Homework #1
Object Detection

Computer Vision Practice with Deep Learning
NTU, Spring 23

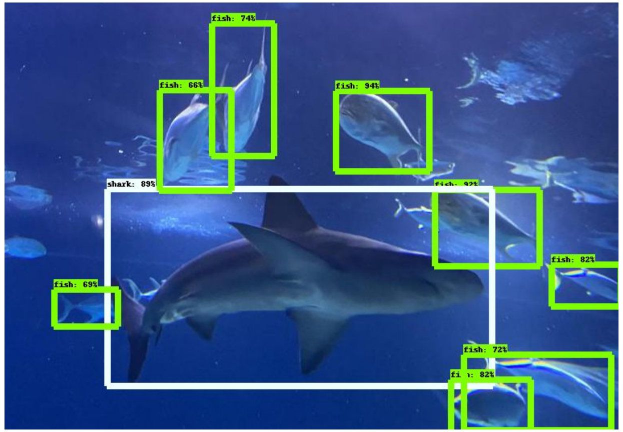
Outline

- Problem
- Dataset
- Evaluation
- Report
- Grading
- Submission
- Rules

Problem - Object Detection

- Object detection - localize objects and predict class for each image.
  - Input : RGB image
  - Output : a sequence of bounding boxes with class label and confidence.

- Perform object detection with the following methods:
  - **A.** CNN-based
  - **B.** Transformer-based
    - The prediction should be generated from the Transformer Decoder
    - e.g. DETR, DINO



Dataset

- The dataset consists of 637 colored images with 8 classes.
- download dataset
- We split the dataset into
  - **training set:** 447 images
  - **validation set:** 127 images
    - DO NOT use validation data to train your model in a fully supervised manner
    - (i.e. using ground truth of validation set for training is prohibited)
  - **testing set:** 63 images
    - DO NOT try to find the ground truth
- Violating any of the rules outlined for this assignment will result in a grade of zero.
- If you are uncertain about the legitimacy of the usage, email the TAs for clarification.

Evaluation

- Evaluation metric
  - We’ll use the metric taught in class - Average Precision
  - Please refer to the course slides or this intro
  - The performance will be evaluated by this function
- Ranking & Baseline
  - AP at IoU = [50:5:95]

Grading

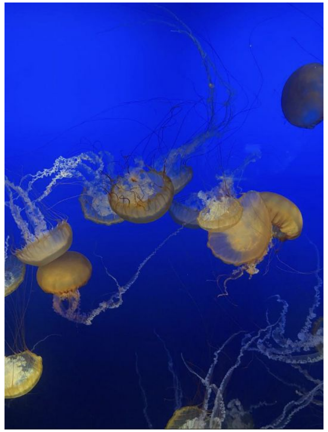
- Baseline(50%)
  - Public baseline(20%): 127 validation data
    - simple baseline(10%): 0.34
    - strong baseline(10%): 0.38
  - Private baseline(30%): 63 testing data
    - Will be announced after deadline (won’t be too difficult)
- Ranking(20%) (no kaggle this time)
  - Public(5%): validation set
  - Private(15%): testing set
- Report(30%)

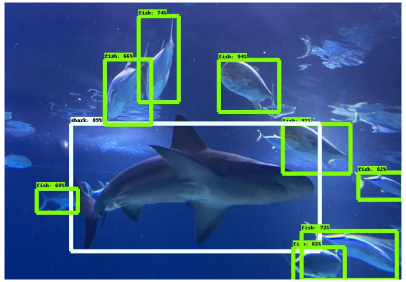
Report

1. (5%) Draw the architectures for both CNN-based and Transformer-based methods
    a. The graph should be **brief** and clear
    b. It would be fine to straight copy the figure from the paper
2. (10%) Report and compare the performance of two methods on validation set
    a. at least with mAP@[50:5:95], mAP@50, mAP@75
    b. use **table** to organize the results

Report

3. (10%) Report the implementation details of both methods
    a. Ex: augmentation, loss function, cross validation method, …etc.
4. (5%) Visualization: draw the bounding boxes of **<u>two</u>** methods on this **<u>test</u>** image.
    a. IMG_2574_jpeg_jpg.rf.ca0c3ad32384309a61e92d9a8bef87b9
    b. Result should be something like this





Submission

- Deadline: **2023/3/28 (Tue.) 23:59 (GMT+8)**
- Compress all files, then submit it to NTU COOL.(hw1_<studentID>.zip)
- Your submission should include the following files:
    - hw1.sh
    - hw1_download.sh
    - hw1_<studentID>.pdf (e.g. hw1_r10921059.pdf)
    - your python files (**Training** & **Inference** code, both required)
- **DO NOT submit the dataset**

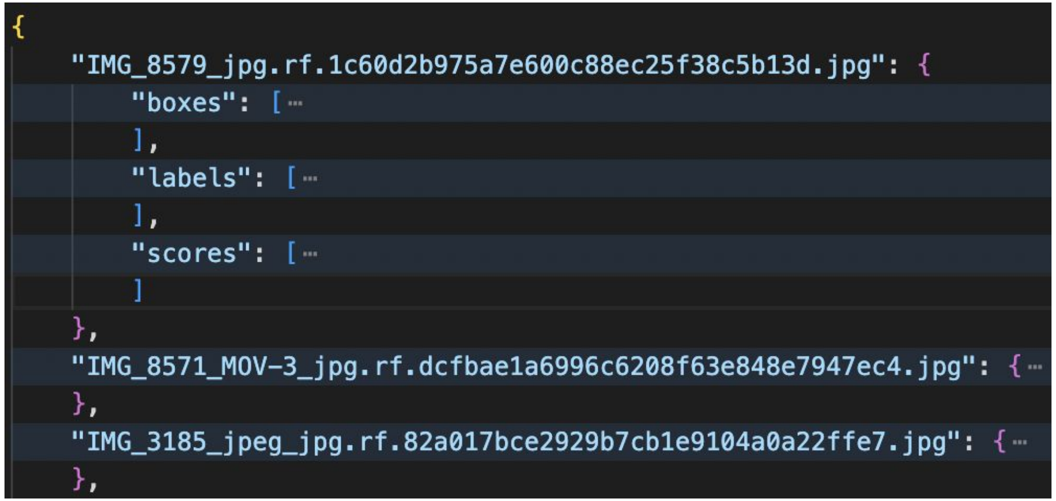
Script - hw1.sh

- Provide a **script** to test images under the specified directory with your model, and save the results in the specified json file.
- TAs will run your script as shown below:
  - bash hw1.sh $1 $2
    - $1: testing images directory with images named 'xxxx**.jpg** (e.g. input/test_dir/)
    - $2: path of output json file (e.g. output/pred.json)
- The output json file **must** have the same format as 'sample_submission.json'
- This section must be finished in **5 mins**, otherwise would be considered as a failed run.
- Your code should only output indicated files
  - Do not use imshow() or show() or save images in your code

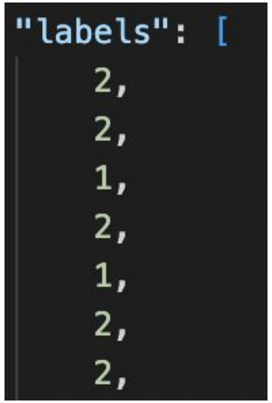
Sample JSON Format

- Overview
  - sample_submission.json is provided

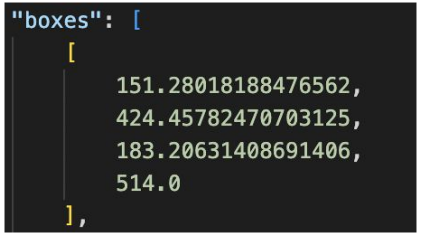
Each prediction contains:
labels, *boxes*, and *scores*
Label: class
Box: [x_min, y_min, x_max, y_max] (according to coordinate before any transformation)
Score: confidence



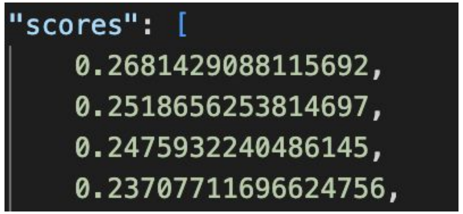
```
{
    "IMG_8579_jpg.rf.1c60d2b975a7e600c88ec25f38c5b13d.jpg": {
        "boxes": [⋯
        ],
        "labels": [⋯
        ],
        "scores": [⋯
        ]
    },
    "IMG_8571_MOV-3_jpg.rf.dcfbae1a6996c6208f63e848e7947ec4.jpg": {⋯
    },
    "IMG_3185_jpeg_jpg.rf.82a017bce2929b7cb1e9104a0a22ffe7.jpg": {⋯
    },
```



```
"labels": [
        2,
        2,
        1,
        2,
        1,
        2,
        2,
```



```
"boxes": [
    [
        151.28018188476562,
        424.4578247070312 5,
        183.2063140869140 6,
        514.0
    ],
```



```
"scores": [
    0.2681429088115692,
    0.2518656253814697,
    0.2475932240486145,
    0.2370771169662475 6,
```

Check your JSON file

- python check_your_prediction_valid.py [your_prediction_path] [target_path]
    - e.g. python3 check_your_prediction_valid.py pred,json data/valid/gt.json
    - use validation set to generate submission and check it with validation ground truth

Model Checkpoint

- Download and preprocess (e.g. unzip, tar zxf, etc.) your model in hw1_download.sh.
    - TAs will run `bash hw1_download.sh` prior to any inference.
- Only one model checkpoint needs to be submitted for baseline and ranking.

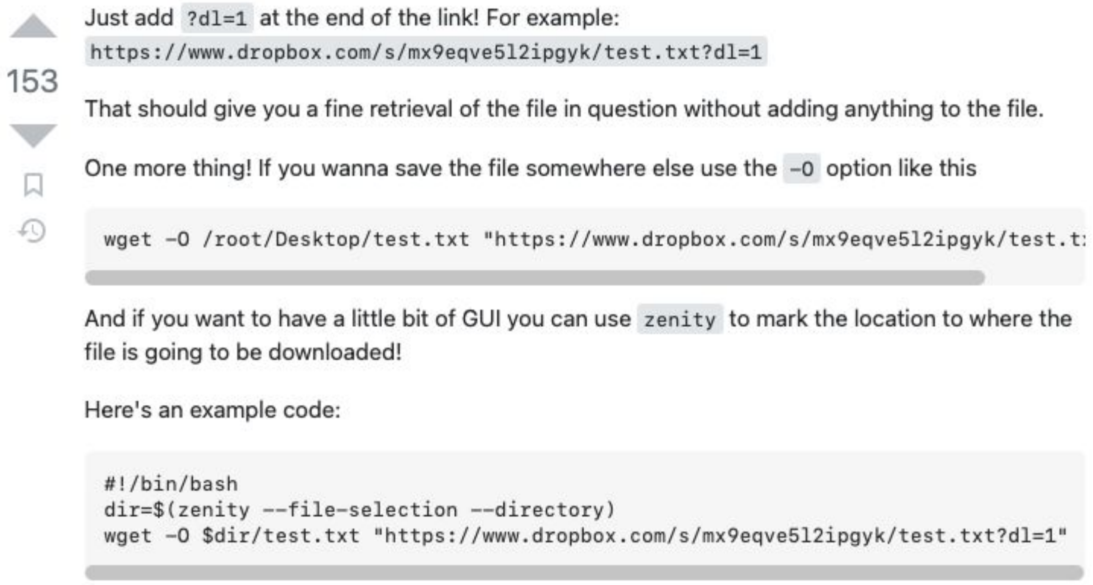
- **DO NOT** delete your model checkpoints before the TAs release your score and before you have ensured that your score is correct.

Download Tutorial

- Please use **wget** to download the model checkpoints from cloud drive (e.g. Dropbox, Google Drive) or your working station.
    - You should use **-O argument** to specify the filename of the downloaded checkpoint.
- Tutorial for Google Drive
    - https://www.matthuisman.nz/2019/01/download-google-drive-files-wget-curl.html

Download Tutorial

- Tutorial for Dropbox
  - ref



153

Just add `?dl=1` at the end of the link! For example: `https://www.dropbox.com/s/mx9eqve512ipgyk/test.txt?dl=1`

That should give you a fine retrieval of the file in question without adding anything to the file.

One more thing! If you wanna save the file somewhere else use the `-O` option like this

```
wget -O /root/Desktop/test.txt "https://www.dropbox.com/s/mx9eqve512ipgyk/test.t:
```

And if you want to have a little bit of GUI you can use `zenity` to mark the location to where the file is going to be downloaded!

Here's an example code:

```
#!/bin/bash
dir=$(zenity --file-selection --directory)
wget -O $dir/test.txt "https://www.dropbox.com/s/mx9eqve512ipgyk/test.txt?dl=1"
```

Rules – Environment

- Ensure your code can be executed successfully on **Linux Ubuntu**
- Language and framework
    - **python3 (3.10)** is the only language you should use
    - **PyTorch (1.13)** is the only allowed framework
- Please keep in mind that any attempt to attack the system, including the use of 'sudo', is not allowed and will result in a 0 for this homework.
- We do NOT recommend using distributed inference, and if you choose to do so, you assume all associated risks if your code cannot be executed properly.
- Do NOT hardcode any absolute paths in your program, but instead use input arguments to specify the paths.

Rules – Code

- If your code cannot be executed, you have one chance to make minor modifications to your code.

	Success to fix the code	Fail to fix the code
deduction	80%	0

- Reproducibility
  - you must make sure that the training results can be reproduced with the code you submit to NTU COOL

Rules – Policy

- **Late policy:**

late (hour)	(0, 24]	(24, 48]	(48, 72]	>72
deduction	70%	50%	30%	0

- **Plagiarism**
  - You must document all of your source material in your report. You must also cite any sources (e.g. GitHub repo, paper, website) from which you obtain numbers, ideas, or other material. Use of ChatGPT or other AI composition software is permitted, but you must specify the usage.
  - We encourage you to discuss ideas with your classmates, but please do NOT share your code directly with them.
  - Plagiarism is a serious offense and will not be treated lightly.

Helps

- Free GPU Resources
  - Google Colab
    - Introduction
    - GPUs: K80, or rarely T4
    - Limits
  - Kaggle
    - Introduction
    - GPUs: T4 x 2 or P100
    - Limits: 30 hours per week

Helps

- Mail
  - If you have any questions, contact TAs via this email
    - ntu.cvpdl.ta@gmail.com
  - Please note that emails sent to TAs' personal email addresses will not receive a response.