

모델 성능 평가 및 3주차 주간 보고

발표자: 최승환

일자: 2023-07-24

목차

01 Loss

02 Metric

03 3주차 활동 내용

04 4주차 활동 계획

손실(Loss)

모델 성능 평가에는 **Loss**와 **Metric**이라는 두 가지 주요 지표가 사용된다.

손실(Loss)

모델 성능 평가에는 **Loss**와 **Metric**이라는 두 가지 주요 지표가 사용된다.

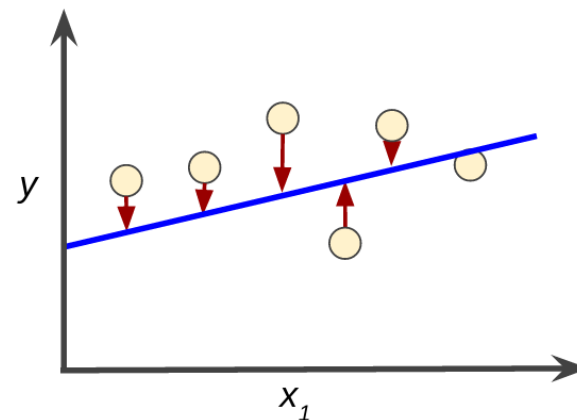
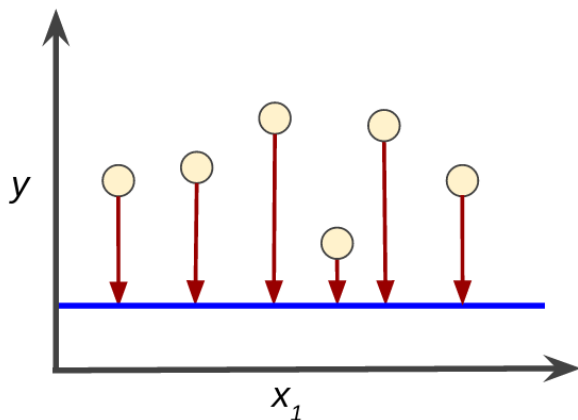
손실은 주로 모델의 학습 과정에서 사용되는 지표로,
학습 데이터와 실제 값(레이블)과의 오차를 최소화 하도록 학습된다.

손실 그래프는 모델의 학습 상태에 대해서 알아볼 수 있는 지표로 활용되고 있다.

손실(Loss)

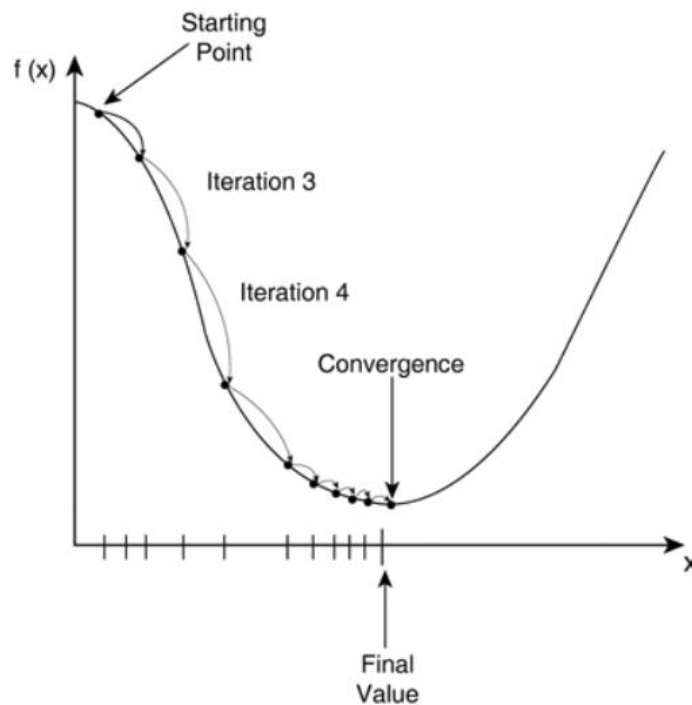
모델 성능 평가에는 **Loss**와 **Metric**이라는 두 가지 주요 지표가 사용된다.

손실은 이러한 오차를 계산하는 함수로, 모델의 예측과 실제 값 간의 차이를 측정한다.



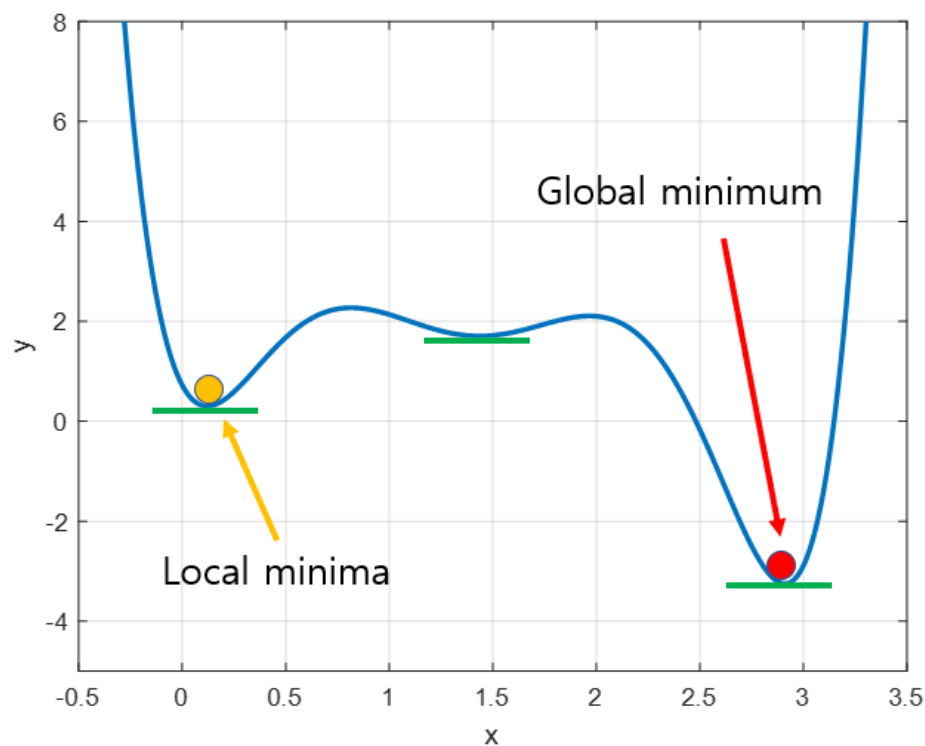
손실(Loss)

모델이 학습하는 동안 모델은 **경사하강법**과 같은 최적화 알고리즘을 사용하여 예측 값이 실제 값에 가깝도록 하는 최적의 가중치를 계속해서 업데이트 한다.



손실(Loss)

최적의 손실 값을 가지기 위해서 **학습률(Learning rate)**와 **배치사이즈(Batch size)**를 잘 조절해야 한다.

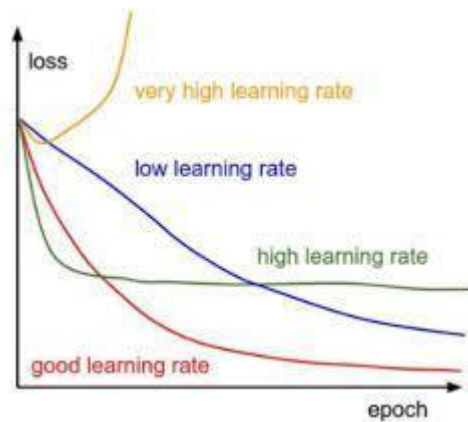


Learning rate : 미분 기울기의 이동 보폭.

Batch size : 전체 데이터에서 쪼개져 학습될 양.

손실(Loss)

최적의 손실 값을 가지기 위해서 **학습률(Learning rate)**와 **배치사이즈(Batch size)**를 잘 조절해야 한다.



손실 곡선 모양	해결방법
Curve가 선형적이다.	Learning rate를 더 높게 설정한다.
Curve 감소 폭이 작다.	Learning rate를 더 낮게 설정한다.
Curve 진동 폭이 크다.	Batch size를 더 높게 설정한다.

손실(Loss)

학습률과 배치사이즈에 따라 학습 결과가 달라질 수 있다.

▼ batch-size & ▼ learning-rate

한 번에 적은 데이터로 조금씩 학습하기 때문에 그 만큼 학습 수가 많아지므로
local minima에 빠져 **global minima에 도달하기 힘들지만**, 초기 방향은 쉽게 잡히고 수렴은 잘 될 것이다.

▼ batch-size & ▲ learning-rate

적은 데이터로 많은 학습을 바라기 때문에 잘 수렴하기 힘들 것이다.

▲ batch-size & ▼ learning-rate

많은 데이터로 조금씩 학습해 나가기 때문에 **과적합**이 쉽게 될 것이다.

▲ batch-size & ▲ learning-rate

한 번에 많은 데이터로 많이 학습하기 때문에 초기 방향을 잡기 힘들어 learning-rate를 잘 조절해야 할 것이다.
그리고 **과적합**의 문제도 존재한다. 하지만 local minima에 걸릴 확률이 적고, **global minima에 수렴할 수 있다**.

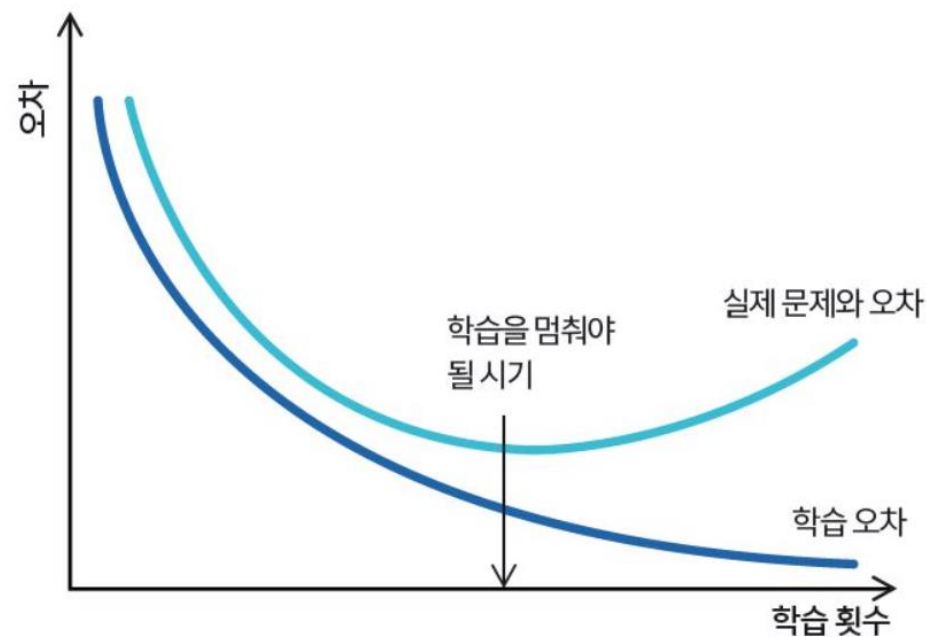
손실(Loss)

손실 그래프를 보면 학습 데이터에 대한 손실은 계속해서 줄어들이지만,
검증 데이터에 대한 손실은 일정 수준 이후 증가하는 경우가 있을 수 있다.

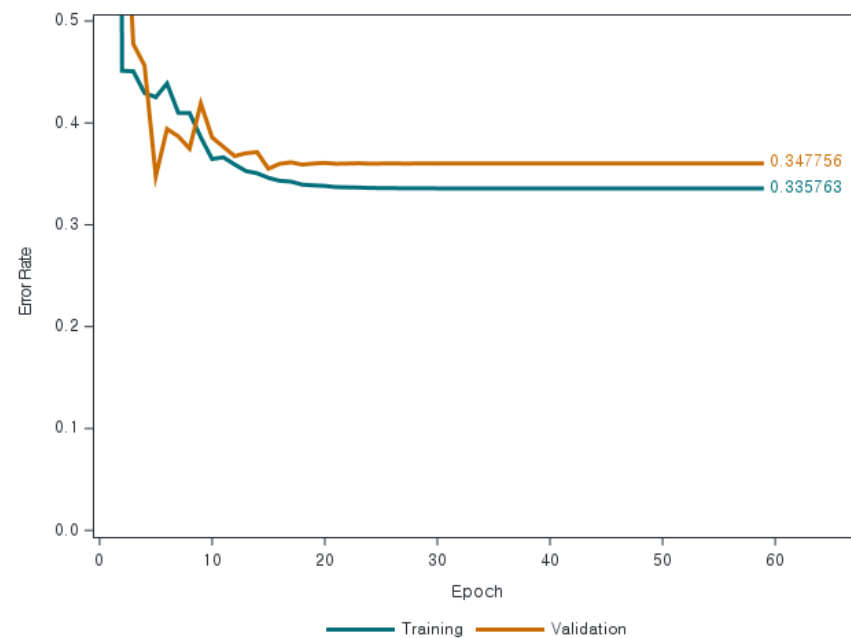
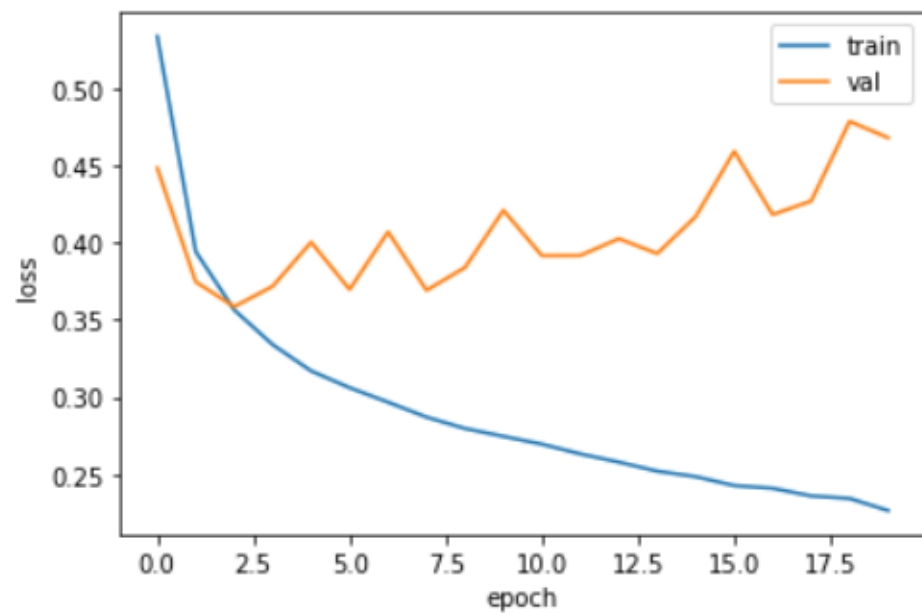
이러한 경우 모델이 학습데이터에 지나치게 적합하여 검증 데이터에 대한 성능이 저하되는 **과적합** 상태일 수 있다.

손실(Loss)

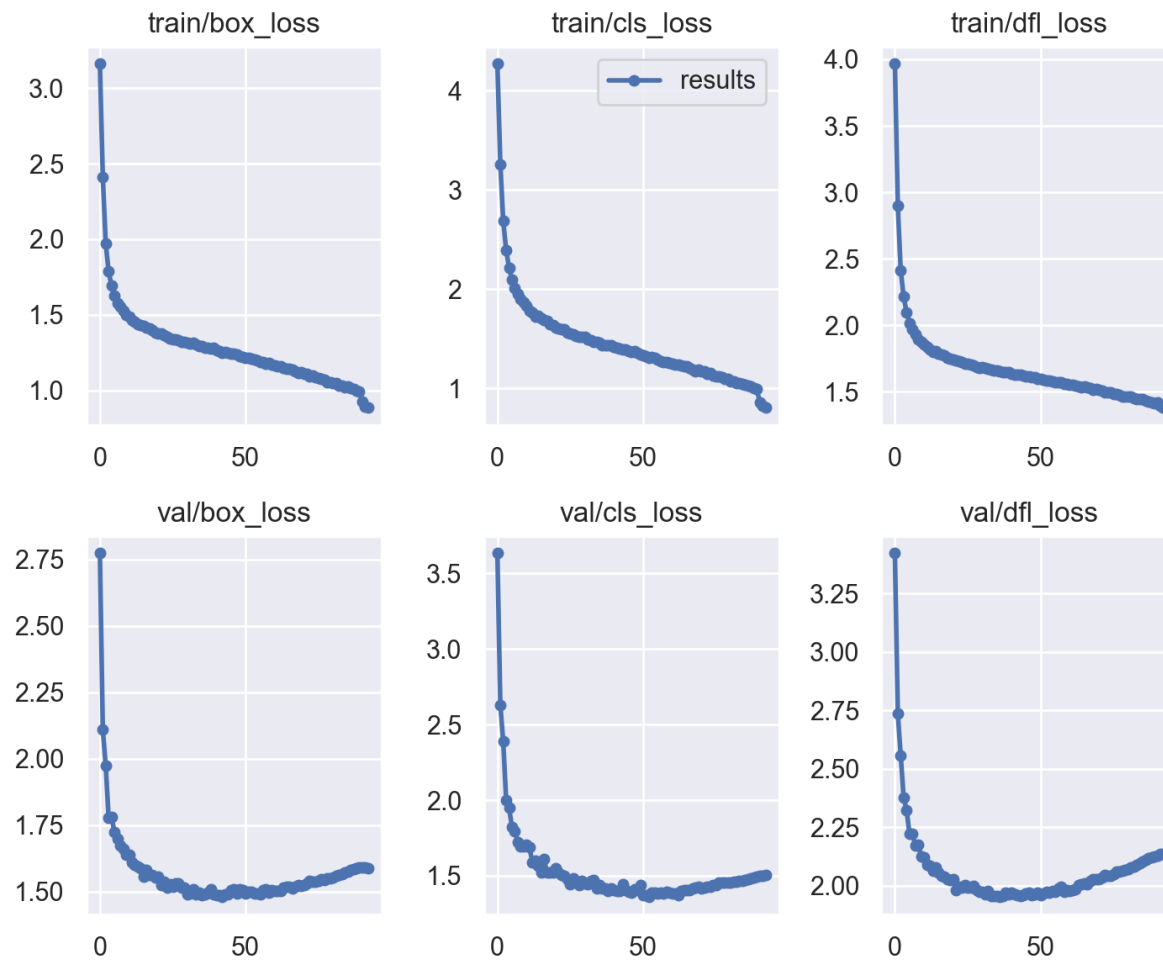
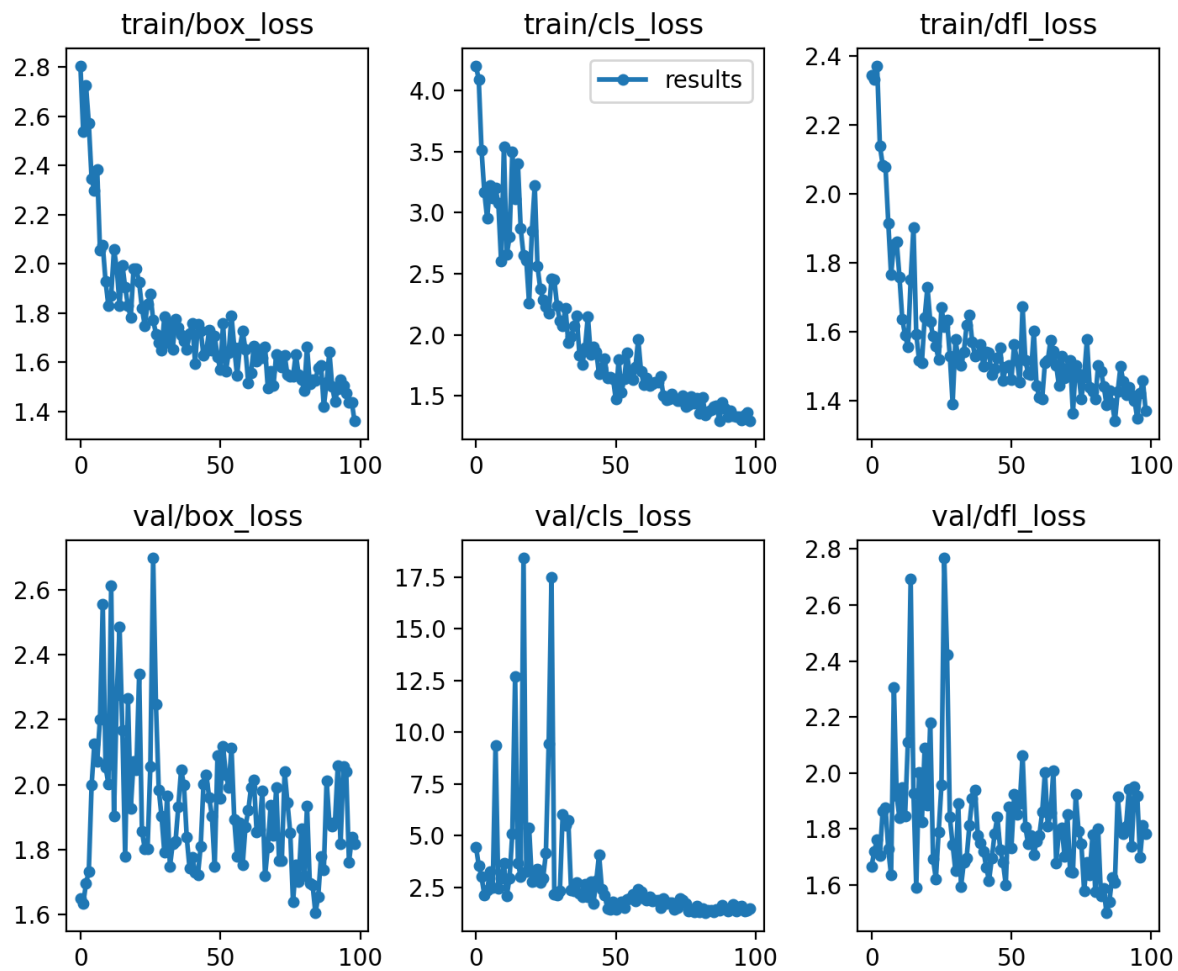
손실 그래프를 보면 학습이 충분히 진행되어 최적의 성능에 도달한 시점을 파악할 수 있다.



손실(Loss)



손실(Loss)



■ 평가지표(Metric)

Metric은 어떤 모델을 평가 혹은 비교하기 위해 모델의 성능을 숫자로 표현하는 것을 말한다.

분류 문제에서 가장 자주 활용되는 Metric은 Accuracy, AUC, F1 score 등이 있다.

이것을 잘 이해하기 위해서 가장 먼저 **혼동 행렬(Confusion matrix)**을 이해하는 것이 도움이 된다.

혼동 행렬(Confusion Matrix)

모델이 예측한 값이 정답인가?

True	Negative	= TN
False	Positive	= FP
False	Negative	= FN
True	Positive	= TP

모델

혼동 행렬(Confusion Matrix)

		예측 값	
		Negative	Positive
실제 값	Negative	TN	FP
	Positive	FN	TP

혼동 행렬(Confusion Matrix)

Actual	T	F	T	F	F	F	F	T	T
Model	T	F	T	T	F	T	F	F	T

		예측 값	
		Negative	Positive
실제 값	Negative	TN	FP
	Positive	FN	TP = 3

혼동 행렬(Confusion Matrix)

Actual	T	F	T	F	F	F	F	T	T
Model	T	F	T	T	F	T	F	F	T

		예측 값	
		Negative	Positive
실제 값	Negative	TN = 3	FP
	Positive	FN	TP = 3

혼동 행렬(Confusion Matrix)

Actual	T	F	T	F	F	F	T	T
Model	T	F	T	T	F	T	F	T

		예측 값	
		Negative	Positive
실제 값	Negative	TN = 3	FP = 2
	Positive	FN	TP = 3

혼동 행렬(Confusion Matrix)

Actual	T	F	T	F	F	F	F	T	T
Model	T	F	T	T	F	T	F	F	T

		예측 값	
		Negative	Positive
실제 값	Negative	TN = 3	FP = 2
	Positive	FN = 1	TP = 3

정확도(Accuracy)

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Accuracy = 0.6666$$

		예측 값	
		Negative	Positive
실제 값	Negative	TN = 3	FP = 2
	Positive	FN = 1	TP = 3

정확도(Accuracy)

- 아래와 같이 **데이터 불균형**이 있다면 이 모델은 무조건 정답이라고만 말하는 바보가 될 것이다.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Accuracy = 0.9999$$

		예측 값	
		Negative	Positive
실제 값	Negative	TN = 1	FP = 0
	Positive	FN = 2	<u>TP = 1,000,000</u>

정밀도(Precision)

- 정답이라고 생각한 것 중에서 얼마나 많이 맞았는가, 과연 실속이 있는가
- 실제로 아닌데 맞다고 생각한 개수에 따라 결과가 달라진다.
- Precision이 낮으면 검출력이 뛰어나겠지만 거짓양성도 많이 섞여 있을 수 있다.
- Precision이 너무 높으면 확실한 경우에만 양성으로 예측할 수 있다.

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = 0.6$$

		예측 값	
		Negative	Positive
실제 값	Negative	TN = 3	FP = 2
	Positive	FN = 1	TP = 3

재현율(Recall)

- 실제로 맞는데 아니라고 생각한 개수에 따라 결과가 달라진다.
- Recall이 낮으면 정답이 아니라고 오인하는 경우가 많아져 정답조차 잘 찾지 못할 수도 있다.
- Recall이 너무 높으면 몇 개 정도는 잘못 판단할지라도 정답은 잘 찾아낼 것이다.

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = 0.25$$

		예측 값	
		Negative	Positive
실제 값	Negative	TN = 3	FP = 2
	Positive	FN = 1	TP = 3

Precision & Recall

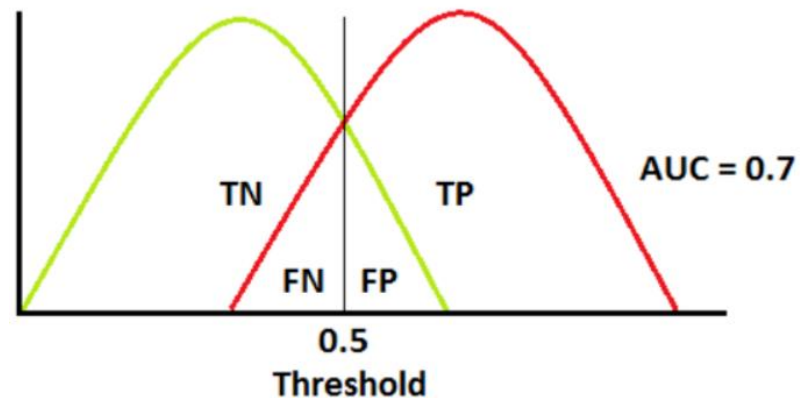
- Precision과 Recall은 모두 True라고 예측한 것과 관련이 있으나, 관점이 다르다.
- 임계값을 조정해 Precision 또는 Recall을 높일 수 있지만, Precision과 Recall은 상호보완적이기 때문에 Recall을 올리면 Precision이 내려가고, Precision을 올리면 Recall이 내려갈 수 밖에 없다.

▼ Precision & ▲ Recall

- 높은 확률로 정답을 잘 찾아내겠지만 많은 경우에 가짜도 포함될 수 있다.
- 필요 없는 것을 뽑을 수 있다. (낮은 임계값)

▼ Precision & ▲ Recall

- 정답만 뽑으려다 보니 미처 못 뽑힌 정답들도 많이 존재할 수 있다.
- 필요한 것을 못 뽑을 수 있다. (높은 임계값)



F1 Score

- Precision과 Recall의 상호보완적 관계를 이용하여 생겨난 것이 F1 score이다.
- Precision과 recall 사이에서 절충이 이루어지며 특이값의 리스크를 회피할 수 있다.
- **클래스 불균형**이 있을 때 활용하는 지표이다.

```
F1-score = 2 * ((Precision*Recall)/(Precision+Recall))
```

```
# 예1) 1, 0.1
```

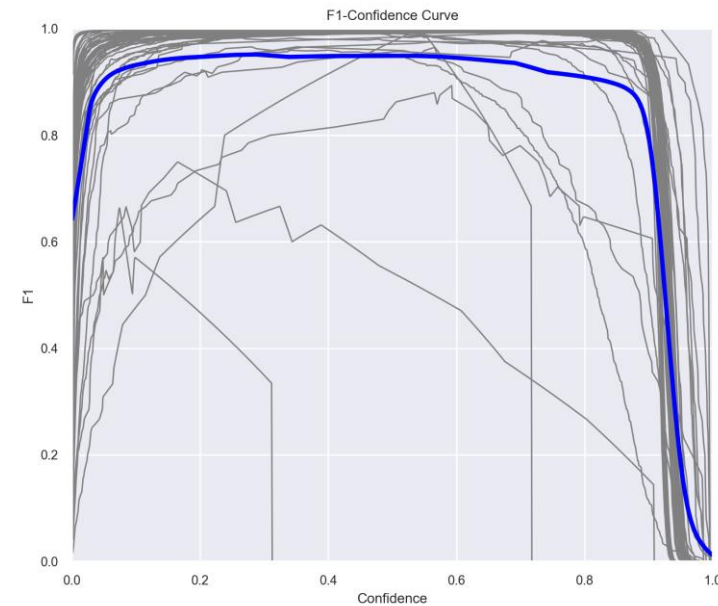
```
2*(0.1/1.1) = 0.1818xxx
```

```
# 예2) 0.1, 0.1
```

```
2*(0.01/0.2) = 0.1
```

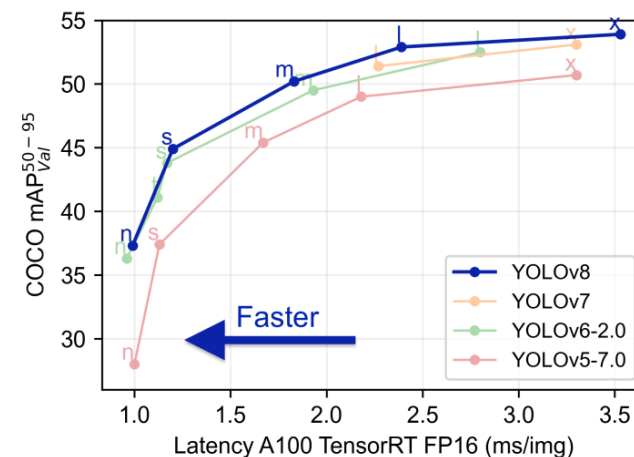
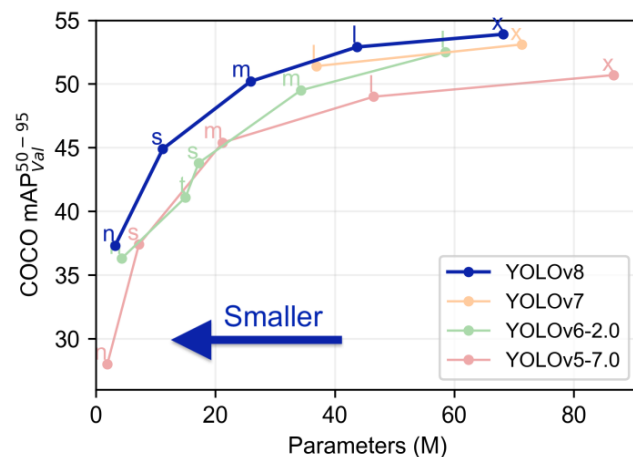
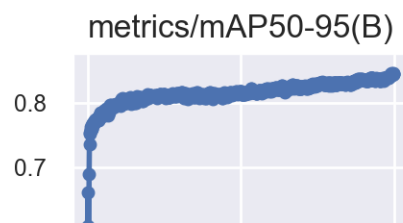
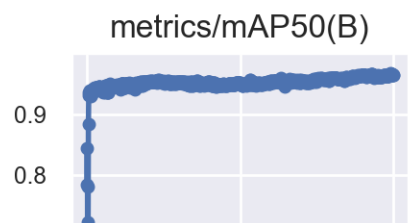
```
# 예3) 1, 1
```

```
2*(1/2) = 1
```




mAP(mean Average Precision)

- **mAP**는 Object Detection 분야에서 모델의 성능을 평가하는 지표이다.
- 객체 검출은 이미지 내에서 객체의 위치와 클래스를 예측하는 작업이므로, 정확도를 측정하기 위해 단순한 정확도 보다는 mAP와 같은 특화된 지표가 사용된다.



IoU(Intersection over Union)

- IoU는 실제 바운딩 박스와 예측 바운딩 박스가 얼마나 일치하는지를 확인하기 위해 사용된다.
- 보통 IoU가 0.5이상일 경우 객체로 판단한다.

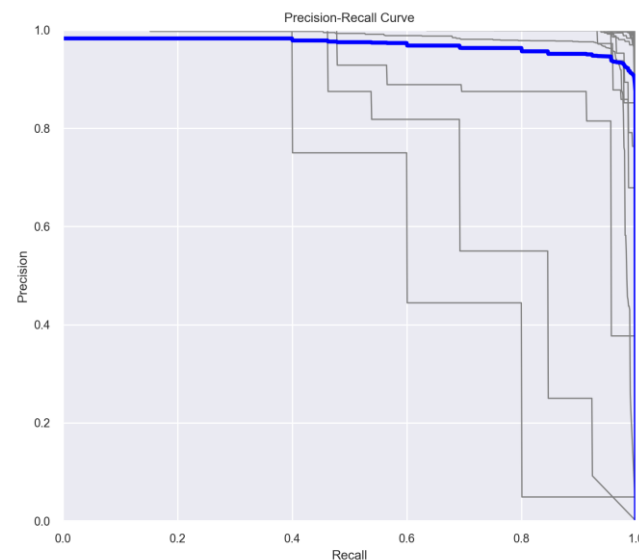
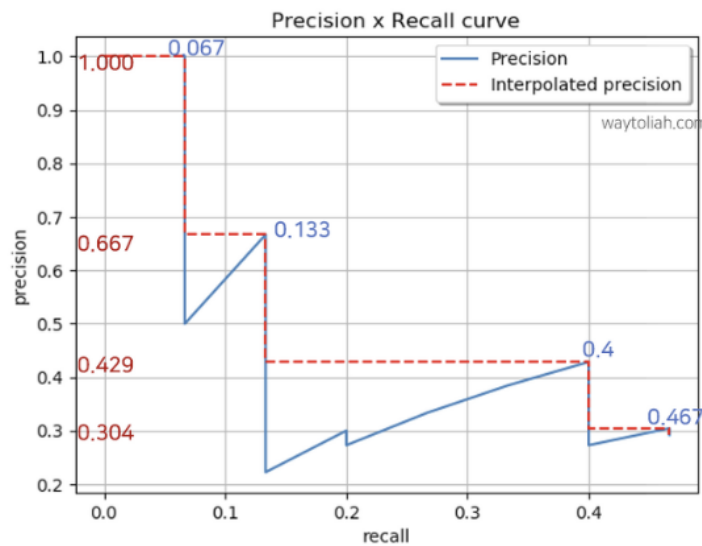

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

AP에 따른 mAP 표기		IOU
AP@IoU=0.50	AP50	0.50
AP@IoU=0.55	AP55	0.55
AP@IoU=0.60	AP60	0.60
AP@IoU=0.65	AP65	0.65
AP@IoU=0.70	AP70	0.70
AP@IoU=0.75	AP75	0.75
AP@IoU=0.80	AP80	0.80
AP@IoU=0.85	AP85	0.85
AP@IoU=0.90	AP90	0.90
AP@IoU=0.95	AP95	0.95

waytoliah.com

P-R curve

- P-R 곡선은 confidence 임계값에 따른 Precision과 Recall의 변화를 그래프로 표현한 것이다.
- Recall이 높아져도 Precision이 유지되는 경우, 특정 클래스 검출을 위한 모델 성능이 좋을 것으로 평가된다.
즉, confidence 임계값을 변경하더라도, Precision과 Recall이 모두 높은 경우 모델 성능이 좋을 것으로 평가된다.
- 관련된 객체만 검출할 수 있는 모델(▲ Precision)도 좋은 모델로 평가할 수 있다.

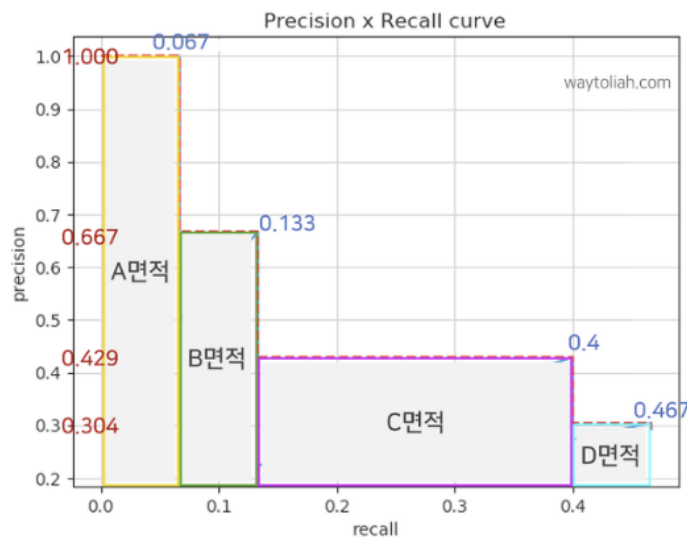


AP(Average Precision)

➤ AP 곡선은 Precision과 Recall을 고려한 종합적 평가 지표이며,

실제로 AP는 0~1 사이의 모든 Recall에 대응하는 평균 Precision이다.

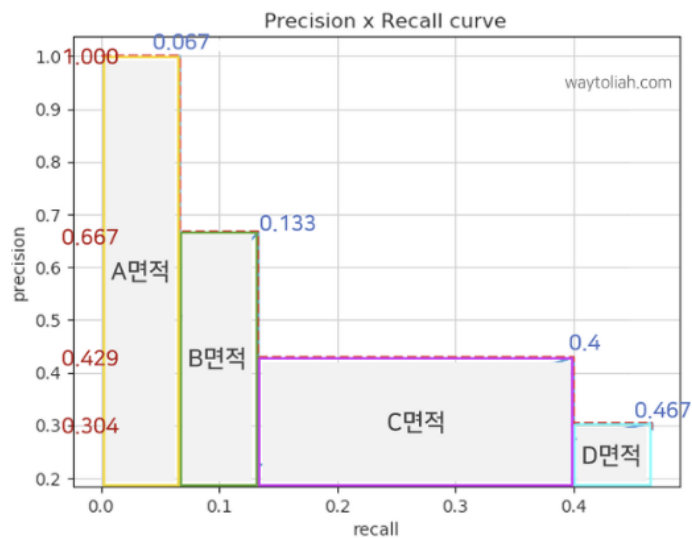
➤ AP는 P-R 곡선의 아래 면적이다. AP가 높을 수록 모델의 성능이 좋다고 판단할 수 있다.



AP에 따른 mAP 표기		IOU
AP@IoU=0.50	AP50	0.50
AP@IoU=0.55	AP55	0.55
AP@IoU=0.60	AP60	0.60
AP@IoU=0.65	AP65	0.65
AP@IoU=0.70	AP70	0.70
AP@IoU=0.75	AP75	0.75
AP@IoU=0.80	AP80	0.80
AP@IoU=0.85	AP85	0.85
AP@IoU=0.90	AP90	0.90
AP@IoU=0.95	AP95	0.95

mAP

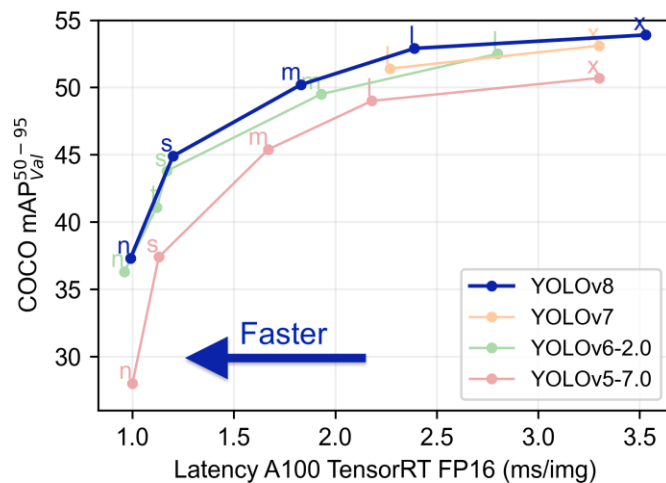
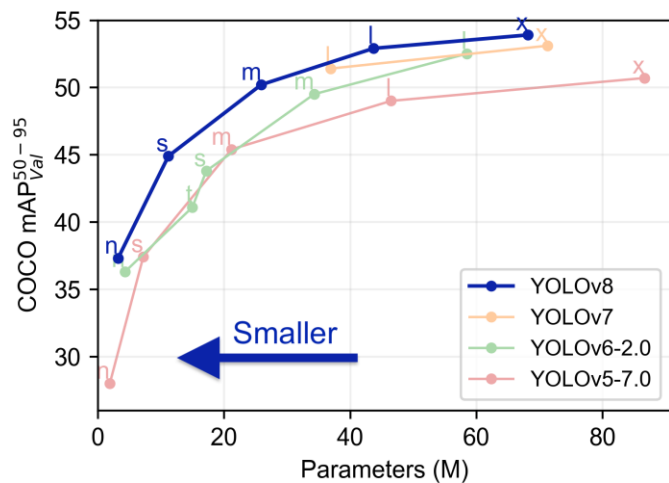
- 각 클래스마다 한 AP를 갖게 되는데 모든 클래스의 AP에 대해 평균값을 낸 것이 **mAP**이다.



AP에 따른 mAP 표기		IOU
AP@IoU=0.50	AP50	0.50
AP@IoU=0.55	AP55	0.55
AP@IoU=0.60	AP60	0.60
AP@IoU=0.65	AP65	0.65
AP@IoU=0.70	AP70	0.70
AP@IoU=0.75	AP75	0.75
AP@IoU=0.80	AP80	0.80
AP@IoU=0.85	AP85	0.85
AP@IoU=0.90	AP90	0.90
AP@IoU=0.95	AP95	0.95

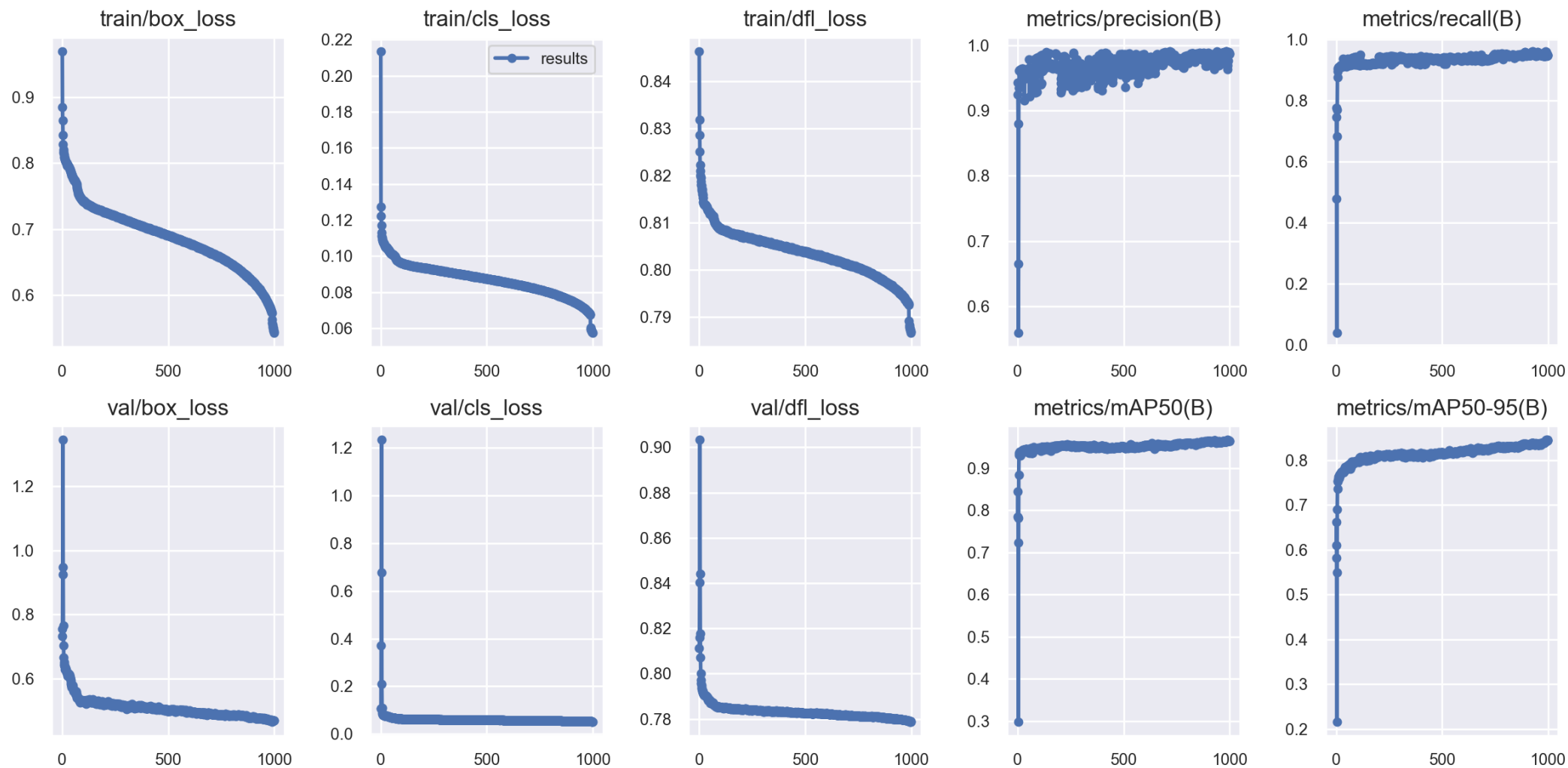
YOLO

➤ YOLO에서 제공하는 잘 설계된 많은 모델이 있다.



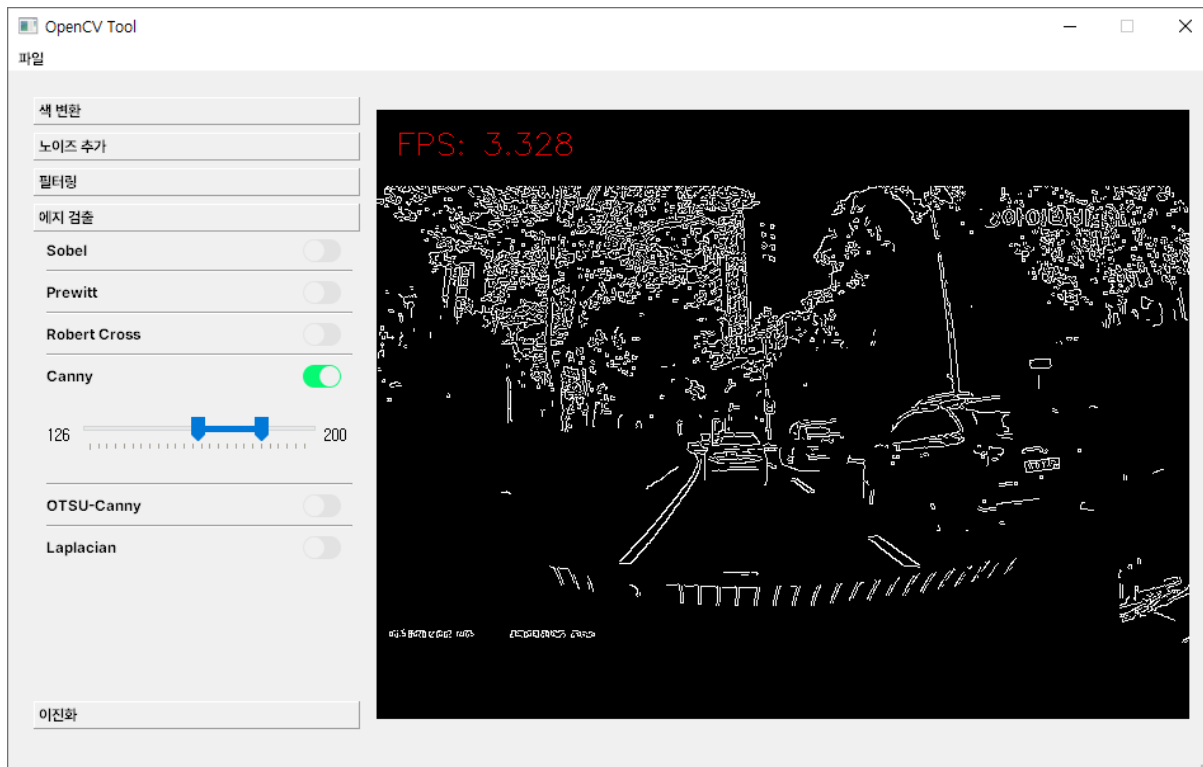
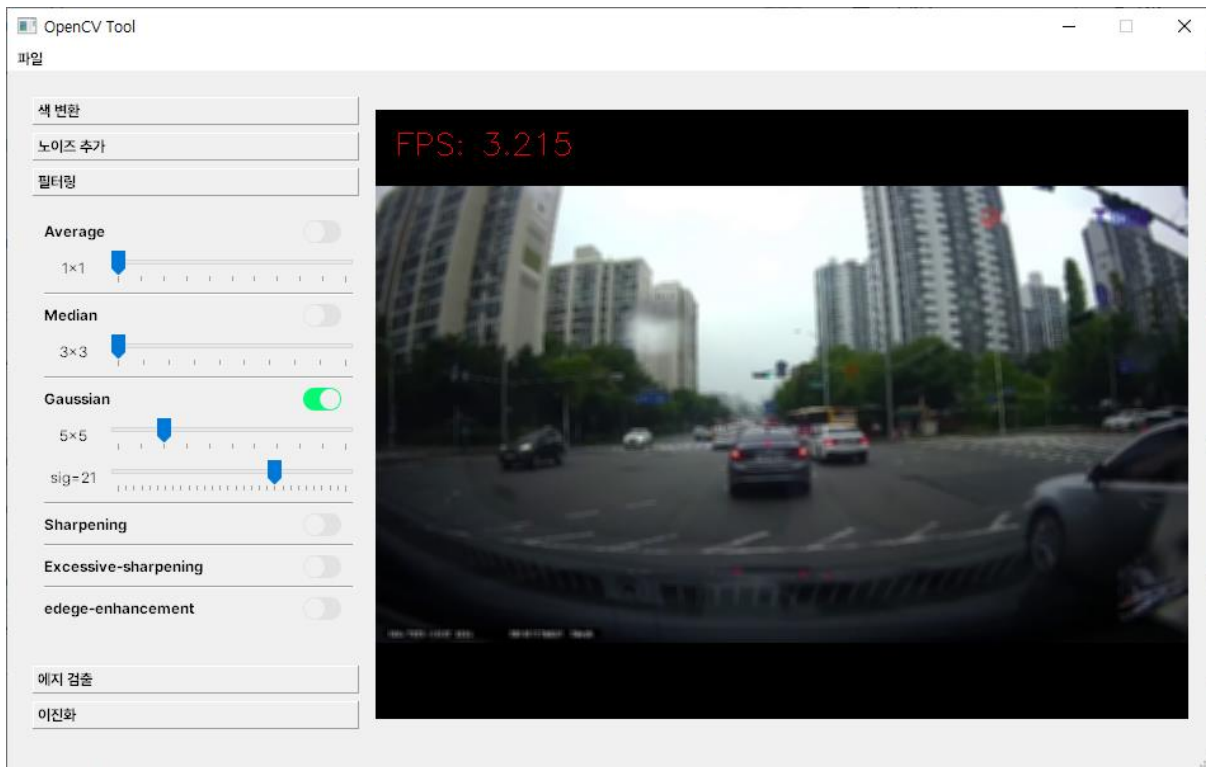
Model	size (pixels)	mAP ^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

YOLO

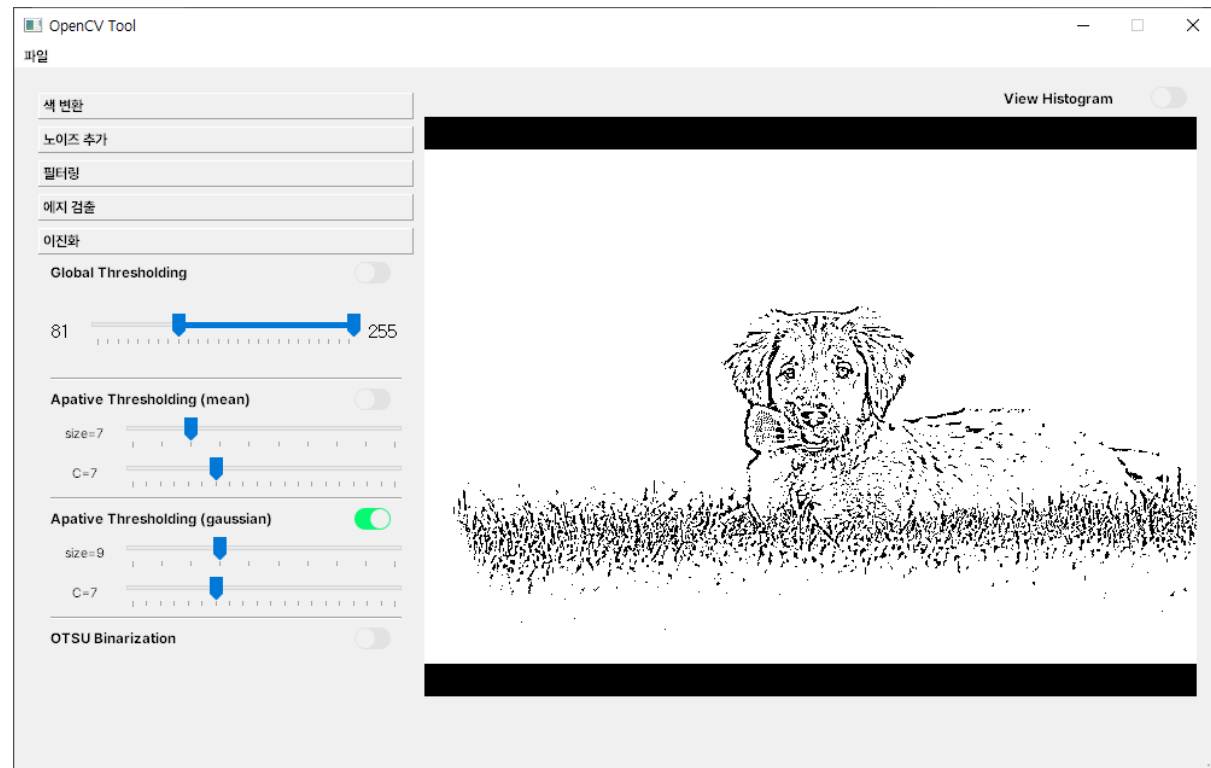
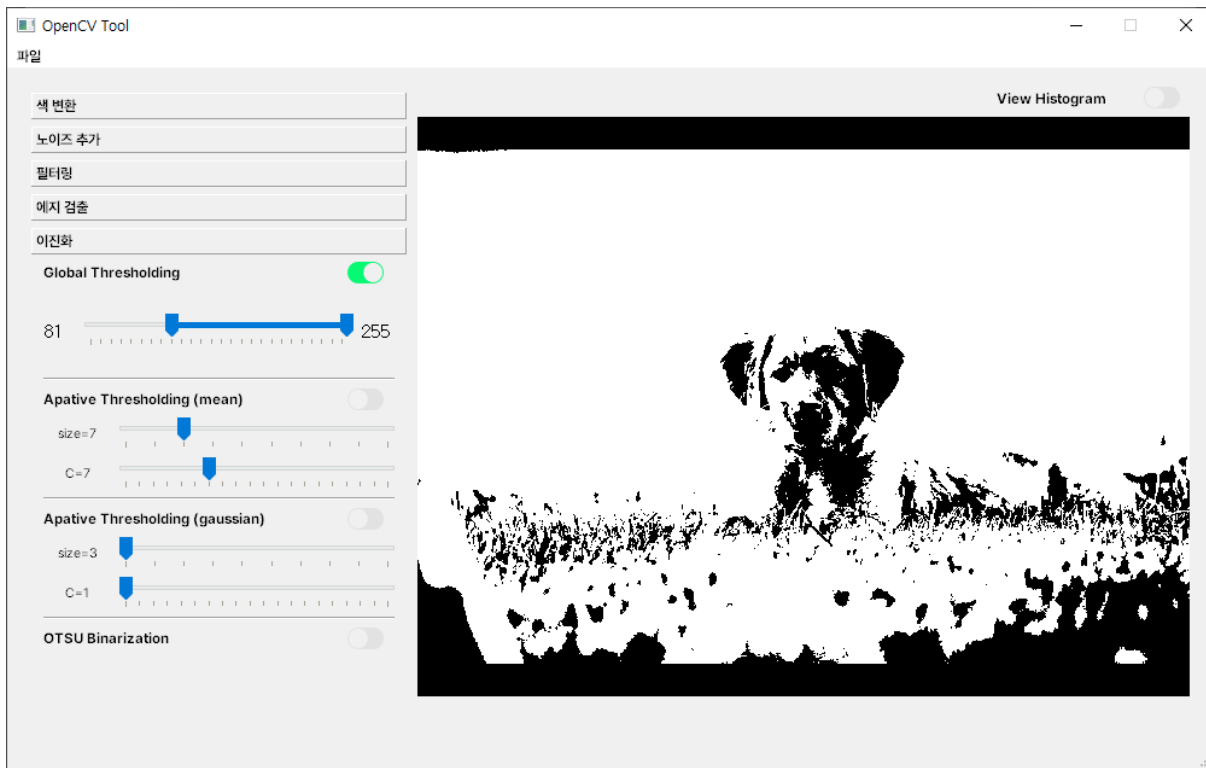


3주차 활동 내용

- PyQt에서 OpenCV 기능 구현
- 파라미터 상세 조정 가능

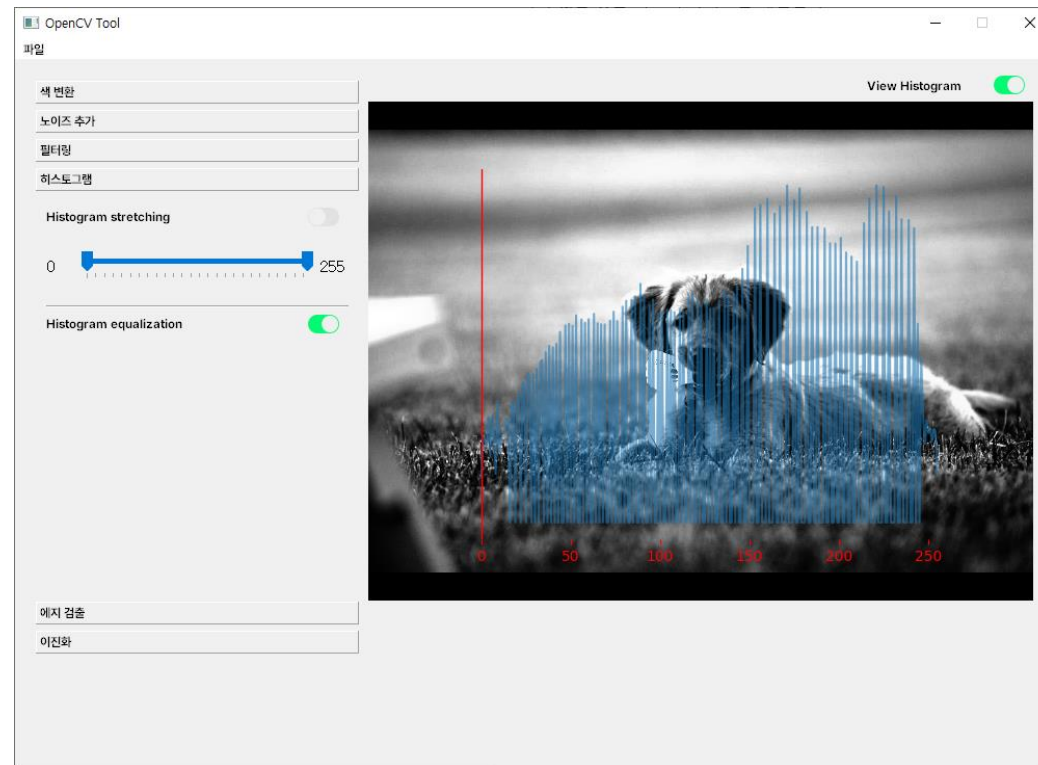
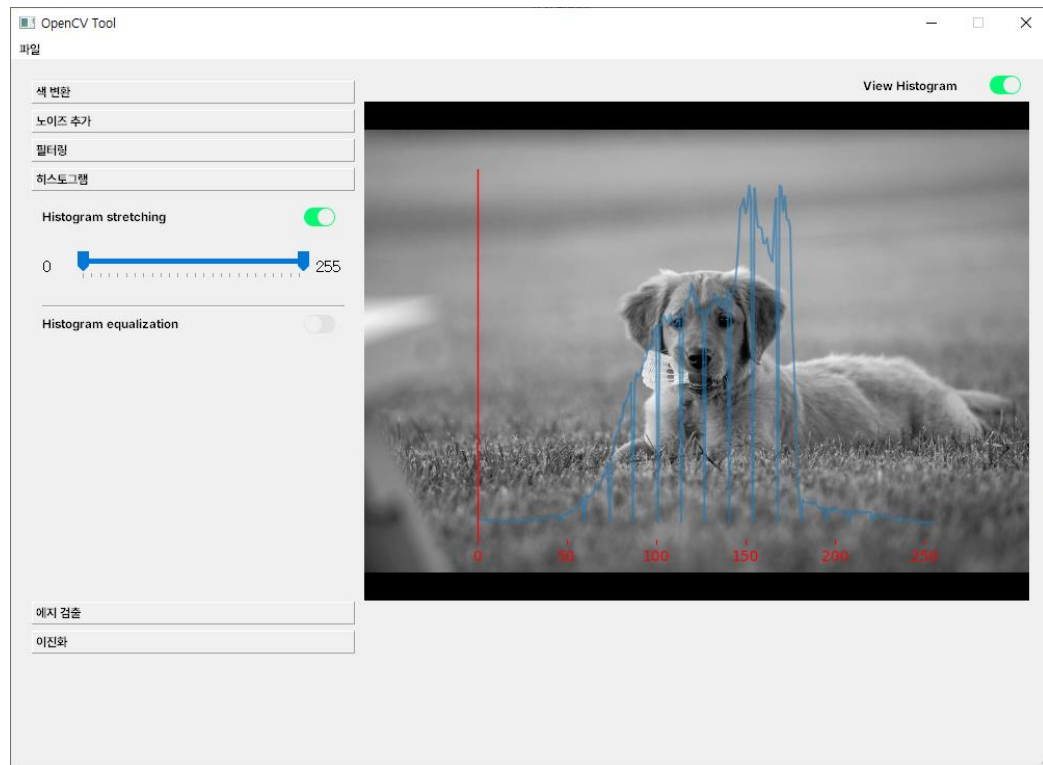


3주차 활동 내용



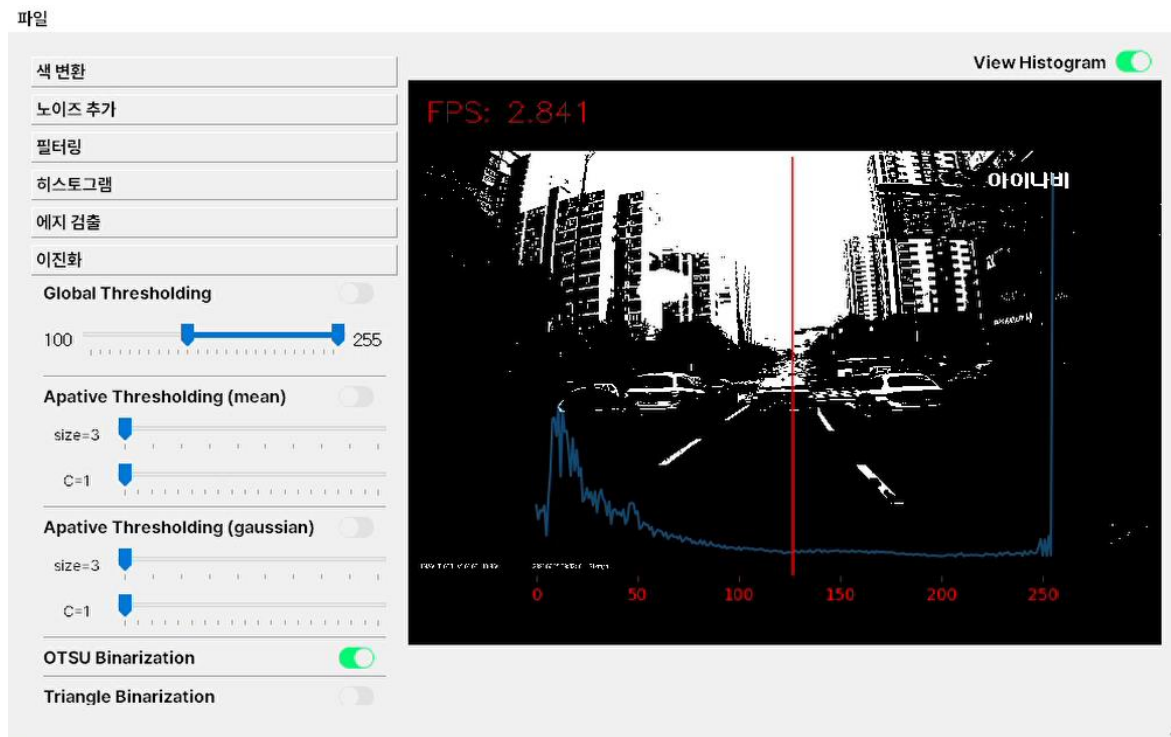
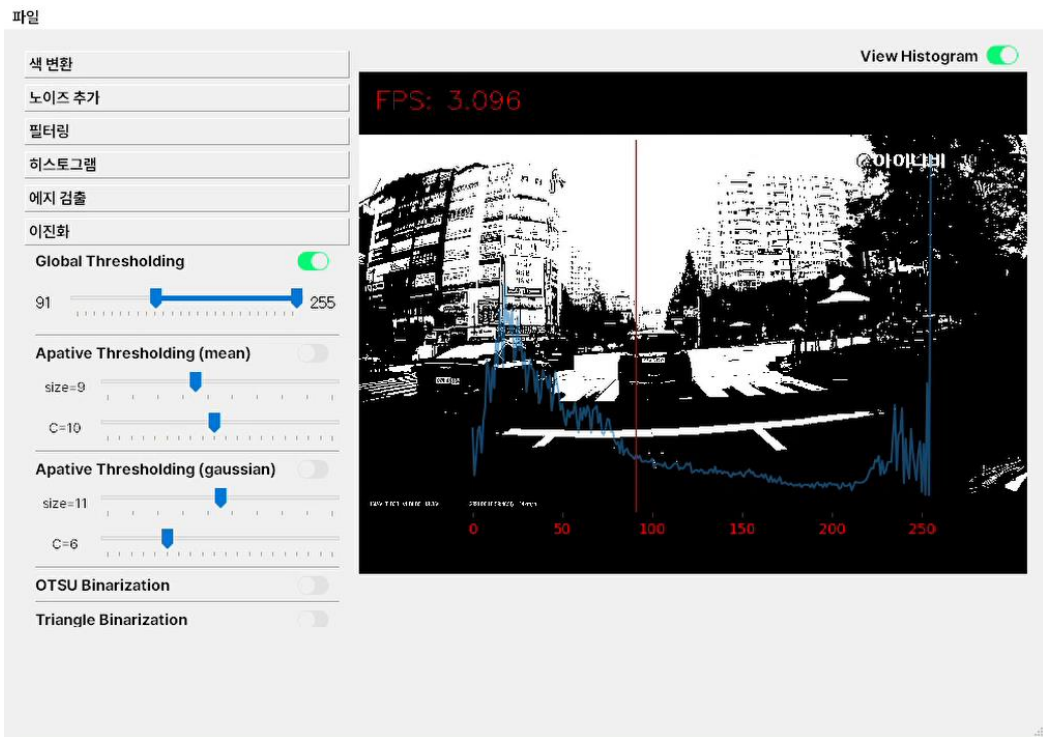
3주차 활동 내용

➤ 영상의 히스토그램 분포를 볼 수 있도록 하였다.



3주차 활동 내용

- 이진화 임계값 적용을 실시간으로 볼 수 있도록 하였다.



4주차 활동 계획

- 라벨링 자동화 프로그램을 사용하여 라벨링 33,709장
- YOLO → 머신러닝 매뉴얼 작성으로 확대



머신 러닝 수행 절차

☰ 리스트

- 📄 1. 문제 정의
- 📄 2. 데이터 수집
- 📄 4. 데이터 전처리
- 📄 3. 데이터 분석 및 특성 추출
- 📄 5. 알고리즘 선택
- 📄 6. 학습
- 📄 7. 모델 평가 및 하이퍼 파라미터 조정
- 📄 8. 모델 배포 및 유지보수

+ 새로 만들기

QnA