

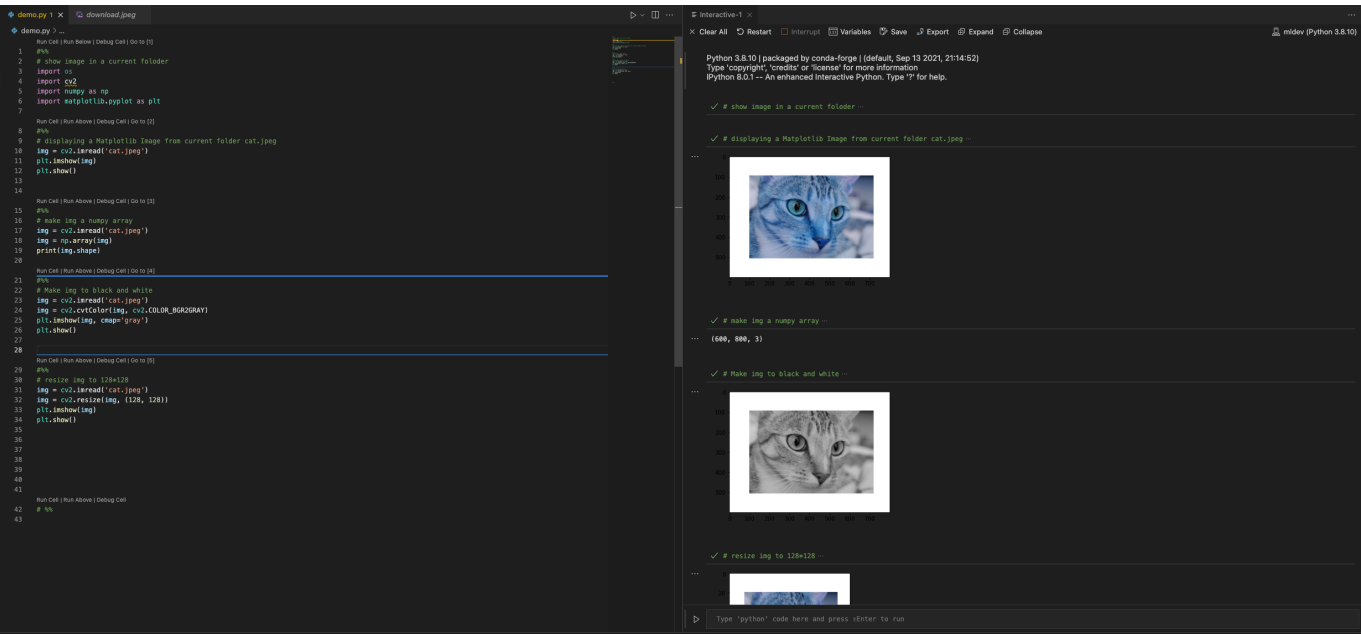
Python应用实验

实验一

在 lab2 文件夹下创建一个 demo.py 依次输入以下语句

- 1. show image in a current foloder
- 2. displaying a Matplotlib Image from current folder cat.jpeg
- 3. make img a numpy array
- 4. Make img to black and white
- 5. resize img to 128*128

结果如下：



实验二

在 lab2 文件夹中创建一个 demo2.py 输入如下

- 1.

```
'''  
  
function to unzip a file  
  
'''  
import zipfile  
import os  
import sys  
import shutil
```

2.

```
'''  
  
function to zip a file  
  
'''
```

```
'''  
  
function to unzip a file  
  
'''  
import zipfile  
import os  
import sys  
import shutil  
  
def unzip(file_name, path):  
    with zipfile.ZipFile(file_name, 'r') as zip_ref:  
        zip_ref.extractall(path)  
  
'''  
  
function to zip a file  
  
'''  
def zip(file_name, path):  
    with zipfile.ZipFile(file_name, 'w') as zip_ref:  
        zip_ref.write(path)
```

实验三

```
''''  
  
Create an image classifier using PyTorch.  
1. Load the data  
2. Define the model  
3. Define the loss function  
4. Define the optimizer  
5. Train the model  
6. Save the model  
  
''''
```

Playground

Load a preset...

SaveView codeShare

1
2
3 Create an image classifier using PyTorch.
4 1. Load the data
5 2. Define the model
6 3. Define the loss function
7 4. Define the optimizer
8 5. Train the model
9 6. Save the model
10
11
12
13 import torch
14 from torch import nn
15 from torch import optim
16 import torch.nn.functional as F
17 from torchvision import datasets, transforms, models
18 from collections import OrderedDict
19 import numpy as np
20 import matplotlib.pyplot as plt
21 import json
22 import argparse
23
24 # Define the arguments
25 parser = argparse.ArgumentParser(description='Train a neural network to classify images')
26 parser.add_argument('--data_dir', action='store', help='Path to the data directory')
27 parser.add_argument('--save_dir', action='store', dest='save_dir', default='checkpoint.pth', help='Path to the save directory')
28 parser.add_argument('--arch', action='store', dest='arch', default='vgg16', help='Model architecture')
29 parser.add_argument('--learning_rate', action='store', dest='learning_rate', default=0.001, type=float, help='Learning rate')
30 parser.add_argument('--hidden_units', action='store', dest='hidden_units', default=512, type=int, help='Number of hidden units')
31 parser.add_argument('--epochs', action='store', dest='epochs', default=5, type=int, help='Number of epochs')
32 parser.add_argument('--gpu', action='store_true', dest='gpu', default=False, help='Use GPU for training')
33
34 # Parse the arguments
35 args = parser.parse_args()
36
37 # Load the data
38 data_dir = args.data_dir
39 train_dir = data_dir + '/train'
40 valid_dir = data_dir + '/valid'
41 test_dir = data_dir + '/test'
42
43 # Define transforms for the training, validation, and testing sets
44 train_transforms = transforms.Compose([transforms.RandomRotation(30),
45 transforms.RandomResizedCrop(224),
46 transforms.RandomHorizontalFlip(),
47 transforms.ToTensor(),
48 transforms.Normalize([0.485, 0.456, 0.406],

Mode

Engine

code-davinci-002

Temperature0

Maximum length5215

Stop sequences

Enter sequence and press Tab

Top P1

Frequency penalty0

Presence penalty0

Best of1

Inject start text

☒

Inject restart text

☒

Show probabilities

Off

4 / 4