Preparatory stage

1. Please download weight files

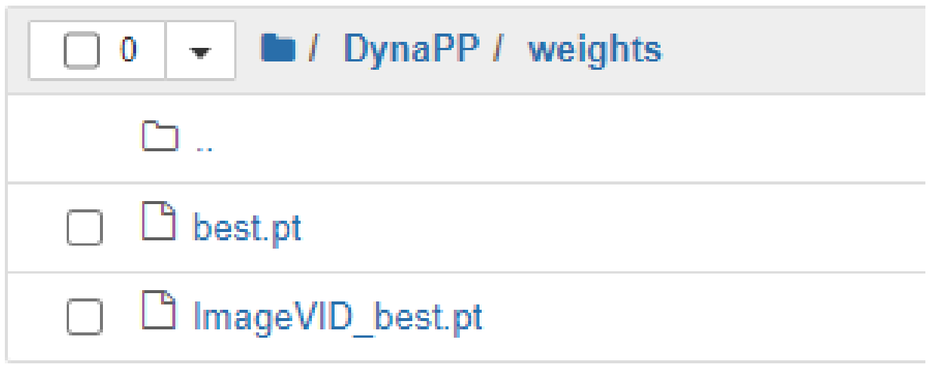
Please download weight files below

(Put the files in 'weights' folder)

https://drive.google.com/file/d/1LTSKE19bpygugyIP9jMk2dtjdgcQZ1vu/view?usp=share_link

https://drive.google.com/file/d/19zlMTZzF9tqOnpDBxMkoKz6u7S3-x7CW/view?usp=share_link

2. Put those files in the ‘weights’ folder.



3. Download datasets

Put the files in the directory of your choice.

Please download datasets below

(Put the files in directory you want, and modify the code inside 'Run.ipynb'

AUAIR

https://drive.google.com/file/d/1syHeOWTO5clw3pjE68TWQdhzZPfTsHTv/view?usp=share_link

VisDrone

https://drive.google.com/file/d/1f02BSNxu0QAkimABYEJeLMSR01Tk1Tnr/view?usp=share_link

UAVDT

https://drive.google.com/file/d/1MpPPzEgjuRH3DjwFE0jhDxscSzqMjPpW/view?usp=share_link

ImageVID

https://drive.google.com/file/d/1w_K7uV4C_VxM5NryFpJFQC8OtSZbPlde/view?usp=share_link

Test hardware

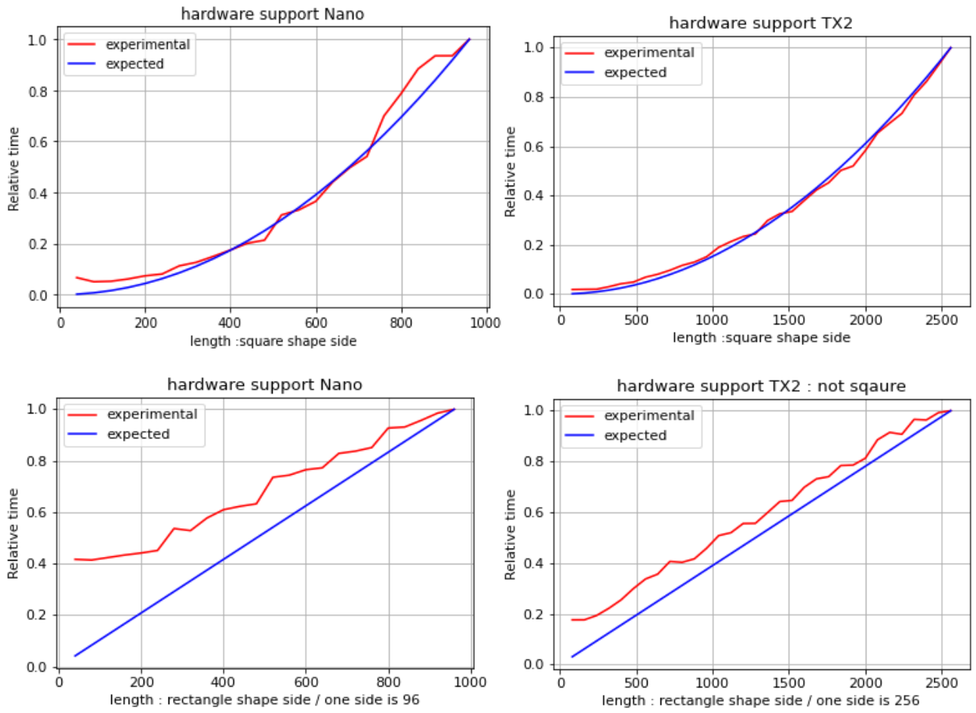
1. Run ‘Test your hardware.ipynb’

2. Check the files in the ‘hardware_support’ folder

constant

- Nano.png : inference time checking
  960 × 960 → … → 40 × 40
- not_square_Nano.png :
  960 × 96 → … → 40 × 96

- Nano.png :
  2560 × 2560 → … → 80 × 80
- not_square_Nano.png :
  2560 × 256 → … → 80 × 256



Jetson Nano **TX2**

We would like to kindly inform you that the acceleration may not be the same depending on the hardware.

Therefore, we strongly recommend using Nvidia Jetson TX2 and Jetson Nano or similar specification hardware to reproduce our experiments.

However, if using Nvidia Jetson TX2 and Jetson Nano or similar specification hardware is not feasible, we suggest experimenting with your existing hardware.

If you would like to see our acceleration results indirectly, you can refer to the 'excel_results/files’ and ‘Time measured by our hardwares.xlsx’. This will enable you to calculate the results indirectly by multiplying the average resolution by the corresponding time.

Evaluate (DynaPP / baseline / Pack and Detect)

1. Go to ‘Run.ipynb’

2. Write dataset directory

!! Please write the directory you put datasets in.

In [ ]:

```
# Please modify.
UAVDT_directory = '../data/datasets/UAVDT/UAV-benchmark-M'
VisDrone_directory = '../data/datasets/VisDroneVID/sequences'
AUAIR_directory = '../data/datasets/AUAIR/videos'
ImageVID_directory = '../data/datasets/ImageVID_yolo_form'
```

3. Run the code

4. Results are saved in excel inside the ‘excel_result’ folder and the ‘runs/test’ folder.

Note please.

DynaPP

Square
Baseline

Pack and
Detect

```
# Run YOLOv5x on AUAIR
for idx, video in enumerate(AUAIR_video_list
    AUAIR_change_yaml(video)
    # The following runs DynaPP, Baseline, Pa

    !python3 val.py --pack --dataset_name 'AU
    !python3 val.py --dataset_name 'AUAIR' --
#     !python3 val_packanddetect.py --pack --
```

Result Analysis

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Videos	Baseline mAP	DynaPP maP		Baseline inference time	DynaPP inference time		Baseline mAP average	DynaPP mAP average		acceleration		average acceleration	mAP loss
2								#DIV/0!	#DIV/0!		#DIV/0!		#DIV/0!	#DIV/0!
3											#DIV/0!			
4											#DIV/0!			
5											#DIV/0!			
6											#DIV/0!			
7											#DIV/0!			
8											#DIV/0!			
9											#DIV/0!			
10											#DIV/0!			
11											#DIV/0!			
12											#DIV/0!			
13											#DIV/0!			
14											#DIV/0!			
15											#DIV/0!			
16											#DIV/0!			
17											#DIV/0!			
18											#DIV/0!			
19											#DIV/0!			
20											#DIV/0!			
21											#DIV/0!			
22											#DIV/0!			
23											#DIV/0!			
24											#DIV/0!			
25											#DIV/0!			
26											#DIV/0!			
27											#DIV/0!			
28											#DIV/0!			
29											#DIV/0!			
30											#DIV/0!			
31											#DIV/0!			
32											#DIV/0!			
33											#DIV/0!			
34											#DIV/0!			
35											#DIV/0!			
36											#DIV/0!			
37											#DIV/0!			
38											#DIV/0!			
39											#DIV/0!			
40											#DIV/0!			
41											#DIV/0!			
42											#DIV/0!			
43											#DIV/0!			
44											#DIV/0!			
45											#DIV/0!			
46											#DIV/0!			
47											#DIV/0!			
48											#DIV/0!			
49											#DIV/0!			
50											#DIV/0!			
51											#DIV/0!			
52											#DIV/0!			

Copy and paste the data

Copy and paste the data

Here, results are pop up

Results analysis.xlsx

SSD to a dynamic resolution model

1. Go to ‘SSD_to_dynamic/Run.ipynb’

2. Adjust resolution [width, height]

Adjust Resolution -> Please change resolution [width ,height]

```
resolution=[150 , 81] # width, height
input2 = []
for i in range(3):
    input2.append(inputs[i][150-resolution[1]//2:150-resolution[1]//2+resolution[1]:,150
tensor2 = utils.prepare_tensor(input2)
```

3. Check results

Ablation Study 1: Training

1. Go to lines 229 - 268 in ‘models/yolo.py’

2. Change AB distance methods (choose one method)

3. Run ‘Ablation_Run.ipynb’

Ablation Study 2: Dynamic Resolution inference

Grid based anchor boxes

1. Go to line 56 in ‘models/yolo.py’

2. Comment lines 56-77 and uncomment lines 79-105

3. Run ‘Run.ipynb’ on ImageNet VID (others too if you want).

Manipulating upper left features

1. Go to the line 236, 329 in ‘models/common.py’

2. Comment the original class Concat and class Focus;

Uncomment the corresponding classes under the original class Concat and class Focus.

3. Run ‘Run.ipynb’ on ImageNet VID (others too if you want).

Where to check our modifications in codes

- canvas_DynaPP.py
  : Check all : Main to Check

- canvas_packanddetect.py
  : Check all

- val.py
  : Check line 49-52, 103-110, 138-143, 202-204, 221, **231-437**, 463-466, 491-499, 531-561, 632-637

- val_packanddetect.py
  : Check line 49-52, 103-110, 138-143, 217, **227-358**, 384-387, 412-420, 452-482, 553-557

- models/common.py
  : Check class Focus, class Concat

- models/yolo.py &
  : Check class Detect, line 229-268

- models/yolov5_nopad_Focus.yaml
  : Check all

- SSD_to_Dynamic/SSD_utils.py
  : Check line 227, 239-240, 261, 282-286, **299-338**

- SSD_to_Dynamic/SSD_model.py
  : Check line 109-110