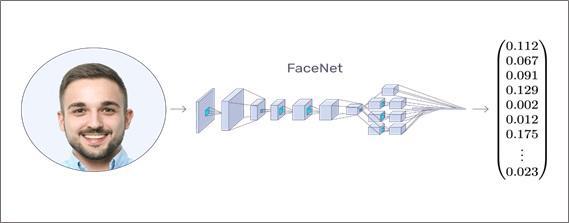
<모델링 및 알고리즘>

1. Face Recognition(FR)

**1.1 Face Recognition**

얼굴을 활용해 개인의 신분을 식별 또는 확인하는 얼굴 분석이다. 이는 이미지의 얼굴 특성 식별 및 측정을 통해 작동한다.  
얼굴 인식 기술은 이미지 또는 동영상에서 사람의 얼굴을 식별하여 두 이미지의 얼굴이 동일한 사람인지를 판단하는데 사용된다.

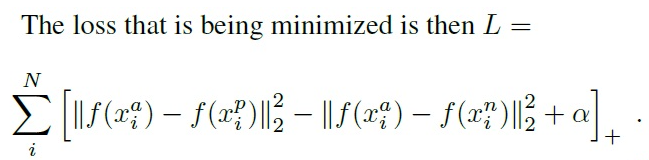
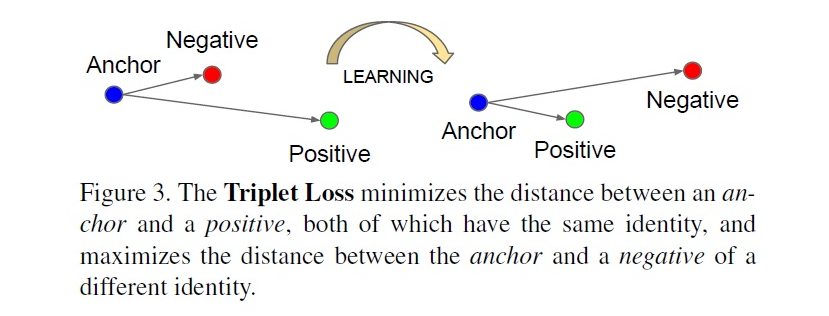
**선정 모델** : facenet\_pytorch



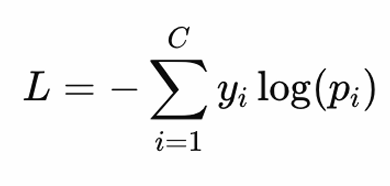
**1.2 모델 설명**

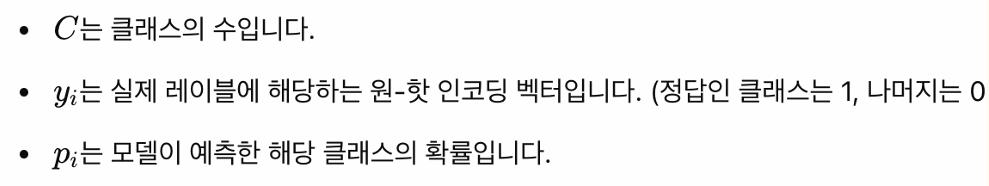
facenet\_pytorch는 각각의 얼굴 이미지를 128차원으로 임베딩하여 유클리드 공간에서 이미지간의 거리(가까울 수록 유사도가 높음)를 통해 분류하는 모델이다. 간단히 말하면 얼굴 사진에서 그 사람에 대한 특징값을 구해주는 모델이다. 모델을 학습하는 방법은 다음과 같다.

* Anchor(기준 사람 이미지), Positive(같은 사람 이미지), Negative(다른 사람 이미지)를 지정해 놓는다.
* Embedding된 값들의 유클리드 거리를 구해 아래 그림과 같이 Anchor와 Positive의 거리를 가깝게, Anchor와 Negative의 거리는 멀게 학습하는 triplet loss 함수를 사용하여 loss값을 최소화하도록 학습한다.
* Triplet Loss 함수



* Cross Entropy Loss 함수





**1.3 알고리즘 개요**

1. **Embedding**
   * FaceNet 모델을 통해 사용자의 얼굴 이미지를 고차원 벡터로 변환하여 얼굴의 고유한 특성을 압축하여 표현.
   * 벡터간의 유클리드 거리를 통해 두 얼굴을 비교하여 동일 인물인지 아닌지 판단.
2. **Cross Entropy Loss Function**
   * 초기에 FaceNet 모델은 Triplet Loss 함수를 사용하여 학습을 진행했으나, 성능 문제로 인해 Cross Entropy Loss 함수로 변경하여 학습을 진행함.
   * Cross Entropy Loss는 모델이 각 클래스(사람의 얼굴)마다 확률을 예측하도록 하고, 예측된 확률과 실제 정답 간의 차이를 최소화하도록 학습시킴.
3. **InceptionResnetV1 모델**
   * 깊은 Feature Extractor 능력을 활용한 Inception 모델과 overfitting에 강한 ResNet 모델의 특성을 함께 적용한 모델로 facenet\_pytorch에서 쓰이는 모델.

**1.4 알고리즘 flow**

* vggface2로 pretrained된 InceptionResnetV1 모델을 load.
* resnet = InceptionResnetV1(
* classify=True,
* pretrained='vggface2',
* num\_classes=len(dataset.class\_to\_idx)
* ).to(device)
  + classify=True로 설정하여 output을 class별 확률값으로 반환. (softmax)
* Fine Tuning.
* resnet.train()
* training.pass\_epoch(
* model, loss\_fn, train\_loader, optimizer, scheduler,
* batch\_metrics=metrics, show\_running=True, device=device,
* writer=writer)
  + 미리 지정해놓은 device, model, dataset, dataloader, parameters(optimizer, learning rate, LR scheduler, loss\_function)를 facenet\_pytorch 모듈에 있는 training.py에 정의되어 있는 pass\_epoch함수의 인자로 넣어줌.
* Embedding / Loss 계산
* for i\_batch, (x, y) in enumerate(loader):
* x = x.to(device)
* y = y.to(device)
* y\_pred = model(x)
* loss\_batch = loss\_fn(y\_pred, y)
  + facenet\_pytorch 모듈의 training.py 안의 pass\_epoch 함수 내부 코드.
  + Dataloader로 받아온 이미지들을 모델 인자로 받아온 InceptionResnetV1 모델에 넣어서 Embedding값을 추출.

**1.5 성능**

* 1차 학습 (자체 fine tuning)

Batch size: 512

Epochs: 10

Learning rate : 0.001

Optimizer: Adam

텍스트, 도표, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명Loss함수: nn.TripletMarginLoss(margin=1.0, p=2)

**Train Loss** : 0.56062 / **Validation Loss** : 0.48366  
**Train Accuracy** : 0.77160 / **Validation Accuracy** : 0.82940  
1epoch 당 학습 소요 시간 : 197.26s ~ 202.76s  
총 학습시간 : 2011.52s

* 2차 학습 (자체 fine tuning)

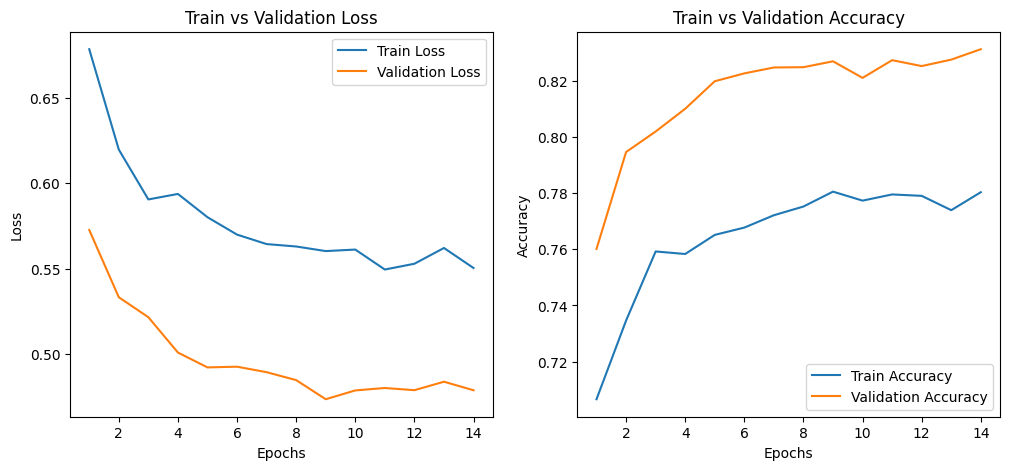
Batchsize: 512

Epochs: 20 (early stopping 적용)

Learning rate : 0.0005

Optimizer: Adam

Loss함수: nn.TripletMarginLoss(margin=1.0, p=2)



**Train Loss** : 0.55031 / **Validation Loss** : 0.47857   
**Train Accuracy** : 0.78030 / **Validation Accuracy** : 0.83120   
1epoch 당 학습 소요 시간 : 175.26s ~ 219.28s   
총 학습시간 : 2666.27s

* 3차 학습 (Github의 facenet\_pytorch fine tuning사용)

Batchsize : 64

Epochs : 20

Num\_workers : 4

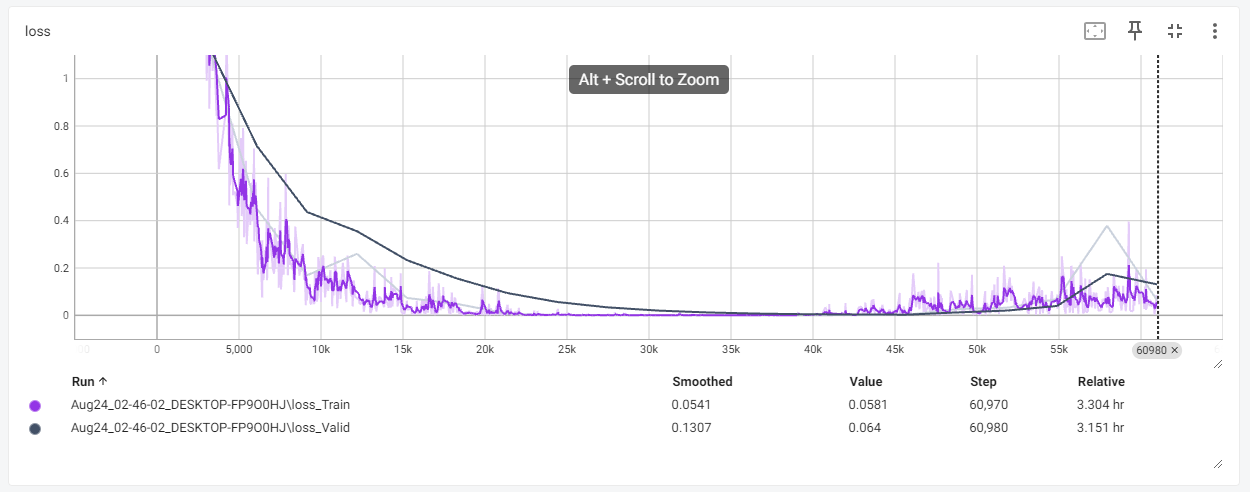
Learning rate : 0.001

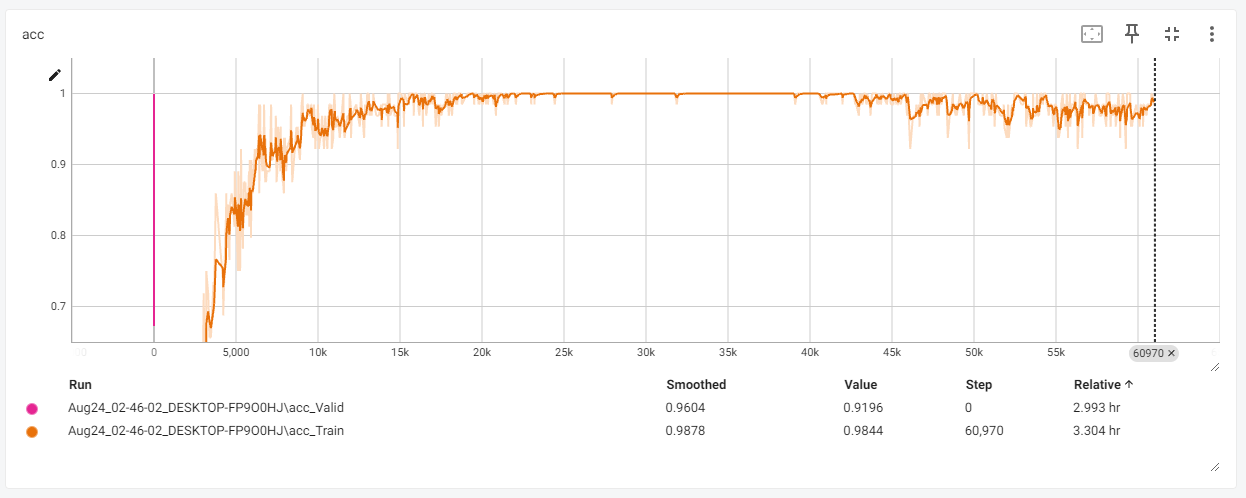
Optimizer : Adam

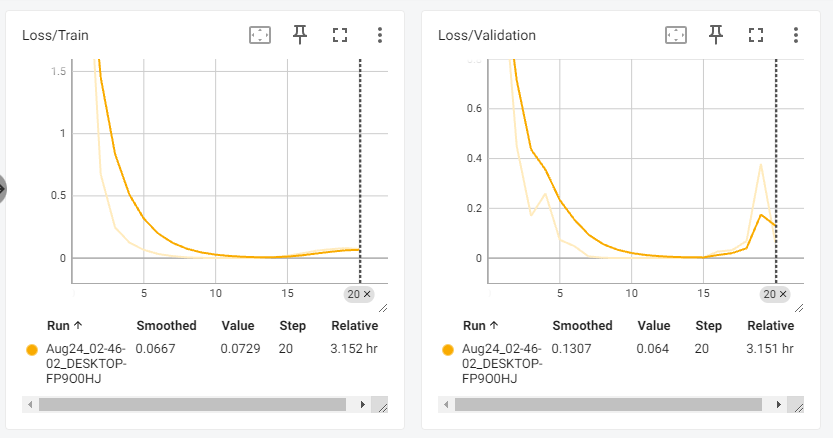
Learning scheduler = CosineAnnealingLR(optimizer, T\_max=10)

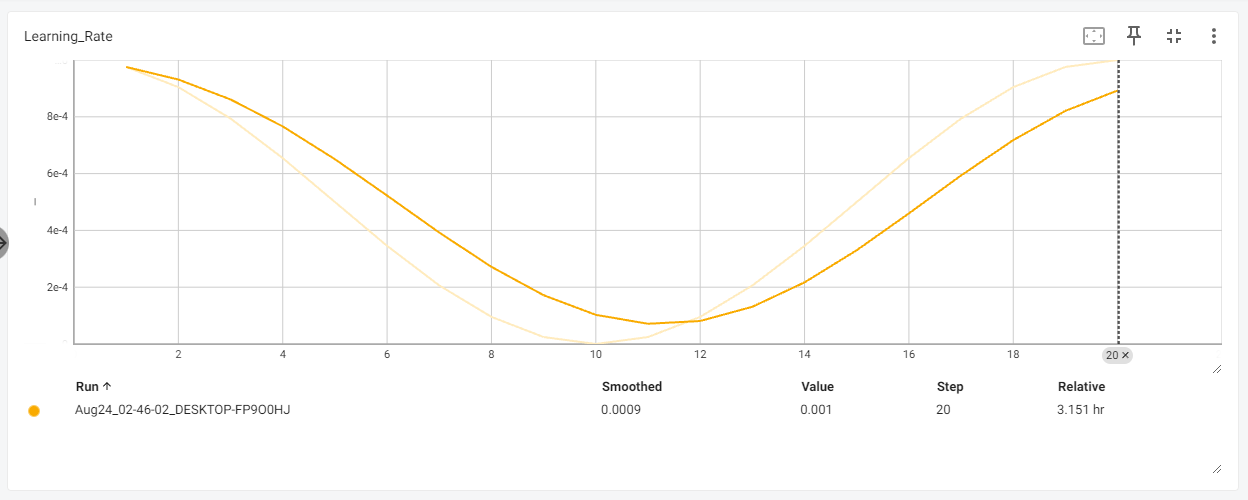
Loss 함수 = torch.nn.CrossEntropyLoss()

총 걸린시간 : 200분







  
  
evaluation 결과값 =========================================================

Valid | 1073/1073 | loss: 0.0007 | fps: 1021.1518 | acc: 0.9999

Test Loss: 0.001 fps: 1021.152 acc: 1.000 =========================================================

* 정확도가 100% 가 나와서 모델 학습을 다시 시키기로 함.
* 모델 데이터셋을 더 많이 늘리고 learning rate를 더 낮춰서 학습을 시키기로 함.
* 4-1차학습 (데이터셋 2배로 늘려서 학습)

Batchsize : 64

Epochs : 20

Num\_workers : 4 (기존 0 )

Learning rate : 0.001

Optimizer : Adam

Learning scheduler : Multi Step LR(optimizer, [5, 10])

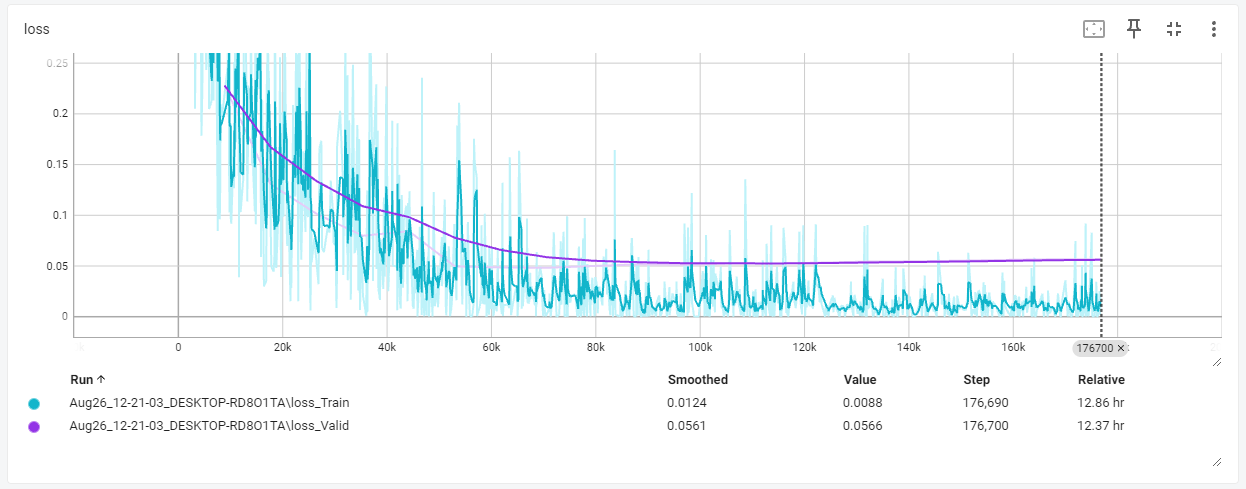
Loss 함수 : torch.nn.CrossEntropyLoss() ========================================================

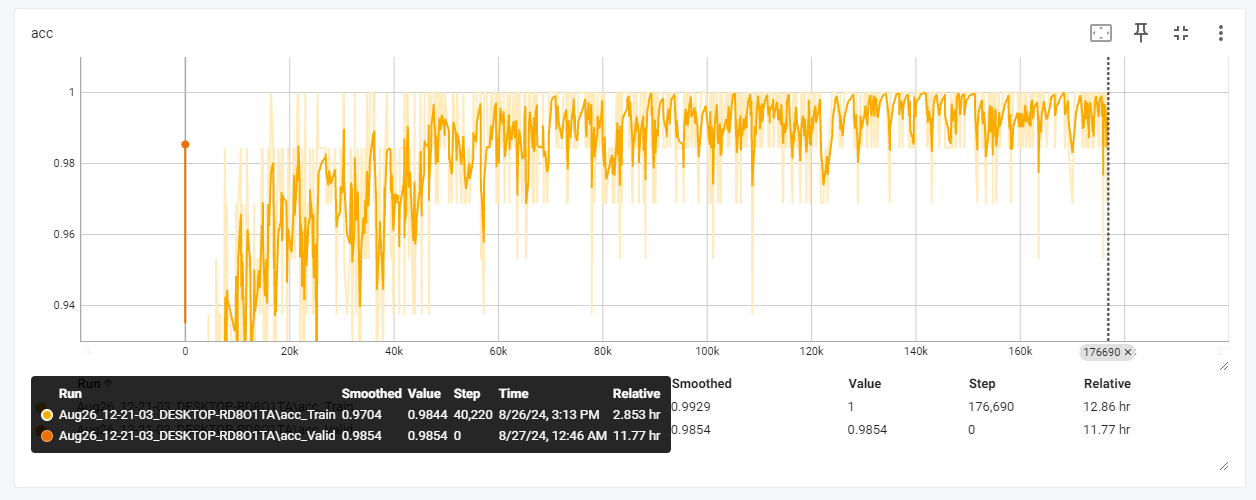
Train | 8835/8835 | loss: 0.1148 | fps: 381.8197 | acc: 0.9531

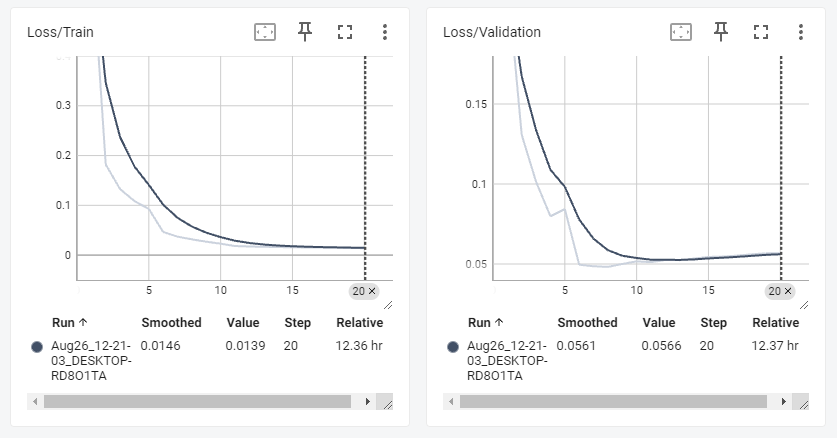
Valid | 5300/5300 | loss: 0.0601 | fps: 1183.0337 | acc: 0.9818

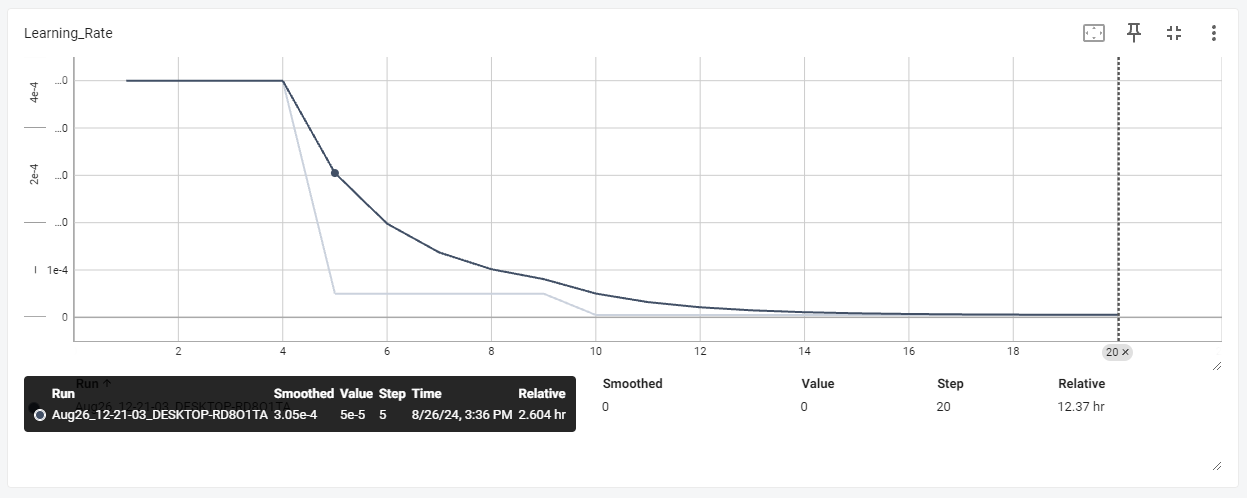
New best model saved with validation loss: 0.060 (7 epoch)

========================================================









evaluation 결과값 =========================================================

Valid | 5393/5393 | loss: 0.0601 | fps: 818.4009 | acc: 0.9818

Test Loss: 0.060 fps: 818.401 acc: 0.982

=========================================================

* 4-2차학습 (데이터셋 2배로 늘려서 학습)

Batchsize : 64

Epochs : 20

Num\_workers : 4 (기존 0 )

Optimizer : optim.Adam(resnet.parameters(), lr=0.001)

Learning scheduler : CosineAnnealingLR(optimizer, T\_max=10)

Loss 함수 : torch.nn.CrossEntropyLoss()

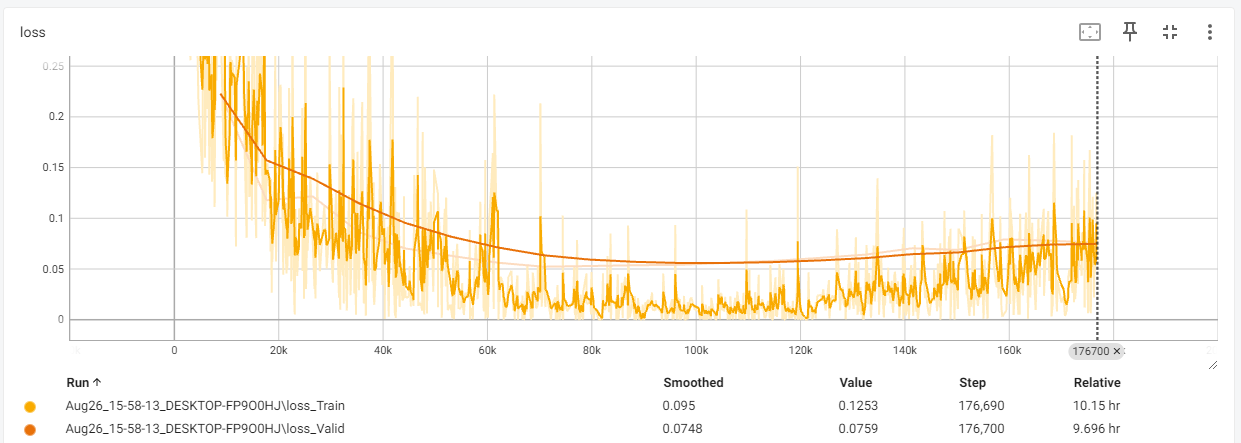
===================================================

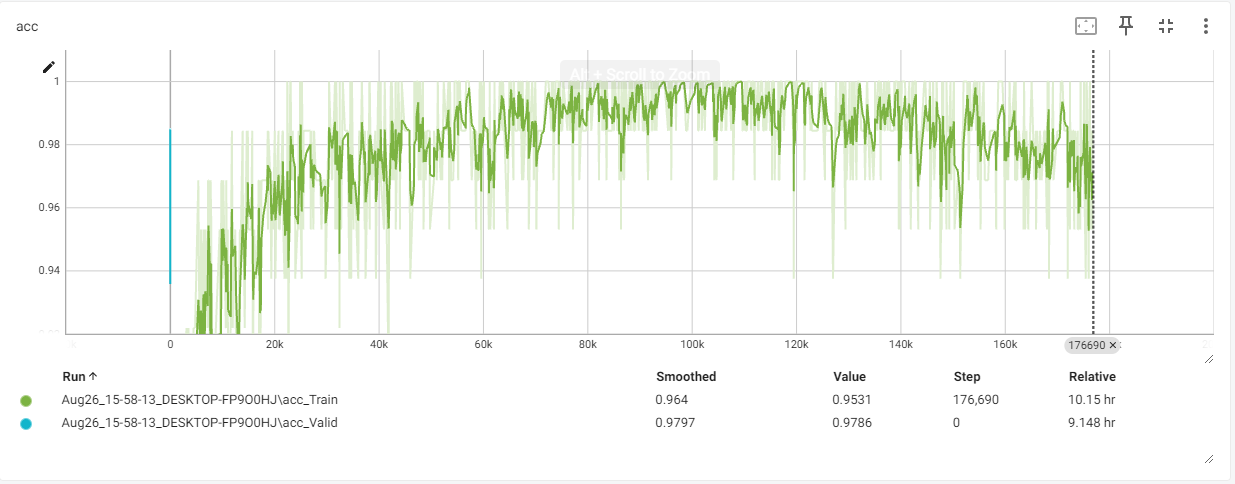
Train | 8835/8835 | loss: 0.0276 | fps: 381.8197 | acc: 0.9885

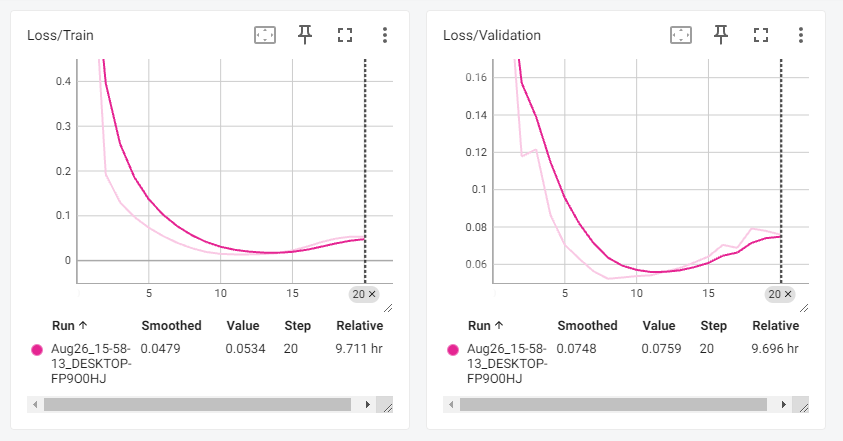
Valid | 5300/5300 | loss: 0.0523 | fps: 1183.0337 | acc: 0.9843

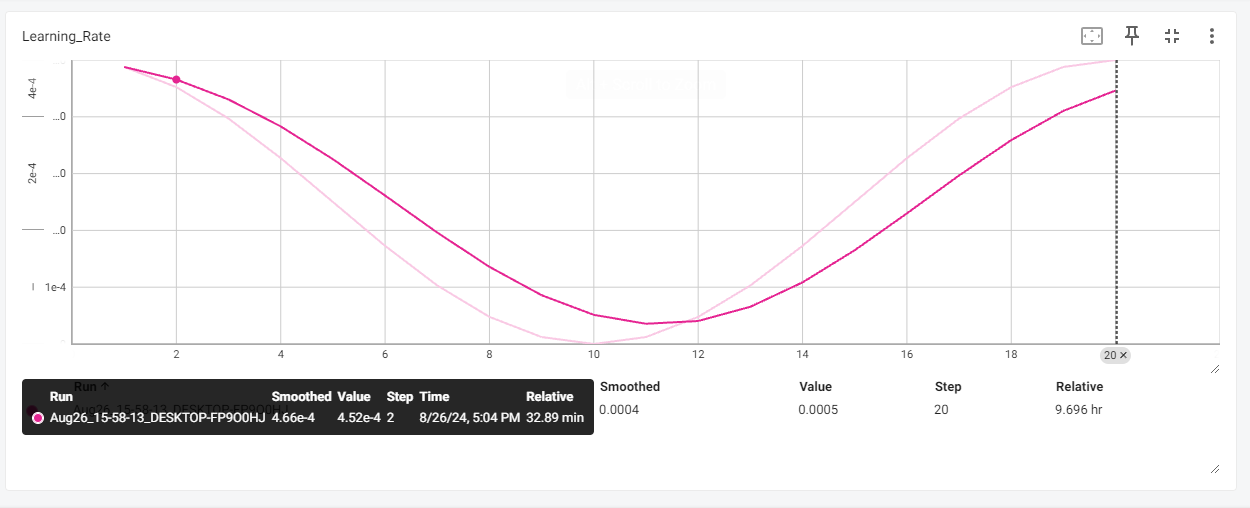
New best model saved with validation loss: 0.052 (8 epoch)

===================================================









evaluation 결과값 =========================================================

Valid | 5393/5393 | loss: 0.0660 | fps: 810.2586 | acc: 0.9814

Test Loss: 0.066 fps: 810.259 acc: 0.981

=========================================================

* 8 epoch 에서 모델이 최종 저장되었고 evaluation 결과도 98.14로 매우 좋게 나옴.

2. Face Pose (FP)

* **Face Pose** : Face Recognition을 하기 전에 User의 얼굴을 등록하는 과정이다. 사용자의 얼굴 이미지를 아래의 조건과 같이 저장하기 위해 웹캠을 통해 기울여야 하는 각도를 표시를 한다. (아래의 숫자는 각도를 의미)
  + yaw(좌우) : -60, -40, -20, 0, 20, 40, 60
  + pitch(상하) : -20, -30, 20, 30
  + roll(좌우 꺾임) : -60, -40, -20, 20, 40, 60
* **알고리즘 Flow**

텍스트, 스크린샷, 흑백, 모노크롬이(가) 표시된 사진

자동 생성된 설명

**3. Face Classification (FC)**

**3.1 Face Classification 개요**  
Face Classification은 얼굴 이미지를 인종, 성별, 나이와 같은 속성으로 분류하는 작업이다.

이는 얼굴 인식 시스템이 다양한 인구 집단에서 편향 없이 정확하게 작동하도록 보장하기 위해 필수적이다.

이러한 분류는 특히 인종, 성별, 나이 분포의 불균형이 존재하는 경우에 중요한 역할을 한다.

**3.2 데이터셋 및 모델 설명**  
Face Classification에 사용된 모델은 **res34\_fair\_align\_multi\_7\_20190809**이다.

이 모델은 FairFace 데이터셋을 기반으로 학습되었다. FairFace 데이터셋은 기존의 얼굴 이미지 데이터셋들이 백인 얼굴에 편중된 문제를 해결하기 위해 만들어졌으며, 총 108,501개의 이미지로 구성되어 있다.

이 데이터셋은 백인, 흑인, 인도인, 동아시아인, 동남아시아인, 중동인, 라틴계 등 7개의 인종 그룹으로 균형 있게 구성되어 있다.

모델 학습 시 ResNet-34 아키텍처를 기반으로 하여 인종, 성별, 나이 분류 작업을 수행한다.

데이터셋의 균형 잡힌 구성을 통해 모델이 특정 인종이나 성별에 치우치지 않고 일관된 성능을 발휘할 수 있도록 설계되었다. 모델 학습은 ADAM 옵티마이저를 사용하며, 학습률은 0.0001로 설정된다.

FairFace 데이터셋의 다양한 인종, 성별, 나이 정보를 반영하여 모델이 각 그룹에서 균등한 성능을 발휘하도록 한다.

**3.3 알고리즘 개요**

**res34\_fair\_align\_multi\_7\_20190809** 모델은 다음과 같은 과정으로 학습된다:

* **데이터 전처리**: 입력 이미지들은 ResNet-34 모델에 맞게 정렬 및 크기 조정된다. FairFace 데이터셋의 다양성을 고려하여 각 이미지에 대해 적절한 전처리 작업이 이루어진다.
* **인종 분류**: 모델은 학습된 특징을 바탕으로 주어진 얼굴 이미지를 7개의 인종 그룹 중 하나로 분류한다. FairFace 데이터셋은 이 과정을 위해 각 인종 그룹에 대한 충분한 데이터를 제공한다.
* **성별 분류**: 얼굴 이미지는 남성 또는 여성으로 분류되며, 모델은 성별에 따른 편향을 최소화하도록 학습된다.
* **나이 분류**: 모델은 얼굴 이미지를 연령대별로 분류하며, 모든 연령대에 걸쳐 균형 잡힌 정확도를 달성한다.

**3.4 성능 평가**  
모델의 성능은 FairFace 데이터셋을 사용한 학습과 다양한 크로스 데이터셋 평가를 통해 검증된다. **res34\_fair\_align\_multi\_7\_20190809** 모델은 FairFace 데이터셋을 통해 학습된 결과, 인종, 성별, 나이에 걸쳐 높은 일관성과 정확도를 보여준다.

특히, 다양한 외부 데이터셋에서 성능을 테스트한 결과, 모델이 새로운 데이터에 대해서도 우수한 일반화 성능을 발휘함이 확인되었다.

FairFace 데이터셋은 학습 과정에서 인종별, 성별, 나이별로 균형 잡힌 분포를 제공하므로, 모델이 특정 그룹에 편향되지 않고, 공정하고 일관된 성능을 유지할 수 있다. 이는 인종과 성별의 균형 잡힌 정확도와 낮은 성능 격차로 나타난다.