**기말 대체 과제 보고서**

**객체 인식 YOLO 활용 - 마스크 착용 상태 파악**

|  |  |  |
| --- | --- | --- |
| **학 과** |  | **컴퓨터공학과** |
| **학 번** |  | **202121059** |
| **이 름** |  | **정보경** |
| **담당**  **교수님** |  | **백우진 교수님** |



# **개발 환경 설정**

학과 서버 cslab에서 진행하였습니다.

라이브러는 yolov5에서 제공하는 requirements.txt 파일을 사용하였습니다. 주요 라이브러리 버전은 아래와 같습니다.

|  |
| --- |
| 1. matplotlib>=3.3 2. numpy>=1.23.5 3. opencv-python>=4.1.1 4. pillow>=10.3.0 5. PyYAML>=5.3.1 6. requests>=2.32.0 7. scipy>=1.4.1 8. thop>=0.1.1 9. torch>=1.8.0 10. torchvision>=0.9.0 11. tqdm>=4.64.0 12. ultralytics>=8.2.34 13. pandas>=1.1.4 14. seaborn>=0.11.0 |

파이썬은 3.11.5를 사용하였습니다.

# **모델 훈련 과정 및 하이퍼 파라미터 튜닝**

기본 모델 훈련은 배치 사이즈 16에 에폭 100으로 진행하였습니다. 이후 하이퍼 파라미터 튜닝은 총 6번에 걸쳐 batch size, epochs, learning rate를 변화하였습니다.

|  |
| --- |
| 1. batch size 16, epochs 100 2. batch size 16, epochs 300 3. batch size 32, epochs 300 4. batch size 8, epochs 300 5. batch size 16, epochs 300, Learning rate 0.1 6. batch size 16, epochs 300, Learning rate 0.001 7. batch size 16, epochs 300, Learning rate 0.001 ~ 0.01 |

1. 먼저 1번 모델(batch size 16, epochs 100)의 훈련 추이 입니다.

텍스트, 도표, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

그래프를 통해 총 4개, loss, 정밀도, 재현율, [mAP@0.5](mailto:mAP@0.5)를 분석하면 아래와 같습니다.

1. Loss : 값이 모두 감소하고 있는 것을 보아 학습이 잘 진행되고 있음을 알 수 있습니다.

2. 정밀도 : 0.9에 가까운 값으로 모델이 클래스를 잘 구분하고 있음을 알 수 있습니다.

3. 재현율 : 0.8에 가까운 값으로 모델이 클래스를 잘 구분하고 있음을 알 수 있습니다.

4. mAP\_0.5 : 0.8에 가까운 값으로 모델이 전반적으로 높은 예측 성능을 가지고 있음을 알 수 있습니다.

정확도와 재현율을 반영한 수치인 [mAP@0.5](mailto:mAP@0.5)를 기준으로 모델을 평가합니다. 정확한 수치를 위해 Precision-Recall Curve 그래프를 출력하여 확인합니다.

텍스트, 도표, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

- 모델의 성능이 0.81로 높은 수치를 갖고 있어 1번 모델의 성능이 좋다고 판단할 수 있습니다.

다음 2번 모델(batch size 16, epochs 300)의 훈련 추이 입니다.

텍스트, 그래프, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

그래프를 통해 총 4개, loss, 정밀도, 재현율, [mAP@0.5](mailto:mAP@0.5)를 분석하면 아래와 같습니다.

1. Loss : 값이 모두 감소하고 있는 것을 보아 학습이 잘 진행되고 있음을 알 수 있습니다.

2. 정밀도 : 초반에는 변동이 크지만, 0.9에 가까운 값으로 모델이 클래스를 잘 구분하고 있음을 알 수 있습니다.

3. 재현율 : 0.8에 가까운 값으로 모델이 클래스를 잘 구분하고 있음을 알 수 있습니다.

4. mAP\_0.5 : 0.8 이상의 값으로 모델이 전반적으로 높은 예측 성능을 가지고 있음을 알 수 있습니다.

정확도와 재현율을 반영한 수치인 [mAP@0.5](mailto:mAP@0.5)를 기준으로 모델을 평가합니다. 정확한 수치를 위해 Precision-Recall Curve 그래프를 출력하여 확인합니다.

텍스트, 도표, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

모델의 성능이 0.86으로 높은 수치를 갖고 있어 2번 모델의 성능이 좋다고 판단할 수 있습니다. 또한, 1번 모델보다 높은 점수를 갖고 있어 현재 최적의 모델은 2번 모델입니다.

다음 3번 모델(batch size 32, epochs 300)의 훈련 추이 입니다.

텍스트, 그래프, 도표, 스크린샷이(가) 표시된 사진

자동 생성된 설명

그래프를 통해 총 4개, loss, 정밀도, 재현율, [mAP@0.5](mailto:mAP@0.5)를 분석하면 아래와 같습니다.

1. Loss : 값이 모두 감소하고 있는 것을 보아 학습이 잘 진행되고 있음을 알 수 있습니다.

2. 정밀도 : 0.8 이상의 값으로 모델이 클래스를 잘 구분하고 있음을 알 수 있습니다.

3. 재현율 : 0.8에 가까운 값으로 모델이 클래스를 잘 구분하고 있음을 알 수 있습니다.

4. mAP\_0.5 : 0.8에 가까운 값으로 모델이 전반적으로 높은 예측 성능을 가지고 있음을 알 수 있습니다.

정확도와 재현율을 반영한 수치인 [mAP@0.5](mailto:mAP@0.5)를 기준으로 모델을 평가합니다. 정확한 수치를 위해 Precision-Recall Curve 그래프를 출력하여 확인합니다.

텍스트, 도표, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

모델의 성능이 0.849로 높은 수치를 갖고 있어 3번 모델의 성능이 좋다고 판단할 수 있습니다.

그러나, 2번 모델의 수치가 0.86이었기 때문에 현재 최적의 모델은 2번 모델입니다.

다음 4번 모델(batch size 8, epochs 300)의 훈련 추이 입니다.

텍스트, 스크린샷, 도표, 그래프이(가) 표시된 사진

자동 생성된 설명

그래프를 통해 총 4개, loss, 정밀도, 재현율, [mAP@0.5](mailto:mAP@0.5)를 분석하면 아래와 같습니다.

1. Loss : 값이 모두 감소하고 있는 것을 보아 학습이 잘 진행되고 있음을 알 수 있습니다.
2. 정밀도 : 변동을 보이기는 하지만, 0.9에 가까운 값이 되었다가 0.8에 가까운 값으로 끝난 것을 볼 수 있습니다. 그러나 0.8 또한 높은 수치이기 때문에 모델이 클래스를 잘 구분한다고 볼 수 있습니다.
3. 재현율 : 0.8에 가까운 값으로 모델이 클래스를 잘 구분하고 있음을 알 수 있습니다.
4. mAP\_0.5 : 0.8 이상의 값으로 모델이 전반적으로 높은 예측 성능을 가지고 있음을 알 수 있습니다.

정확도와 재현율을 반영한 수치인 [mAP@0.5](mailto:mAP@0.5)를 기준으로 모델을 평가합니다. 정확한 수치를 위해 Precision-Recall Curve 그래프를 출력하여 확인합니다.

텍스트, 도표, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

모델의 성능이 0.838로 높은 수치를 갖고 있어 4번 모델의 성능이 좋다고 판단할 수 있습니다. 그러나 2번 모델의 수치가 0.86이었기 때문에 현재 최적의 모델은 2번 모델입니다.

다음 5번 모델(batch size 16, epochs 100, learning rate 0.1)입니다.

5번 모델의 경우, learning rate를 변경하기 때문에 yaml 파일을 생성합니다. Learning rate는 처음과 끝으로 나누어 지정해야 하나, 이번에는 둘 다 0.1로 설정했습니다.

텍스트, 그래프, 도표, 스크린샷이(가) 표시된 사진

자동 생성된 설명

그래프를 통해 총 4개, loss, 정밀도, 재현율, [mAP@0.5](mailto:mAP@0.5)를 분석하면 아래와 같습니다.

1. Loss : train loss는 잘 감소하고 있으나 valid loss는 증가하는 것을 보아 overfitting 일 수 있다는 것을 알 수 있습니다.

2. 정밀도 : 값이 0.8 이상을 유지하다가 0.6 구간으로 떨어진 후 0.6 ~ 0.8 구간에 위치하는 것을 볼 수 있습니다.

3. 재현율 : 0.6에 가까운 값으로 모델이 긍정클래스의 대부분을 잘 구분하고 있다고 볼 수 있습니다.

4. mAP\_0.5 : 0.6에 가까운 값으로 낮은 성능은 아니지만 높은 성능을 가진 모델도 아니라고 볼 수 있습니다. 즉, 중간 정도의 성능을 가진 모델이라고 볼 수 있습니다.

정확도와 재현율을 반영한 수치인 [mAP@0.5](mailto:mAP@0.5)를 기준으로 모델을 평가합니다. 정확한 수치를 위해 Precision-Recall Curve 그래프를 출력하여 확인합니다.

텍스트, 도표, 라인, 번호이(가) 표시된 사진

자동 생성된 설명

모델의 성능이 0.616으로 중간 정도의 수치를 갖고 있어 5번 모델의 성능은 나쁘지도 않지만 좋지도 않은 중간 모델이라고 할 수 있습니다. 2번 모델의 수치가 0.86이었기 때문에 현재 최적의 모델은 2번 모델입니다.

다음 6번 모델(batch size 16, epochs 100, learning rate 0.001)입니다.

6번 모델의 경우, learning rate를 변경하기 때문에 yaml 파일을 생성합니다. Learning rate는 처음과 끝으로 나누어 지정해야 하나, 이번에는 둘 다 0.001로 설정했습니다.

도표, 그래프, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

그래프를 통해 총 4개, loss, 정밀도, 재현율, [mAP@0.5](mailto:mAP@0.5)를 분석하면 아래와 같습니다.

1. Loss : train loss는 잘 감소하고 있습니다. valid loss는 조금 증가하는 것처럼 보이지만 그래도 비교적 잘 감소하고 있습니다.

2. 정밀도 : 값이 0.8 이상의 값으로 모델이 대부분을 잘 구분하고 있다고 볼 수 있습니다.

3. 재현율 : 0.5 이상의 가까운 값으로 0.6 보다 낮기 때문에 낮은 재현율을 보인다고 볼 수 있습니다.

4. mAP\_0.5 : 0.5 ~ 0.6 정도의 값으로 낮은 성능은 아니지만 높은 성능을 가진 모델도 아니라고 볼 수 있습니다. 즉, 중간 정도의 성능을 가진 모델이라고 볼 수 있습니다.

정확도와 재현율을 반영한 수치인 [mAP@0.5](mailto:mAP@0.5)를 기준으로 모델을 평가합니다. 정확한 수치를 위해 Precision-Recall Curve 그래프를 출력하여 확인합니다.

텍스트, 라인, 도표, 그래프이(가) 표시된 사진

자동 생성된 설명

모델의 성능이 0.554으로 중간 정도의 수치를 갖고 있어 6번 모델의 성능은 나쁘지도 않지만 좋지도 않은 중간 모델이라고 할 수 있습니다. 2번 모델의 수치가 0.86이었기 때문에 현재 최적의 모델은 2번 모델입니다.

다음 7번 모델(batch size 16, epochs 100, learning rate 0.001 ~ 0.1)입니다.

7번 모델의 경우, learning rate를 변경하기 때문에 yaml 파일을 생성합니다. Learning rate 처음은 0.001, 끝은 0.1로 설정했습니다.

도표, 그래프, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

그래프를 통해 총 4개, loss, 정밀도, 재현율, [mAP@0.5](mailto:mAP@0.5)를 분석하면 아래와 같습니다.

1. Loss : train loss는 잘 감소하고 있으나 valid loss는 증가하는 것으로 overfitting 일 수있다는 것을 알 수 있습니다.

2. 정밀도 : 값이 0.8 이상의 값으로 모델이 긍정클래스의 대부분을 잘 구분하고 있다고 볼 수 있습니다.

3. 재현율 : 0.5 이상의 가까운 값으로 0.6 보다 낮기 때문에 낮은 재현율을 보인다고 볼 수 있습니다.

4. mAP\_0.5 : 0.5 ~0.6에 가까운 값으로 낮은 성능은 아니지만 높은 성능을 가진 모델도 아니라고 볼 수 있습니다. 즉, 중간 정도의 성능을 가진 모델이라고 볼 수 있습니다.

정확도와 재현율을 반영한 수치인 [mAP@0.5](mailto:mAP@0.5)를 기준으로 모델을 평가합니다. 정확한 수치를 위해 Precision-Recall Curve 그래프를 출력하여 확인합니다.

텍스트, 도표, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

모델의 성능이 0.552로 높은 수치를 갖고 있어 7번 모델의 성능은 중간 정도라고 볼 수 있습니다. 그러나 2번 모델의 수치가 0.86이었기 때문에 현재 최적의 모델은 2번 모델입니다.

# **객체 인식 결과 및 분석**

**1번 케이스**

|  |  |
| --- | --- |
|  |  |

1번 케이스의 원본 사진은 1명이 마스크를 쓰고 있는 사진입니다. 결과는 얼굴에 박스를 잘 그렸고, 0, good mask로 마스크를 잘 쓰고 있는 것으로 예측하였습니다. 위 케이스는 잘 예측한 케이스입니다.

**2번 케이스**

|  |  |
| --- | --- |
|  |  |

위 사진도 1명이 마스크를 쓰고 있는 사진입니다. 결과는 얼굴에 박스를 잘 그렸고, 0, good mask로 마스크를 잘 쓰고 있는 것으로 예측하였습니다. 위 케이스는 잘 예측한 케이스입니다.

1명인 사진들은 대부분 맞게 잘 예측하는 것을 볼 수 있었습니다.

**3번 케이스**

|  |  |
| --- | --- |
|  |  |

원본 사진은 2명이 마스크를 쓰고 있는 사진입니다. 결과는 두 명 모두 얼굴에 박스를 잘 그렸고, 0, good mask로 마스크를 잘 쓰고 있는 것으로 예측하였습니다. 이번 케이스는 잘 예측한 케이스입니다.

**4번 케이스**

|  |  |
| --- | --- |
| 인간의 얼굴, 스크린샷, 사람, 의류이(가) 표시된 사진  자동 생성된 설명 | 스크린샷, 의류, 사람, 인간의 얼굴이(가) 표시된 사진  자동 생성된 설명 |

원본 사진은 3명이 있는 사진입니다. 결과는 한 명은 얼굴에 박스를 잘 그리고 마스크를 잘 쓰고 있는 것으로 올바른 예측을 하였으나, 아래 사람들에 대해서는 잘못된 예측을 하고 있습니다. 맨 왼쪽 사람의 경우 코는 보이지만 입이 보이지 않습니다. 이런 경우 마스크를 잘못 쓰고 있는 경우일 수도 있지만, 1, 마스크를 쓰지 않고 있음으로 예측하였습니다. 가운데 사람과 오른쪽 사람의 경우 얼굴의 3분의 1정도만 보이고 코와 입이 아예 보이지 않는데 마스크를 쓰지 않음으로 인식하였습니다. 이는 잘못 예측한 경우라고 볼 수 있습니다. 이번 케이스는 잘못 예측한 케이스라고 할 수 있습니다.

**5번 케이스**

|  |  |
| --- | --- |
|  |  |

원본 사진은 5명이 있는 사진입니다. 결과는 5명 모두 얼굴에 박스를 그리고, 마스크를 쓰고 있음으로 잘 예측하였습니다. 이번 케이스는 잘 예측한 케이스입니다. 여러 명이 있는 사진임에도 사람 사이의 간격있고, 사람이 잘 구분이 되어 있다면 잘 인식하는 것을 알 수 있습니다.

**6번 케이스**

|  |  |
| --- | --- |
|  |  |

원본 사진은 1명이 마스크를 쓰고 있는 사진입니다. 결과는 중앙에 있는 사람은 마스크를 잘 쓰고 있다고 예측하였습니다. 원본 라벨링은 1명으로 되어 있지만, 모델은 3명을 인식하고 예측하였습니다. 사진의 크기가 작아 글자와 박스가 잘리지만, 잘 예측한 것으로 예상됩니다. 이번 케이스는 잘 예측한 케이스입니다.

**7번 케이스**

|  |  |
| --- | --- |
|  |  |

원본 사진은 1명이 마스크를 잘못 쓰고 있는 사진입니다. 결과는 박스를 2개 그리고 두 개 모두 0 마스크를 잘 쓰고 있음으로 예측하였습니다. 하지만 해당 사진은 마스크를 잘못 쓰고 있는 사진이라는 점, 그리고 사람이 한 명인데 한 명에 바운딩 박스를 그려 2명으로 인식했다는 것으로 보아 잘못 예측한 케이스라는 것을 알 수 있습니다. 이번 케이스는 잘못 예측한 케이스입니다.

**8번 케이스**

|  |  |
| --- | --- |
|  |  |

원본 사진은 1명이 마스크를 쓰고 있는 사진입니다. 결과는 바운딩 박스를 잘 그리고, 0 마스크를 잘 쓰고 있음으로 예측했습니다. 손으로 마스크를 가리고 있어 이상하게 예측할 수 있었지만, 잘 예측하였습니다. 이번 케이스는 잘 예측한 케이스이며, 다른 장애물이 있어도 잘 예측할 수 있을 정도의 성능을 갖고 있다는 것을 알 수 있습니다.

**9번 케이스**

|  |  |
| --- | --- |
|  |  |

원본 사진은 2명이 있는 사진입니다. 결과는 두 명 모두 바운딩 박스를 잘 그렸고, 한 명은 잘 쓰고 있음, 한 명은 마스크 없음으로 예측하였습니다. 이번 케이스는 잘 예측한 케이스입니다. 한 명은 측면으로 있음에도 불구하고 잘 예측한 것으로 보아 높은 성능을 갖고 있다는 것을 알 수 있습니다.

**10번 케이스**

|  |  |
| --- | --- |
|  |  |

원본 사진은 3명이 있는 사진입니다. 결과는 4명을 인식하였고 두 명은 잘 쓰고 있음, 2명은 안 쓰고 있음으로 예측하였습니다. 앞 두 명과, 뒤 2명 중 오른쪽 사람은 잘 인식했다고 할 수 있습니다. 뒤 중 왼쪽 사람은 1 마스크를 쓰고 있지 않음으로 인식하였지만 작고 잘 보이지 않음에도 진위 여부를 확인 할 수 없습니다. 또한, 사진을 확대를 해보아도 사람으로 보이지 않아, 잘못 예측한 것으로 생각됩니다. 이번 케이스 3명은 잘 예측하였고, 1명은 잘못 예측한 것으로 볼 수 있기 때문에 그래도 잘 예측한 케이스락 볼 수 있습니다.

**11번 케이스**

|  |  |
| --- | --- |
|  |  |

원본 사진은 14명이 있는 사진입니다. 박스를 여러 개를 그리기는 했지만, 결과들이 겹쳐서 제대로 확인 할 수 없었습니다. 사진 중에서 몇 명의 사람을 결과를 분석해보겠습니다. 먼저 왼쪽 아래 빨간 옷을 입은 아이는 마스크를 잘못 쓰고 있는 것으로 잘 예측하였습니다. 아이 옆에 있는 아주머니의 경우에는 마스크를 잘못 쓰고 있는 경우지만, 0 마스크를 잘 쓰고 있음과 2 마스크를 잘못 쓰고 있음으로 중복으로 예측하였습니다. 이를 통해 사람이 많은 사진이면 예측하기 어려우며, 모델의 성능에 아쉬운 부분이 있다고 생각합니다.

# **모델 성능 평가**

아래 지표를 통해 모델의 최종 성능을 평가합니다.

* Precision-Recall Curve
* mAP@50
* confusion\_matrix

텍스트, 도표, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

라벨에 따르면 각 선은 다음과 같은 클래스를 가집니다.

* Good Mask (연한 파란색 선): 마스크 잘 착용
* No Mask (주황색 선): 마스크 미 착용
* Wrong Mask (초록색 선): 마스크 잘 못 착용
* All Classes (굵은 파란색 선): 모든 클래스의 성능을 종합한 mAP@0.5

각 클래스는 다음과 같은 값과 의미를 가집니다.

* Good Mask (0.951): 마스크를 잘 착용한 사람을 인식하고 예측하는 것을 의미하고 0.9는 높은 값이므로, 높은 precision과 recall 값을 가진다는 것을 알 수 있습니다.
* No Mask (0.851): 마스크를 착용하지 않은 사람을 인식하고 예측하는 것을 의미하고 0.851은 높은 값이므로, 높은 precision과 recall 값을 가진다는 것을 알 수 있습니다.
* Wrong Mask (0.801): 마스크를 잘 못 착용한 사람을 인식하고 예측하는 것을 의미하고 0.801는 높은 값이므로, 높은 precision과 recall 값을 가진다는 것을 알 수 있습니다.
* All Classes (0.867 mAP@0.5): mAP@0.867은 높은 값으로 전반적으로 높은 수치를 갖고 있어 좋은 성능을 보인다는 것을 알 수 있습니다.

스크린샷, 도표, 직사각형, 사각형이(가) 표시된 사진

자동 생성된 설명

1. confusion\_matrix 구성 요소

* True Labels : 실제 클래스 레이블
* Predicted Labels : 모델이 예측한 클래스 레이블
* 색상 : 정확도 값

2. good\_mask 분석

* 실제 마스크를 잘 쓰고 있는 것을 잘 쓰고 있음으로 예측한 것은 0.94로 높은 수치를 갖고 있습니다. 이를 통해 마스크를 잘 착용한 사진들은 잘 인식하는 것을 알 수 있습니다.
* 실제 마스크를 잘 쓰고 있는 것을 쓰고 있지 않음과 잘 못 쓰고 있음으로 예측한 것은 수치는 나오지 않지만 색으로 분석할 수 있습니다. 두 개 모두 흰 색인 것으로 보아 쓰고 있지 않음과 잘못 쓰고 있음으로 예측하지 않은 것을 알 수 있습니다.
* 실제 마스크를 잘 쓰고 있음을 배경으로 예측한 것은 연한 하늘 색으로 몇 개는 실제 마스크를 쓰고 있는 것을 배경으로 인식한 것이 있음을 알 수 있습니다.

3. no\_mask 분석

* 실제 마스크를 쓰지 않고 있는 것을 잘 쓰고 있음으로 예측한 것은 0.07로 낮은 수치를 갖고 있습니다. 이를 통해 마스크를 쓰지 않고 있는데 마스크를 쓰고 있다고 예측한 것이 조금 있다는 것을 알 수 있습니다.
* 실제 마스크를 쓰지 않고 있는 것을 쓰지 않고 있음으로 예측한 것은 수치는 나오지 않지만 색으로 분석할 수 있습니다. 색이 남색인 것으로 보아 약 0.8 정도의 수치를 갖고 있다고 볼 수 있습니다. 이는 올바르게 예측한 것이 0.8임을 알 수 있습니다.
* 실제 마스크를 쓰지 않고 있는 것을 잘 못 쓰고 있음으로 예측한 것은 수치는 나오지 않지만 색으로 분석할 수 있습니다. 색이 연한 하늘 색인 것으로 보아 0.0~0.1 사이의 수치를 갖고 있다고 볼 수 있습니다.
* 실제 마스크를 쓰지 않고 있는 것을 배경으로 예측한 것은 조금 진한 하늘 색으로 몇 개는 실제 마스크를 쓰고 있는 것을 배경으로 인식한 것이 있음을 알 수 있습니다.

4. wrong\_mask 분석

* 실제 마스크를 잘못 쓰고 있는 것을 잘 쓰고 있음으로 예측한 것은 0.07로 낮은 수치를 갖고 있습니다. 이를 통해 마스크를 잘못 쓰고 있는데 마스크를 쓰고 있다고 예측한 것이 조금 있다는 것을 알 수 있습니다.
* 실제 마스크를 잘못 쓰고 있는 것을 쓰지 않고 있음으로 예측한 것은 수치는 나오지 않지만 색으로 분석할 수 있습니다. 색이 흰색인 것으로 보아 쓰지 않고 있음으로 예측한 것이 없다고 볼 수 있습니다.
* 실제 마스크를 잘못 쓰고 있는 것을 잘못 쓰고 있음으로 예측한 것은 수치는 나오지 않지만 색으로 분석할 수 있습니다. 색이 진한 남색인 것으로 보아 약 0.8 정도의 수치를 갖고 있다고 볼 수 있습니다. 이는 올바르게 예측한 것이 0.8임을 알 수 있습니다.
* 실제 마스크를 잘못 쓰고 있는 것을 배경으로 예측한 것은 연한 하늘 색으로 몇 개는 마스크를 잘못 쓰고 있는 것을 배경으로 인식한 것이 있음을 알 수 있습니다.

5. background 분석

* 배경을 잘 쓰고 있음으로 예측한 것은 0.61로 조금 높은 수치를 갖고 있습니다. 이를 통해 라벨링이 되어 있지 않은 배경에서 객체를 인식하고 마스크를 쓰고 있다고 예측한 것이 있다는 것을 알 수 있습니다. 이는 모델이 객체 인식 하는 성능이 좋다는 것을 의미합니다.
* 배경을 쓰지 않고 있음으로 예측한 것은 수치는 나오지 않지만 색으로 분석할 수 있습니다. 색이 연한 파란색인 것으로 보아 약 0.3 정도의 수치를 갖고 있다고 볼 수 있습니다. 이는 모델이 객체 인식 하는 성능이 좋다는 것을 의미합니다.
* 배경을 잘못 쓰고 있음으로 예측한 것은 수치는 나오지 않지만 색으로 분석할 수 있습니다. 색이 연한 하늘색인 것으로 보아 약 0.1 정도의 수치를 갖고 있다고 볼 수 있습니다. 이는 모델이 객체 인식 하는 성능이 좋다는 것을 의미합니다.

# **최종 성능 평가**

해당 모델은 마스크 잘 착용 > 마스크 미 착용 > 마스크 잘 못 착용 순서로 잘 인식하고 예측하는 것을 알 수 있습니다. 또한, mAP@0.5의 수치가 높은 것으로 보아 `좋은 모델`이라고 할 수 있습니다.

# **오류 사례 및 분석**

테스트 데이터로 확인했을 때는 다음과 같은 오류 사례가 있었습니다.

1. 바운딩 박스를 잘못 그리는 경우
2. 바운딩 박스를 여러 개 그리는 경우
3. 잘못 쓰고 있는 것을 안 쓰고 있다고 예측한 경우
4. 사람으로 볼 수 없지만 사람이라고 인식한 경우

오류가 발생하는 이유는 다음과 같다고 생각합니다.

1. 코를 눈으로 인식하는 등 신체 부위를 실제와 다르게 인식하여 마스크를 썼다고 예측한 것
2. 배경에서 사람을 인식할 때에는 배경색, 주변 사람의 존재 유무 등으로 사람을 인식한 것

# **실제 환경에서 발생할 수 있는 문제점**

실제 환경에서 발생할 수 있는 문제점은 아래와 같이 경우가 존재합니다.

1. 조명 조건의 변화 : 조명에 따라 사람을 배경이라고 인식할 수 있습니다.
2. 다양한 배경 : 모델 테스트를 통해 배경을 사람으로 인식하는 경우가 있다는 것을 알 수 있습니다. 벽화와 같이 사람처럼 생긴 배경이 있다면, 이를 사람으로 잘못 인식할 수 있습니다.
3. 사진의 화질 : 실제 환경에서도 카메라로 인식하는 것이 아닌 사진으로 인식하게 되면 화질에 따라 인식을 못할 수도 있습니다.
4. 마스크 종류 : 마스크 색상 등에 대해서는 잘 인식하지만 만약, 사람이나 캐릭터가 그려진 마스크라면 2명으로 인식할 수도 있습니다.
5. 피부와 마스크 색상 : 피부와 마스크 색상이 동일하다면 잘못 인식할 수도 있습니다.

# **개선 방향**

1. 데이터 불균형 해결

* + 이번에 사용한 데이터셋은 데이터의 불균형이 존재하는데 3개의 분류 중 마스크를 잘 쓰고 있는 사진의 비율이 많습니다.
  + 이를 해결하기 위해 마스크를 잘못 쓰고 있는 사진과 마스크를 잘 쓰고 있는 사진의 수를 늘려 두 케이스의 학습을 늘려 개선할 수 있습니다.

2. 다양한 데이터 사용

* 캐릭터가 그려진 마스크나, 다양한 마스크 색상을 가진 데이터를 추가하여 학습하면 모델 성능을 개선할 수 있습니다.
* 여러 배경을 가진 데이터를 활용하면 모델 성능을 개선할 수 있습니다.

3. 여러 화질 데이터 사용

* 동일한 사진을 화질을 낮추어 학습해보는 것도 모델의 성능을 올릴 수 있는 방법입니다.