Kissipo Learning for Deep Learning

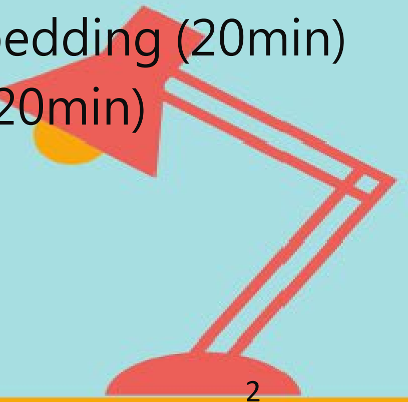
Topic 19: AOI simple Pipeline (B) (20min)

Hsueh-Ting Chu

KLDL-W7-T19

Topics

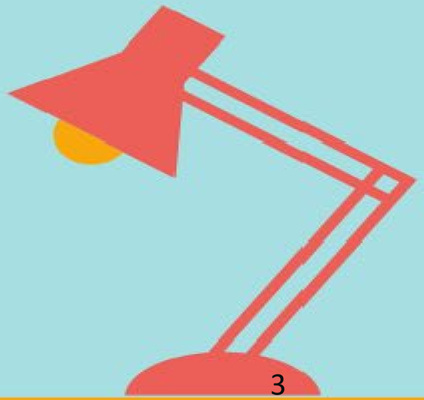
- Topic 01: Introduction to Deep Learning (20min)
- Topic 02: Kissipo Learning for Deep Learning (20min)
- Topic 03: Python quick tutorial (20min)
- Topic 04: Numpy quick tutorial (15min)
- Topic 05: Pandas quick tutorial (15min)
- Topic 06: Scikit-learn quick tutorial (15min)
- Topic 07: OpenCV quick tutorial (15min)
- Topic 08: Image Processing basics (20min)
- Topic 09: Machine Learning basics (20min)
- Topic 10: Deep Learning basics (20min)
- Topic 11: TensorFlow overview (20min)
- Topic 12: CNN with TensorFlow (20min)
- Topic 13: RNN with TensorFlow (20min)
- Topic 14: PyTorch overview (20min)
- Topic 15: CNN with PyTorch (20min)
- Topic 16: RNN with Pytorch (20min)
- Topic 17: Introduction to AOI (20min)
- Topic 18: AOI simple Pipeline (A) (20min)
- Topic 19: AOI simple Pipeline (B) (20min)
- Topic 20: Introduction to Object detection (20min)
- Topic 21: YoloV5 Quick Tutorial (20min)
- Topic 22: Using YoloV5 for RSD (20min)
- Topic 23: Introduction to NLP (20min)
- Topic 24: Introduction to Word Embedding (20min)
- Topic 25: Name prediction project (20min)



Course Schedule

- W1 - Course Introduction
- W2 - DL Programming Basics(1)
- W3 - DL Programming Basics(2)
- W4 - DL with TensorFlow
- W5 - Midterm
- W6 - DL with PyTorch
- W7 - AOI hands-on project
- W8 - RSD hands-on project
- W9 - NLP hands-on project
- W10 - Final exam

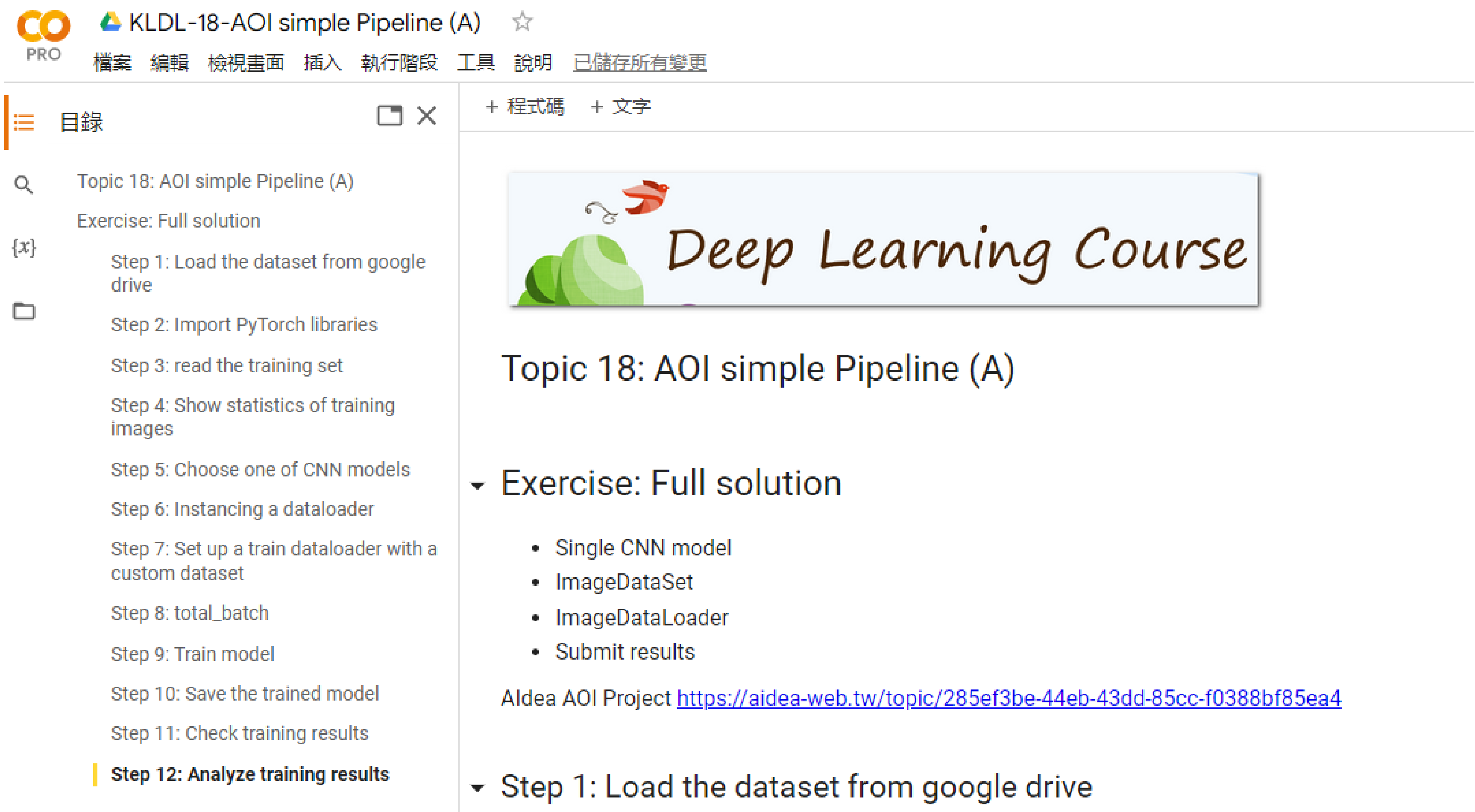
DL: Deep Learning
AOI: Automated Optical Inspection
RSD: Road Sign Detection
NLP: Natural Language Processing



Week 7 Topics

- Topic 17: Introduction to AOI (20min)
- Topic 18: AOI simple Pipeline (A) (20min)
- Topic 19: AOI simple Pipeline (B) (20min)

AOI pipeline (B)



KLDL-18-AOI simple Pipeline (A)

Topic 18: AOI simple Pipeline (A)

Exercise: Full solution

Step 1: Load the dataset from google drive

Step 2: Import PyTorch libraries

Step 3: read the training set

Step 4: Show statistics of training images

Step 5: Choose one of CNN models

Step 6: Instancing a dataloader

Step 7: Set up a train dataloader with a custom dataset

Step 8: total_batch

Step 9: Train model

Step 10: Save the trained model

Step 11: Check training results

Step 12: Analyze training results

Topic 18: AOI simple Pipeline (A)

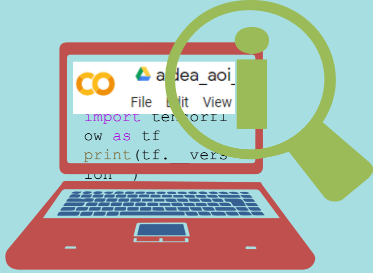
Exercise: Full solution

- Single CNN model
- ImageDataSet
- ImageDataLoader
- Submit results

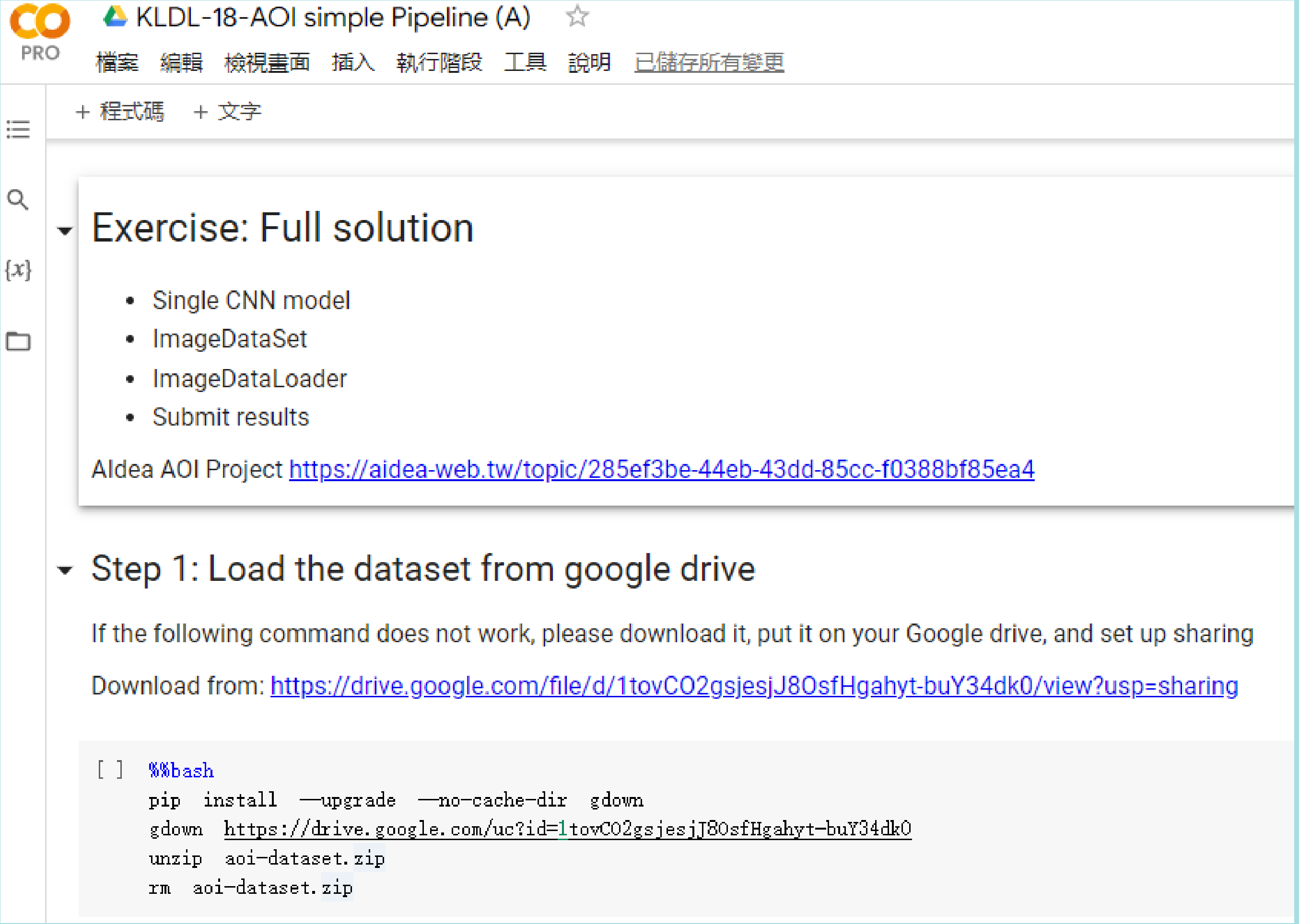
AIdea AOI Project https://aidea-web.tw/topic/285ef3be-44eb-43dd-85cc-f0388bf85ea4

Step 1: Load the dataset from google drive

Step 1: Load the dataset from google drive



AUAOI Ex2.



CO PRO
KLDL-18-AOI simple Pipeline (A)
檔案 編輯 檢視畫面 插入 執行階段 工具 說明 已儲存所有變更
+ 程式碼 + 文字

Exercise: Full solution

- Single CNN model
- ImageDataSet
- ImageDataLoader
- Submit results

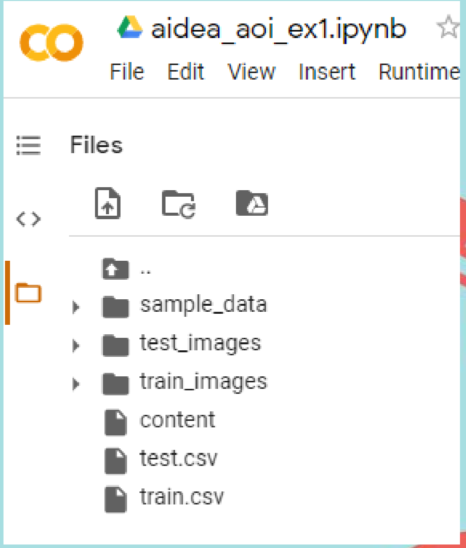
AIdea AOI Project https://aidea-web.tw/topic/285ef3be-44eb-43dd-85cc-f0388bf85ea4

Step 1: Load the dataset from google drive

If the following command does not work, please download it, put it on your Google drive, and set up sharing

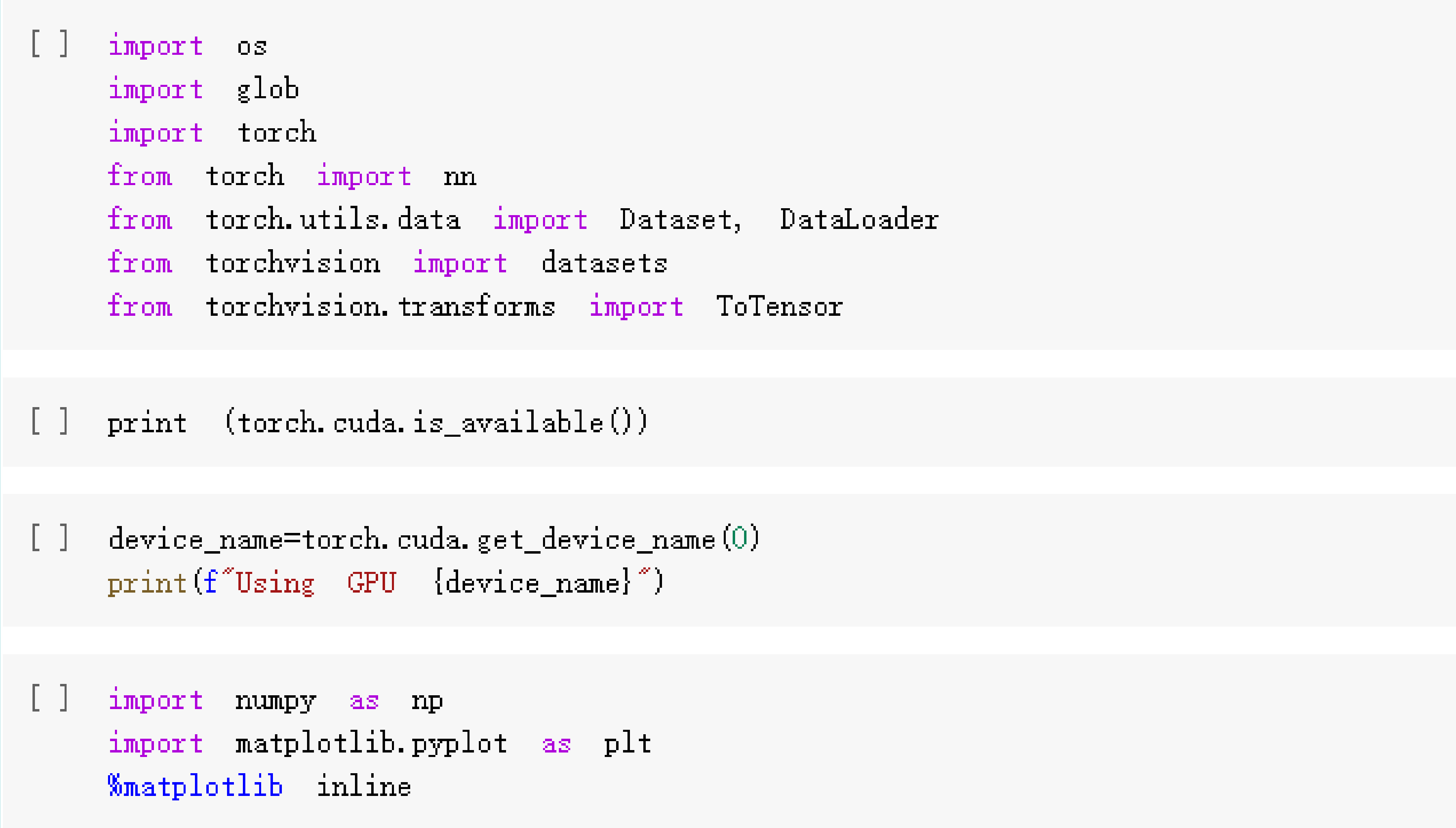
Download from: https://drive.google.com/file/d/1tovCO2gsjesjJ8OsfHgahyt-buY34dk0/view?usp=sharing

```
[ ]  %%bash
     pip install --upgrade --no-cache-dir gdown
     gdown https://drive.google.com/uc?id=1tovCO2gsjesjJ8OsfHgahyt-buY34dk0
     unzip aoi-dataset.zip
     rm aoi-dataset.zip
```



Step 2: Import python libraries

AUAOI Ex2.



```
[ ]  import os
     import glob
     import torch
     from torch import nn
     from torch.utils.data import Dataset, DataLoader
     from torchvision import datasets
     from torchvision.transforms import ToTensor
```

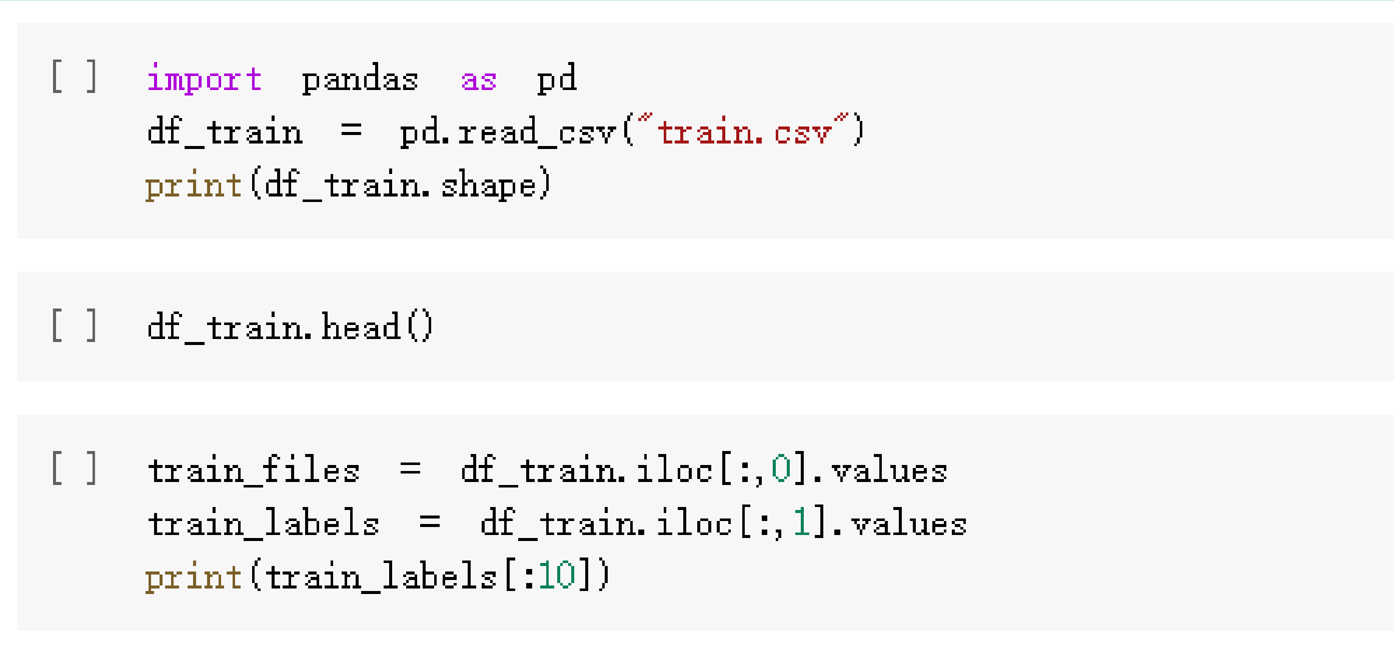
```
[ ]  print (torch.cuda.is_available())
```

```
[ ]  device_name=torch.cuda.get_device_name(0)
     print(f"Using GPU {device_name}")
```

```
[ ]  import numpy as np
     import matplotlib.pyplot as plt
     %matplotlib inline
```

Step 3: read the training set

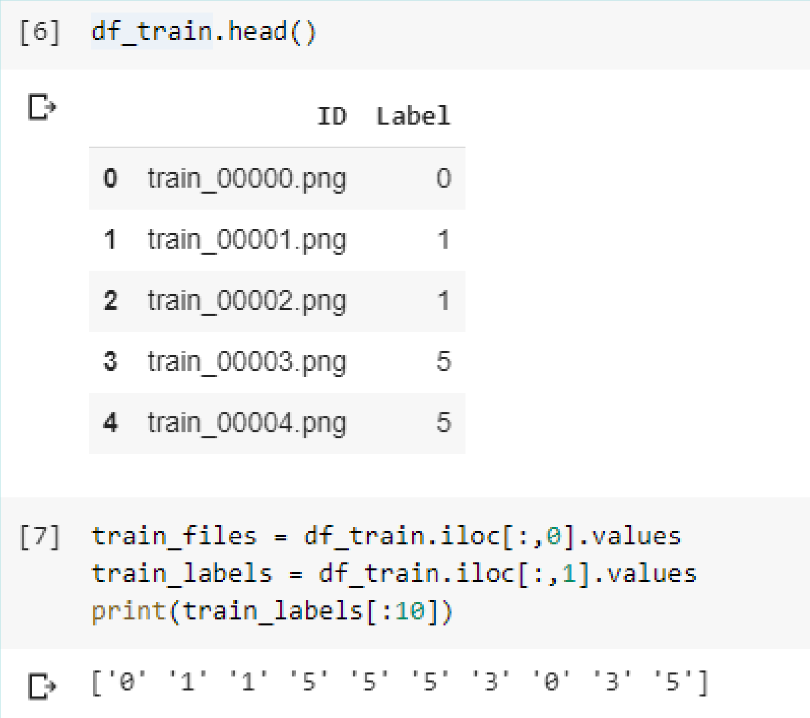
AUAOI Ex2.



```
[ ]  import pandas as pd
     df_train = pd.read_csv("train.csv")
     print(df_train.shape)
```

```
[ ]  df_train.head()
```

```
[ ]  train_files = df_train.iloc[:,0].values
     train_labels = df_train.iloc[:,1].values
     print(train_labels[:10])
```



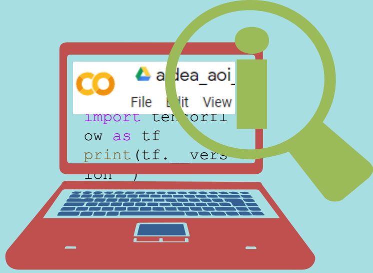
```
[6]  df_train.head()
```

	ID	Label
0	train_00000.png	0
1	train_00001.png	1
2	train_00002.png	1
3	train_00003.png	5
4	train_00004.png	5

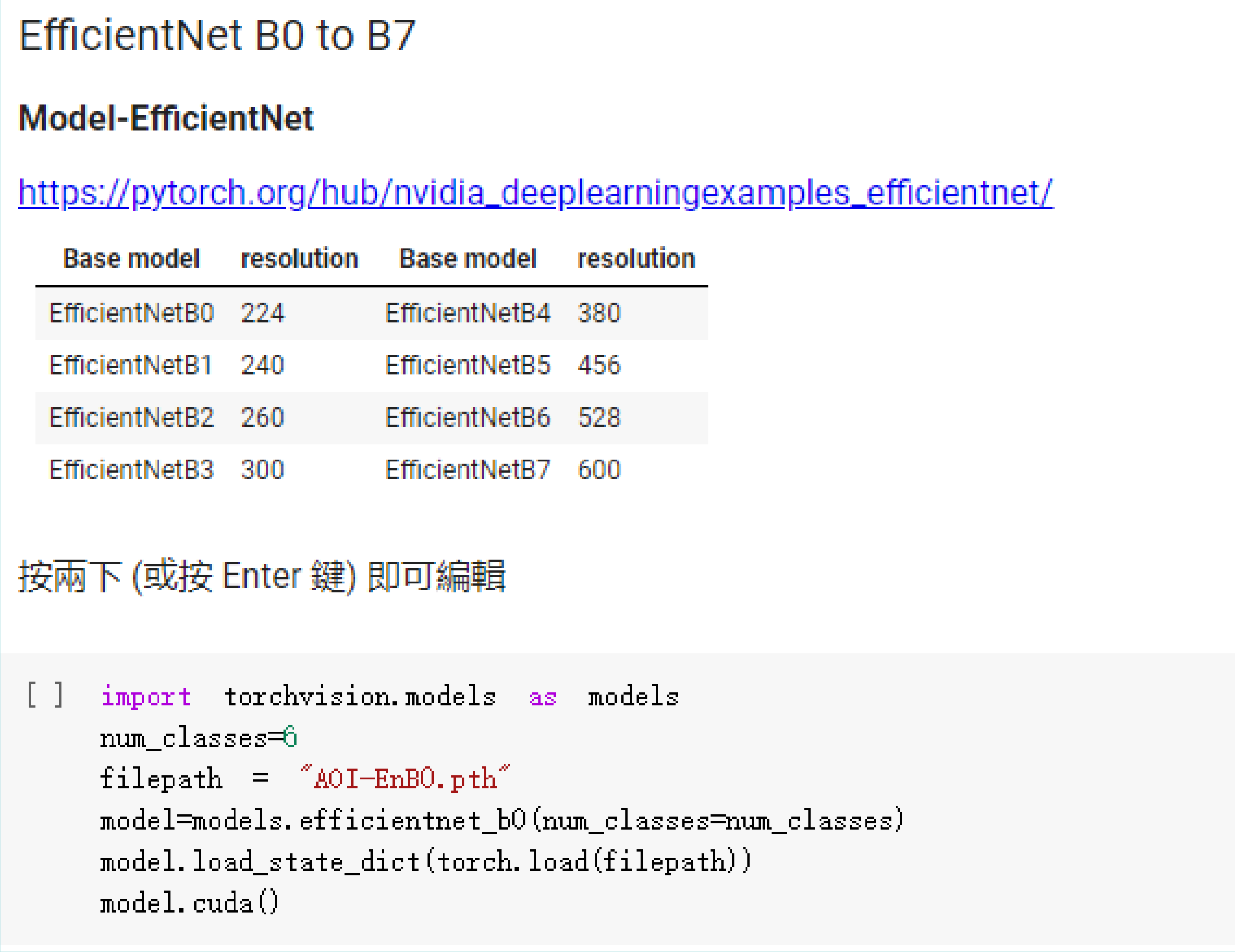
```
[7]  train_files = df_train.iloc[:,0].values
     train_labels = df_train.iloc[:,1].values
     print(train_labels[:10])

['0' '1' '1' '5' '5' '5' '3' '0' '3' '5']
```

Step 3: Choose one of CNN models



AUAOI Ex2.



EfficientNet B0 to B7

Model-EfficientNet

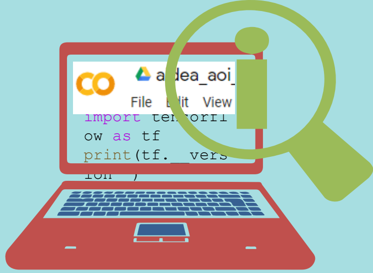
https://pytorch.org/hub/nvidia_deeplearningexamples_efficientnet/

Base model	resolution	Base model	resolution
EfficientNetB0	224	EfficientNetB4	380
EfficientNetB1	240	EfficientNetB5	456
EfficientNetB2	260	EfficientNetB6	528
EfficientNetB3	300	EfficientNetB7	600

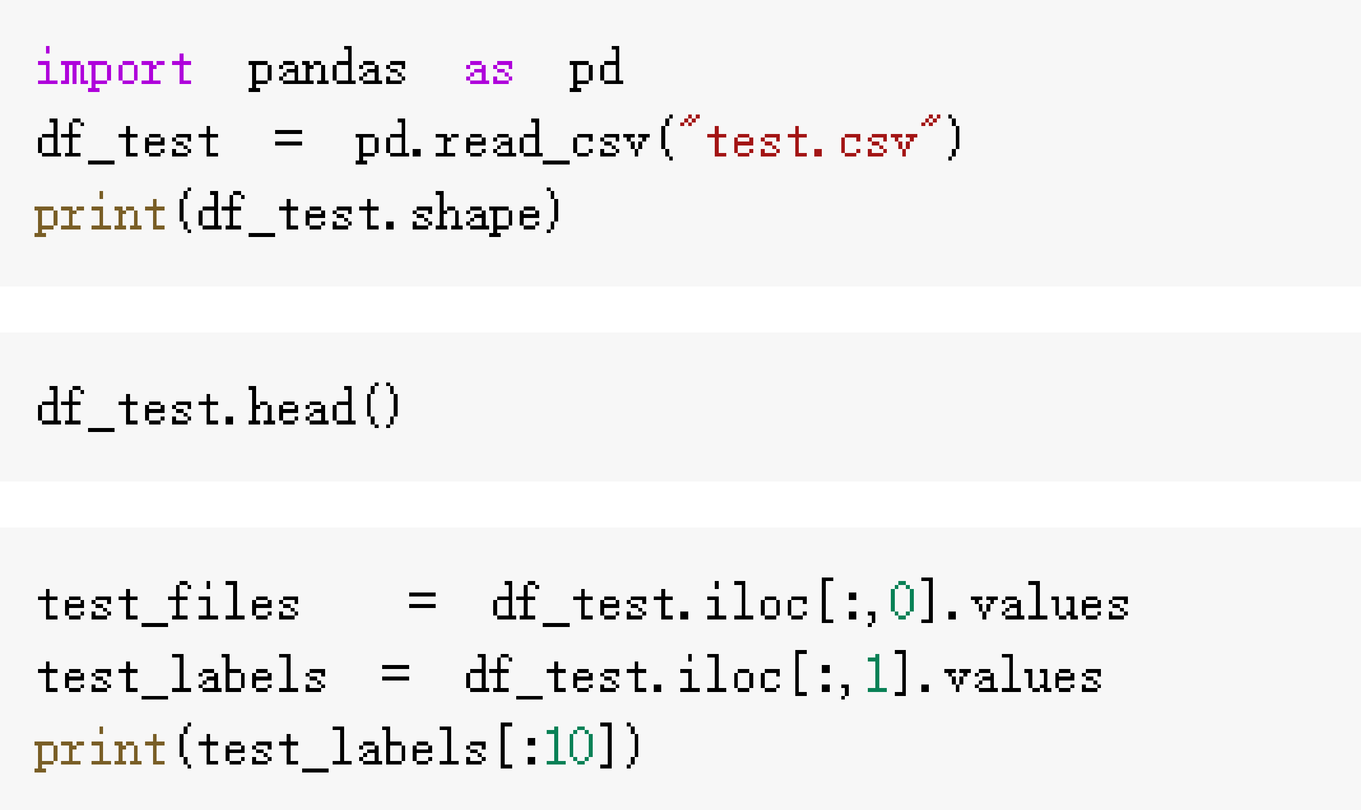
按兩下 (或按 Enter 鍵) 即可編輯

```
import torchvision.models as models
num_classes=6
filepath = "AOI-EnB0.pth"
model=models.efficientnet_b0(num_classes=num_classes)
model.load_state_dict(torch.load(filepath))
model.cuda()
```

Step 4: Load the test set



AUAOI Ex2.



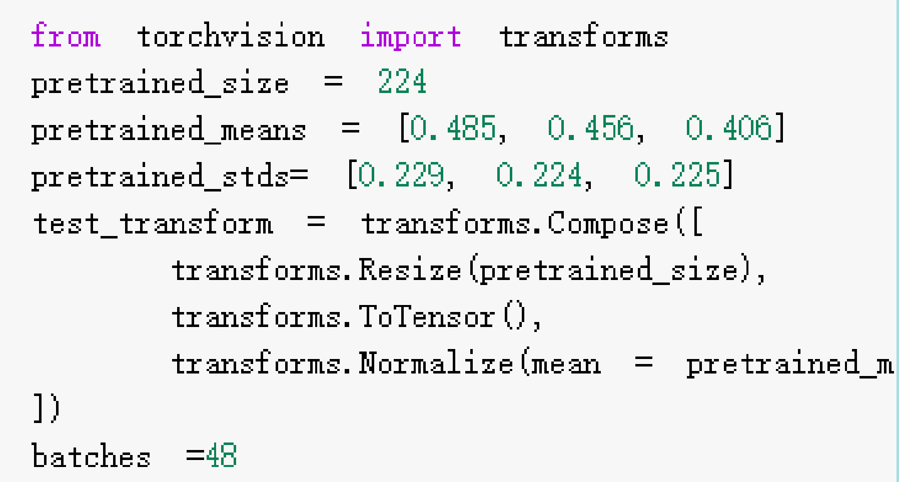
```
import pandas as pd
df_test = pd.read_csv(“test.csv”)
print(df_test.shape)
```

```
df_test.head()
```

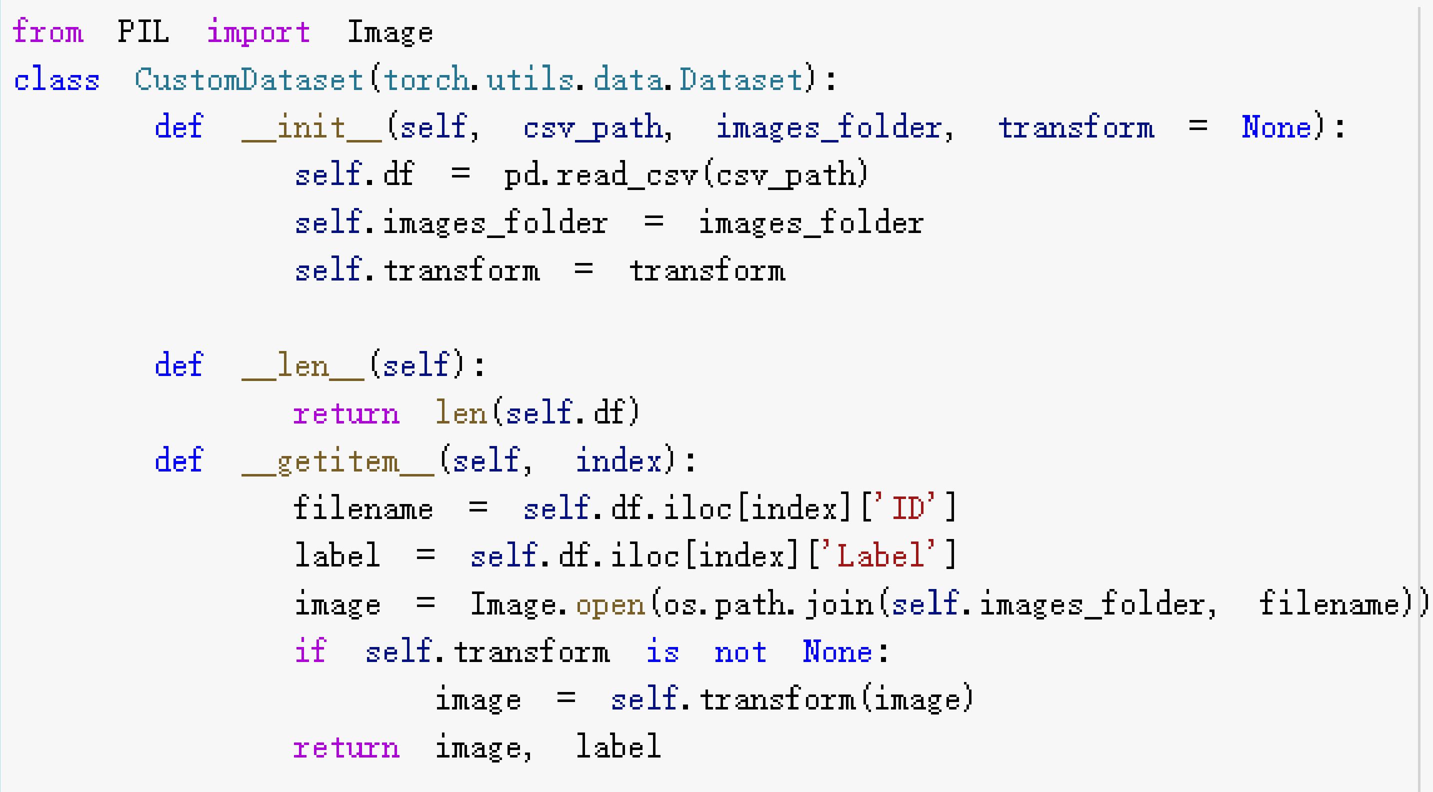
```
test_files  = df_test.iloc[:,0].values
test_labels = df_test.iloc[:,1].values
print(test_labels[:10])
```

Step 5: Set up a test_dataloader with test_dataset

AUAOI Ex2.



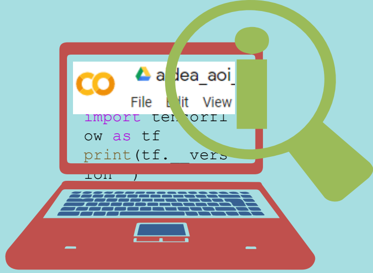
```
from torchvision import transforms
pretrained_size = 224
pretrained_means = [0.485, 0.456, 0.406]
pretrained_stds= [0.229, 0.224, 0.225]
test_transform = transforms.Compose([
        transforms.Resize(pretrained_size),
        transforms.ToTensor(),
        transforms.Normalize(mean = pretrained_m
])
batches =48
```



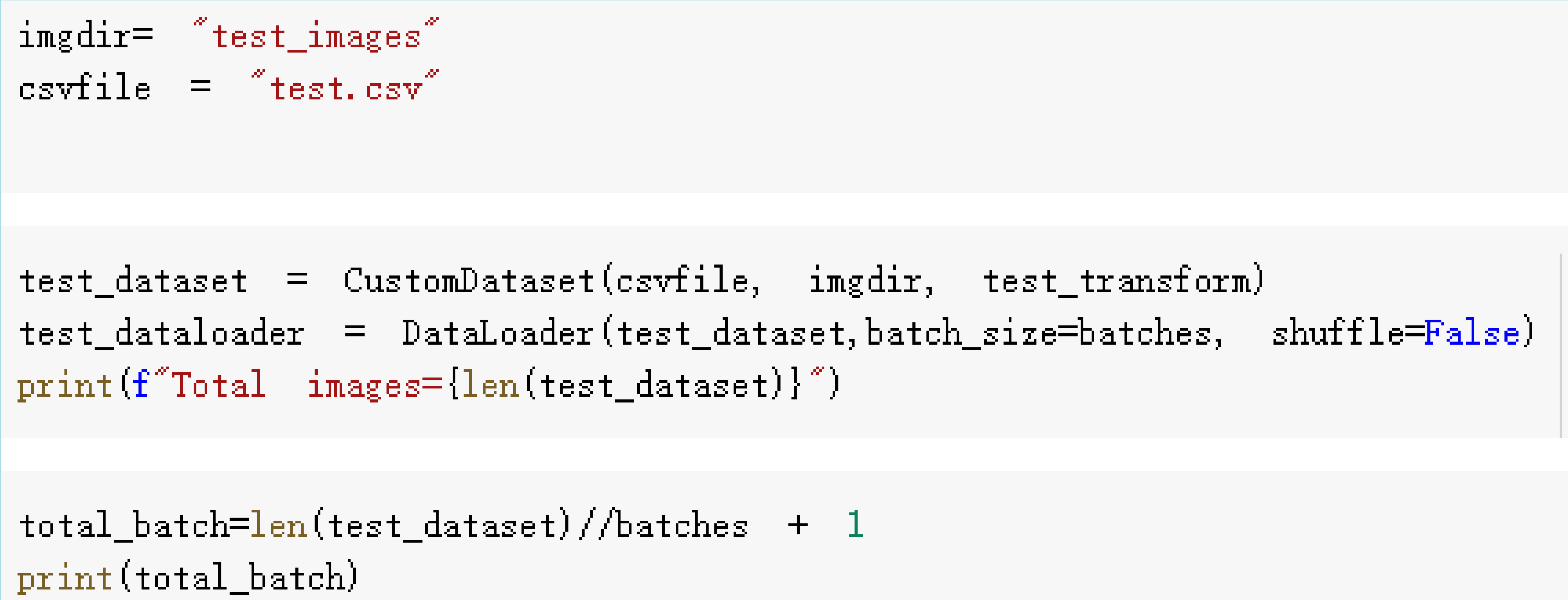
```
from PIL import Image
class CustomDataset(torch.utils.data.Dataset):
    def __init__(self, csv_path, images_folder, transform = None):
        self.df = pd.read_csv(csv_path)
        self.images_folder = images_folder
        self.transform = transform

    def __len__(self):
        return len(self.df)
    def __getitem__(self, index):
        filename = self.df.iloc[index]['ID']
        label = self.df.iloc[index]['Label']
        image = Image.open(os.path.join(self.images_folder, filename))
        if self.transform is not None:
            image = self.transform(image)
        return image, label
```

Step 5: Initialize the test_dataloader



AUAOI Ex2.

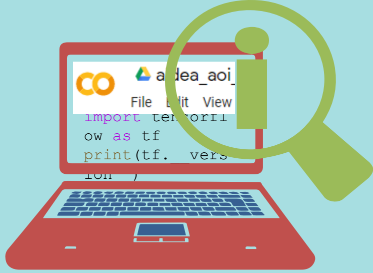


```
imgdir=  ″test_images″
csvfile  =  ″test.csv″
```

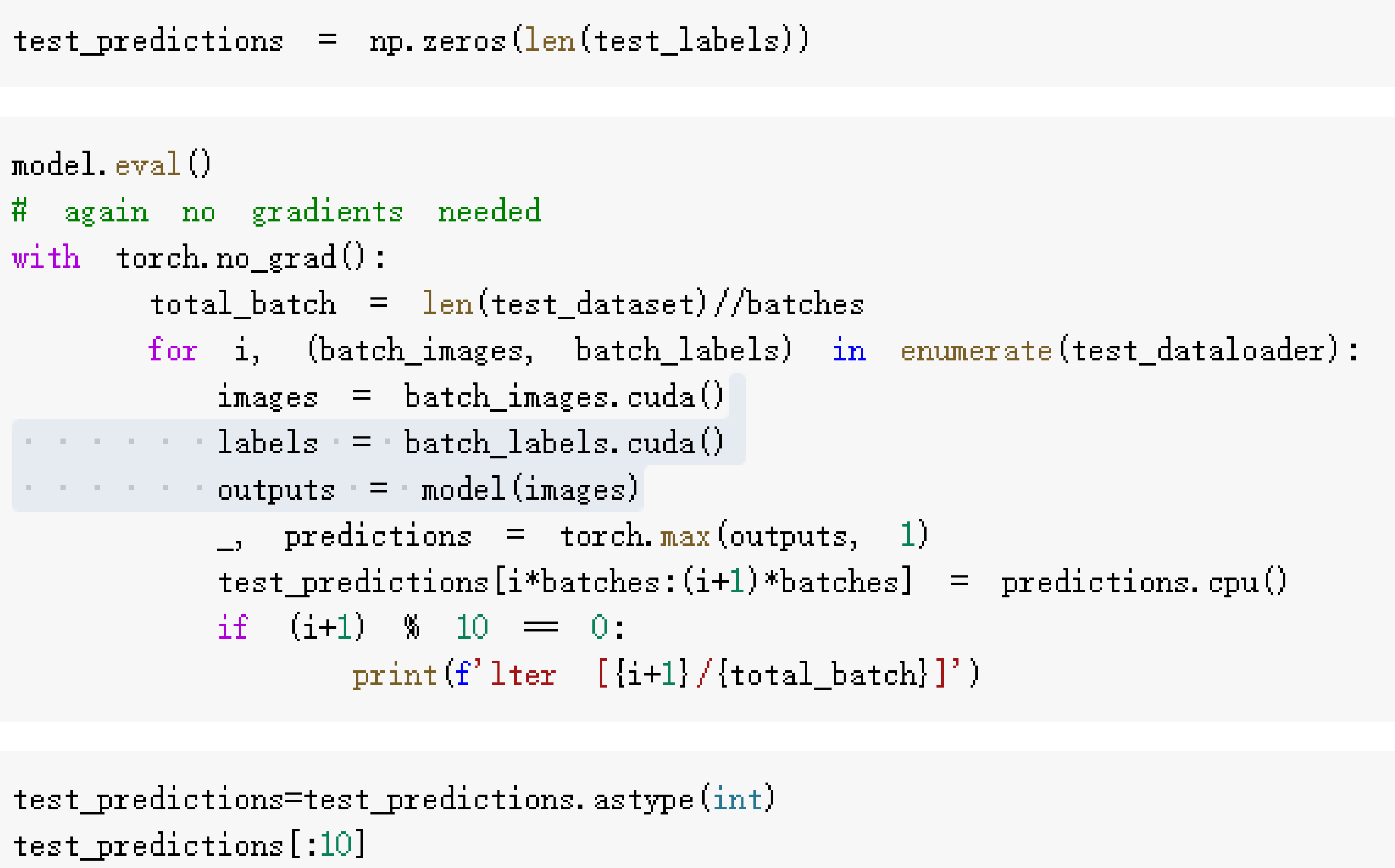
```
test_dataset  =  CustomDataset(csvfile,  imgdir,  test_transform)
test_dataloader  =  DataLoader(test_dataset,batch_size=batches,  shuffle=False)
print(f″Total  images={len(test_dataset)}″)
```

```
total_batch=len(test_dataset)//batches  +  1
print(total_batch)
```

Step 6: Check test results



AUAOI Ex2.

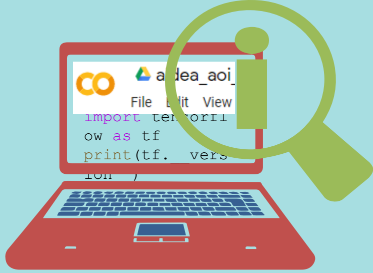


```
test_predictions = np.zeros(len(test_labels))
```

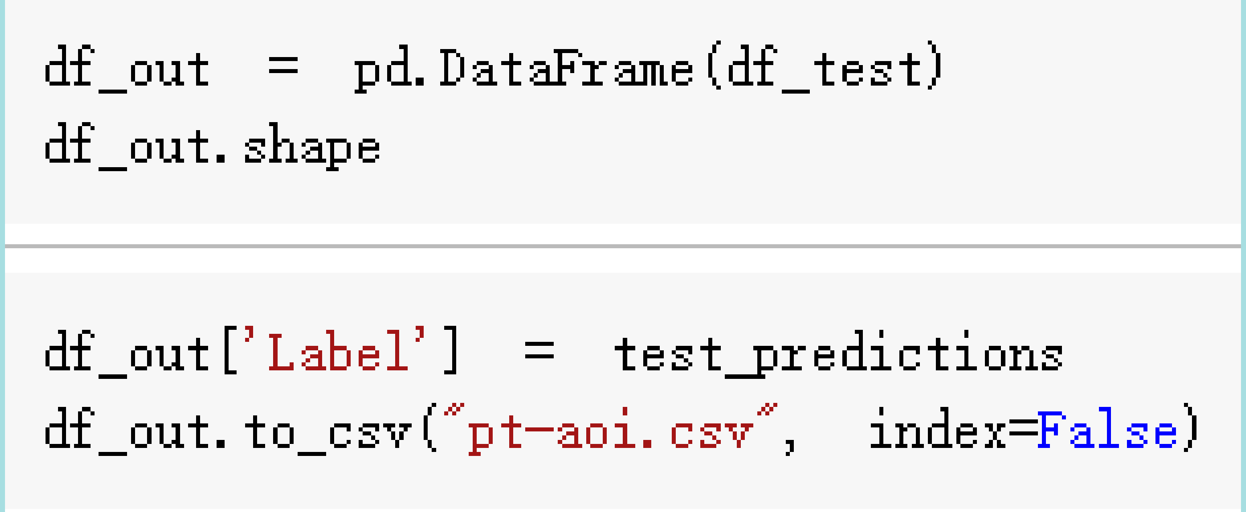
```
model.eval()
# again no gradients needed
with torch.no_grad():
    total_batch = len(test_dataset)//batches
    for i, (batch_images, batch_labels) in enumerate(test_dataloader):
        images = batch_images.cuda()
        labels = batch_labels.cuda()
        outputs = model(images)
        _, predictions = torch.max(outputs, 1)
        test_predictions[i*batches:(i+1)*batches] = predictions.cpu()
        if (i+1) % 10 == 0:
            print(f’Iter [{i+1}/{total_batch}]’)
```

```
test_predictions=test_predictions.astype(int)
test_predictions[:10]
```

Step 7: Output test results



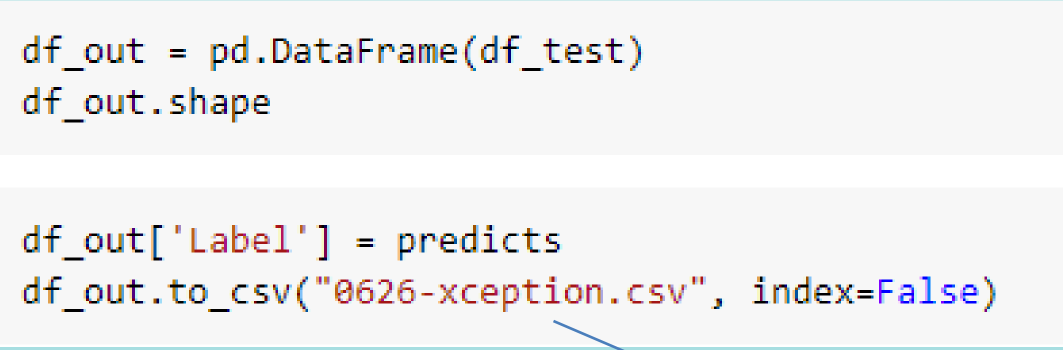
AUAOI Ex2.



```
df_out = pd.DataFrame(df_test)
df_out.shape
```

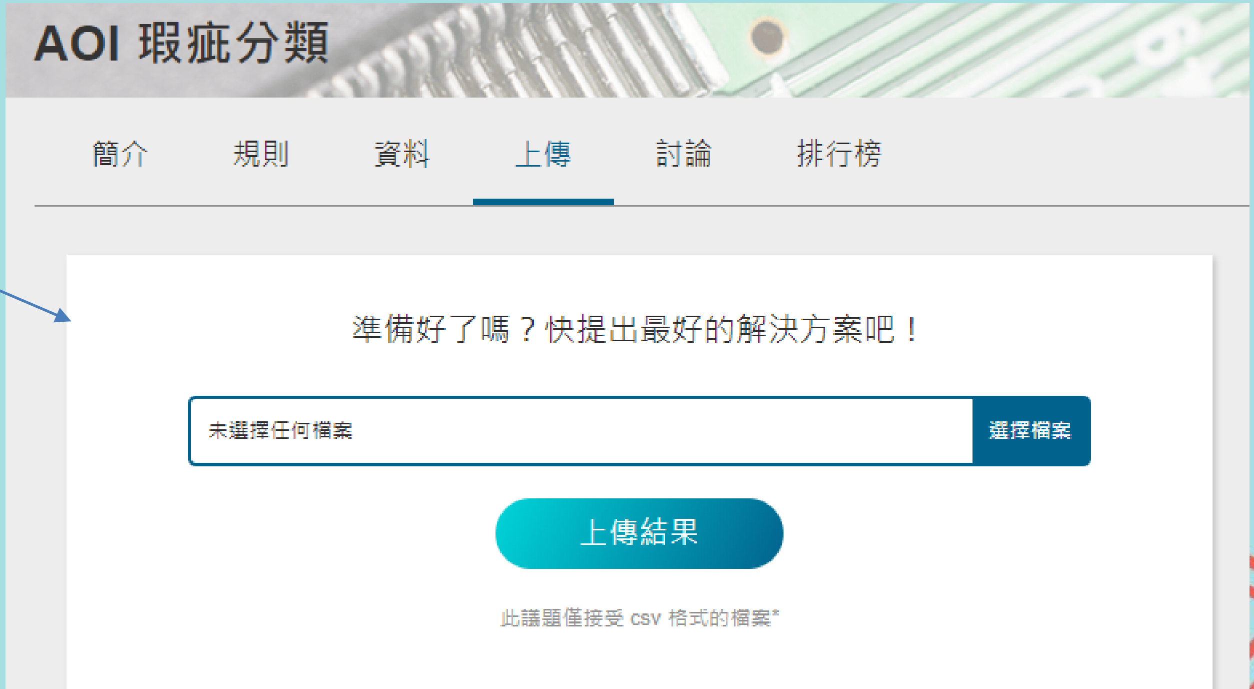
```
df_out[’Label’] = test_predictions
df_out.to_csv(“pt-aoi.csv”, index=False)
```

Step 8: Submit the result to AIdea



```
df_out = pd.DataFrame(df_test)
df_out.shape
```

```
df_out['Label'] = predicts
df_out.to_csv("0626-xception.csv", index=False)
```



AOI 瑕疵分類

準備好了嗎？快提出最好的解決方案吧！

未選擇任何檔案

此議題僅接受 csv 格式的檔案*

Thanks! Q&A