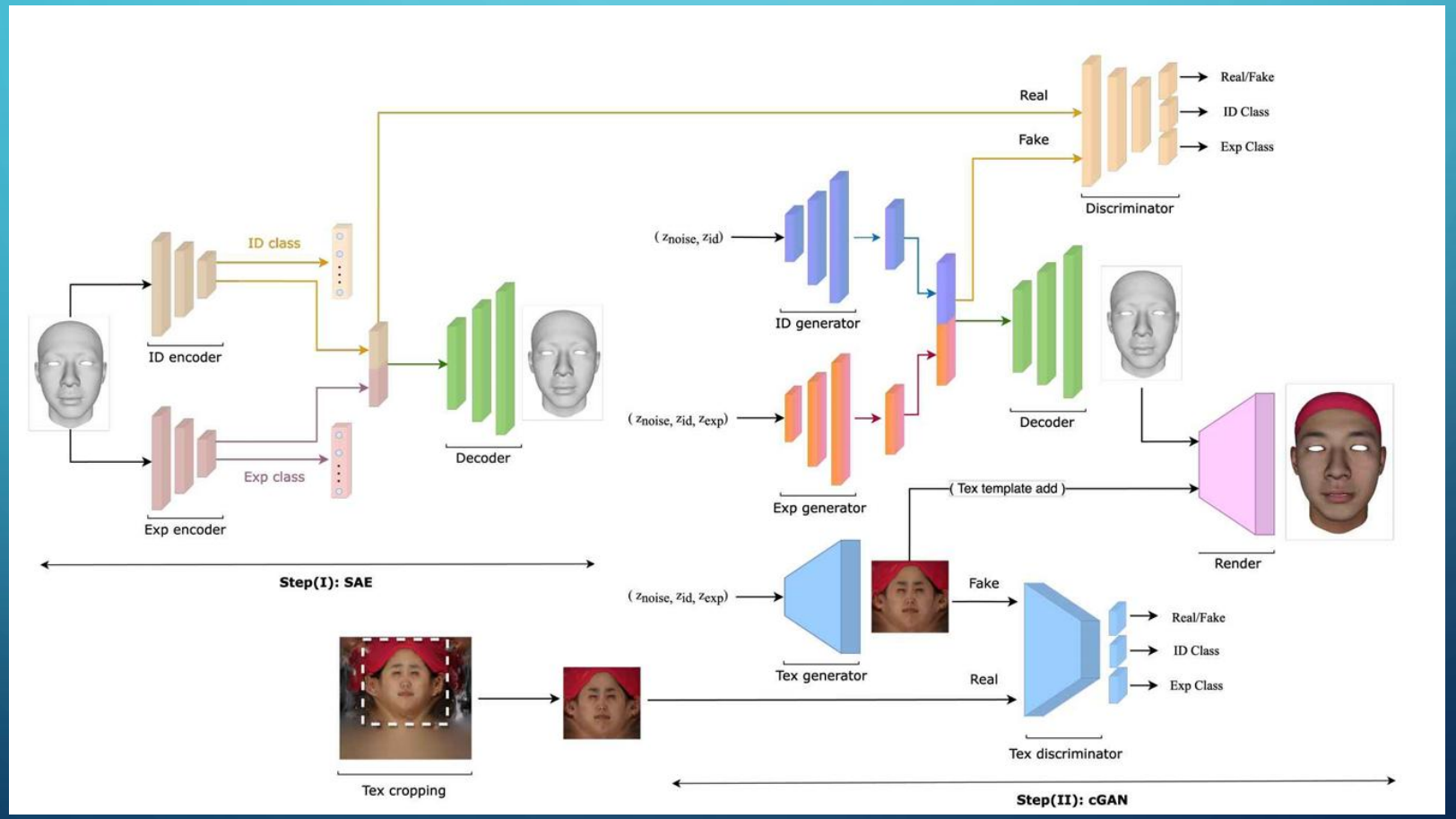
3DFACECAM-MAIN 코드 분석

논문: **CONTROLLABLE 3D GENERATIVE ADVERSARIAL FACE MODEL VIA DISENTANGLING SHAPE AND APPEARANCE**

FACEBOOK/META

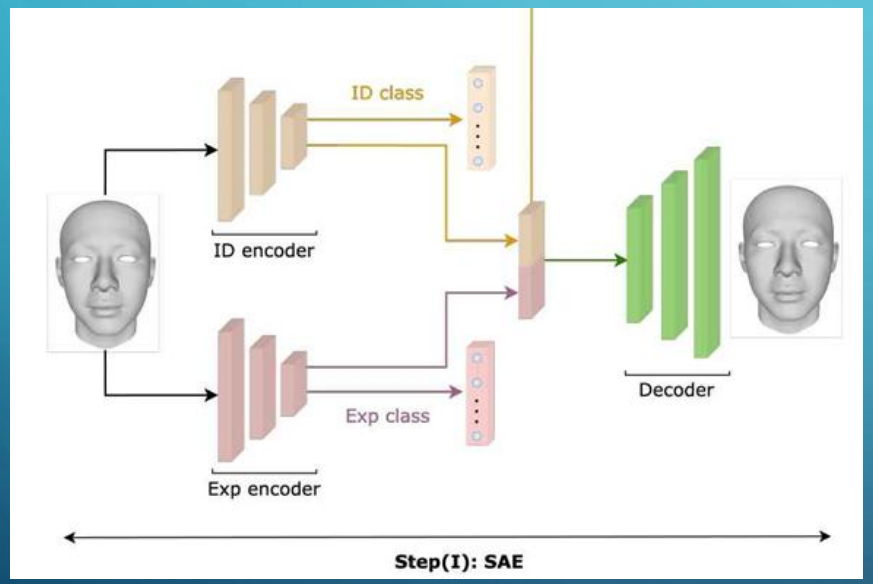
전체 구조도


Real
Fake
Real/Fake
ID Class
Exp Class
Discriminator
ID class
ID encoder
Exp class
Exp encoder
Decoder
Step(I): SAE
(znoise, zid)
ID generator
(znoise, zid, zexp)
Exp generator
Decoder
(Tex template add)
Render
(znoise, zid, zexp)
Tex generator
Fake
Real
Real/Fake
ID Class
Exp Class
Tex discriminator
Tex cropping
Step(II): cGAN

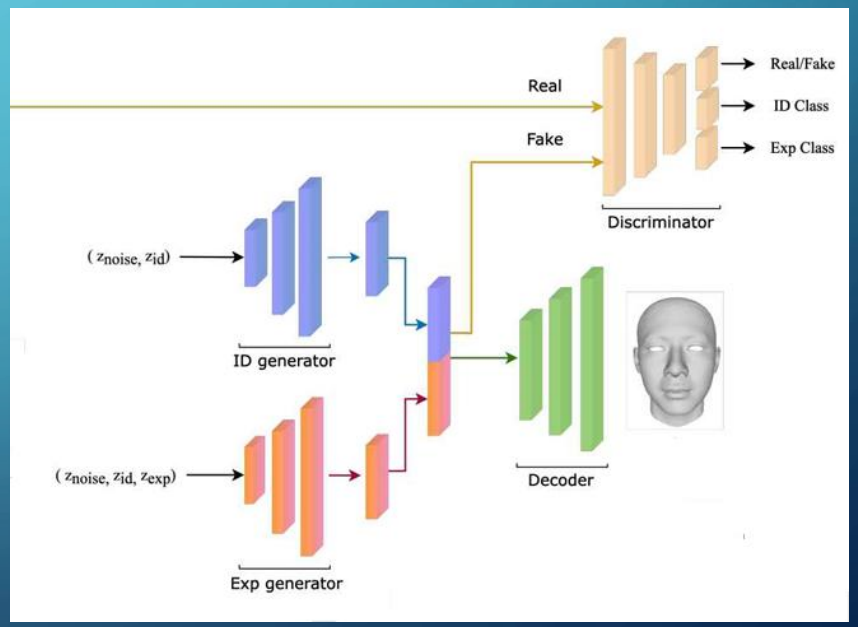
SHAPE MODEL

SHAPE MODEL

AE



GAN3D



SHAPE_MODEL/ARCHITECTURES.PY

- 전체 구성 요약

클래스 이름	설명
`Encoder_identity`	사람의 **identity (정체성)**을 인코딩 (즉, 얼굴의 개인적인 고유 특징 추출)
`Encoder_expression`	얼굴의 **expression (표정)**을 인코딩
`Decoder`	인코딩된 identity + expression 특징을 바탕으로 다시 원래 얼굴 데이터 복원
`Generator`	랜덤 latent vector로부터 identity 혹은 expression 특징 생성
`Discriminator`	GAN 구조에서 generator가 만든 특징이 진짜 같은지 판별
`generator_network`	identity와 expression을 각각 생성하는 두 개의 generator 묶음

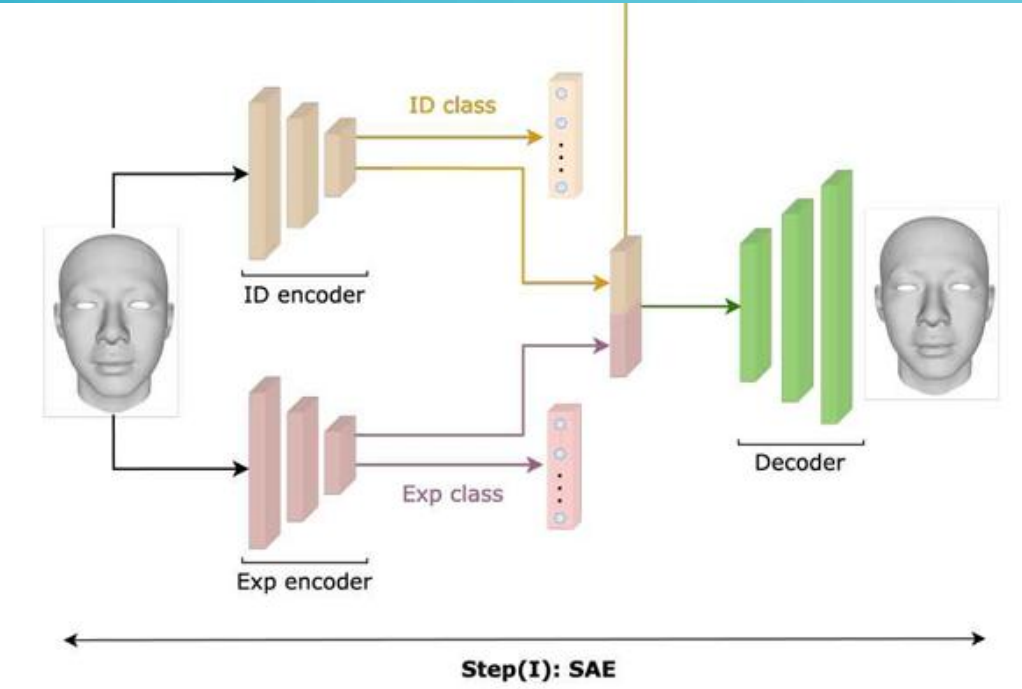
SHAPE_MODEL/MESH_OBJ.PY

정리: 전체 동작 원리

구성 요소	설명
`__init__()`	`.obj` 파일을 파싱하여 정점, 면, 법선 등을 메모리에 저장
`create()`	외부에서 넘겨받은 리스트로 직접 메쉬 구성
`get_adjacent()`	정점 인접 리스트 계산 (지역 연결 정보 활용 가능)
`export()`	다시 `.obj` 형식으로 저장 (텍스처/색상 포함 가능)
`read_mtl()`	`.mtl` 파일 파싱하여 재질 정보 저장

AE

AE/MODELS/AE.PY



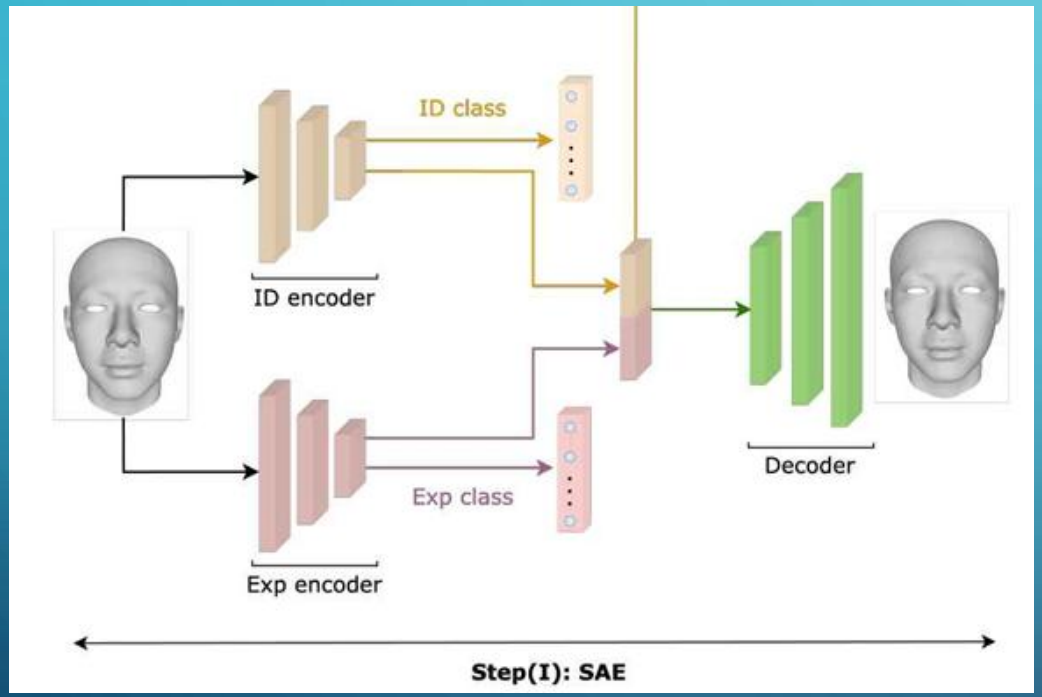
1. Network 클래스
오토인코더 :

•Encoder_identity:
아이덴티티(개인 식별 정보) 특성 인코딩

•Encoder_expression:
표정 특성 인코딩

•Decoder:
위 두 인코딩을 기반으로 원본 입력을 복원

AE/MODELS/AE.PY



2. AE 클래스

오토인코더의 학습 및 평가 루프를 정의

주요 구성 요소

- self.device: GPU 사용 ("cuda:7")
- self.model: 앞서 정의한 네트워크 구조
- self.data: 사용자 정의 데이터셋 클래스
- self.optimizer: Adam 옵티마이저 사용

2. AE 클래스

학습 함수: train(self, epoch)

1.데이터 불러오고 GPU로 전송

2.모델 순전파 → 복원된 결과, ID/표정 예측 값 얻기

3.손실 계산(def loss_function 이용):
- **복원 손실 (L1)**: 입력과 출력 mesh 비교
- **ID 분류 손실 (CrossEntropy)**: 실제 ID vs 예측
- **Expression 분류 손실**: 실제 표정 vs 예측

4.역전파 및 옵티마이저 갱신

5.주기적으로 모델 저장

2. AE 클래스 동작 요약

- **입력 데이터**는 3D 얼굴 메시(mesh)이며, 78951차원의 플랫 벡터
- 오토인코더는 이를 **ID**와 **표정 특성**으로 인코딩하고 다시 복원
- 학습 시:
  - 재구성 오류를 줄이고,
  - ID/표정 분류 성능도 동시에 최적화
- 주기적으로 모델을 저장하여 추후 텍스처/메시 생성 등에 활용

AE/GEN_REDUCED_DATA.PY

동작 원리 요약

1.모델 로딩

- Identity encoder, Expression encoder, Decoder를 정의하고 CPU에 로딩함.
- 학습된 파라미터 (./checkpoints/.../2000)를 불러옴.

2.데이터 로딩

- ./data/displace_data.npy 파일을 불러오며, 각 row는 [78951차원 displacement + ID + Expression]의 구조로 되어 있음.

3.인코딩 (Latent Representation 생성)

- 각 3D 얼굴 데이터를 인코더에 넣어, identity (100차원) + expression (30차원)의 latent 벡터를 추출함.
- label은 1-indexing → 0-indexing으로 조정.

4.결과 저장

- [z_id, z_exp, id_label, exp_label] 형태의 132차원 벡터로 저장하여, .npy로 저장함.
- 이 결과는 나중에 GAN 등의 downstream 모델에서 학습 데이터로 활용됨.

AE/DATASETS.PY

동작 원리 요약

- **데이터 로딩**
  - args.dataset에 지정된 .npy 파일을 로딩해.
  - 각 row는 [78951 features (3D mesh displacement) + identity label + expression label]로 구성되어 있음.

- **Tensor 변환**
  - numpy array를 PyTorch Tensor로 변환
  - 이때 features는 tensor_x, ID는 tensor_y, expression은 tensor_z로 나눠짐

- **TensorDataset 생성**
  - 세 가지 tensor를 묶은 TensorDataset을 만들어.
  - 이 데이터셋은 __getitem__ 시 feature, id, expression을 함께 반환함.

- **DataLoader 구성**
  - 학습용 train_loader를 생성.
  - shuffle=True로 데이터를 섞어서 batch 단위로 반환함.
  - args.cuda가 True일 경우 worker 수 및 pin memory를 최적화.

- **Test Loader**
  - 현재는 사용하지 않음 (None으로 설정됨).

GAN3D

GAN3D/RENDERER_PT3D.PY

3D 얼굴 메쉬 데이터를 **PyTorch3D**로 시각화

➢ 이미지로 변환

GAN3D/RENDERER_PT3D.PY

- **전체 흐름 요약**

1.입력 데이터 준비
- verts: 3D vertex 좌표들 (예: 5023개의 점 × 3차원)
- face_v: 삼각형으로 이뤄진 face index들 (ex: 9976개의 (v1, v2, v3))

2.색상 지정
- 얼굴에 기본 RGB 색상을 입힘 (살색 톤으로 고정)

3.Mesh 생성
- Meshes 객체를 만들어서 PyTorch3D에서 이해 가능한 구조로 변환

4.카메라 및 조명 설정
- 카메라는 고정 위치(250 거리)에서 정면을 바라보며,
- 조명은 앞쪽에서 비추는 directional light를 사용함

5.렌더링 수행
- MeshRasterizer로 메쉬 → 이미지로 변환
- HardGouraudShader로 조명과 색상 적용
- MeshRenderer로 최종 이미지 생성

6.이미지 저장
- 렌더링된 이미지를 .png로 저장

TEXTURE MODEL

PREPROCESS.PY

- 텍스처 모델 학습에 필요한 이미지 데이터를 전처리하고, 커스텀 데이터셋을 정의

전체 동작 원리 요약

구성 요소	역할
`accumulate(model1, model2)`	모델 파라미터를 EMA 방식으로 부드럽게 업데이트
`imagefolder_loader(path)`	이미지 폴더에서 데이터셋 로더를 생성
`sample_data(dataloader, image_size)`	리사이즈 및 정규화된 이미지를 반환하는 로더 생성
`my_dictionary`	확장 가능한 딕셔너리 클래스 (거의 기본 `dict`와 유사)
`MyDataset`	이미지 파일 경로 리스트를 기반으로, 라벨을 추출하고 이미지와 함께 반환하는 PyTorch용 커스텀 데이터셋 클래스

MAIN