

課程名稱：Video streaming and tracking
作業名稱：Homework 2 - Object Detection
系級：資訊科學與工程研究所 二年級
學號：310551135
姓名：張宇辰

註：綠色底色為檔案或路徑，橘色底色為指令

1. Environment:

- OS: Ubuntu 20.04
- GPU: GTX 1080Ti, CUDA Version 11.4
- Python 3.8.13
- Anaconda

numpy==1.23.4	torch==1.10.0
image==1.4.1	protobuf==3.20.0
opencv-python==4.6.0.66	setuptools==59.5.0
tensorboard==2.9.0	matplotlib==3.6.1

(其餘 package 請見 `Code/environment.yml`)

2. Experiment Setup and Explain (完整實作請見第三節)

- **Data pre-process (before training)**

YOLOX 只支援 COCO format 或是 VOC format，而本次作業所提供的資料格式為 YOLO format，因此訓練之前須做轉換，我選擇將 YOLO format 轉換為 COCO format 來進行訓練。轉換方式參考[該網站\[1\]](#)，並稍做改寫，完整程式碼請見 `Code/yolo2coco.py`。

- **Data pre-process (before calculate validation mAP)**

此次作業須提交 validation set 的 mAP，計算參考自 [Object-Detection-Metrics\[2\]](#)。助教提供的 validation set label 格式為 (x_center, y_center, width, height) 且已經歸一化，而上述計算 mAP 的專案須提供 (left, top, right, bottom) pixel level 的格式，因此須將 validation set label 轉換為後者的格式。完整程式碼請見 `Code/yolo2xyrb.py`。

- **Model architecture**

作業其中一項要求是加入 SE module，我將助教提供的 sample code 寫在 `Code/SE/YOLOX/yolox/models/attention.py`。並將 SE module 加入同路徑底下的 `yolo_pafpn.py`，分別在 Backbone Darknet53 的 dark5, dark4, dark3 分支後方加入 SE module，如下圖所示：

```
85 self.se1 = SELayer(int(in_channels[2] * width))
86 self.se2 = SELayer(int(in_channels[1] * width))
87 self.se3 = SELayer(int(in_channels[0] * width))
```

```
98 # backbone
99 out_features = self.backbone(input)
100 features = [out_features[f] for f in self.in_features]
101 [x2, x1, x0] = features
102
103 x0 = self.se1(x0)
104 x1 = self.se2(x1)
105 x2 = self.se3(x2)
```

- **Hyperparameters**

Name	Value	Name	Value
Epoch	300	Batch Size	16
Pre-train Weight	yolox_s.pth	Scheduler	yoloxwarmcos
Learning Rate	0.01/64.0	Weight Decay	0.00004

3. Re-produce training and inference (Step 4 為計算 val 的 mAP，可跳過)

請注意：在 pdf 複製指令時，換行處可能會出現空白，導致指令執行失敗。強烈建議到 [GitHub](https://github.com/EricZhang1028/Video-Streaming-and-Tracking/tree/main/hw2_object_detection) 閱讀第三小節：

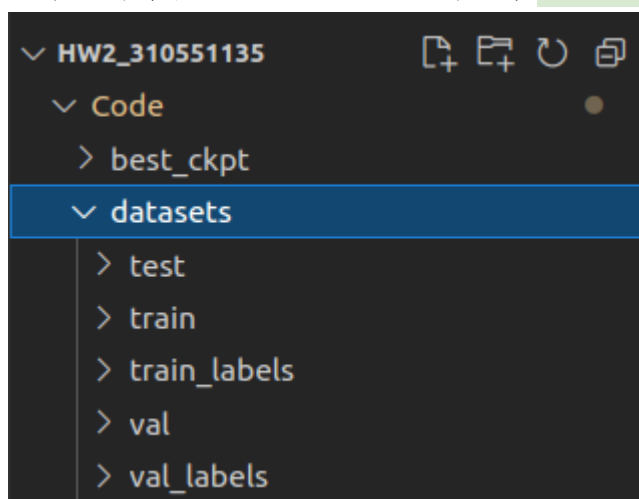
https://github.com/EricZhang1028/Video-Streaming-and-Tracking/tree/main/hw2_object_detection

Import environment

- `Code/environment.yml` 為本次作業的環境，可事先匯入安裝。
- `conda env create -f environment.yml` 若有路徑錯請修改檔案內最後的 prefix
- 以下套件請在上述環境，手動安裝：
`pip install opencv-python==4.6.0.66`
`pip install tensorboard==2.9.0`

Prepare datasets

- 將本次作業提供的 dataset 放置到目錄 `Code/datasets` 底下，如下圖所示：



切換到目錄 `Code` 底下，執行下方指令，將 `train` 和 `val` 的資料集，從 YOLO format 轉換為 COCO format：

`python yolo2coco.py`

轉換完畢後，即可在 `datasets/annotations` 資料夾底下看到 `train.json` 和 `val.json`。

Original

- Step 1: Set YOLOX path
切換到目錄 `Code/Original/YOLOX`。
先執行安裝指令，使系統路徑指向 `Original` 底下的 YOLOX：
`pip3 install -v -e .`
- Step 2: Training command
在目錄 `Code/Original/YOLOX` 執行訓練指令：
`python tools/train.py -f exps/example/custom/hw2_exp.py -d 1 -b 16 --fp16 -o -c yolox_s.pth`
- Step 3: Inference **test** data (with best checkpoint)
在目錄 `Code/Original/YOLOX` 執行推論指令：
`python tools/demo.py image -f exps/example/custom/hw2_exp.py -c ../../best_ckpt/original_best_ckpt.pth --path ../../datasets/test --conf 0.25 --nms 0.45 --tsize 640 --save_result --device gpu --txt_path ../../Original`
預測結果儲存在 `Original`
- (Optional) Step 4: Inference **validation** data (with best checkpoint)
在目錄 `Code/Original/YOLOX` 執行推論指令：
`python tools/demo.py image -f exps/example/custom/hw2_exp.py -c ../../best_ckpt/original_best_ckpt.pth --path ../../datasets/val --conf 0.25 --nms 0.45 --tsize 640 --save_result --device gpu --txt_path ../../origi_val_infer_res`
預測結果儲存在 `Code/origi_val_infer_res`

計算 validation set 的 mAP 之前，要先轉換 `val_labels` 的格式 (`xywh` -> `xyrb`)，切換到目錄 `Code` 底下，執行指令：

```
python yolo2xyrb.py
結果將儲存在 datasets/val_labels_xyrb。
```

計算 validation set 的 mAP，切換到目錄 `Code/Object-Detection-Metrics` 底下，執行指令：

```
python pascalvoc.py -t 0.85 -gtformat xyrb -detformat xyrb -np --gt ../../datasets/val_labels_xyrb/ --det ../../origi_val_infer_res
```

with SE module

- Step 1: Set YOLOX path

切換到目錄 `Code/SE/YOLOX`。

先執行安裝指令，使系統路徑指向 SE 底下的 YOLOX：

```
pip3 install -v -e .
```

- Step 2: Training command

在目錄 `Code/SE/YOLOX` 執行訓練指令：

```
python tools/train.py -f exps/example/custom/hw2_se_exp.py -d 1 -b 16 --fp16 -o -c yolox_s.pth
```

- Step 3: Inference **test** data (with best checkpoint)

在目錄 `Code/SE/YOLOX` 執行推論指令：

```
python tools/demo.py image -f exps/example/custom/hw2_se_exp.py -c ../../best_ckpt/se_best_ckpt.pth --path ../../datasets/test --conf 0.25 --nms 0.45 --tsize 640 --save_result --device gpu --txt_path ../../SE
```

預測結果儲存在 `SE`

- (Optional) Step 4: Inference **validation** data (with best checkpoint)

在目錄 `Code/SE/YOLOX` 執行推論指令：

```
python tools/demo.py image -f exps/example/custom/hw2_se_exp.py -c ../../best_ckpt/se_best_ckpt.pth --path ../../datasets/val --conf 0.25 --nms 0.45 --tsize 640 --save_result --device gpu --txt_path ../../se_val_infer_res
```

預測結果儲存在 `Code/se_val_infer_res`

計算 validation set 的 mAP 之前，要先轉換 val_labels 的格式 (xywh -> xyrb)，切換到目錄 `Code` 底下，執行指令：

```
python yolo2xyrb.py
```

結果將儲存在 `datasets/val_labels_xyrb`。

計算 validation set 的 mAP，切換到目錄 `Code/Object-Detection-Metrics` 底下，執行指令：

```
python pascalvoc.py -t 0.85 -gtformat xyrb -detformat xyrb -np --gt ../../datasets/val_labels_xyrb/ --det ../../se_val_infer_res
```

4. Validation results

Original:

```
(video_hw2_env2) hsc_yuchen@hsc-yuchen-System:~/Course/Video-Streaming-and-Tracking/HW2_310551135/Code/Object-Detection-Metrics$ python pascalvoc.py -t 0.85 -gtformat xyrb -detformat xyrb -np --gt ../datasets/val_labels_xyrb/ --det ../origi_val_infer_res

#####
#
# THE CURRENT VERSION WAS UPDATED WITH A VISUAL INTERFACE, INCLUDING MORE METRICS AND SUPPORTING
# OTHER FILE FORMATS.
#
# PLEASE ACCESS IT ACCESSED AT:
# https://github.com/rafaelpadilla/review_object_detection_metrics
#
# @Article{electronics10030279,
#   author      = {Padilla, Rafael and Passos, Wesley L. and Dias, Thadeu L. B. and Netto,
#                 Sergio L. and da Silva, Eduardo A. B.},
#   title       = {A Comparative Analysis of Object Detection Metrics with a Companion
#                 Open-Source Toolkit},
#   journal     = {Electronics},
#   volume      = {10},
#   year        = {2021},
#   number      = {3},
#   article-number = {279},
#   url         = {https://www.mdpi.com/2079-9292/10/3/279},
#   issn        = {2079-9292},
#   doi         = {10.3390/electronics10030279}, }
#
#####

Folder /home/hsc_yuchen/Course/Video-Streaming-and-Tracking/HW2_310551135/Code/Object-Detection-Metrics/resu
lts already exists and may contain important results.

Enter 'Y' to continue. WARNING: THIS WILL REMOVE ALL THE CONTENTS OF THE FOLDER!
Or enter 'N' to abort and choose another folder to save the results.
Y
AP: 93.57% (0)
mAP: 93.57%
```

with SE module

```
(video_hw2_env2) hsc_yuchen@hsc-yuchen-System:~/Course/Video-Streaming-and-Tracking/HW2_310551135/Code/Object-Detection-Metrics$ p
ython pascalvoc.py -t 0.85 -gtformat xyrb -detformat xyrb -np --gt ../datasets/val_labels_xyrb/ --det ../se_val_infer_res

#####
#
# THE CURRENT VERSION WAS UPDATED WITH A VISUAL INTERFACE, INCLUDING MORE METRICS AND SUPPORTING
# OTHER FILE FORMATS.
#
# PLEASE ACCESS IT ACCESSED AT:
# https://github.com/rafaelpadilla/review_object_detection_metrics
#
# @Article{electronics10030279,
#   author      = {Padilla, Rafael and Passos, Wesley L. and Dias, Thadeu L. B. and Netto,
#                 Sergio L. and da Silva, Eduardo A. B.},
#   title       = {A Comparative Analysis of Object Detection Metrics with a Companion
#                 Open-Source Toolkit},
#   journal     = {Electronics},
#   volume      = {10},
#   year        = {2021},
#   number      = {3},
#   article-number = {279},
#   url         = {https://www.mdpi.com/2079-9292/10/3/279},
#   issn        = {2079-9292},
#   doi         = {10.3390/electronics10030279}, }
#
#####

Folder /home/hsc_yuchen/Course/Video-Streaming-and-Tracking/HW2_310551135/Code/Object-Detection-Metrics/results already exists and
may contain important results.

Enter 'Y' to continue. WARNING: THIS WILL REMOVE ALL THE CONTENTS OF THE FOLDER!
Or enter 'N' to abort and choose another folder to save the results.
Folder /home/hsc_yuchen/Course/Video-Streaming-and-Tracking/HW2_310551135/Code/Object-Detection-Metrics/results already exists and
may contain important results.

Enter 'Y' to continue. WARNING: THIS WILL REMOVE ALL THE CONTENTS OF THE FOLDER!
Or enter 'N' to abort and choose another folder to save the results.
Y
AP: 93.64% (0)
mAP: 93.64%
```

5. Discussion

- 第一次使用 YOLOX，與 YOLOv5 在使用上相比較不直覺。第一個遇到的問題是資料格式轉換，YOLOX 不支援 xywh 的資料格式，需要轉 COCO format，在最後計算 validation set 的 mAP 之前，也需要轉一份程式碼做轉換。因此在訓練之前需要花點時間做資料格式的轉換。
- 我是在助教課堂講解前就開始做這份作業，不曉得 YOLOX 不會將 bbox 的座標和信心指數輸出成檔案，花了時間去搜尋相關資料才知道預設沒有這項功能。因此去找預測結果的 code，並加入一段程式碼將預測結果都匯成 txt 檔。
- 以我目前的硬體規格只能選擇 yolox_m 的 pre-trained model，batch 最多只能到 16，因此無法在這方面做更多嘗試。
- SE module 的加入方式已在 2. Experiment Setup and Explain 的 model architecture 做說明。

6. References

- [1]. <https://github.com/Taeyoung96/Yolo-to-COCO-format-converter>
- [2]. <https://github.com/rafaelpadilla/Object-Detection-Metrics>