Spring 2023
Introduction to
Artificial Intelligence

Homework 1

Mar. 14, 2023

Introduction

There are two tasks in this assignment,In task A, you need to

1) learn how to prepare a dataset for machine learning
2) build and train traditional ML models on image classification tasks
3) get a taste of a complete supervised learning process

In task B, you need to

1) load a pretrained model and directly apply it to the image (.png) to detect the object
2) learn how to finetune yolov7 model
3) achieve greater than 90% performance on the test dataset

When you do this assignment, please:

- **Only add your codes between # Begin your code and # End your code.**
- **Do not remove these two comments # Begin your code and # End your code.**
- **Do not import any additional packages.**
- **Use Python 3.10 in Task A and the default Python version of Colab in Task B.**

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. If you have never used Google Colab before, please refer to this official tutorial (shorturl.at/bdfBT).

Requirements

Task A

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., 

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 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: Build and Train Models (25%)

- Build and train KNN Classifier, Random Forest Classifier and Adaboost Classifier in model.py.
- Please use scikit-learn to implement this part.
- 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 understand these three algorithms, and briefly explain the following questions in your report.
  - Explain the difference between parametric and non-parametric models.
  - What is ensemble learning? Please explain the difference between bagging, boosting and stacking.

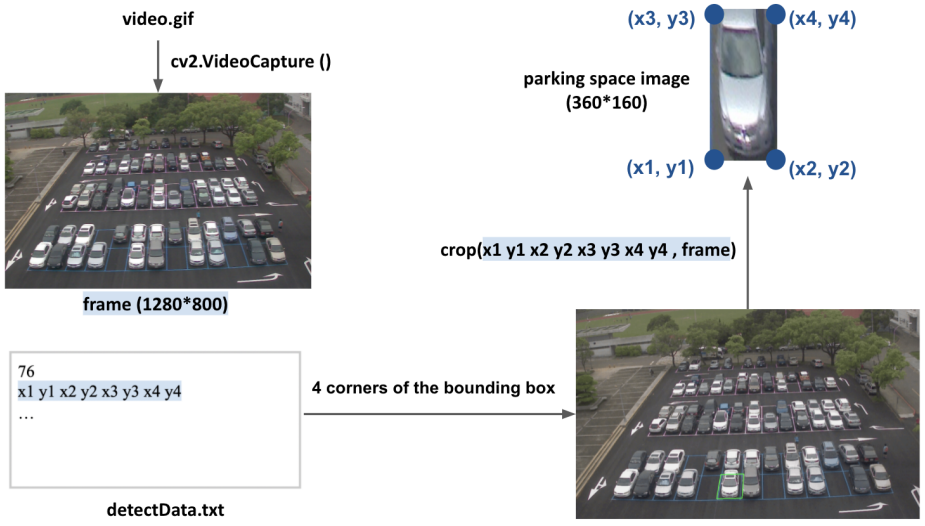
- Explain the meaning of the “n_neighbors” parameter in KNeighborsClassifier, “n_estimators” in RandomForestClassifier and AdaBoostClassifier.
- Explain the meaning of four numbers in the confusion matrix.
- In addition to “Accuracy”, “Precision” and “Recall” are two common metrics in classification tasks, how to calculate them, and under what circumstances would you use them instead of “Accuracy”.

Part 3: Additional experiments (10%)

- Please change the parameter “n_neighbors” in the KNeighborsClassifier and “n_estimators” in RandomForestClassifier and AdaBoostClassifier, and compare the corresponding detection performance. Of course, feel free to tune additional hyperparameters to improve the performance of your models.
- 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 find the best classifier with its optimal hyperparameters (ex. RandomForestClassifier with n_estimators=100), it will be used in part 4.

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.
- Run main.py or main.ipynb to test your implementation.
- 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.



Task B

DataSet: HW1_material

- 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.
- HW1.yaml
  - the file that YoloV7 use to find the path of dataset
- DETECT
  - parking_area.png: Apply YoloV7 to detect the object on this image
- DATASET
  - Used to fine tuning the Yolov7 model.

Part 1: Load a pretrained model and directly apply it to the .png to detect the object.(10%)

- Implement the cells with this kind of comment `''' comment'''`
- Successfully run the code in part1.
- Show the result like the following picture in your report.

- If you have any problem when running your code, check the path first and feel free to change the path we give you.
- You can find all the files by clicking  on the left side of the colab.

Part 2: Learn to fine tuning yolov7 model (10%)

- You need to input the training instruction which can be found in the training part of this link.

- Feel free to add parameters in instruction but do not change the parameters that we already give.
- You need to get 90% accuracy and paste the accuracy screenshot on your report.
- If you work on google colab, then you need to keep the output of each cell and submit the file which download by clicking File>Download>Download .ipynb

Part 3: (Optional)

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.
- Only one report should be submitted, write all you need in this file.
- Save the report as a report_{學號}**.pdf** file (ex. report_111XXX.pdf)
- The report should include:
  1. Screenshot of your code if needed 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 3/27 (Mon.) 23:59:00.**
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: dataset.py, model.py, detection.py, Yolov7.ipynb and report_{學號}.pdf.
2. Please follow the file hierarchy and the naming rules specified below. The wrong format will cause -10 points as well.

- hw1_111XXXXXX.zip
    - dataset.py
    - model.py
    - detection.py
    - Yolov7.ipynb
    - report_{學號}.pdf

Files

Model	File name	Description
ML_Models	main.py/main.ipynb	Test all of the code in ML_Models.
	dataset.py	Load images.

	model.py	The main file that implements the KNN, Random Forest and AdaBoost.
	detection.py	Detect images with your classifier.
	data folder	Train, test and detect datasets.
Yolov7	Yolov7_sample_code.ipynb	Execute Yolov7
	HW1_materials.zip	Include all the data used by Yolov7_sample_code.ipynb