress: " + str(step) + "/" + str(steps))
    step += 1
```

```
['./VOCtrainval_11-May-2012/VOCtrainval_11-May-2012/VOCdevkit/VOC2012/JPEGImages']
Inserting progress: 1/1
```

In [ ]:

```
random_ids = [int(red.randomkey()) for x in range(3)]
search_images = [x.decode("utf-8") for x in red.mget(random_ids)]
```

In [ ]:

```
# Process and search for the query image

query_vectors = towhee.dc(search_images) \
                    .image_decode() \
                    .image_embedding.timm(model_name="resnet50") \
                    .tensor_normalize() \
                    .to_list()
```

In [ ]:

```
# Searching

import time
search_params = {"metric_type": "L2", "params": {"nprobe": 32}}
start = time.time()
results = collection.search(query_vectors, "vector", param=search_params, limit=3, expr=None)
end = time.time() - start

print("Search took a total of: ", end)
```

```
Search took a total of:  0.7474019527435303
```

In [ ]:

```python
#Helper display function

import matplotlib.pyplot as plt
from PIL import Image

def show_results(query, results, distances):

    fig_query, ax_query = plt.subplots(1,1, figsize=(5,5))
    ax_query.imshow(Image.open(query))
    ax_query.axis('off')
    ax_query.set_title("Searched Image")

    res_count = len(results)
    fig, ax = plt.subplots(1,res_count,figsize=(10,10))
    for x in range(res_count):
        ax[x].imshow(Image.open(results[x]))
        ax[x].axis('off')
        dist =  str(distances[x])
        dist = dist[0:dist.find('.')+4]
        ax[x].set_title("D:  " +dist)
```

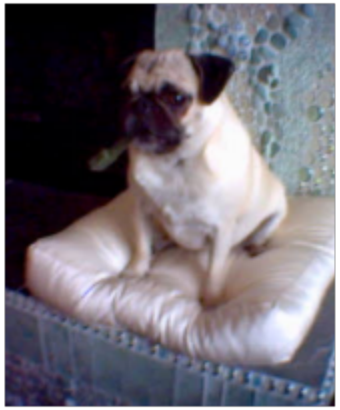
In [ ]:

```python
for x in range(len(results)):
    query_file = search_images[x]
    result_files = [red.get(y.id).decode('utf-8') for y in results[x]]
    distances = [y.distance for y in results[x]]
    show_results(query_file, result_files, distances)
```

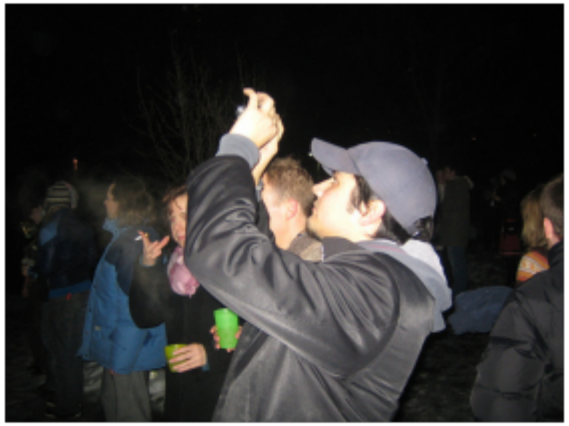
D: 0.0

D: 0.766

D: 0.894



Searched Image

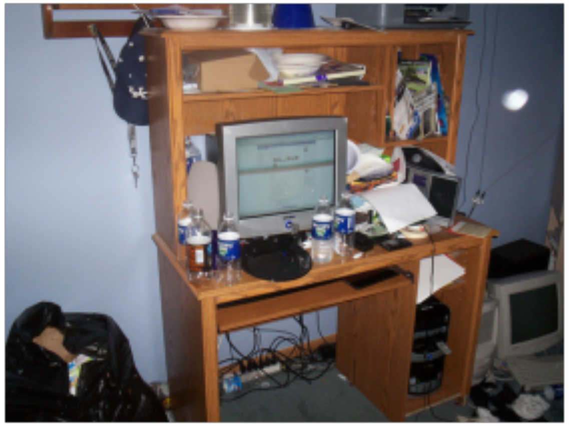


D: 0.0

D: 0.699

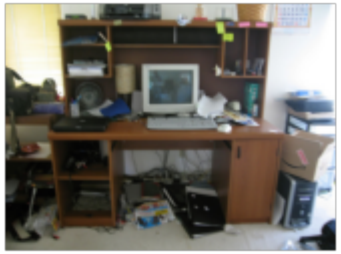
D: 0.715

Searched Image

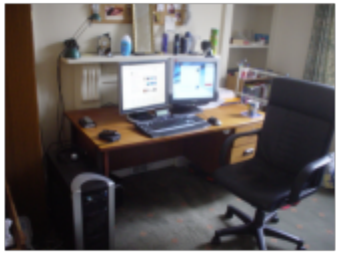


D: 0.0

D: 0.254



D: 0.467



```
In [ ]:
```