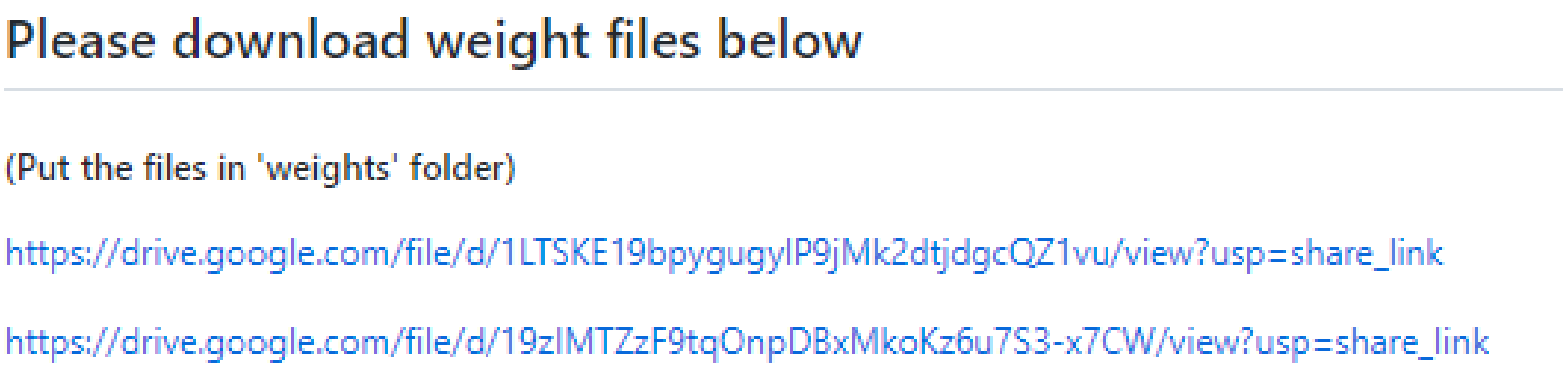
Preparatory stage

1. Download weight files



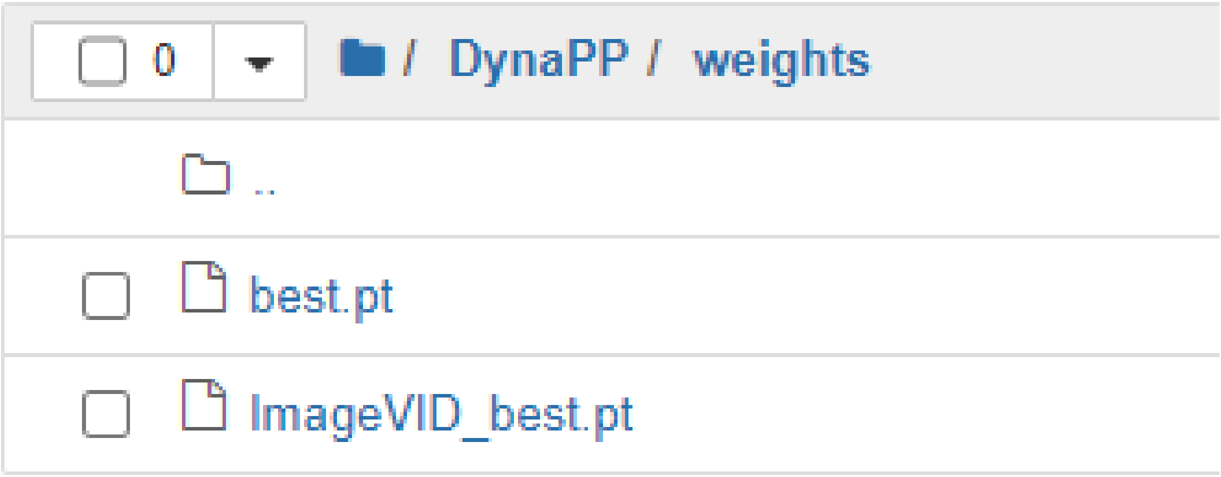
Please download weight files below

(Put the files in 'weights' folder)

https://drive.google.com/file/d/1LTSKE19bpygugylP9jMk2dtjdgcQZ1vu/view?usp=share_link

https://drive.google.com/file/d/19zIMTZzF9tqOnpDBxMkoKz6u7S3-x7CW/view?usp=share_link

2. Put those files in ‘weights’ folder.



□ 0 ▾ / DynaPP / weights

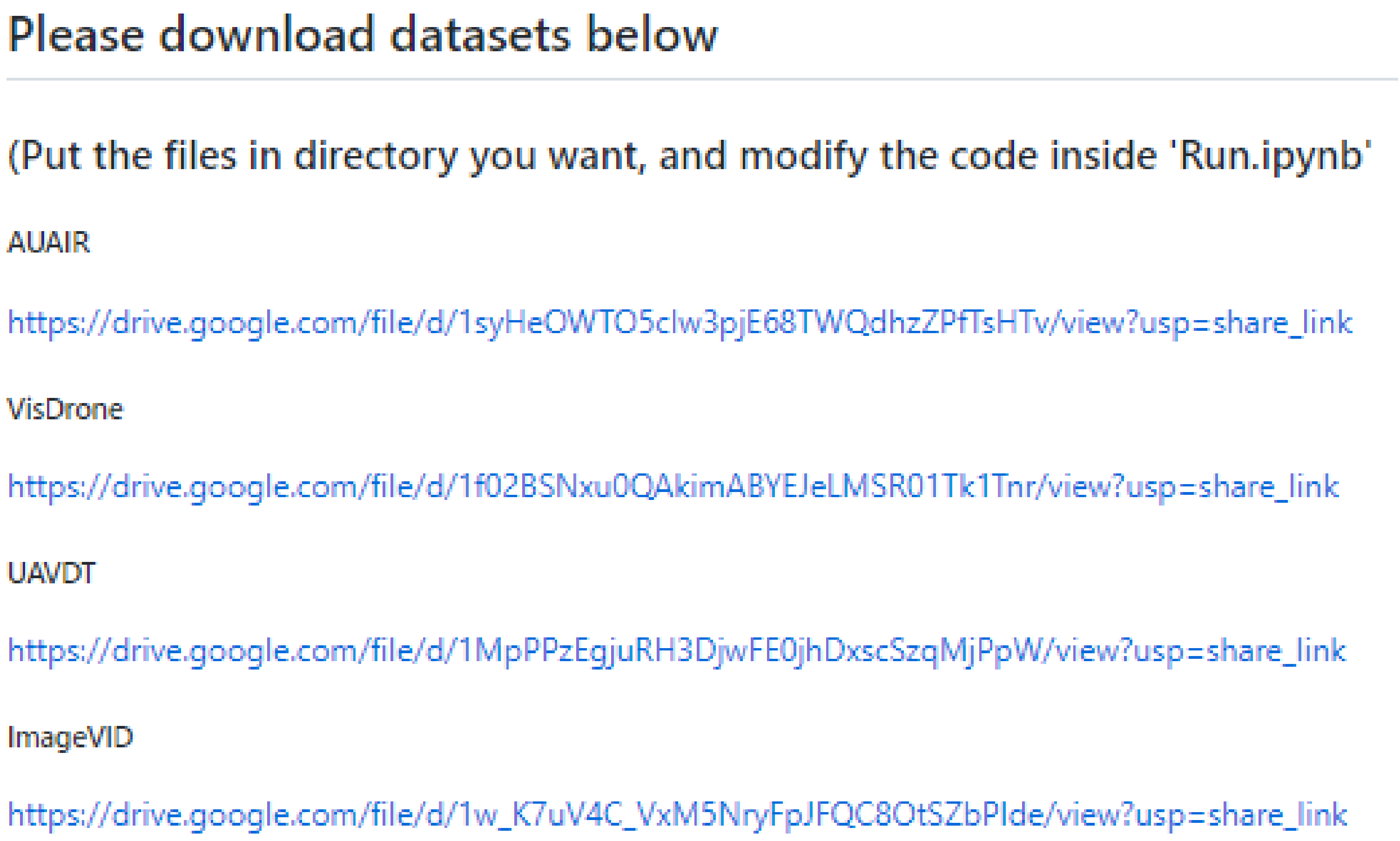
..

□ best.pt

□ ImageVID_best.pt

3. Download datasets

Put the files in directory you want.



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_VxM5NryFpJFQC8OtSZbPIde/view?usp=share_link

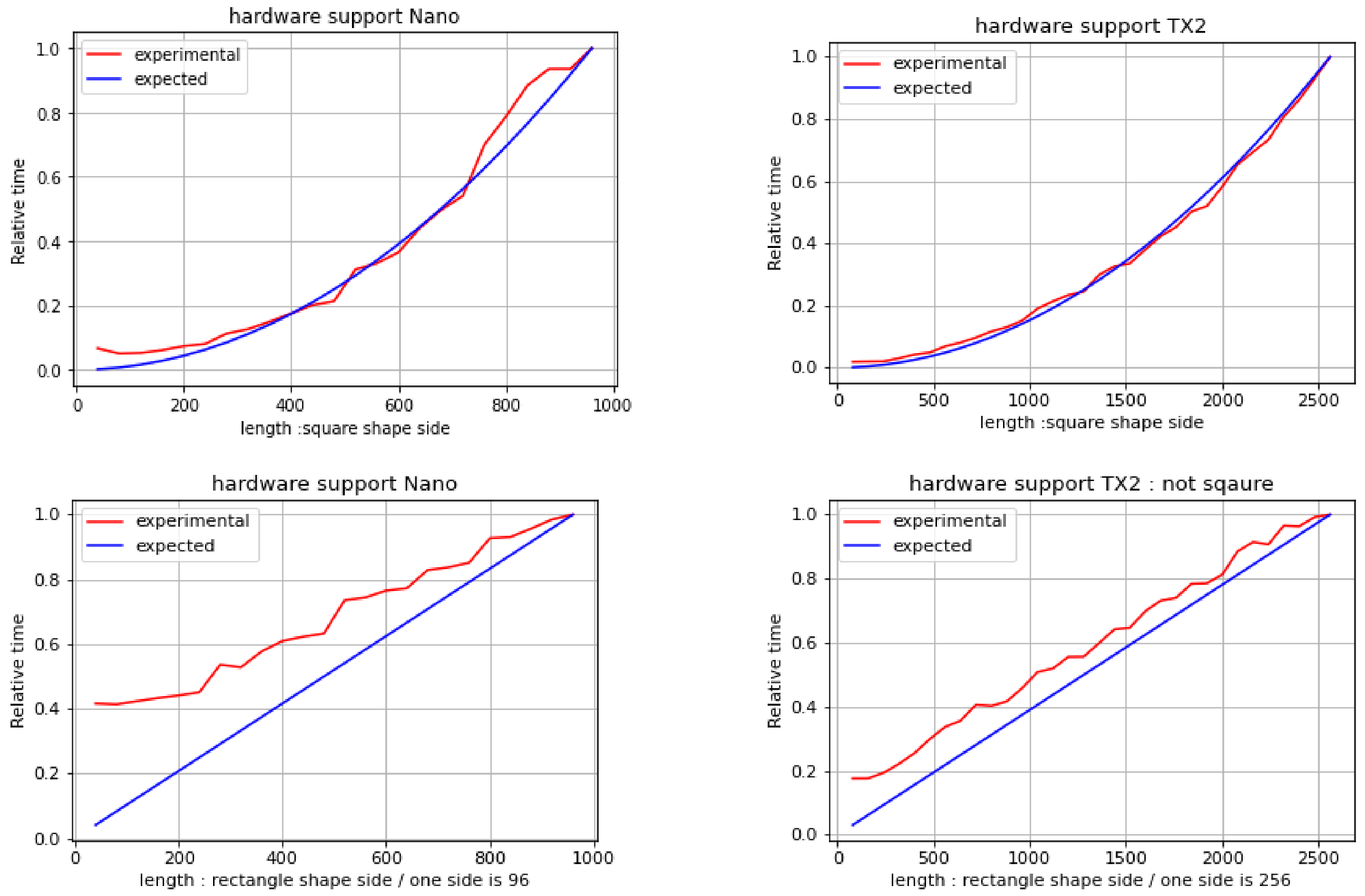
Test hardware

1. Run ‘Test your hardware.ipynb’

2. Check files in ‘hardware_support’ folder

   - 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**

Note that depending on the hardware, the inference time may not be accelerated as much as the operation costs reduction.

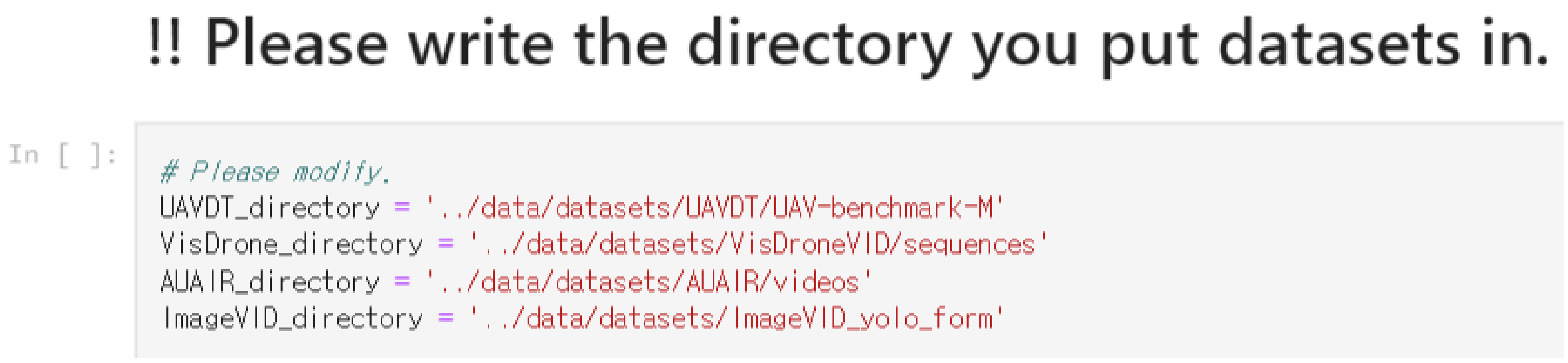
Therefore, we strongly recommend using Nvidia Jetson TX2 and Jetson Nano or with hardware of a similar specification for reproducing our experiments.

However, if there is none, please experiment with existing hardware and refer to the result of acceleration indirectly with average resolution in 'excel_results/files'.

Evaluate (DynaPP / baseline / Pack and Detect)

1. Go to ‘Run.ipynb’

2. Write dataset directory

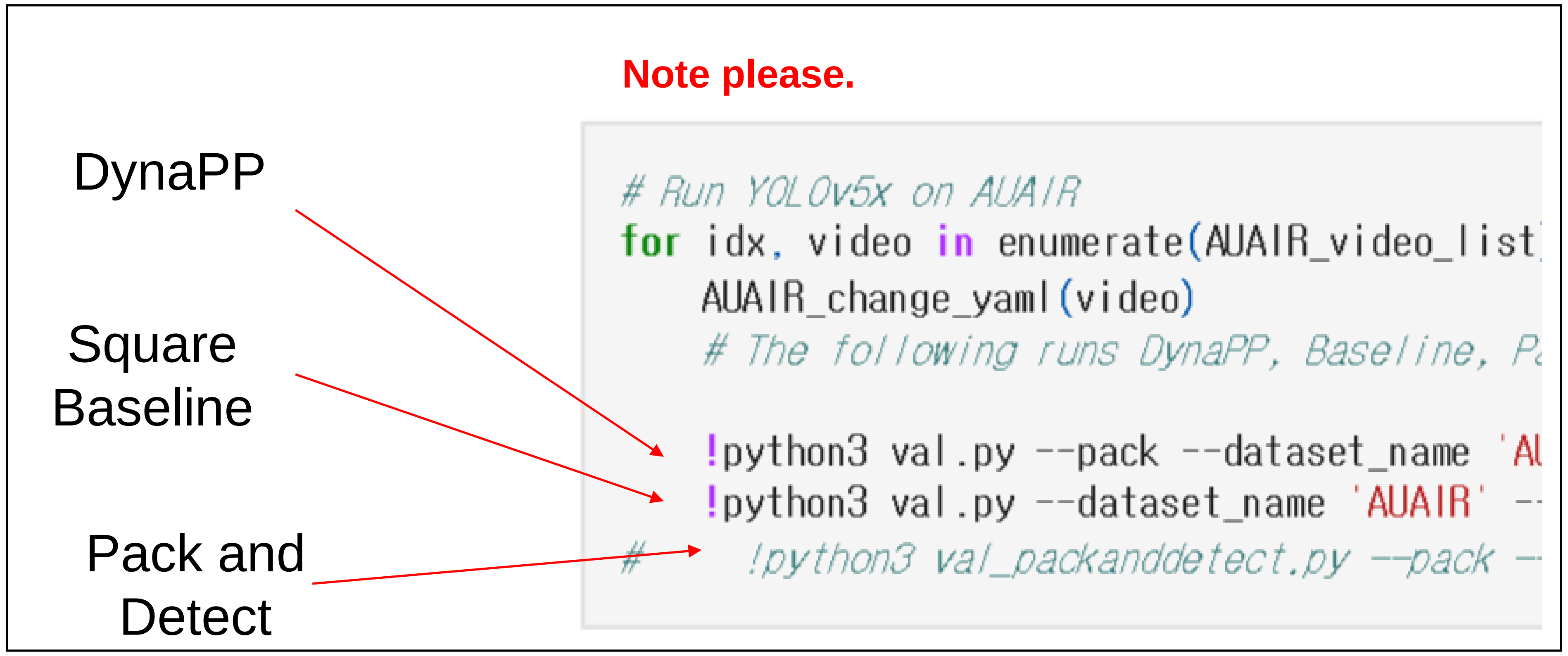


!! Please write the directory you put datasets in.

```
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 ‘excel_result’ folder and ‘runs/test’ folder.



Note please.

DynaPP

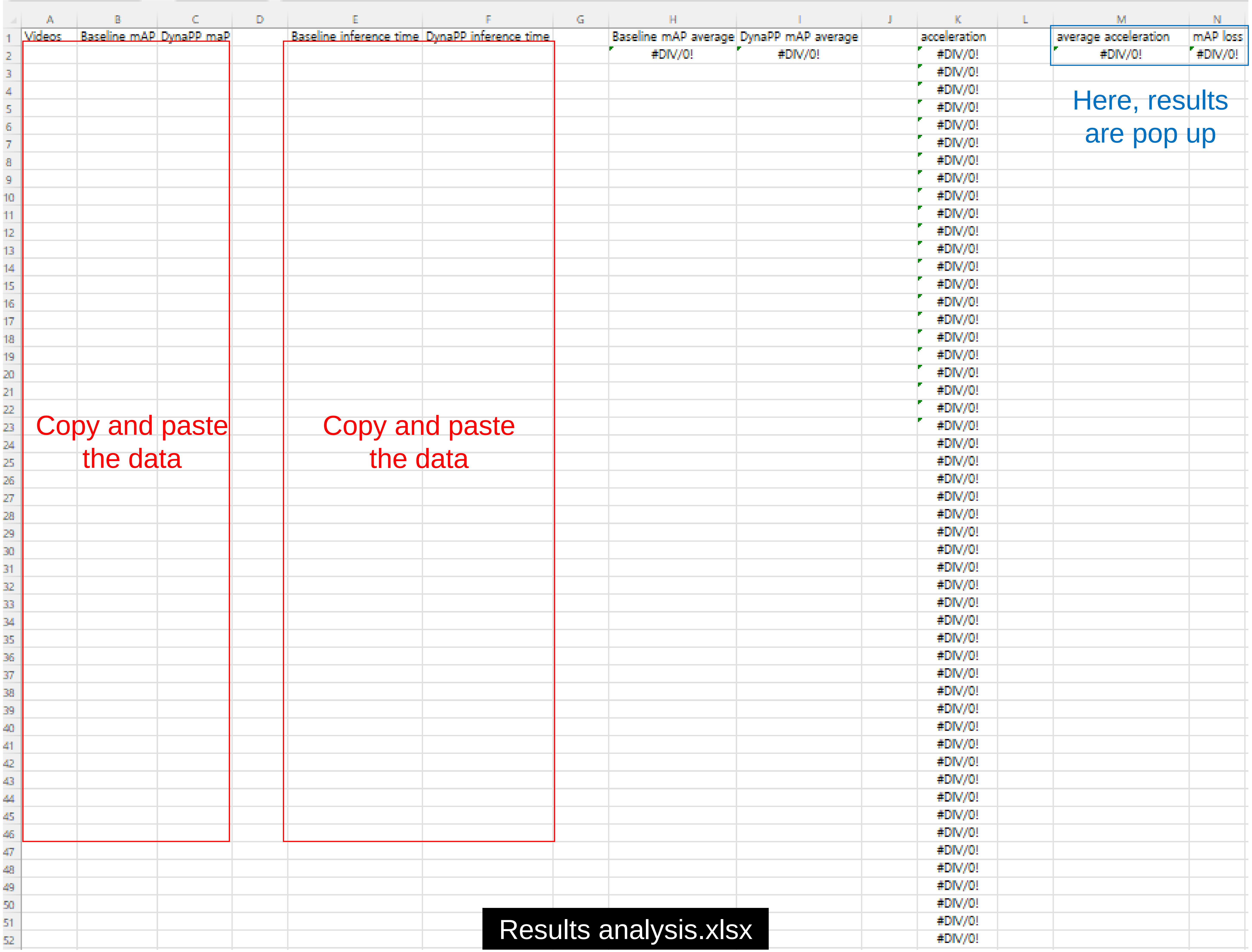
Square Baseline

Pack and Detect

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

    !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

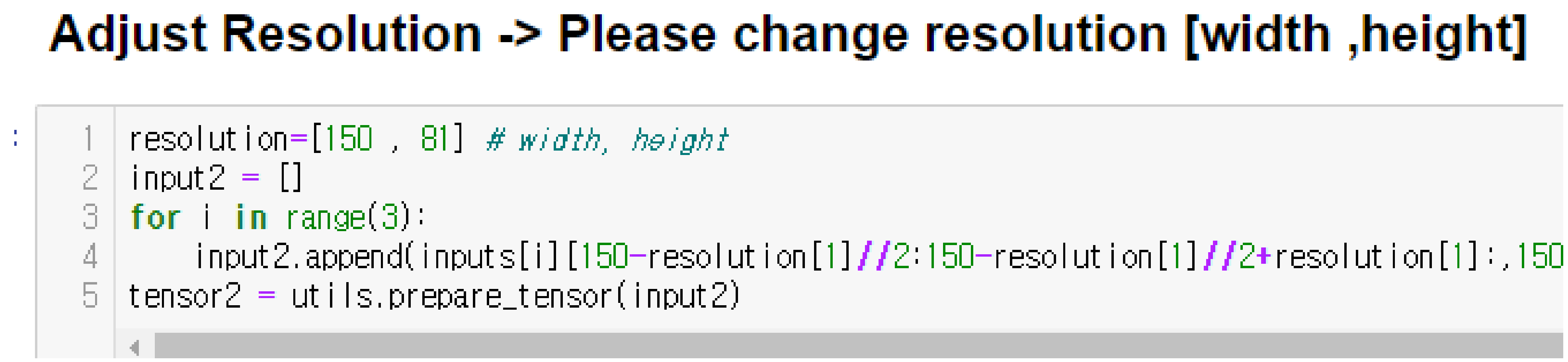
Here, results are pop up

Results analysis.xlsx

SSD to a dynamic resolution model

1. Go to ‘SSD_to_dynamic/Run.ipynb’

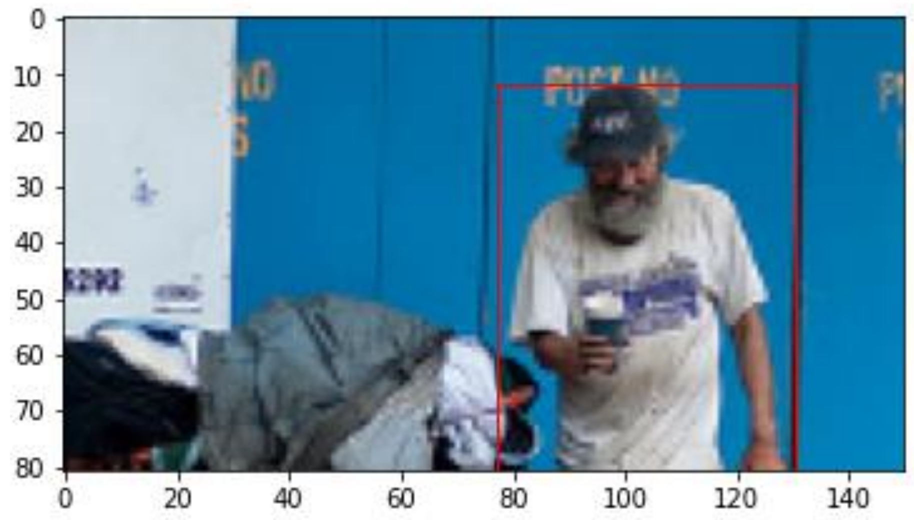
2. Adjust resolution [width, height]



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



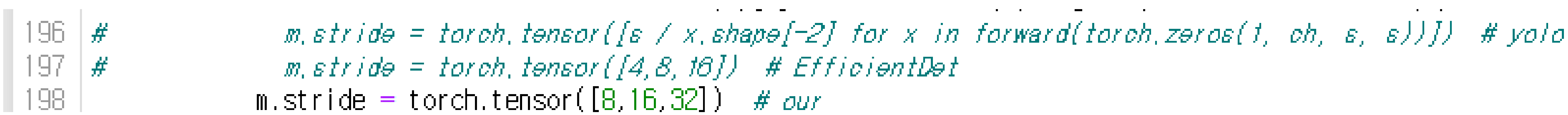
Note that it is best to implement with trained SSD, which AB Distance set as our idea,

but for easy implementation, using existing model provided by Nvidia is fine.

Ablation Study 1: Training

1. Go to line 196-198 in ‘Ablation_Training/models/yolo.py’

2. Change AB distance methods (choose one method)



```
196 #            m.stride = torch.tensor([s / x.shape[-2] for x in forward(torch.zeros(1, ch, s, s))])  # yolo
197 #            m.stride = torch.tensor([4,8,16])  # EfficientDet
198             m.stride = torch.tensor([8,16,32])  # our
```

3. Run ‘Ablation_Training/Run.ipynb’

Ablation Study 2: Dynamic Resolution inference

Grid based anchor boxes

1. Go to line 86 in ‘models/yolo.py’

2. Off the original class Detect and activate line 86-136

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

Manipulating upper left features

1. Go to line 86 in ‘models/common.py’

2. Off the original class Concat and class Focus;

Activate 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 code write

- canvas_DynaPP.py
  : Check all

- canvas_packanddetect.py
  : Check all

- val.py
  : Check line 26-34, 102-105, **118-290**, 307-314, 393-402, 429-455, 522-526

- val_packanddetect.py
  : Check line 26-34, 102, **115-220**, 236-242, 324-332, 359-385, 452-456

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

- models/yolo.py &
  : Check class Detect

- Ablation_Training/models/yolo.py
  : Check line 196-198

- Ablation_Training/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