

## Assignment 4

### Task 1



### จัดทำโดย

นายจตุพัฒน์ ทวีกสิกรรม รหัสนิสิต 62361000

นายเจษฎา น่วมประวัติ รหัสนิสิต 6261321

### เสนอ

อาจารย์รัฐภูมิ วรานุสาสน์

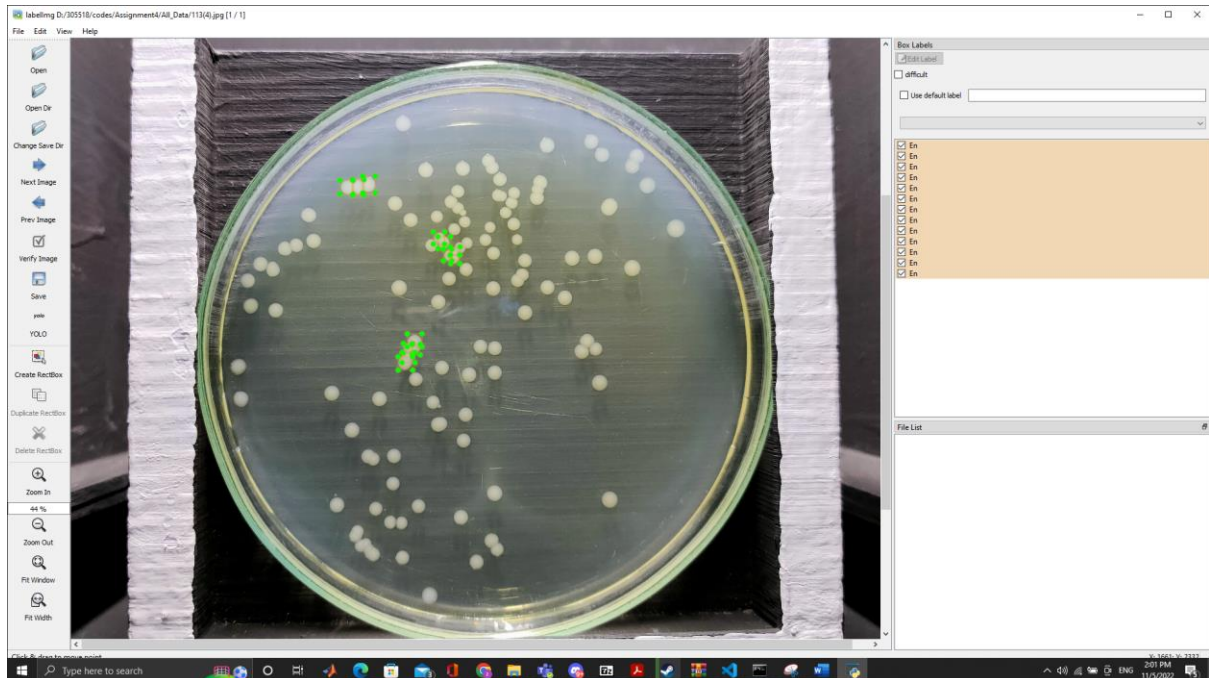
รายงานนี้เป็นส่วนหนึ่งของวิชา Special Topics in Image Processing

รหัสวิชา 305456 สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและ  
คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธนบุรี

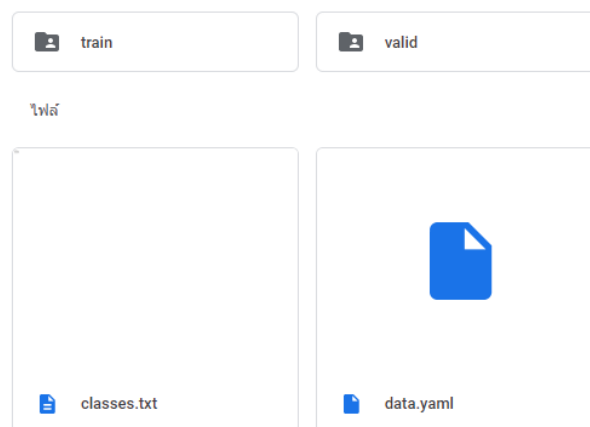
ปีการศึกษา 2565

## Assignment 4 task1: Bacteria colonies counting

1. จัดทำชุดข้อมูลเพื่อนำไปฝึกโมเดลโดยการมอบ label ให้กับวัตถุด้วย library ที่มีชื่อว่า labelling



















































2. จัดหมวดหมู่ชุดข้อมูลโดยให้ train เก็บไฟล์รูปภาพไว้ใช้ในการฝึก ให้ valid เก็บไฟล์เพื่อใช้ในการทดสอบ ให้ classes.txt บันทึกชื่อของคลาส และ data.yaml เป็นเหมือนสารบัญของข้อมูลเหล่านี้



En

```
1 train: ../split_En/train
2 val: ../split_En/valid
3
4 nc: 1
5 names: ['En']
```

ชื่อ ↑	เจ้าของ	แก้ไขล่าสุด	ขนาดไฟล์
 9(1).jpg 	ฉัน	18 ต.ค. 2022 ฉัน	2.6 MB
 9(1).txt 	ฉัน	28 ต.ค. 2022 ฉัน	380 ไบต์
 9(2).jpg 	ฉัน	18 ต.ค. 2022 ฉัน	2.6 MB
 9(2).txt 	ฉัน	28 ต.ค. 2022 ฉัน	342 ไบต์
 9(3).jpg 	ฉัน	18 ต.ค. 2022 ฉัน	2.4 MB
 9(3).txt 	ฉัน	28 ต.ค. 2022 ฉัน	380 ไบต์
 9(4).jpg 	ฉัน	18 ต.ค. 2022 ฉัน	2.6 MB
 9(4).txt 	ฉัน	28 ต.ค. 2022 ฉัน	380 ไบต์
 9(5).jpg 	ฉัน	18 ต.ค. 2022 ฉัน	2.7 MB
 9(5).txt 	ฉัน	28 ต.ค. 2022 ฉัน	380 ไบต์
 14(1).jpg 	ฉัน	18 ต.ค. 2022 ฉัน	2.6 MB

ชื่อ ↑	เจ้าของ	แก้ไขล่าสุด	ขนาดไฟล์
 9(6).jpg 	ฉัน	18 ต.ค. 2022 ฉัน	2.5 MB
 9(6).txt 	ฉัน	27 ต.ค. 2022 ฉัน	342 ไบต์
 14(6).jpg 	ฉัน	18 ต.ค. 2022 ฉัน	3.2 MB
 14(6).txt 	ฉัน	27 ต.ค. 2022 ฉัน	494 ไบต์
 21(3).jpg 	ฉัน	18 ต.ค. 2022 ฉัน	1.9 MB
 21(3).txt 	ฉัน	27 ต.ค. 2022 ฉัน	798 ไบต์
 38(6).jpg 	ฉัน	18 ต.ค. 2022 ฉัน	1.8 MB
 38(6).txt 	ฉัน	29 ต.ค. 2022 ฉัน	1 KB
 51(6).jpg 	ฉัน	18 ต.ค. 2022 ฉัน	3.1 MB
 51(6).txt 	ฉัน	29 ต.ค. 2022 ฉัน	2 KB
 58(6).jpg 	ฉัน	18 ต.ค. 2022 ฉัน	3.1 MB
 58(6).txt 	ฉัน	1 พ.ย. 2022 ฉัน	2 KB
 59(6).jpg 	ฉัน	18 ต.ค. 2022 ฉัน	1.9 MB

#### 4. ฝึกโมเดลด้วย yolo7

```
%cd /content/drive/MyDrive/YOLO_Training/yolov7

/content/drive/.shortcut-targets-by-id/10a1BpUP2EqZI9wCR100onwCIIdH7m6ucd/YOLO_Training/yolov7

!python train.py --data ../split_En/data.yaml --epoch 300 --hyp /content/drive/MyDrive/YOLO_Training/yolov7/data/hyp.

val: Scanning '../split_En/valid.cache' images and labels... 12 found, 0 missing, 0 empty, 0 corrupted: 100% 12/12 [00:00<?, ?it/s]

autoanchor: Analyzing anchors... anchors/target = 1.87, Best Possible Recall (BPR) = 0.9949
Image sizes 640 train, 640 test
Using 2 dataloader workers
Logging results to runs/train/exp6
Starting training for 300 epochs...

Epoch 0/299  gpu_mem 7.86G  box 0.05108  obj 0.02472  cls 0  total 0.0758  labels 514  img_size 640: 100% 4/4 [01:06<00:00, 16.61s/it]
              Class Images Labels P R mAP@.5 mAP@.5:.95: 0% 0/1 [00:00<?, ?it/s]/usr/local/lib/
return _VF.meshgrid(tensors, **kwargs) # type: ignore[attr-defined]
              Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 1/1 [00:04<00:00, 4.30s/it]
              all 12 661 0.00287 0.00303 1.69e-05 2.62e-06

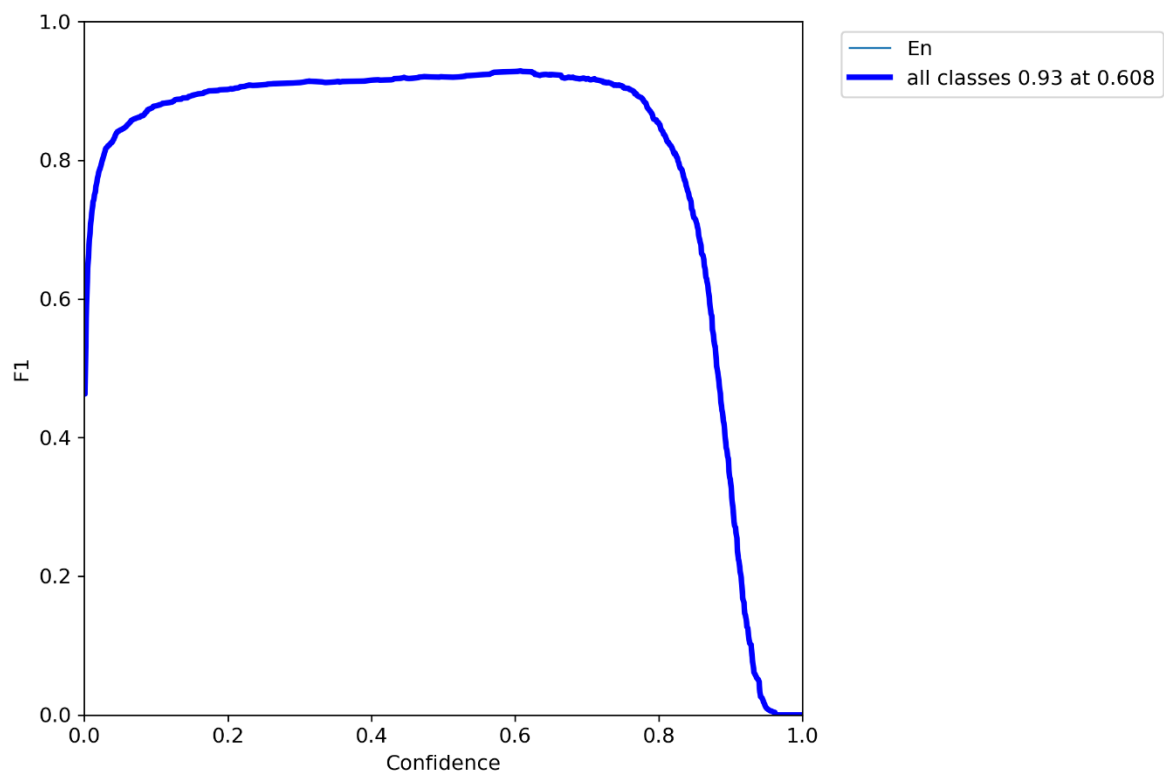
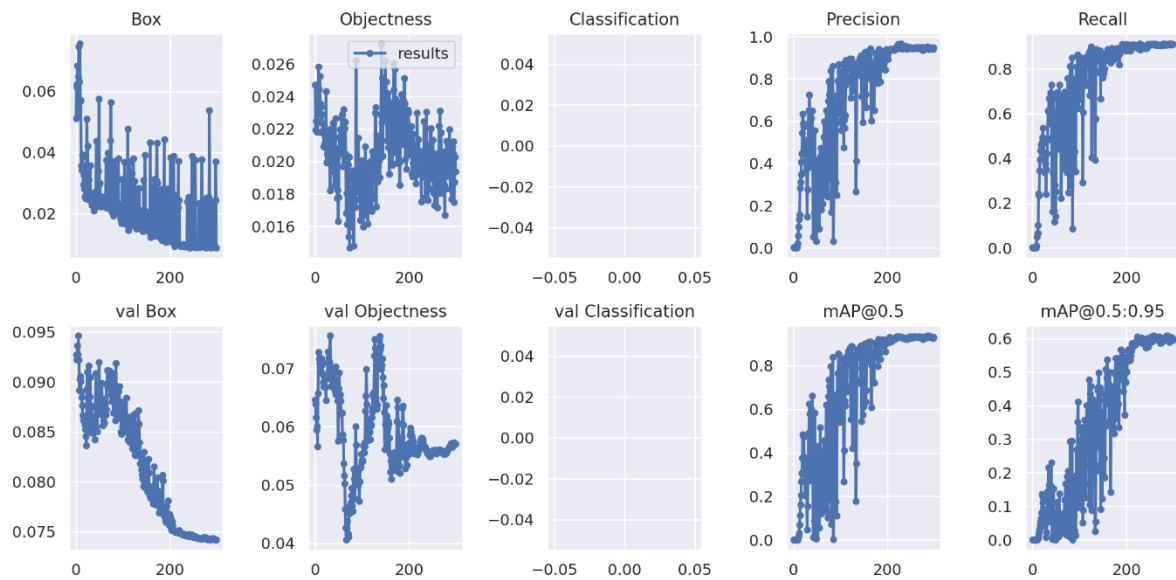
Epoch 1/299  gpu_mem 10.8G  box 0.06226  obj 0.02195  cls 0  total 0.08421  labels 850  img_size 640: 100% 4/4 [00:34<00:00, 8.75s/it]
              Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00, 1.20it/s]
              all 12 0 0 0 0 0

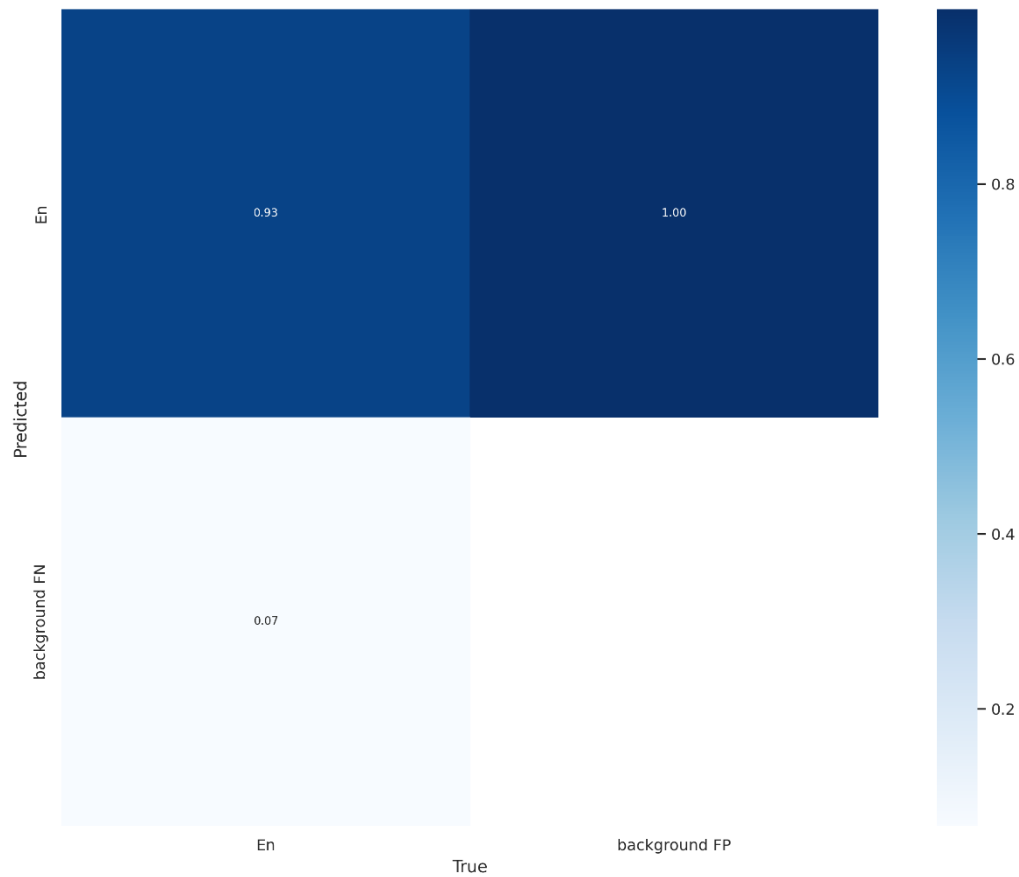
Epoch 2/299  gpu_mem 10.8G  box 0.06146  obj 0.02249  cls 0  total 0.08396  labels 509  img_size 640: 100% 4/4 [00:41<00:00, 10.40s/it]
              Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00, 1.47it/s]
              all 12 0 0 0 0 0
```

## 5. นำไฟล์ที่ได้หลังจากฝึกไปใช้

Optimizer stripped from runs/train/exp6/weights/best.pt, 74.8MB

## 6. การประเมินผลการฝึก





7. เขียนโปรแกรมเพื่อดึงค่าจากไฟล์ที่ได้จากข้อที่ 5 ในที่นี้ชื่อ En\_count.py

```
import argparse
import time
from pathlib import Path

import numpy
import cv2
import torch
import torch.backends.cudnn as cudnn
from numpy import random

from models.experimental import attempt_load
```

```

from utils.datasets import LoadStreams, LoadImages

from utils.general import check_img_size, non_max_suppression, \
    scale_coords

from utils.plots import plot_one_box

from utils.torch_utils import time_synchronized, TracedModel


def detect(source):

    weights, view_img, imgsz, trace = opt.weights, opt.view_img, opt.img_size, not opt.no_trace

    predicted_val_list = []

    Observed_val_list = []

    # Initialize

    device = torch.device('cpu')

    half = device.type != 'cpu' # half precision only supported on CUDA

    print(device.type)


    # Load model

    model = attempt_load(weights, map_location=device) # load FP32 model

    stride = int(model.stride.max()) # model stride

    imgsz = check_img_size(imgsz, s=stride) # check img_size


    if trace:

        model = TracedModel(model, device, imgsz)


    # Set Dataloader

    dataset = LoadImages(source, img_size=imgsz, stride=stride)


    # Get names and colors

```

```

names = model.module.names if hasattr(model, 'module') else model.names

colors = [[random.randint(0, 255) for _ in range(3)] for _ in names]

# Run inference

old_img_w = old_img_h = imgsz

old_img_b = 1

t0 = time.time()

for path, img, im0s, vid_cap in dataset:

    img = torch.from_numpy(img).to(device)

    img = img.half() if half else img.float() # uint8 to fp16/32

    img /= 255.0 # 0 - 255 to 0.0 - 1.0

    if img.ndimension() == 3:

        img = img.unsqueeze(0)

# Warmup

for i in range(3):

    model(img, augment=False)[0]

# Inference

t1 = time_synchronized()

with torch.no_grad(): # Calculating gradients would cause a GPU memory leak

    pred = model(img, augment=False)[0]

t2 = time_synchronized()

# Apply NMS

```



```

pred = non_max_suppression(pred, opt.conf_thres, opt.iou_thres , classes=None, agnostic=False)

t3 = time_synchronized()

# Process detections

for i, det in enumerate(pred): # detections per image

    p, s, im0, frame = path, ", im0s, getattr(dataset, 'frame', 0)

    p = Path(p) # to Path
    p_list = str(p).split("\\")
    p_list.reverse()
    True_label = p_list[0].split('/')
    True_Colonies = int(True_label[0])

    print("Name = {}".format(True_Colonies))

    Observed_val_list.append(True_Colonies)

    gn = torch.tensor(im0.shape)[[1, 0, 1, 0]] # normalization gain whwh

    if len(det):

        # Rescale boxes from img_size to im0 size

        det[:, :4] = scale_coords(img.shape[2:], det[:, :4], im0.shape).round()

        # Print results

        for c in det[:, -1].unique():

            n = (det[:, -1] == c).sum() # detections per class

            s += f"{n} {names[int(c)]}'s' * (n > 1)}, " # add to string

        predicted_val_list.append(int(n))

    # Draw_Box

    for *xyxy, conf, cls in reversed(det):

```

```

        if view_img: # Add bbox to image

            label = f'{names[int(cls)]} {conf:.2f}'

            plot_one_box(xyxy, im0, label=label, color=colors[int(cls)], line_thickness=1)

    # Print time (inference + NMS)

    print(f'{s}Done. ({(1E3 * (t2 - t1)):.1f}ms) Inference, ({(1E3 * (t3 - t2)):.1f}ms) NMS')

    # Stream results

    if view_img:

        im0 = cv2.resize(im0,(int(old_img_h*1.5),int(old_img_w*1.5)),interpolation= cv2.INTER_AREA)

        cv2.putText(im0, '{0} colonies'.format(n), (50,50), cv2.FONT_HERSHEY_DUPLEX, 1.0, (125, 246,
55), 1, cv2.LINE_AA)

        cv2.imshow(str(p), im0)

        cv2.waitKey(0) # 1 millisecond

    if opt.regression :

        return predicted_val_list ,Observed_val_list

if __name__ == '__main__':

    parser = argparse.ArgumentParser()

    parser.add_argument('--weights', nargs='+', type=str, default='../En_model/best.pt', help='model.pt
path(s)')

    parser.add_argument('--source', type=str, default='../All_Data/', help='source') # file/folder, 0 for
webcam

    parser.add_argument('--img-size', type=int, default=640, help='inference size (pixels)')

    parser.add_argument('--conf-thres', type=float, default=0.5, help='object confidence
threshold(default = 0.5)')

```

```

parser.add_argument('--iou-thres', type=float, default=0.45, help='IOU threshold for NMS(default = 0.45)')

parser.add_argument('--regression', action='store_true', help='display regression')

parser.add_argument('--view-img', action='store_true', help='display results')

parser.add_argument('--no-trace', action='store_true', help='don't trace model')

opt = parser.parse_args()

print(opt)

#check_requirements(exclude=('pycocotools', 'thop'))

with torch.no_grad():

    if not opt.regression :

        detect(opt.source)

    if opt.regression :

        source= "../All_Data/"

        predicted_val_list ,Observed_val_list = detect(source)

        predicted_array = numpy.array(predicted_val_list)

        Observed_array = numpy.array(Observed_val_list)

        error_array = predicted_array - Observed_array

        sum_of_error_pow2 = sum(error_array**2)

        number_of_value = len(error_array)

        rmse = (sum_of_error_pow2/number_of_value)**0.5

        print("predicted = {}".format(predicted_array))

        print("Observed = {}".format(Observed_array))

        print("error = {}".format(error_array))

        print("RMSE tested by {0} = {1}".format(source,rmse))

        Observed_mean =sum(Observed_array)/number_of_value

        predicted_mean =sum(predicted_array)/number_of_value

```

```

factor = ((sum((predicted_array - predicted_mean)**2)) * (sum((Observed_array -
Observed_mean)**2)))**0.5

correl = sum((predicted_array - predicted_mean) * (Observed_array - Observed_mean)) / factor

print("Correlation tested by {0} = {1}".format(source, correl))

print("MAE =", sum(abs(error_array)) / number_of_value)

```

## 8. ใช้งานโปรแกรม

```

(A4venv3_9) D:\305518\codes\Assignment4\yolov7>python En_count.py --help
usage: En_count.py [-h] [--weights WEIGHTS [WEIGHTS ...]] [--source SOURCE] [--img-size IMG_SIZE]
                  [--conf-thres CONF_THRES] [--iou-thres IOU_THRES] [--regression] [--view-img] [--no-trace]

optional arguments:
  -h, --help            show this help message and exit
  --weights WEIGHTS [WEIGHTS ...]
                        model.pt path(s)
  --source SOURCE        source
  --img-size IMG_SIZE    inference size (pixels)
  --conf-thres CONF_THRES
                        object confidence threshold(default = 0.5)
  --iou-thres IOU_THRES
                        IOU threshold for NMS(default = 0.45)
  --regression            display regression
  --view-img             display results
  --no-trace             don't trace model

```

```

D:\305518\codes\Assignment4\A4venv3_9\lib\site-packages\torch\functional.py:504
UserWarning: torch.meshgrid: in an upcoming release, it will be required to p
ass the indexing argument. (Triggered internally at C:\actions-runner\work\pyt
orch\pytorch\builder\windows\pytorch\aten\src\ATen\native\TensorShape.cpp:3191.
)
return VF.meshgrid(tensors, **kwargs) # type: ignore[attr-defined]
Convert model to Traced-model...
traced script module saved!
model is traced!

Name = 113
106 Ens, Done. (717.1ms) Inference, (27.2ms) NMS
Name = 113
106 Ens, Done. (685.3ms) Inference, (1.0ms) NMS
Name = 113
107 Ens, Done. (875.2ms) Inference, (1.0ms) NMS
Name = 113
106 Ens, Done. (835.7ms) Inference, (1.0ms) NMS
Name = 113
109 Ens, Done. (705.7ms) Inference, (1.0ms) NMS
Name = 113
106 Ens, Done. (816.2ms) Inference, (7.0ms) NMS
Name = 116
105 Ens, Done. (839.1ms) Inference, (0.0ms) NMS
Name = 116
104 Ens, Done. (904.4ms) Inference, (1.0ms) NMS
Name = 116
103 Ens, Done. (915.5ms) Inference, (2.0ms) NMS
Name = 116
101 Ens, Done. (1012.6ms) Inference, (1.0ms) NMS
Name = 116
108 Ens, Done. (1052.5ms) Inference, (1.0ms) NMS
Name = 116
78 Ens, Done. (847.2ms) Inference, (1.0ms) NMS
Name = 122
121 Ens, Done. (805.0ms) Inference, (1.0ms) NMS
Name = 122
122 Ens, Done. (947.6ms) Inference, (2.1ms) NMS

```

## 9. การประเมินผลการคำนวณกับภาพที่มีอนามิคของแบคทีเรียตั้งแต่ 9 ถึง 265 ตัว

```
predicted = [106 106 107 106 109 106 105 104 103 101 108 78 121 122 123 122 12
2 109 14 14 14 14 14 13 21 20 21 205 205 204 199 201 208 211 212 215 2
15 211 199 38 38 38 38 39 39 48 47 47 49 48 45 56 57 57 57 55
53 53 53 53 52 53 55 60 61 60 61 60 66 64 63 63 63 64 64 76
76 75 75
75 73 83 83 83 84 85 10 10 10 10 10 10]
Observed = [113 113 113 113 113 113 116 116 116 116 116 116 122 122 122 122 12
2 122 14 14 14 14 14 14 21 21 21 261 261 261 261 261 261 265 265 265 2
65 265 265 38 38 38 38 38 38 38 51 51 51 51 51 51 58 58 58 58 58
58 59 59 59 59 59 59 62 62 62 62 62 62 62 66 66 66 66 66 66 77
77 77 77
77 77 84 84 84 84 84 84 9 9 9 9 9 9]
error = [ -7 -7 -6 -7 -4 -7 -11 -12 -13 -15 -8 -38 -1 0 1 0 0 -
13 0 0 0 0 0 -1 0 -1 0 -56 -56 -57 -62 -60 -53 -54 -53 -50 -50
-54 -66 0 0 0 0 1 1 -3 -4 -4 -2 -3 -6 -2 -1 -1 -1 -3 -5
-6 -6 -6 -7 -6 -4 -2 -1 -2 -1 -2 4 -2 -3 -3 -3 -2 -2 -1 -
1 -2 -2
-2 -4 -1 -1 -1 -1 0 1 1 1 1 1 1 1]
RMSE tested by ../All_Data/ = 20.94822085498545
Correlation tested by ../All_Data/ = 0.9918886223046026
MAE = 10.161290322580646
```