Spring 2021
Introduction to Artificial Intelligence

Homework 1: Face Detection

Due Date: 2021/3/23 23:55

Introduction

The goal of this programming assignment is to 1) learn how to prepare a dataset for machine learning, 2) implement Viola-Jones face detection, especially **Adaboost algorithm for feature selection**, and 3) get a taste of a complete supervised learning process. Please make sure you understand the concept of Viola-Jones algorithm and examples discussed in class before working on HW1.

The codebase includes most parts of the Viola-Jones face detection algorithm. The details of each file are listed at the end of this document. Please finish codes between # Begin your code and # End your code.

If you run python code locally(Anaconda), please use main.py to test your implementation. For Google Colab users, please use main.ipynb and see Appendix A for more details.

Data

The dataset can be found from the “data” folder. There are three subfolders. The “TRAIN” folder is for training. The “TEST” folder is for testing. The “DETECT” folder includes nature images. The details are as follow:

- TRAIN and TEST:
    - The two subfolders contain face images and non-face images, respectively.
    - The size of an image is 19 x 19, and the format is grayscale PGM.
- DETECT:
    - detectData.txt: The location and size of each face in a natural image. The format is:

```
image1_name face_num
x y width height
…
image2_name face_num
x y width height
```

Requirements

Part 1: Load and prepare your dataset (10%)

- Implement the “loadImages” function in dataset.py that loads all images in the folder. Please convert images into a list of tuples. The first element is a numpy array that stores the image. The shape of the numpy array is (height, width). The second element is its **Label** (1 or 0).
- Please feel free to import packages implemented in dataset.py.
- Please run main.py or main.ipynb to test your implementation. If your implementation is correct, one face image and one non-face image will be displayed.
- Please read the comments of the “loadImages” function in dataset.py line 4.

Part 2: Implement Adaboost algorithm (30%)

- A strong classifier is a linear combination of weak classifiers.
- Please implement the “selectBest” function in adaboost.py to select the best weak classifier.
- Please run main.py or main.ipynb to test your implementation. If your implementation is correct, you will see your classifier is being trained and evaluated.
- Please read the comments in the “selectBest” function, which can be found in adaboost.py line 133 and the “WeakClassifier” class in classifier.py.

Part 3: Additional experiments (20%)

- Please change the parameter T in the Adaboost algorithm. Please compare the corresponding detection performance.
- Please test the parameter T between 1 to 10.
- Please run main.py or main.ipynb to test your implementation. If your implementation is correct, you will see your classifier is being trained and evaluated.
- Please read the comments in the “Adaboost” class, which can be found in adaboost.py line 10.

Part 4: Detect face (15%)

- Please implement the “detect” function in detection.py to load the images in the “DETECT” folder. The function will detect faces we assigned.
- Please read detectData.txt to understand the format. Load the image and get the face images. Resize and convert the face images to 19 x 19 grayscale images. Then, please use clf.classify() function to detect faces.
- Display face detection results. If the classification is “Face,” draw a green box on an image. Otherwise, draw a red box on an image.
- Please feel free to import packages that you need in detection.py.

- Run main.py or main.ipynb to test your implementation.
- Please read the comments in the “classify” function, which can be found in adaboost.py line 155.

Part 5: Test classifier on your own images (5%)

- Please choose your own images with faces. Please create a text file with the same format as in detectData.txt.
- Please run main.py or main.ipynb to check if your classifier can classify face and non-face!

Part 6: Implement another classifier (Bonus) (10%)

- You can implement the “selectBest” function in adaboost.py with another method. Please compare detection results with results obtained from Part 2 and Part 3.

Report (20%)

- You are required to submit a report.
- The report should be written in **English**.
- Save the report as a **.pdf** file
    - font size: 12
- Put the **screenshot of your code** and briefly explain it.
- Explain your implementation of the above requirements **in detail**.
- Include **screenshots of the results** in the report.
- Describe problems you meet and how you solve them.

Submission

Please prepare your source code and report (.pdf) into STUDENTID_hw1.zip.

e.g. 309123456_hw1.zip

Late Submission Policy

20% off per late day

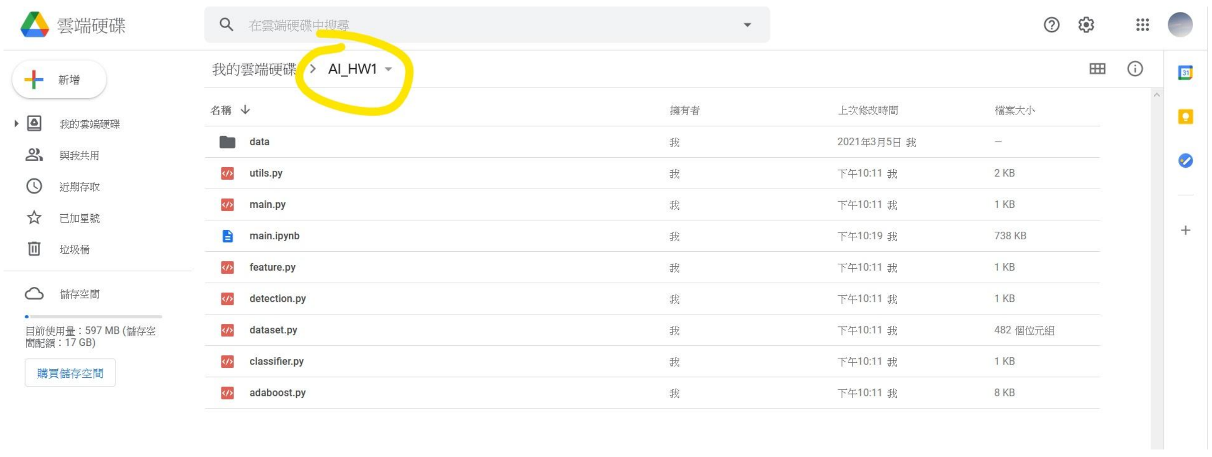
Files

File name	Description
main.py/main.ipynb	Test all of code.
dataset.py	Load images.
feature.py	Define data structures about features for implementing the Adaboost algorithm.
classifier.py	Define data structures about weak classifiers for implementing the Adaboost algorithm.
adaboost.py	The main file that implements the Adaboost algorithm.
detection.py	Detection images.
utils.py	Contain functions to compute the integral image and evaluate classifiers.

Appendix A: For using Google Colab

Since Google Colab has usage limits for users i.e. the files and codes you've uploaded and modified will be cleaned up after timeout, we recommend uploading project files to your Google Drive and use it as a storage of your work. Here are instructions for setting up Colab environment:

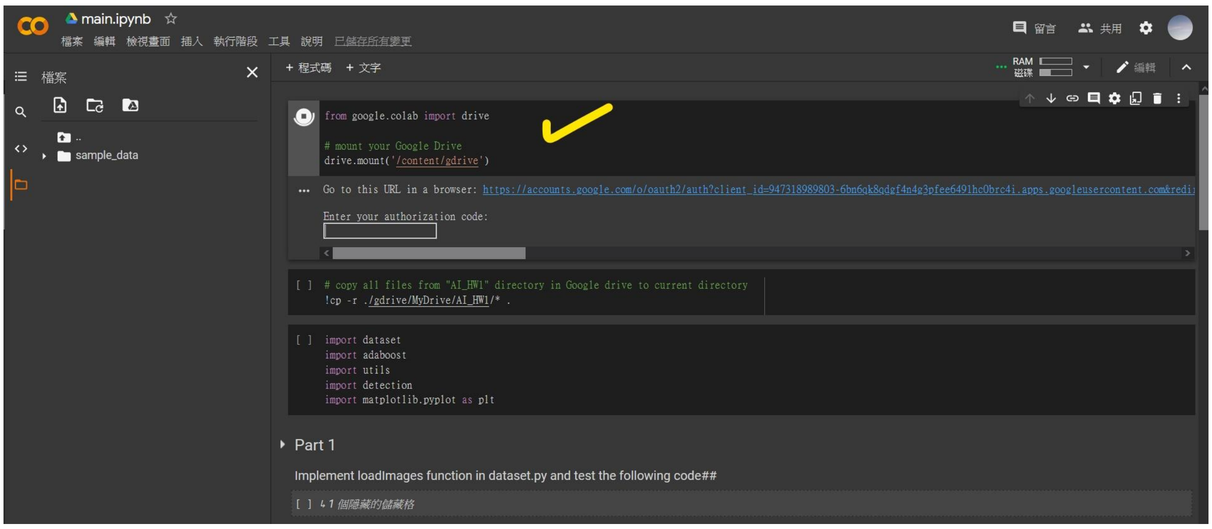
1. Upload the project file to your Google Drive, and name the folder: "AI_HW1"



我的雲端硬碟 > AI_HW1

名稱	擁有者	上次修改時間	檔案大小
data	我	2021年3月5日 我	–
utils.py	我	下午10:11 我	2 KB
main.py	我	下午10:11 我	1 KB
main.ipynb	我	下午10:19 我	738 KB
feature.py	我	下午10:11 我	1 KB
detection.py	我	下午10:11 我	1 KB
dataset.py	我	下午10:11 我	482 個位元組
classifier.py	我	下午10:11 我	1 KB
adaboost.py	我	下午10:11 我	8 KB

2. Open the main.ipynb.
   Execute the first cell to mount your Google Drive. Copy authorization code through the link and pasted it, then press ‘Enter’.



```
from google.colab import drive

# mount your Google Drive
drive.mount('/content/gdrive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redi

Enter your authorization code:
```

```
[ ] # copy all files from "AI_HW1" directory in Google drive to current directory
    !cp -r ./gdrive/MyDrive/AI_HW1/* .
```

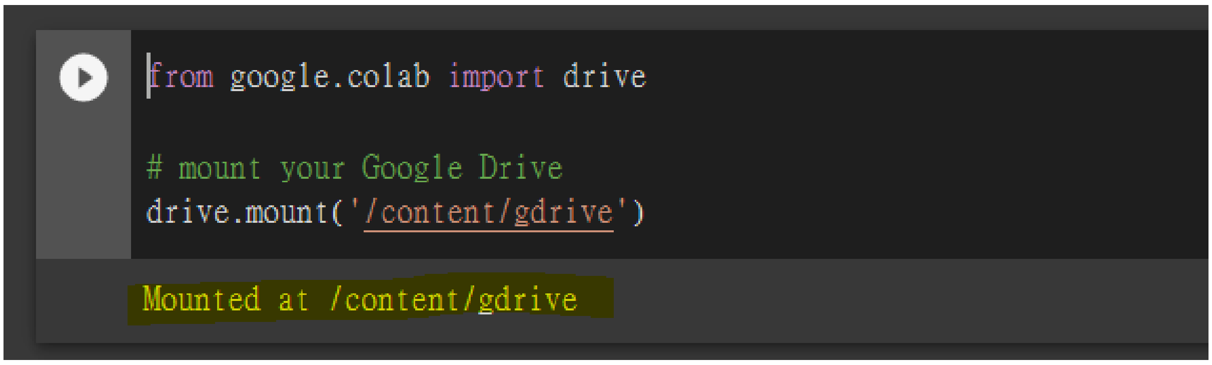
```
[ ] import dataset
    import adaboost
    import utils
    import detection
    import matplotlib.pyplot as plt
```

Part 1

Implement loadImages function in dataset.py and test the following code##

[ ] ↳ 1 個隱藏的儲存格

You will see the following messages if Google drive is mounted successfully.

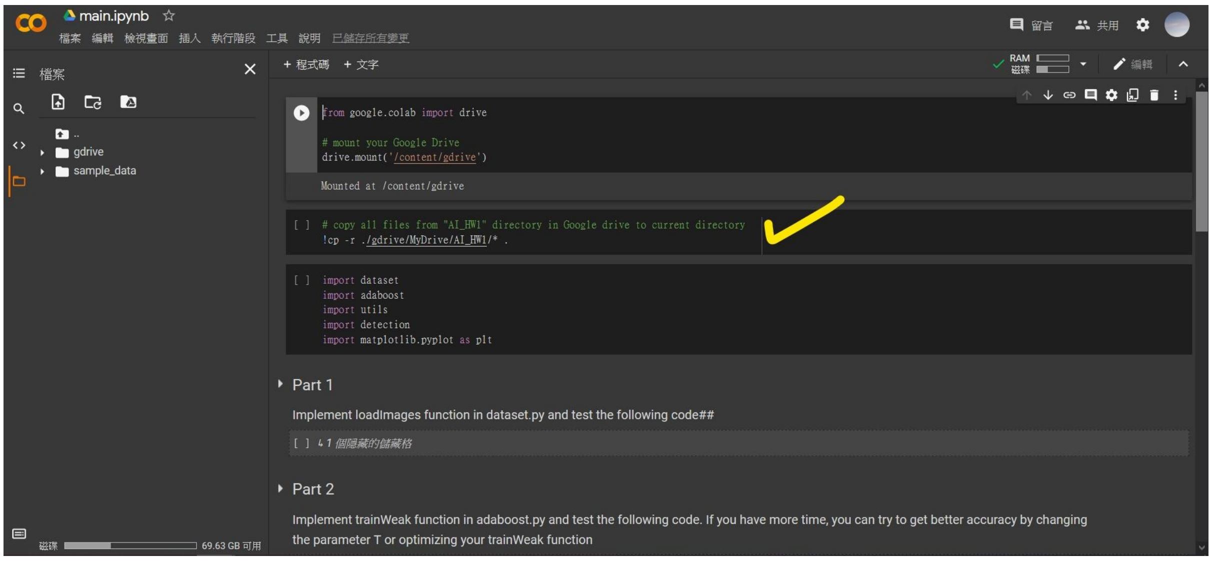


```
from google.colab import drive

# mount your Google Drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

3. Execute the next cell to copy project files to the current working directory.



```
from google.colab import drive

# mount your Google Drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

```
[ ] # copy all files from "AI_HW1" directory in Google drive to current directory
    !cp -r ./gdrive/MyDrive/AI_HW1/* .
```

```
[ ] import dataset
    import adaboost
    import utils
    import detection
    import matplotlib.pyplot as plt
```

Part 1

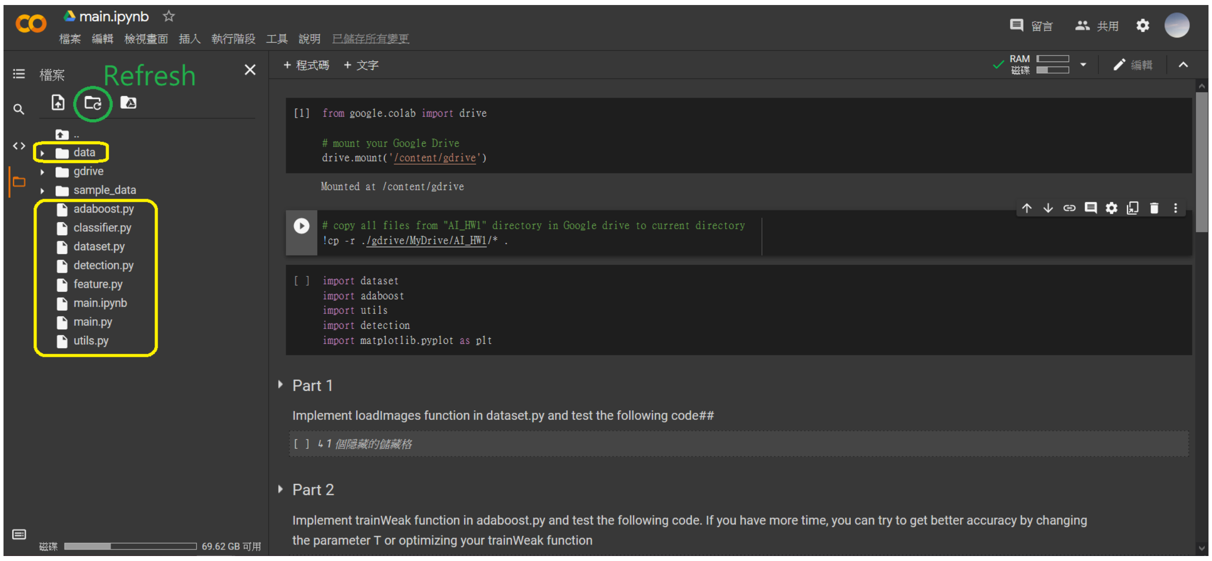
Implement loadImages function in dataset.py and test the following code##

[ ] ↳ 1 個隱藏的儲存格

Part 2

Implement trainWeak function in adaboost.py and test the following code. If you have more time, you can try to get better accuracy by changing the parameter T or optimizing your trainWeak function

After copying, you will see project files on the left. (Press Refresh button to update file structure)



```
[1] from google.colab import drive

    # mount your Google Drive
    drive.mount('/content/gdrive')

    Mounted at /content/gdrive
```

```
# copy all files from "AI_HW1" directory in Google drive to current directory
!cp -r ./gdrive/MyDrive/AI_HW1/* .
```

```
[ ] import dataset
    import adaboost
    import utils
    import detection
    import matplotlib.pyplot as plt
```

Part 1

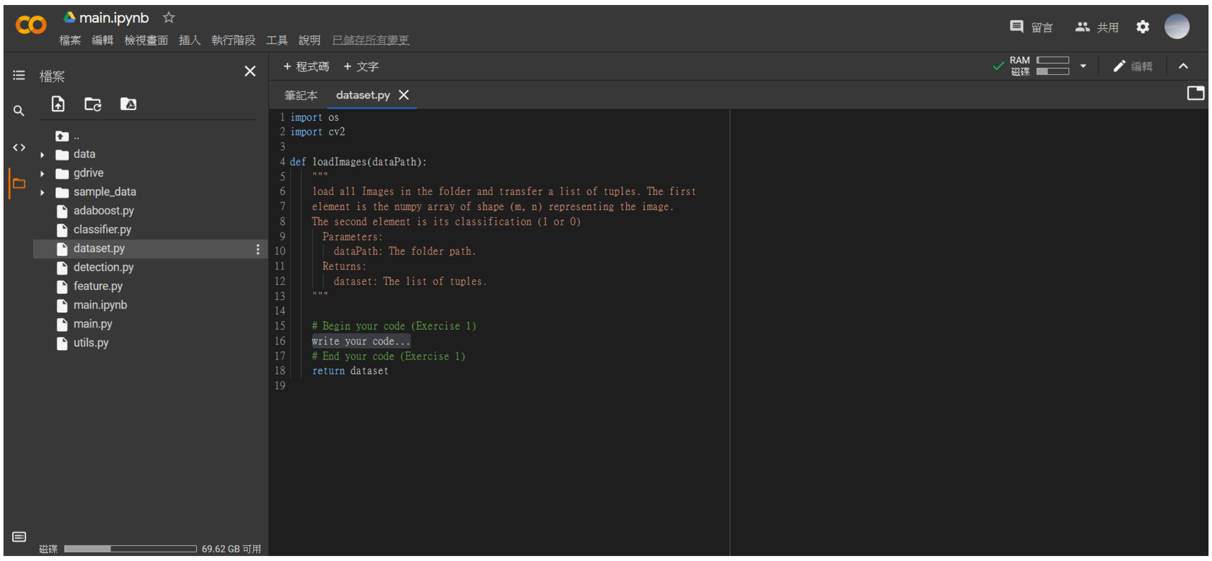
Implement loadImages function in dataset.py and test the following code##

[ ] ↳ 1 個隱藏的儲藏格

Part 2

Implement trainWeak function in adaboost.py and test the following code. If you have more time, you can try to get better accuracy by changing the parameter T or optimizing your trainWeak function

4. Write your code. (Press ‘ctrl+s’ in case auto-saving is not working)



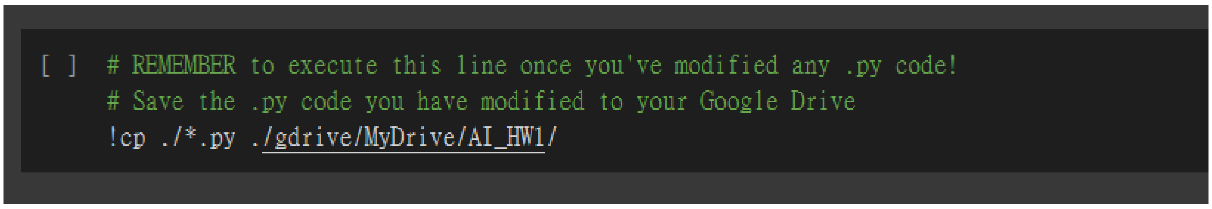
```
import os
import cv2

def loadImages(dataPath):
    """
    load all Images in the folder and transfer a list of tuples. The first
    element is the numpy array of shape (m, n) representing the image.
    The second element is its classification (1 or 0)
      Parameters:
        dataPath: The folder path.
      Returns:
        dataset: The list of tuples.
    """

    # Begin your code (Exercise 1)
    write your code...
    # End your code (Exercise 1)
    return dataset

```

5. ***IMPORTANT***
Please REMEMBER to execute the last cell in order to copy .py files back to your Google Drive. If you forget to save your work back to your Google Drive, your code will be gone after Colab timeout! So make sure you execute this cell whenever you’ve done your current work.



```
[ ] # REMEMBER to execute this line once you've modified any .py code!
    # Save the .py code you have modified to your Google Drive
    !cp ./*.py ./gdrive/MyDrive/AI_HW1/
```