

YOLOv5 성능 개선

YOLOv5 - GPU 활용

Cuda 사용을 위한 라이브러리 설치

```
# ROCM 5.4.2 (Linux only)
pip install torch==2.0.0+rocm5.4.2 torchvision==0.15.1+rocm5.4.2 torchaudio==2.0.1 --index-url https://download.pytorch.org/whl/rocm5.4.2
# CUDA 11.7
pip install torch==2.0.0+cu117 torchvision==0.15.1+cu117 torchaudio==2.0.1 --index-url https://download.pytorch.org/whl/cu117
# CUDA 11.8
pip install torch==2.0.0+cu118 torchvision==0.15.1+cu118 torchaudio==2.0.1 --index-url https://download.pytorch.org/whl/cu118
# CPU only
pip install torch==2.0.0+cpu torchvision==0.15.1+cpu torchaudio==2.0.1 --index-url https://download.pytorch.org/whl/cpu
```

모델에 적용

```
yolo_model.to(torch.device('cuda'))
```

YOLOv5 - 모델 선택

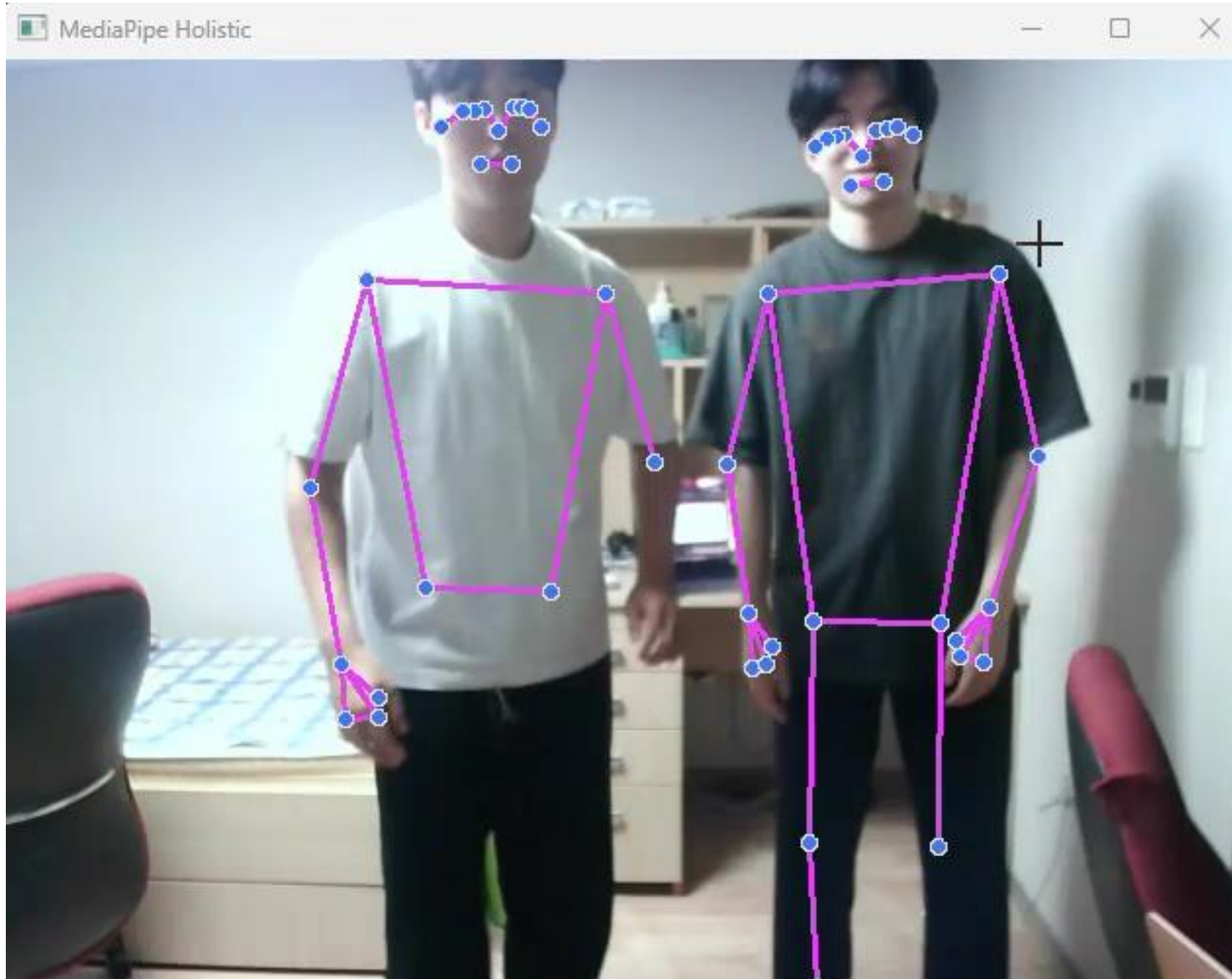
Pretrained Checkpoints								
Model	size (pixels)	mAP ^{val} 50-95	mAP ^{val} 50	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	45	6.3	0.6	1.9	4.5
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7
YOLOv5n6	1280	36.0	54.4	153	8.1	2.1	3.2	4.6
YOLOv5s6	1280	44.8	63.7	385	8.2	3.6	12.6	16.8
YOLOv5m6	1280	51.3	69.3	887	11.1	6.8	35.7	50.0
YOLOv5l6	1280	53.7	71.3	1784	15.8	10.5	76.8	111.4
YOLOv5x6 + TTA	1280 1536	55.0 55.8	72.7 72.7	3136 -	26.2 -	19.4 -	140.7 -	209.8 -

코드 최적화

```
83 while cap.isOpened():
84     # 프레임을 읽기
85     success, image = cap.read()
86
87     # 프레임을 못 읽은 경우 기다리기 (캠 로딩 문제)
88     if not success:
89         continue
90
91     # 이미지를 저장할 빈 리스트 초기화
92     img_list = []
93
94     # 여백 값 설정
95     MARGIN = 10
96
97     image.flags.writeable = False
98     image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
99
100    # YOLOv5 모델 사용해 객체 탐지 수행
101    result = yolo_model(image)
102
103    image.flags.writeable = True
104    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
105
106    # 감지된 객체(사람)들에 대해서 반복
107    for (xmin, ymin, xmax, ymax, confidence, clas) in result.xyxy[0].
tolist():
108
109        # 자세 추정을 위한 mediapipe holistic 객체를 초기화
110        with mp_holistic.Holistic(min_detection_confidence=0.3,
min_tracking_confidence=0.3) as holistic:
111            # 감지된 객체 이미지 부분 처리
112            results = holistic.process(image[int(ymin)+MARGIN:int(ymax)
+MARGIN, int(xmin)+MARGIN:int(xmax)+MARGIN:])
```

```
109 holistic.append(mp_holistic.Holistic(
110     min_detection_confidence=0.5,
111     min_tracking_confidence=0.9,
112     model_complexity=1,
113     smooth_segmentation=False,
114     enable_segmentation=False,
115     refine_face_landmarks=False))
116
117 # 웹캠으로부터 프레임을 처리하는 루프를 시작
118 while cap.isOpened():
119     # 프레임을 읽기
120     success, image = cap.read()
121
122     # 프레임을 못 읽은 경우 기다리기 (캠 로딩 문제)
123     if not success:
124         continue
125
126     # 이미지를 저장할 빈 리스트 초기화
127     img_list = []
128
129     # 여백 값 설정
130     MARGIN = 1
131
132     image.flags.writeable = False
133     image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
134
135     # YOLOv5 모델 사용해 객체 탐지 수행
136     result = yolo_model(image)
137
138     image.flags.writeable = True
139     image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
140
141     # 감지된 객체(사람)들에 대해서 반복
142     part_data = [] # 한 프레임의 랜드마크 모음
143
144     for i in range(len(result.xyxy[0].tolist())):
145         xmin, ymin, xmax, ymax, confidence, clas = result.xyxy[0].tolist
```

개선 전 영상 - Python



개선 후 영상 - Python

