Spring 2022
Introduction to Artificial Intelligence

Homework 1

Mar. 8, 2022

Introduction

The goal of this programming assignment is to 1) learn how to prepare a dataset for machine learning, 2) implement Viola-Jones object detection, especially **Adaboost algorithm for feature selection**, 3) get a taste of a complete supervised learning process, and 4) compare the result and performance with SOTA method. Please make sure you understand the concept of Viola-Jones algorithm before working on HW1.

The codebase includes most parts of the Viola-Jones object 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 (e.g., 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 images in the dataset were taken from the parking area beside the gymnasium in NYCU. You can find it in the "data" folder. There are three subfolders. The "TRAIN" folder is for training. The "TEST" folder is for testing. The "DETECT" folder includes a global view video (video.gif) and the parking space xy coordinates (detectData.txt). The details are as follows:

- TRAIN and TEST:
  - The two subfolders contain car images and non-car (just parking space) images, respectively.
  - The size of an image is 360 x 160 , and the format is PNG.
- DETECT:
  - detectData.txt: The location of each parking space in a natural image. There are a total of 76 parking spaces in the image. The format is:

```
76
x1 y1 x2 y2 x3 y3 x4 y4
…
```

E.g., 

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 of each tuple 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) indicating the space is full or not.
- 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 car image and one non-car (just parking space) image will be displayed.
- Please read the comments of the "loadImages" function in dataset.py line 4.

Part 2: Implement Adaboost algorithm (30%)

- 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 134 and the "WeakClassifier" class in classifier.py.

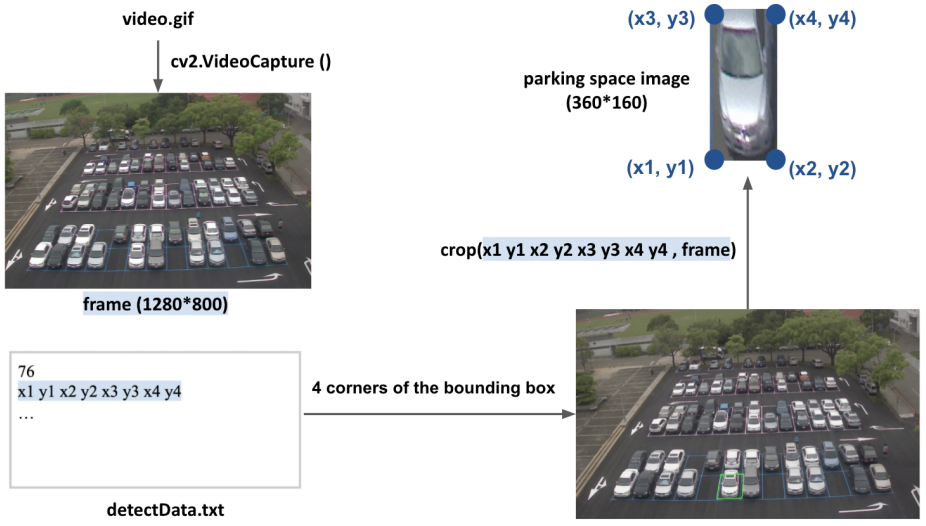
Part 3: Additional experiments (15%)

- 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.

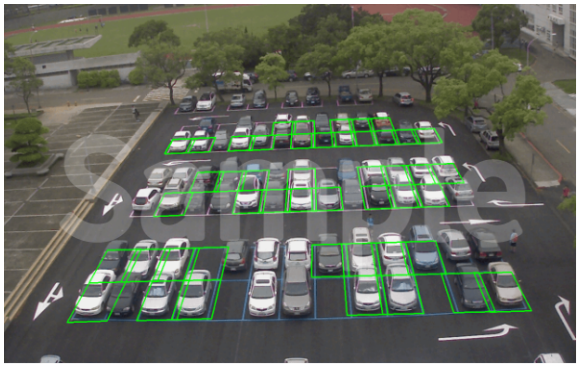
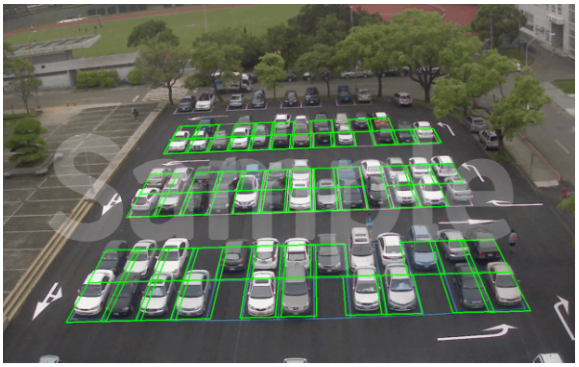
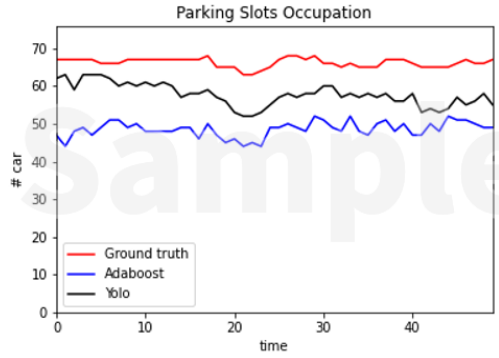
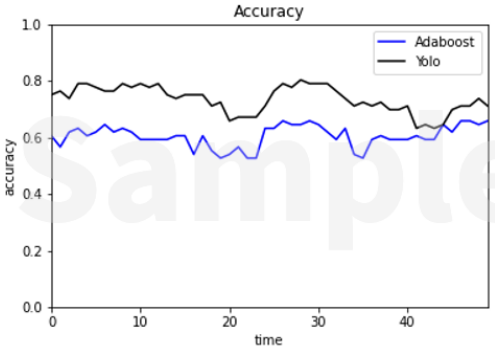
Part 4: Detect car (15%)

- Please implement the "detect" function in detection.py to load the images in the "DETECT" folder. The function will detect cars we are assigned.
- Please read detectData.txt to understand the format. Load the video.gif and get the parking space images. Resize and convert the parking space images to 36 x 16 grayscale images. Then, please use the clf.classify() function to detect cars.
- Display car detection results. If the classification is "Car," draw a green 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 156.
- The following figure is a pipeline of detecting cars. crop() function will help you extract the corresponding parking space image from each frame of video.gif according to detectData.txt.



Part 5: Discuss the difference between Adaboost and Yolov5 and draw a scatter plot/line graph to show the temporal parking slots occupation (10%)

- Execute yolov5_sample_code.ipynb (see Appendix B for more details)
- Draw a plot to show the temporal parking slots occupation and accuracy
- Write some discussions in the report
    - E.g., Image with bounding box, statistic plot

Adaboost	SOTA (E.g. Yolov5)
	
Parking slots occupation	
	

Part 6: Try other SOTA methods (Bonus) (10%)

- Implement other SOTA methods to detect the vehicle.
- Compare to Adaboost and do some discussions in your report.
- In this part, feel free to use any package if needed.

Report (20%)

- You are required to submit a report and it can be written in Chinese or English.
- Save the report as a **.pdf** file
- The report should include:
  1. Screenshot of your code and brief explanation
  2. Your implementation of the above requirements in detail
  3. Discuss what you observed with accuracy, F1-score and parking slots occupation plot of different methods in the report.
  4. Describe problems you meet and how you solve them

Discussion

TAs had opened a channel HW1 討論區 on Microsoft Teams of the course, you can ask questions about the homework in the channel. TAs will answer questions in the channel as soon as possible.

Discussion rules:

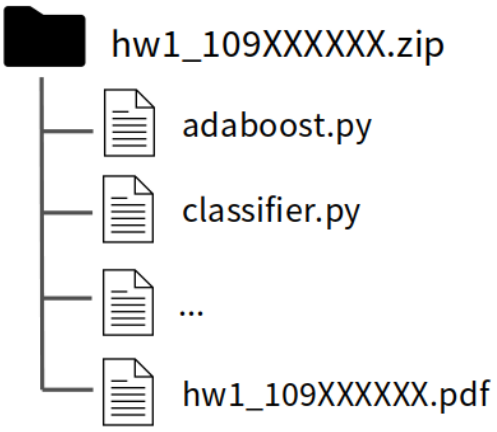
1. Do not ask for the answer to the homework.
2. Check if someone has asked the question you have before asking.
3. We encourage you to answer other students' questions, but again, do not give the answer to the homework. Reply to the messages to answer questions.
4. Since we have this discussion channel, do not send emails to ask questions about the homework unless the questions are personal and you do not want to ask publicly.

Submission

1. **The deadline for this homework is 3/21 (Mon.) 23:55:00.**
2. Please submit one zip file that contains all the Python code (i.e., **.py**) files, and report (**.pdf**). Note that you should not include the dataset in your zip file.
3. Submit the zip file with the filename of HW#number_StudentID.zip (e.g., HW1_109123456.zip). Please refer to the File Organization Section to name each submission file.
4. Late submission leads to a score of (original score)*0.85^{days}, for example, if you submit your homework right after the deadline, you will get (original score)*0.85 points.
5. **We only accept one zip file**, wrong format or naming format cause -10 points to your score (after considering late submission penalty).
6. If there is anything you are not sure about submission, ask in the discussion forum.

File Organization

1. Files needed to be submitted: adaboost.py, classifier.py, dataset.py, detection.py, feature.py, utils.py, main.py, hw1_109XXXXXX.pdf
2. Please follow the file hierarchy and the naming rules specified below. The wrong format will cause -10 points as well.



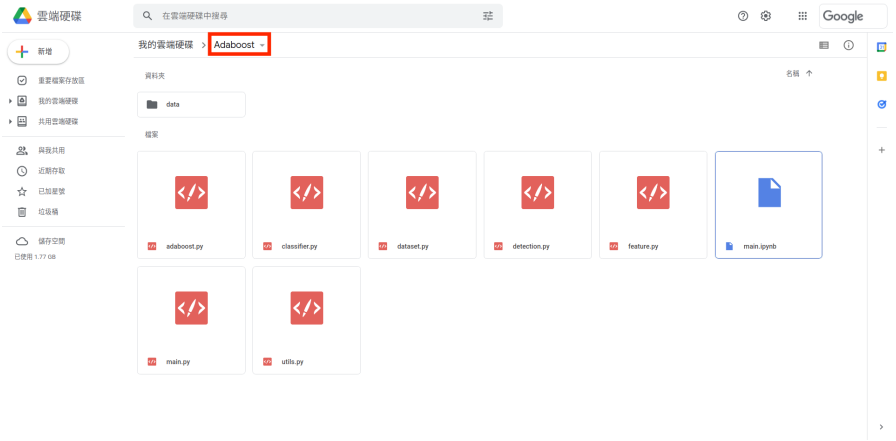
Files

Model	File name	Description
Adaboost	main.py/main.ipynb	Test all of the code in Adaboost.
	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.
	data folder	Train, test and detect datasets.
Yolov5	Yolov5_sample_code.ipynb	Execute yolov5
	HW1_materials.zip	Include all the data used by Yolov5_sample_code.ipynb
	GroundTruth.txt	Groundtruth of video.gif

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: "Adaboost"



2. Open the main.ipynb.
   Execute the first cell to mount your Google Drive. You will see the following messages if Google Drive is mounted successfully.

```
# for google colab
from google.colab import drive
# mount your Google Drive
drive.mount('/content/gdrive')
```

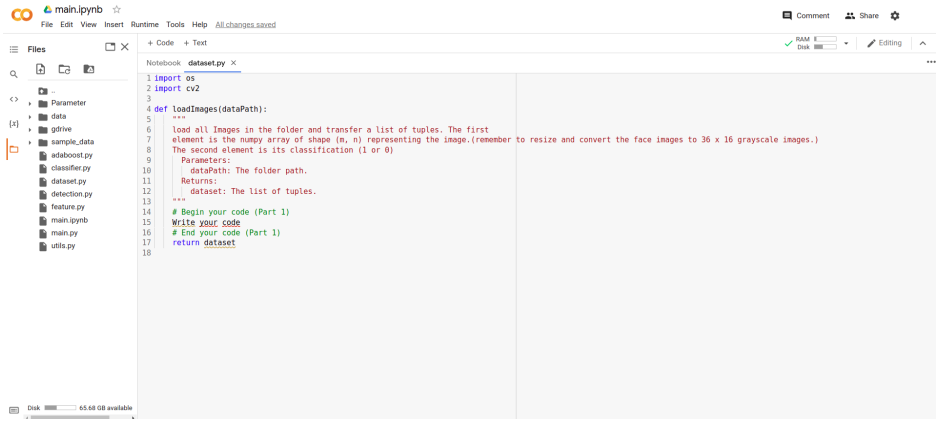
```
Mounted at /content/gdrive
```

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

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

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

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



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.

```
[ ]  # for google colab
     # 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/Adaboost/
```

Appendix B: For executing yolov5_sample_code.ipynb

Because Yolov5 kit and GPU resources are required, we recommend using Colab to run this code. Follow the below instructions in order and you can successfully execute the code.

1. Upload Yolov5_HW1 folder to your Google Drive, and don't need to change the folder name

2. Open the yolov5_sample_code.ipynb. As mentioned in Appendix A, execute the first cell to install related packages and mount your Google Drive.

```
[1]  1 !git clone https://github.com/ultralytics/yolov5  # clone
     2 %cd yolov5
     3 %pip install -qr requirements.txt  # install
     4 import os
     5 import torch
     6 import cv2
     7 import numpy as np
     8 from os import walk
     9 from os.path import join
    10 from datetime import datetime
    11 from yolov5 import utils
    12 display = utils.notebook_init()  # checks
    13 import matplotlib.pyplot as plt
    14 from google.colab import drive
    15 from google.colab.patches import cv2_imshow
    16 drive.mount('/content/gdrive')

     YOLOv5 🚀 v6.1-14-g8a66eba torch 1.10.0+cu111 CPU
     Setup complete ✅ (2 CPUs, 12.7 GB RAM, 41.9/107.7 GB disk)
     Mounted at /content/gdrive
```

3. Check if path is correct in second cell (the folder path to place HW1_material.zip)

```
[2]  1 '''
     2 This sample code takes '/content/gdrive/MyDrive/Yolov5_HW1' as the folder path to store HW1_material.zip,
     3 If the path used below is not your Google Drive path, please replace it with the path where you put HW1_material.zip
     4 '''
     5 yourPath = '/content/gdrive/MyDrive/Yolov5_HW1'
```

4. Continue to execute the remaining cell in order, please read the command and try to modify some parameters, so that you can get better results and also do some discussions.
5. In the last cell, please use Yolov5 model to detect cars, the code is similar to Part 4, you can follow the hint to finish between **# Begin your code** and **# End your code**

Appendix C: Some references

- Adaboost: ML Lecture 22: Ensemble
- Viola/Jones Face Detector: https://www.cs.ubc.ca/~lowe/425/slides/13-ViolaJones.pdf