Hand Landmark Detection Experiments

Aug 16th~18th, 2023

To. professor

Index_

1. Introduction of dataset for experiment

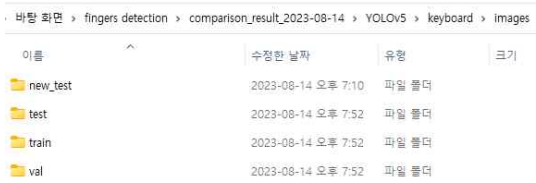
1. Introduction of dataset for experiment

- There are 2,605 images from divided in 5 technique related videos.
- First, I split dataset 8:1:1 for train, test, and val respectively using “train_val_test_split.py” founded by searching on GitHub.

```
!wget https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/master/util_scripts/train_val_test_split.py
!python train_val_test_split.py
```

```
--2023-08-09 09:33:26--  https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/master/util_scripts/train_val_test_split.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2803 (2.7K) [text/plain]
Saving to: 'train_val_test_split.py'

train_val_test_spli 100%[===================>]   2.74K  --.-KB/s    in 0s

2023-08-09 09:33:26 (26.5 MB/s) - 'train_val_test_split.py' saved [2803/2803]

Total images: 2605
Images moving to train: 2084
Images moving to validation: 260
Images moving to test: 261
```

- Next, I downloaded these data files in my computer. Then, these are moved to YOLOv5 folder so that train the model for same data.



2. Evaluation Metric for Object Detection – (YOLOv5, SSD–MobileNet)

2. Guidance on experimental procedures

- **YOLOv5**
  - I obtained an evaluation metric for object detection according to IoU threshold using val.py provided.
  - Used IoU threshold are 0.5, 0.6, 0.7, 0.8, 0.9 and 0.95.

- **SSD-MobileNet**
- I obtained an evaluation metric for object detection according to IoU threshold using calculate_map_cartucho.py founded by searching on GitHub.
  - Used IoU threshold are from 0.5 to 0.95 in increments of 0.05.

2. Evaluation Metric for Object Detection – YOLOv5

2. Evaluation Metric for Object Detection – YOLOv5 ~ IoU threshold: 0.5

- The image below is the command to get YOLOv5 model's evalutaion metric in cmd.

```
(base) C:\Users\User\Desktop\fingers detection\comparison_result_2023-08-14\YOLOv5>python ./yolov
5/val.py --task "test" --data ./test.yaml --weights ./yolov5/runs/train/piano_yolov5s_results/wei
ghts/best.pt --device 0 --save-conf --iou-thres 0.5
val: data=./test.yaml, weights=['./yolov5/runs/train/piano_yolov5s_results/weights/best.pt'], bat
ch_size=32, imgsz=640, conf_thres=0.001, iou_thres=0.5, max_det=300, task=test, device=0, workers
=8, single_cls=False, augment=False, verbose=False, save_txt=False, save_hybrid=False, save_conf=
True, save_json=False, project=yolov5\runs\val, name=exp, exist_ok=False, half=False, dnn=False
fatal: cannot change to 'C:\Users\User\Desktop\fingers': No such file or directory
YOLOv5  2023-8-3 Python-3.10.11 torch-2.0.1 CUDA:0 (NVIDIA GeForce RTX 4080, 16376MiB)
```

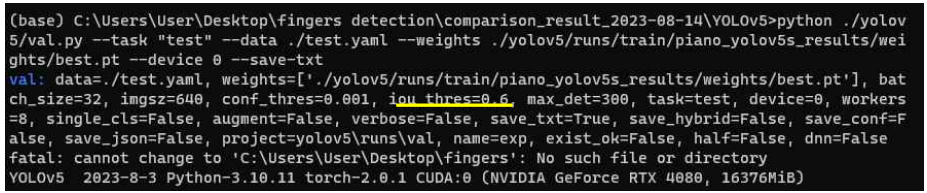
- The image below is the result of the YOLOv5 model's evalutaion metric.

```
YOLOv5s summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
test: Scanning C:\Users\User\Desktop\fingers detection\comparison_result_2023-08-14\YOLOv5\keybo
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|
                   all        261        261          1          1      0.995      0.819
Speed: 0.2ms pre-process, 3.6ms inference, 1.8ms NMS per image at shape (32, 3, 640, 640)
```

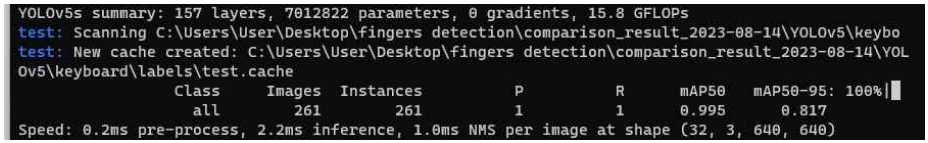
- **Precision: 1**
- **Recall: 1**
- **mAP50: 0.995**
- **mAP50-95: 0.819**

2. Evaluation Metric for Object Detection – YOLOv5 ~ IoU threshold: 0.6

- The image below is the command to get YOLOv5 model's evalutaion metric in cmd.



- The image below is the result of the YOLOv5 model's evalutaion metric.



- **Precision: 1**
- **Recall: 1**
- **mAP50: 0.995**
- **mAP50-95: 0.817**

2. Evaluation Metric for Object Detection – YOLOv5 ~ IoU threshold: 0.7

- The image below is the command to get YOLOv5 model's evalutaion metric in cmd.

```
(base) C:\Users\User\Desktop\fingers detection\comparison_result_2023-08-14\YOLOv5>python ./yolov
5/val.py --task "test" --data ./test.yaml --weights ./yolov5/runs/train/piano_yolov5s_results/wei
ghts/best.pt --device 0 --save-conf --iou-thres 0.7
val: data=./test.yaml, weights=['./yolov5/runs/train/piano_yolov5s_results/weights/best.pt'], bat
ch_size=32, imgsz=640, conf_thres=0.001, iou_thres=0.7, max_det=300, task=test, device=0, workers
=8, single_cls=False, augment=False, verbose=False, save_txt=False, save_hybrid=False, save_conf=
True, save_json=False, project=yolov5\runs\val, name=exp, exist_ok=False, half=False, dnn=False
fatal: cannot change to 'C:\Users\User\Desktop\fingers': No such file or directory
YOLOv5  2023-8-3 Python-3.10.11 torch-2.0.1 CUDA:0 (NVIDIA GeForce RTX 4080, 16376MiB)
```

- The image below is the result of the YOLOv5 model's evalutaion metric.

```
YOLOv5s summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
test: Scanning C:\Users\User\Desktop\fingers detection\comparison_result_2023-08-14\YOLOv5\keybo
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|
                   all        261        261          1          1      0.995      0.814
Speed: 0.2ms pre-process, 3.6ms inference, 2.1ms NMS per image at shape (32, 3, 640, 640)
```

- **Precision: 1**
- **Recall: 1**
- **mAP50: 0.995**
- **mAP50-95: 0.814**

2. Evaluation Metric for Object Detection – YOLOv5 ~ IoU threshold: 0.8

- The image below is the command to get YOLOv5 model's evalutaion metric in cmd.

```
(base) C:\Users\User\Desktop\fingers detection\comparison_result_2023-08-14\YOLOv5>python ./yolov
5/val.py --task "test" --data ./test.yaml --weights ./yolov5/runs/train/piano_yolov5s_results/wei
ghts/best.pt --device 0 --save-conf --iou-thres 0.8
val: data=./test.yaml, weights=['./yolov5/runs/train/piano_yolov5s_results/weights/best.pt'], bat
ch_size=32, imgsz=640, conf_thres=0.001, iou_thres=0.8, max_det=300, task=test, device=0, workers
=8, single_cls=False, augment=False, verbose=False, save_txt=False, save_hybrid=False, save_conf=
True, save_json=False, project=yolov5\runs\val, name=exp, exist_ok=False, half=False, dnn=False
fatal: cannot change to 'C:\Users\User\Desktop\fingers': No such file or directory
YOLOv5  2023-8-3 Python-3.10.11 torch-2.0.1 CUDA:0 (NVIDIA GeForce RTX 4080, 16376MiB)
```

- The image below is the result of the YOLOv5 model's evalutaion metric.

```
YOLOv5s summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
test: Scanning C:\Users\User\Desktop\fingers detection\comparison_result_2023-08-14\YOLOv5\keybo
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|
                   all        261        261          1          1      0.995      0.811
Speed: 0.2ms pre-process, 3.7ms inference, 1.7ms NMS per image at shape (32, 3, 640, 640)
```

- **Precision: 1**
- **Recall: 1**
- **mAP50: 0.995**
- **mAP50-95: 0.811**

2. Evaluation Metric for Object Detection – YOLOv5 ~ IoU threshold: 0.9

- The image below is the command to get YOLOv5 model's evalutaion metric in cmd.

```
(base) C:\Users\User\Desktop\fingers detection\comparison_result_2023-08-14\YOLOv5>python ./yolov
5/val.py --task "test" --data ./test.yaml --weights ./yolov5/runs/train/piano_yolov5s_results/wei
ghts/best.pt --device 0 --save-conf --iou-thres 0.9
val: data=./test.yaml, weights=['./yolov5/runs/train/piano_yolov5s_results/weights/best.pt'], bat
ch_size=32, imgsz=640, conf_thres=0.001, iou_thres=0.9, max_det=300, task=test, device=0, workers
=8, single_cls=False, augment=False, verbose=False, save_txt=False, save_hybrid=False, save_conf=
True, save_json=False, project=yolov5\runs\val, name=exp, exist_ok=False, half=False, dnn=False
fatal: cannot change to 'C:\Users\User\Desktop\fingers': No such file or directory
YOLOv5  2023-8-3 Python-3.10.11 torch-2.0.1 CUDA:0 (NVIDIA GeForce RTX 4080, 16376MiB)
```

- The image below is the result of the YOLOv5 model's evalutaion metric.

```
YOLOv5s summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
test: Scanning C:\Users\User\Desktop\fingers detection\comparison_result_2023-08-14\YOLOv5\keybo
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|
                   all        261        261      0.995          1      0.995      0.813
Speed: 0.2ms pre-process, 3.9ms inference, 2.0ms NMS per image at shape (32, 3, 640, 640)
```

- **Precision:** 0.995
- **Recall:** 1
- **mAP50:** 0.995
- **mAP50-95:** 0.813

2. Evaluation Metric for Object Detection – YOLOv5 ~ IoU threshold: 0.95

- The image below is the command to get YOLOv5 model's evalutaion metric in cmd.

```
(base) C:\Users\User\Desktop\fingers detection\comparison_result_2023-08-14\YOLOv5>python ./yolov
5/val.py --task "test" --data ./test.yaml --weights ./yolov5/runs/train/piano_yolov5s_results/wei
ghts/best.pt --device 0 --save-conf --iou-thres 0.95
val: data=./test.yaml, weights=['./yolov5/runs/train/piano_yolov5s_results/weights/best.pt'], bat
ch_size=32, imgsz=640, conf_thres=0.001, iou_thres=0.95, max_det=300, task=test, device=0, worker
s=8, single_cls=False, augment=False, verbose=False, save_txt=False, save_hybrid=False, save_conf
=True, save_json=False, project=yolov5\runs\val, name=exp, exist_ok=False, half=False, dnn=False
fatal: cannot change to 'C:\Users\User\Desktop\fingers': No such file or directory
YOLOv5  2023-8-3 Python-3.10.11 torch-2.0.1 CUDA:0 (NVIDIA GeForce RTX 4080, 16376MiB)
```

- The image below is the result of the YOLOv5 model's evalutaion metric.

```
YOLOv5s summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
test: Scanning C:\Users\User\Desktop\fingers detection\comparison_result_2023-08-14\YOLOv5\keybo
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|
                   all        261        261      0.951      0.989      0.992       0.81
Speed: 0.2ms pre-process, 3.2ms inference, 1.0ms NMS per image at shape (32, 3, 640, 640)
```

- **Precision:** 0.951
- **Recall:** 0.989
- **mAP50:** 0.992
- **mAP50-95:** 0.81

2. Evaluation Metric for Object Detection
– SSD–MobileNet

2. Evaluation Metric for Object Detection – SSD-MobileNet

- I coded to create a graph about PR curve using numpy, matplotlib, seaborn and scikit-learn.
- Used IoU threshold are 0.5, 0.6, 0.7, 0.8, 0.9 and 0.95.

```
import numpy as np
from sklearn.metrics import auc
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#output_iou_0.50

# Calculate mAP by taking the average of AP values for all classes
precision_values = ['1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.0
recall_values = ['0.00', '0.01', '0.01', '0.02', '0.02', '0.02', '0.03', '0.03',

# 문자열을 실수로 변환
precision_values = [float(value) for value in precision_values]
recall_values = [float(value) for value in recall_values]

precision = np.array(precision_values, dtype=float)
recall = np.array(recall_values, dtype=float)

# Calculate AP using the area under the precision-recall curve
ap = auc(recall, precision)

# Calculate AR by averaging recall values
ar = np.mean(recall)

# Calculate mAP by taking the average of AP values for all classes
mAP = ap

# Set seaborn style
sns.set(style="whitegrid")

# Plot Precision-Recall curve
plt.figure(figsize=(10, 6))
sns.lineplot(x=recall, y=precision, marker='o', color='b')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve (AP = {:.2f}, IoU_0.50)'.format(ap))
plt.grid(True)
plt.show()

print("Average Precision (AP):", ap)
print("Average Recall (AR):", ar)
print("Mean Average Precision (mAP):", mAP)
```

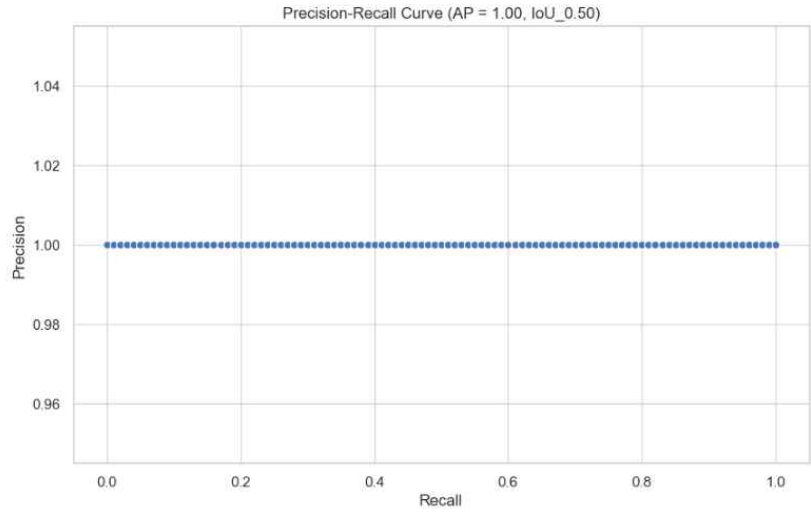
2. Evaluation Metric Summary – SSD-MobileNet

```
Calculating mAP at 0.50 IoU threshold...
100.00% = keyboard AP
mAP = 100.00%
Calculating mAP at 0.55 IoU threshold...
100.00% = keyboard AP
mAP = 100.00%
Calculating mAP at 0.60 IoU threshold...
100.00% = keyboard AP
mAP = 100.00%
Calculating mAP at 0.65 IoU threshold...
99.43% = keyboard AP
mAP = 99.43%
Calculating mAP at 0.70 IoU threshold...
99.01% = keyboard AP
mAP = 99.01%
Calculating mAP at 0.75 IoU threshold...
96.91% = keyboard AP
mAP = 96.91%
```

```
Calculating mAP at 0.80 IoU threshold...
83.74% = keyboard AP
mAP = 83.74%
Calculating mAP at 0.85 IoU threshold...
53.59% = keyboard AP
mAP = 53.59%
Calculating mAP at 0.90 IoU threshold...
16.38% = keyboard AP
mAP = 16.38%
Calculating mAP at 0.95 IoU threshold...
0.48% = keyboard AP
mAP = 0.48%

***mAP Results***

Class            Average mAP @ 0.5:0.95
----------------------------------------
keyboard                 74.95%

Overall          74.95%
```

2. Evaluation Metric for Object Detection – SSD ~ IoU threshold: 0.5

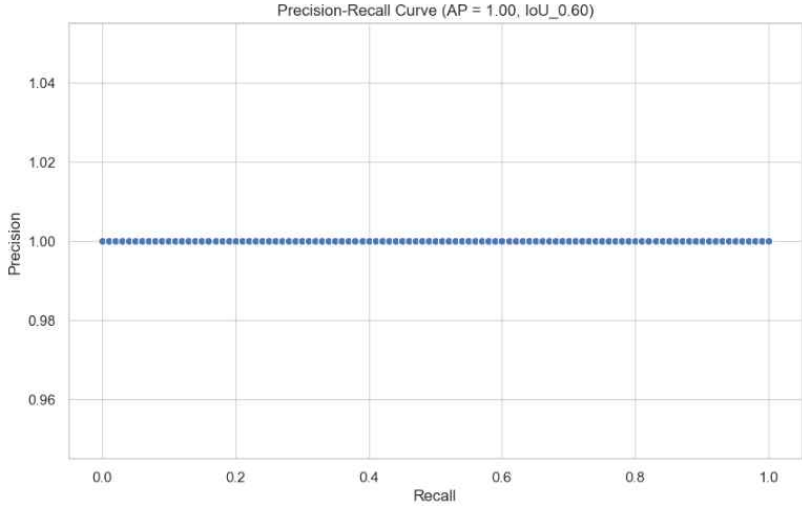
- The image below is the result of the SSD–MobileNet model's evalutaion metric.



Average Precision (AP): 1.0
Average Recall (AR): 0.5019157088122606
Mean Average Precision (mAP): 1.0

2. Evaluation Metric for Object Detection – SSD ~ IoU threshold: 0.6

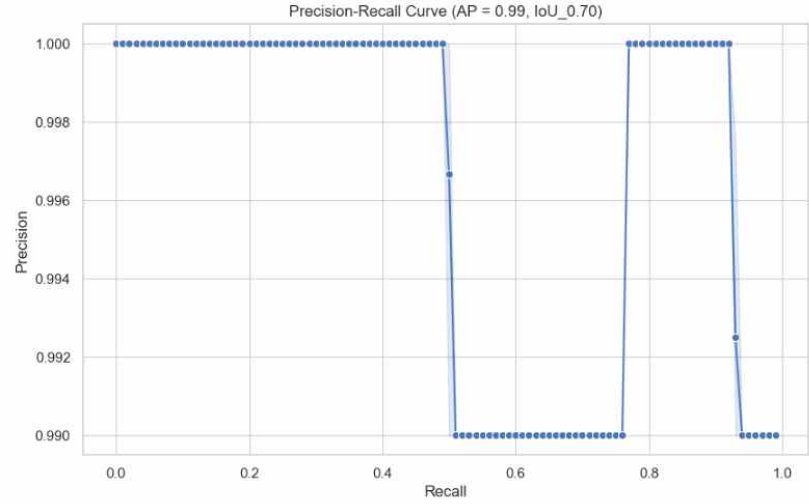
- The image below is the result of the SSD–MobileNet model's evalutaion metric.



Average Precision (AP): 1.0
Average Recall (AR): 0.5019157088122606
Mean Average Precision (mAP): 1.0

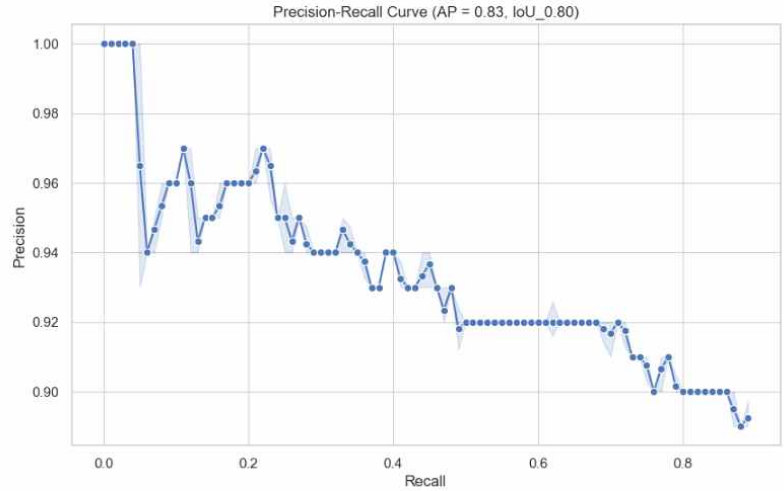
2. Evaluation Metric for Object Detection – SSD ~ IoU threshold: 0.7

- The image below is the result of the SSD–MobileNet model's evalutaion metric.



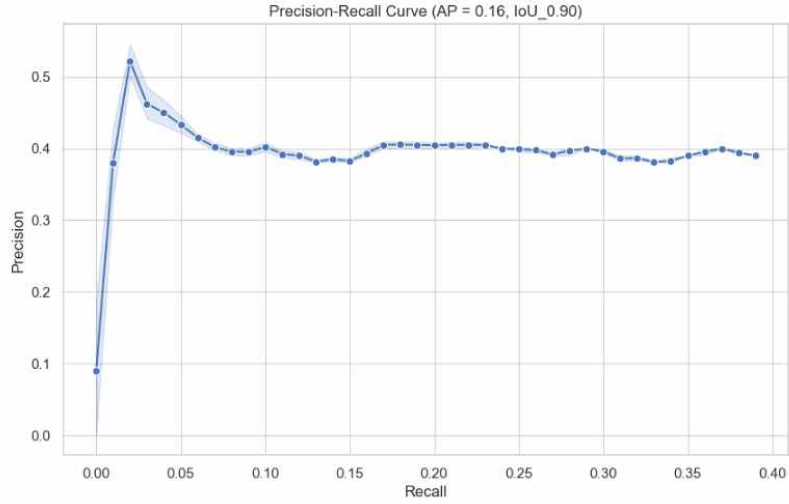
2. Evaluation Metric for Object Detection – SSD ~ IoU threshold: 0.8

- The image below is the result of the SSD–MobileNet model's evalutaion metric.



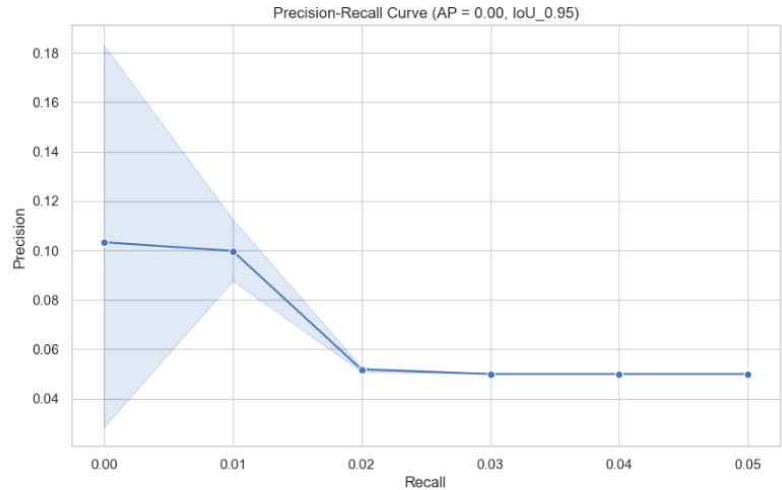
2. Evaluation Metric for Object Detection – SSD ~ IoU threshold: 0.9

- The image below is the result of the SSD–MobileNet model's evalutaion metric.



2. Evaluation Metric for Object Detection – SSD ~ IoU threshold: 0.95

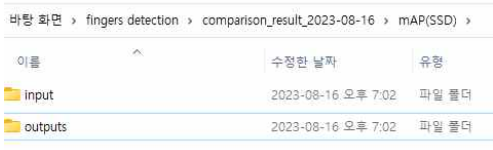
- The image below is the result of the SSD–MobileNet model's evalutaion metric.

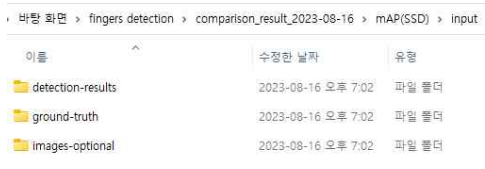


2. Evaluation Metric for Object Detection – SSD–MobileNet

- Files about evalution metric of SSD–MobileNet are attached in this folder with the name ‘mAP(SSD).

- **How to understanding those**





- The images above are directory structures.
  - input
    - detection–result: result files inferenced about test dataset
    - ground–truth: files labeled about test dataset
    - images–optional: images about test dataset
  - outputs: each output files about IoU threshold values

```
!cp /content/images/test/* /content/mAP/input/images-optional # Copy images and xml files
!mv /content/mAP/input/images-optional/*.xml /content/mAP/input/ground-truth/  # Move xml files to the appropriate folder
```

2. Evaluation Metric for Object Detection – SSD–MobileNet

- First, We need to enter the outputs directory.
- Second, open the output.txt file in the output_iou folder.
- This image below is example about IoU threshold 0.5.

```
# AP and precision/recall per class
100.00% = keyboard AP
 Precision: ['1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1
'1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00'
'1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00'
'1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00'
'1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00'
'1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00'
'1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00'
'1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00', '1.00'
'1.00']
 Recall :['0.00', '0.01', '0.01', '0.02', '0.02', '0.02', '0.0
'0.11', '0.11', '0.12', '0.12', '0.13', '0.13', '0.13', '0.14'
'0.22', '0.23', '0.23', '0.23', '0.24', '0.24', '0.25', '0.25'
'0.33', '0.34', '0.34', '0.34', '0.35', '0.35', '0.36', '0.36'
'0.44', '0.45', '0.45', '0.46', '0.46', '0.46', '0.47', '0.47'
'0.56', '0.56', '0.56', '0.57', '0.57', '0.57', '0.58', '0.58'
'0.67', '0.67', '0.67', '0.68', '0.68', '0.69', '0.69', '0.69'
'0.78', '0.78', '0.79', '0.79', '0.79', '0.80', '0.80', '0.80'
'0.89', '0.89', '0.90', '0.90', '0.90', '0.91', '0.91', '0.92'
'1.00']


# mAP of all classes
mAP = 100.00%

# Number of ground-truth objects per class
keyboard: 261

# Number of detected objects per class
keyboard: 261 (tp:261, fp:0)
```

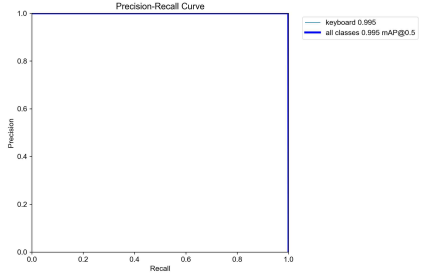
- Precision List: precision values for test dataset in './input/images–optional'
- Recall List: recall values for test dataset in './input/images–optional'
- mAP: mAP value found at each IoU value.

* If you want to understand something additionally, please refer to the 'calculate_map_cartucho.py' and 'main.py' code attached.

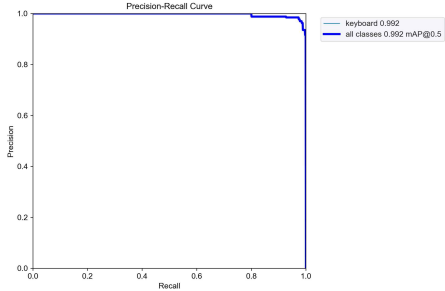
3. Comparison the results (PR–Curve)

3. Comparison the results (PR–Curve)

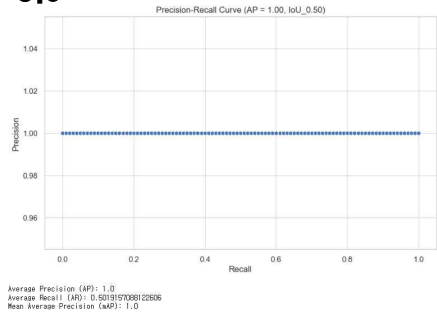
- **YOLOv5 ~ IoU: 0.5**



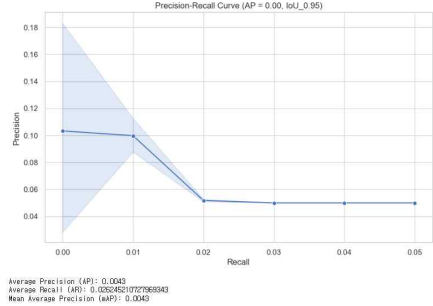
- **YOLOv5 ~ IoU: 0.95**



- **SSD ~ IoU: 0.5**



- **SSD ~ IoU: 0.95**



3. Analysis of evalutaion metric results

- **YOLOv5 & SSD–MobileNet**
  - As explained earlier, I collected 2,605 images from divided in 5 technique related videos. Then I trained the model using these.
  - Although I split dataset for train, test, and val respectively, For example, elements such as the background are too similar, so I think **overfitting** occurs. Also, Precision and Recall values are too high.

- **YOLOv5 vs SSD–MobileNet**
  - **I think the model trained YOLOv5 is better than another.**