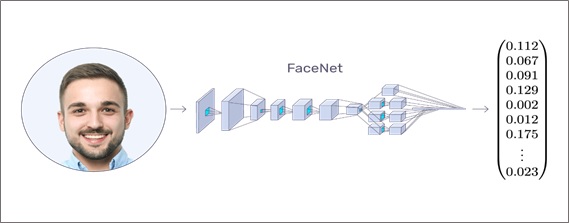
# <모델링 및 알고리즘>

# 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.

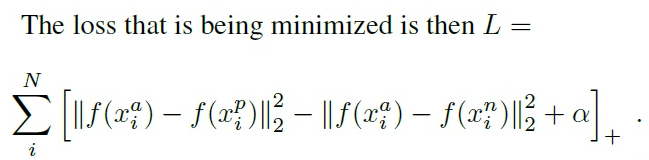
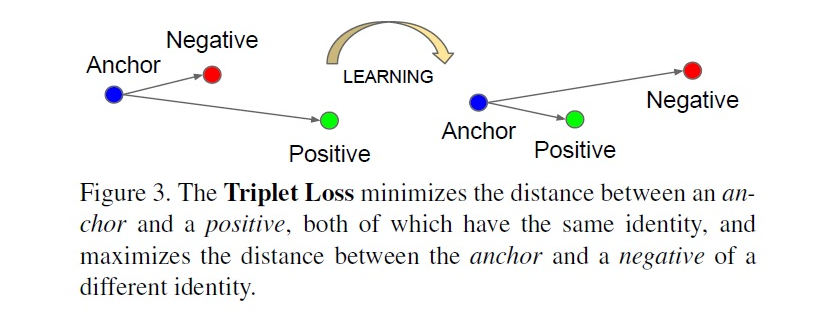
### 선정 모델 : facenet\_pytorch



### 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.



### 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. **Triplet Loss Function**
   * FaceNet모델은 Triplet Loss 함수를 사용하여 학습을 진행.
   * Anchor(사람 이미지), Positive(동일 인물), Negative(다른 인물)
   * 얼굴에 대한 임베딩간의 거리를 구분하기 위해 Anchor-Positive 거리는 가까워지도록, Anchor-Negative 거리는 멀어지도록 학습.
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.
* model = InceptionResnetV1(pretrained='vggface2').to(device)
  + classify=False로 설정하면 output을 embedding으로, classify=True로 설정하면 output을 class별 확률값으로 반환
* Fine Tuning.
* activate\_grad = False  
  for name, param in model.named\_parameters():  
   if 'last\_linear' in name:  
   activate\_grad = True  
   param.requires\_grad = activate\_grad
  + last\_linear layer 이후의 상단 layer들은 requires\_grad=True
* Embedding.
* anchor\_output = model(anchor)  
  positive\_output = model(positive)  
  negative\_output = model(negative)
  + embedding 값으로 반환
* Loss 계산.
* loss = loss\_fn(anchor\_output, positive\_output, negative\_output)  
  loss.backward()
  + epoch을 돌며 loss값을 최소화

### 1.5 성능

learning rate=0.005

optimizer : Adam  
epochs=10  
early stop=5

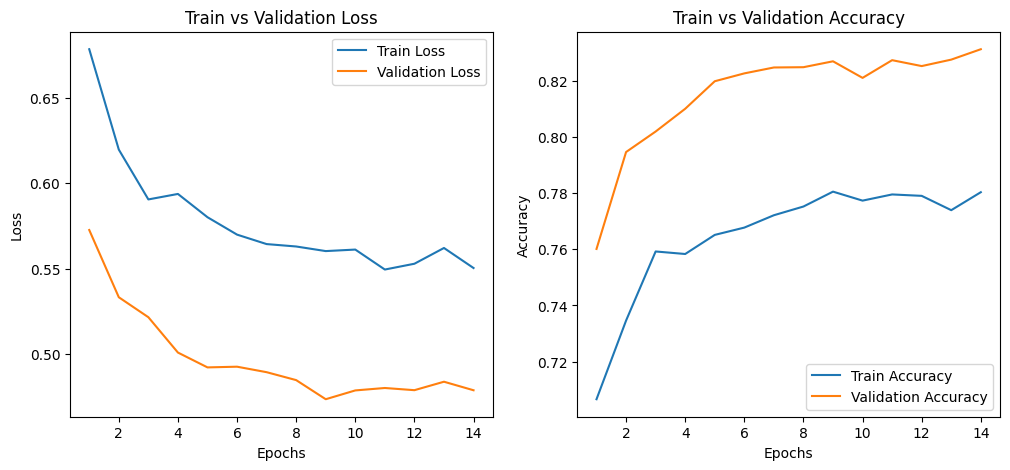
텍스트, 도표, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

**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

learning rate : 0.001

optimizer : Adam  
epochs=20 (early stop in 14 epoch)   
early stop=5



**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

learning rate : 0.001   
optimizer : Adam

epochs=50 (5 epoch때 중단)

train set : 400,000 / valid set : 140,000 / test set : 140,000

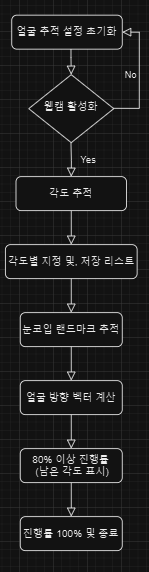
**Train Loss** : 0.56393 / **Validation Loss** : 0.48288   
**Train Accuracy** : 77.58625 / **Validation Accuracy** : 82.11143   
1epoch 당 학습 소요 시간 : 10,492.12s ~ 10,761.27s   
5epoch 학습시간 : 52,993.42 s (약 14시간 40분)

### 1.6 결과

**Train Loss**는 0.01031 감소, **Validation Loss**는 0.00509 감소했다.  
**Train Accuracy**는 0.0087 증가, **Validation Loss**는 0.0018 증가했다.  
Hyper parameter는 learning rate를 0.005 → 0.001, epoch을 10 → 20 변경했다.  
Loss와 Accuracy가 아주 미세하게 성능 향상이 있었지만, Hyper Parameter의 변화때문인지 epoch의 차이때문인지는 아직은 알 수 없다.

## 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**

****

## 3. Face Classification (FC)