

```

gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
    print('Not connected to a GPU')
else:
    print(gpu_info)

```

```

Sun Feb  5 19:30:07 2023

```

```

+-----+
| NVIDIA-SMI 510.47.03   Driver Version: 510.47.03   CUDA Version: 11.6   |
+-----+-----+-----+-----+
| GPU  Name            Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|====================================+=====+
|   0  NVIDIA A100-SXM...  Off      | 00000000:00:04:0 Off |           0          |
| N/A   33C    P0      50W / 400W  |  0MiB / 40960MiB |          0%    Default |
|                                     |                      | MIG M.     Disabled |
+-----+-----+-----+-----+

```

```

+-----+
| Processes:
| GPU   GI    CI          PID    Type    Process name                        GPU Memory
|   ID   ID                                     Usage                        |
+-----+
| No running processes found
+-----+

```

```

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import PIL
import os
import tarfile
import urllib

```

```

from PIL import Image, ImageDraw
import shutil

```

```

!pip install -U monk-colab
from monk.pytorch_prototype import prototype

```

```

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/

```

```

Collecting monk-colab

```

```

  Downloading monk_colab-0.0.1-py3-none-any.whl (515 kB)

```

```

    515.0/515.0 KB 10.7 MB/s eta 0:00:00

```

```

Collecting gluoncv

```

```

  Downloading gluoncv-0.10.5.post0-py2.py3-none-any.whl (1.3 MB)

```

```

    1.3/1.3 MB 52.2 MB/s eta 0:00:00

```

```

Collecting GPUtil

```

```

  Downloading GPUtil-1.4.0.tar.gz (5.5 kB)

```

```

  Preparing metadata (setup.py) ... done

```

```

Collecting mxnet-cu101

```

```

  Downloading mxnet_cu101-1.9.1-py3-none-manylinux2014_x86_64.whl (360.0 MB)

```

```

    360.0/360.0 MB 3.3 MB/s eta 0:00:00

```

```

Collecting pylg

```

```

  Downloading Pylg-1.3.3-py3-none-any.whl (27 kB)

```

```

Requirement already satisfied: opencv-python in /usr/local/lib/python3.8/dist-packages (from gluoncv->monk-colab) (4.6.0.66)

```

```

Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from gluoncv->monk-colab) (1.21.6)

```

```

Requirement already satisfied: matplotlib in /usr/local/lib/python3.8/dist-packages (from gluoncv->monk-colab) (3.2.2)

```

```

Collecting portalocker

```

```

  Downloading portalocker-2.7.0-py2.py3-none-any.whl (15 kB)

```

```

Collecting yacs

```

```

  Downloading yacs-0.1.8-py3-none-any.whl (14 kB)

```

```

Requirement already satisfied: pandas in /usr/local/lib/python3.8/dist-packages (from gluoncv->monk-colab) (1.3.5)

```

```

Collecting autocfg

```

```

  Downloading autocfg-0.0.8-py3-none-any.whl (13 kB)

```

```

Requirement already satisfied: Pillow in /usr/local/lib/python3.8/dist-packages (from gluoncv->monk-colab) (7.1.2)

```

```

Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-packages (from gluoncv->monk-colab) (1.7.3)

```

```

Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages (from gluoncv->monk-colab) (4.64.1)

```

```

Requirement already satisfied: pyyaml in /usr/local/lib/python3.8/dist-packages (from gluoncv->monk-colab) (6.0)

```

```

Requirement already satisfied: requests in /usr/local/lib/python3.8/dist-packages (from gluoncv->monk-colab) (2.25.1)

```

```

Collecting graphviz<0.9.0,>=0.8.1

```

```

  Downloading graphviz-0.8.4-py2.py3-none-any.whl (16 kB)

```

```

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests->gluoncv->monk-colab) (2022.7.1)

```

```

Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests->gluoncv->monk-colab) (1.26.13)

```

```

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests->gluoncv->monk-colab) (2.10)

```

```

Requirement already satisfied: chardet<5,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests->gluoncv->monk-colab) (4.0.0)

```

```

Requirement already satisfied: cyclers>=0.10 in /usr/local/lib/python3.8/dist-packages (from matplotlib->gluoncv->monk-colab) (0.11.0)

```

```

Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib->gluoncv->monk-colab) (1.4.4)

```

```

Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib->gluoncv->monk-colab) (2.8.2)

```

```

Requirement already satisfied: pyparsing!=2.0.4,!>=2.1.2,!>=2.1.6,>=2.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib->gluoncv->monk-colab) (3.0.9)

```

```

Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-packages (from pandas->gluoncv->monk-colab) (2022.7.1)

```

```

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.1->matplotlib->gluoncv->monk-colab) (1.16.0)

```

```

Building wheels for collected packages: GPUtil

```

```

Building wheel for GPUUtil (setup.py) ... done
Created wheel for GPUUtil: filename=GPUUtil-1.4.0-py3-none-any.whl size=7409 sha256=ad71f69c2b12d9f7570d24867924b8d025574932021af47
Stored in directory: /root/.cache/pip/wheels/ba/03/bb/7a97840eb54479b328672e15a536e49dc60da200fb21564d53
Successfully built GPUUtil
Installing collected packages: pylg, GPUUtil, yacs, portlocker, graphviz, autocfg, mxnet-cu101, gluoncv, monk-colab
Attempting uninstall: graphviz
  Found existing installation: graphviz 0.10.1
  Uninstalling graphviz-0.10.1:
    Successfully uninstalled graphviz-0.10.1
Successfully installed GPUUtil-1.4.0 autocfg-0.0.8 gluoncv-0.10.5.post0 graphviz-0.8.4 monk-colab-0.0.1 mxnet-cu101-1.9.1 portlocke

```

```
!pip install --upgrade torch torchvision TorchAudio
```

```

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: torch in /usr/local/lib/python3.8/dist-packages (1.13.1+cu116)
Requirement already satisfied: torchvision in /usr/local/lib/python3.8/dist-packages (0.14.1+cu116)
Requirement already satisfied: TorchAudio in /usr/local/lib/python3.8/dist-packages (0.13.1+cu116)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.8/dist-packages (from torch) (4.4.0)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3.8/dist-packages (from torchvision) (7.1.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from torchvision) (1.21.6)
Requirement already satisfied: requests in /usr/local/lib/python3.8/dist-packages (from torchvision) (2.25.1)
Requirement already satisfied: chardet<5,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests->torchvision) (4.0.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests->torchvision) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests->torchvision) (2022.12.7)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests->torchvision) (2.10)

```

```

from torchvision import transforms as tf
import torch
import torchvision
from torch.autograd import Variable as V
from torchvision import transforms as trn
from torch.nn import functional as F

```

```
#!wget http://data.csail.mit.edu/places/places365/places365standard_easyformat.tar
```

```
tar = tarfile.open("drive/MyDrive/TFG/places365standard_easyformat.tar")
```

```
tar.extractall("places365standard_easyformat")
```

```
!git clone https://github.com/CSAILVision/places365.git
```

```

Cloning into 'places365'...
remote: Enumerating objects: 264, done.
remote: Total 264 (delta 0), reused 0 (delta 0), pack-reused 264
Receiving objects: 100% (264/264), 1.37 MiB | 8.92 MiB/s, done.
Resolving deltas: 100% (145/145), done.

```

Haz doble clic (o pulsa Intro) para editar

```

removeElements = ['train', 'oilrig', 'veterinarians_office', 'racecourse', 'arena-rodeo', 'temple-asia', 'stadium-football', 'department_
goodElements = ['corridor', 'computer_room', 'bathroom', 'kitchen', 'staircase', 'bedroom']
print(len(goodElements))
#listElements = os.walk("places365standard_easyformat/places365_standard/train")
#for element in listElements:
# if os.path.basename(element[0]) not in removeElements:
#   goodElements.append(os.path.basename(element[0]))
print("Elements to train: " + str(goodElements))
print("Elements to remove: " + str(removeElements))

```

```

6
Elements to train: ['corridor', 'computer_room', 'bathroom', 'kitchen', 'staircase', 'bedroom']
Elements to remove: ['train', 'oilrig', 'veterinarians_office', 'racecourse', 'arena-rodeo', 'temple-asia', 'stadium-football', 'de

```

```

trainFile = open('places365standard_easyformat/places365_standard/train.txt', 'r')
lines = trainFile.readlines()

```

```

if not os.path.exists("customPlaces"):
    os.mkdir("customPlaces")

```

```

if not os.path.exists("customPlaces/train"):
    os.mkdir("customPlaces/train")

```

```
for element in goodElements:
```

```
if not os.path.exists("customPlaces/train/" + element):
    os.mkdir("customPlaces/train/" + element)

customTrainFile = open("customPlaces/train.txt", "a")

for line in lines:
    line = line.strip()
    if os.path.exists("places365standard_easyformat/places365_standard/" + line):
        if os.path.basename(os.path.dirname(line)) in goodElements:
            customTrainFile.write(line + "\n")
            shutil.copyfile("places365standard_easyformat/places365_standard/" + line, "customPlaces/" + line)

customTrainFile.close()

valFile = open('places365standard_easyformat/places365_standard/val.txt', 'r')
lines = valFile.readlines()

if not os.path.exists("customPlaces"):
    os.mkdir("customPlaces")

if not os.path.exists("customPlaces/val"):
    os.mkdir("customPlaces/val")

for element in goodElements:
    if not os.path.exists("customPlaces/val/" + element):
        os.mkdir("customPlaces/val/" + element)

customValFile = open("customPlaces/val.txt", "a")

for line in lines:
    line = line.strip()
    if os.path.exists("places365standard_easyformat/places365_standard/" + line):
        if os.path.basename(os.path.dirname(line)) in goodElements:
            customValFile.write(line + "\n")
            shutil.copyfile("places365standard_easyformat/places365_standard/" + line, "customPlaces/" + line)

customTrainFile.close()

!python3 places365/train_placesCNN.py -a resnet18 customPlaces --num_classes 6 --epochs 90 --lr 0.01
```

```

Epoch: [89][40/112]    Time 0.097 (0.194)    Data 0.000 (0.121)    Loss 0.4519 (0.4311)    Prec@1 82.812 (84.918)    Prec@5 91.125 (92.125)
Epoch: [89][50/112]    Time 0.097 (0.183)    Data 0.000 (0.111)    Loss 0.4036 (0.4294)    Prec@1 85.547 (84.995)    Prec@5 91.125 (92.125)
Epoch: [89][60/112]    Time 0.231 (0.176)    Data 0.205 (0.106)    Loss 0.3424 (0.4285)    Prec@1 88.672 (85.041)    Prec@5 91.125 (92.125)
Epoch: [89][70/112]    Time 0.097 (0.173)    Data 0.000 (0.103)    Loss 0.4659 (0.4269)    Prec@1 81.641 (85.024)    Prec@5 91.125 (92.125)
Epoch: [89][80/112]    Time 0.652 (0.173)    Data 0.591 (0.104)    Loss 0.3722 (0.4214)    Prec@1 85.547 (85.214)    Prec@5 91.125 (92.125)
Epoch: [89][90/112]    Time 0.098 (0.167)    Data 0.001 (0.097)    Loss 0.4151 (0.4244)    Prec@1 86.328 (85.126)    Prec@5 91.125 (92.125)
Epoch: [89][100/112]    Time 0.096 (0.165)    Data 0.000 (0.095)    Loss 0.3686 (0.4230)    Prec@1 86.719 (85.164)    Prec@5 91.125 (92.125)
Epoch: [89][110/112]    Time 0.315 (0.163)    Data 0.298 (0.093)    Loss 0.4634 (0.4232)    Prec@1 82.812 (85.167)    Prec@5 91.125 (92.125)
Test: [0/3]            Time 1.283 (1.283)    Loss 0.3815 (0.3815)    Prec@1 87.891 (87.891)    Prec@5 100.000 (100.000)

```

```

# hacky way to deal with the Pytorch 1.0 update
def recursion_change_bn(module):
    if isinstance(module, torch.nn.BatchNorm2d):
        module.track_running_stats = 1
    else:
        for i, (name, module1) in enumerate(module._modules.items()):
            module1 = recursion_change_bn(module1)
    return module

# this model has a last conv feature map as 14x14

model_file = 'resnet18_best.pth.tar'
if not os.access(model_file, os.W_OK):
    os.system('wget https://raw.githubusercontent.com/csailvision/places365/master/wideresnet.py')

import wideresnet
model = wideresnet.resnet18(num_classes=6)
checkpoint = torch.load(model_file, map_location=lambda storage, loc: storage)
state_dict = {str.replace(k, 'module.', ''): v for k, v in checkpoint['state_dict'].items()}
model.load_state_dict(state_dict)

# hacky way to deal with the upgraded batchnorm2D and avgpool layers...
for i, (name, module) in enumerate(model._modules.items()):
    module = recursion_change_bn(module)
model.avgpool = torch.nn.AvgPool2d(kernel_size=14, stride=1, padding=0)

model.eval()

(conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=1)
(downsample): Sequential(
  (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
  (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=1)
)
(1): BasicBlock(
  (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=1)
  (relu): ReLU(inplace=True)
  (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=1)
)
(layer3): Sequential(
  (0): BasicBlock(
    (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=1)
    (relu): ReLU(inplace=True)
    (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=1)
  )

```

```

    (l): BasicBlock(
      (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (avgpool): AvgPool2d(kernel_size=14, stride=1, padding=0)
  (fc): Linear(in_features=512, out_features=6, bias=True)
)

# load the image transformer
centre_crop = trn.Compose([
    trn.Resize((256,256)),
    trn.CenterCrop(224),
    trn.ToTensor(),
    trn.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])

for element in goodElements:
    print(element)

    corridor
    computer_room
    bathroom
    kitchen
    staircase
    bedroom

# load the class label
file_name = 'categories_places365.txt'
if not os.access(file_name, os.W_OK):
    synset_url = 'https://raw.githubusercontent.com/csailvision/places365/master/categories_places365.txt'
    os.system('wget ' + synset_url)

classes = list()
with open(file_name) as class_file:
    for line in class_file:
        classes.append(line.strip().split(' ')[0][3:])
classes = tuple(classes)

# load the test image
img_name_1 = "opencv_frame_0.png"
img_name_2 = "prueba_2.jfif"

img_1 = Image.open(img_name_1)
img_2 = Image.open(img_name_2)
input_img_1 = V(centre_crop(img_1).unsqueeze(0))
input_img_2 = V(centre_crop(img_2).unsqueeze(0))

# forward pass
logit_1 = model.forward(input_img_1)
h_x_1 = F.softmax(logit_1, 1).data.squeeze()
probs_1, idx_1 = h_x_1.sort(0, True)

# forward pass
logit_2 = model.forward(input_img_2)
h_x_2 = F.softmax(logit_2, 1).data.squeeze()
probs_2, idx_2 = h_x_2.sort(0, True)

arch = 'resnet18'

print('{} prediction on {}'.format(arch, img_name_1))
# output the prediction
for i in range(0, 5):
    print('{:.3f} -> {}'.format(probs_1[i], classes[idx_1[i]]))

print('{} prediction on {}'.format(arch, img_name_2))
# output the prediction
for i in range(0, 5):
    print('{:.3f} -> {}'.format(probs_2[i], classes[idx_2[i]]))

resnet18 prediction on opencv_frame_0.png
0.572 -> corridor
0.207 -> bathroom
0.112 -> bedroom
0.046 -> computer_room
0.045 -> staircase
resnet18 prediction on prueba_2.jfif

```

```
0.385 -> computer_room  
0.232 -> bathroom  
0.141 -> bedroom  
0.104 -> staircase  
0.091 -> kitchen
```

