

KDT

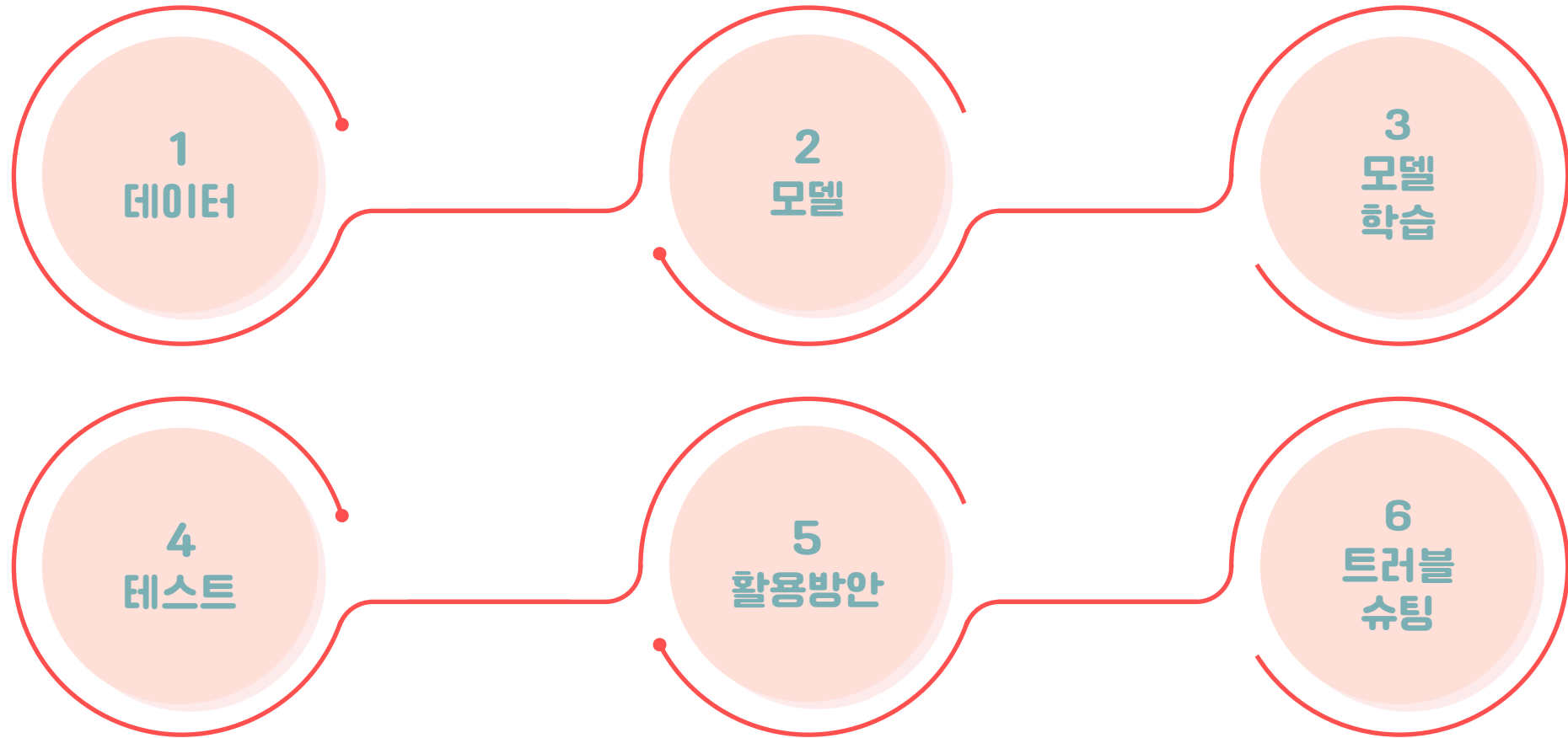
나이 예측 프로그램 구현

이루지명

백지명, 강진영, 고예성, 박현식, 안영준, 조세연



seok830621





seok830621

IMDB-WIKI

- 연령 레이블이 있는 사용 가능한 얼굴 이미지 데이터셋
- 연령 예측을 위한 사전 훈련된 모델을 제공
- 약 6만장의 인물 사진
- 파일 이름은 사진의 연도 + 인물의 태어난 연도(나이)



[출처 이동](#)


seok830621

연령 레이블 생성

- 사진의 연도 - 생년월일의 연도 = 해당 사진 시점의 연령
- 연령으로 새로운 디렉토리를 생성하고 이미지 이동

데이터 이상치 제거

- 사람이 아닌 경우, 두명 이상인 경우, 마스크를 낀 경우 삭제
- 나이 70이상, 93이하로 제한

이미지 처리

- Resize: 너비와 높이를 모두 224픽셀로 조정
- RandomAffine : 임의로 이미지 데이터를 회전/이동/크기조정/전단 변형
- RandomHorizontalFlip: 임의로 이미지 데이터를 좌우 반전

데이터 분리

- 전체 데이터 42,337개 중 train 33905개, validation 8432개로 분리


seok830621

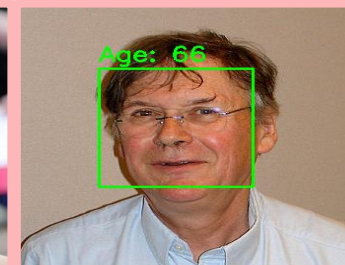
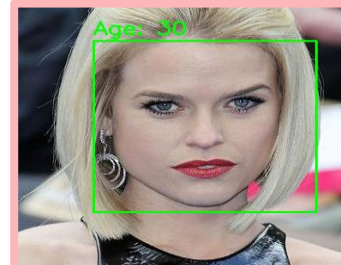
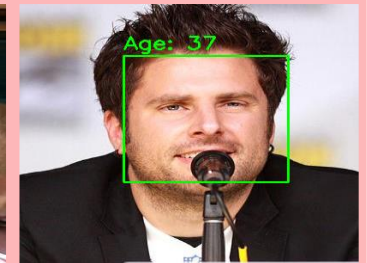
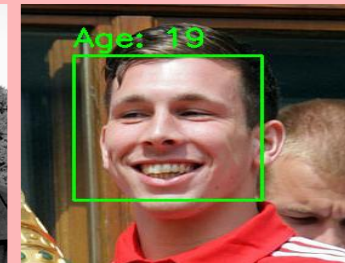
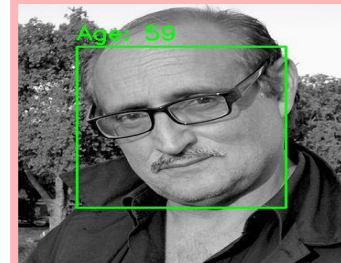
Data transform

- 데이터 변환은 데이터를 전처리하고 준비하는 과정
- 일반적인 데이터 변환작업
- 예)
 - 정규화: 데이터를 일정한 범위로 스케일링
 - 표준화: 데이터를 평균이 0이고 표준편차가 1인 분포로 변환
 - 원핫인코딩: 범주형 데이터를 바이너리 형태로 변환
 - 데이터 임베딩: 텍스트나 범주형 데이터를 숫자 형태로 변환
 - 이미지 전처리 : 이미지 크기 조정, 채널 변환 등의 작업을 수행

```
data_transforms = {  
    'train': transforms.Compose(  
        [  
            transforms.Resize(  
                (224, 224)  
            ),  
            transforms.RandomAffine(0, shear = 10, scale = (0.8, 1.2)),  
            transforms.RandomHorizontalFlip(),  
            transforms.ToTensor()  
        ]  
    ),  
    'validation': transforms.Compose(  
        [  
            transforms.Resize(  
                (224, 224)  
            ),  
            transforms.ToTensor()  
        ]  
    )  
}
```

데이터 정보

- train 데이터 셋의 데이터 수 분포
- 26세, 24세, 29세, 25세, 27세 순으로 이미지 데이터가 가장 많음
- 20대 이미지 데이터 셋이 가장 많음
- python-opencv 패키지를 통해 이미지 데이터셋에서 얼굴을 감지하고 해당 연령 Label로 표시




seok830621

Data Loader

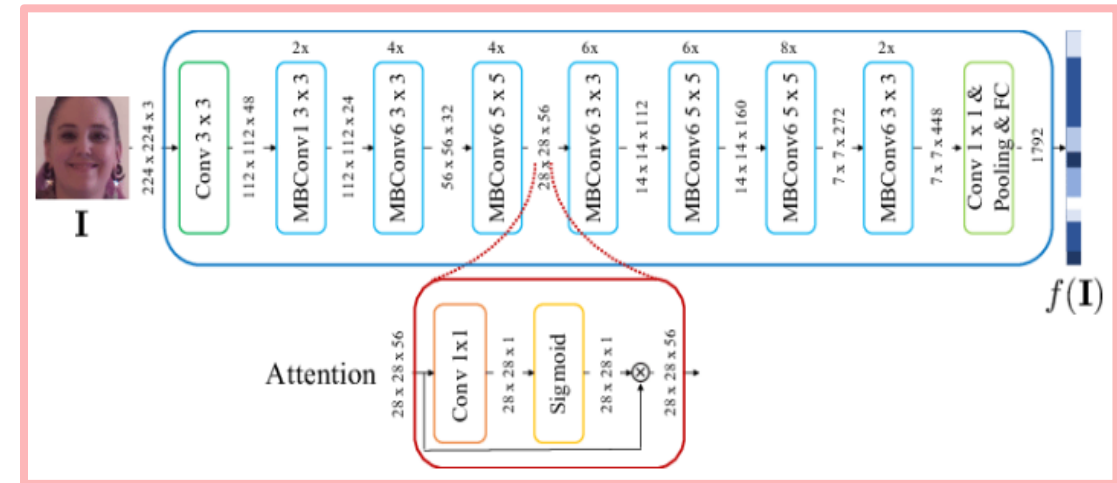
- 데이터 로더는 데이터를 배치단위로 모델에 공급하는 역할
- 데이터셋을 batch기반의 딥러닝모델 학습을 위해 미니배치 형태로 만들어 실제로 학습할 때 이용할 수 있게 하는 형태
- 반복자(iterator)로 데이터에 접근 하도록 하며, batch_size나 shuffle 유무 설정이 가능

```
image_datasets = {  
    'train': datasets.ImageFolder('train', data_transforms['train']),  
    'validation': datasets.ImageFolder('validation', data_transforms['validation'])  
}  
dataloaders = {  
    'train': DataLoader(image_datasets['train'], batch_size = 32, shuffle = True),  
    'validation': DataLoader(image_datasets['validation'], batch_size = 32, shuffle = False)  
}
```



Pretrainde Model

- EfficientNet은 기존의 Convolutional Neural Network (CNN) 모델들보다 더 효율적인 모델 구조를 갖고 있음
- EfficientNet은 Compound Scaling이라는 방법을 사용하여 모델의 규모를 효율적으로 조절
- EfficientNet은 B0, B1, B2, B3, B4, B5, B6, B7과 같이 여러 버전이 있으며, 숫자가 커질수록 더 크고 복잡한 모델
- EfficientNetB4는 이미지 분류, 객체 탐지, 시맨틱 세그멘테이션 등 다양한 컴퓨터 비전 태스크에 사용




seok830621

FC Layer

- 사용 모델: **CNN(Convolutional Neural Network)**
- EfficientB4 사전 학습한 모델 구축
- FC레이어 부분을 수정하여 모델의 마지막 완전히 연결된 레이어를 재구성
- 기존의 마지막 레이어를 대체하기 위해 세 개의 선형 레이어와 활성화 함수 ReLU를 사용
- 입력 1792차원에서 87개의 클래스로 분류

```
model.classifier = nn.Sequential(  
    nn.Linear(1792, 1024),  
    nn.ReLU(),  
    nn.Linear(1024, 512),  
    nn.ReLU(),  
    nn.Linear(512, 87)  
)
```


seok830621

Model Train

- **optimizer**는 **Adam**을 사용
- 에폭 30만큼 반복하여 학습
- 다중 클래스로 분류하기 때문에 손실함수는 **CrossEntropyLoss**
- Val loss는 약 5%

```
optimizer = optim.Adam(model.parameters(), lr=0.001)
```

```
epochs = 30
```

```
loss = nn.CrossEntropyLoss()(y_pred, y_batch.long())
```

Age Prediction

Predict: 25, Actual: 20



Predict: 48, Actual: 79



Predict: 26, Actual: 27



Predict: 29, Actual: 23



Predict: 24, Actual: 24



Predict: 38, Actual: 42



Predict: 17, Actual: 33



Predict: 33, Actual: 34



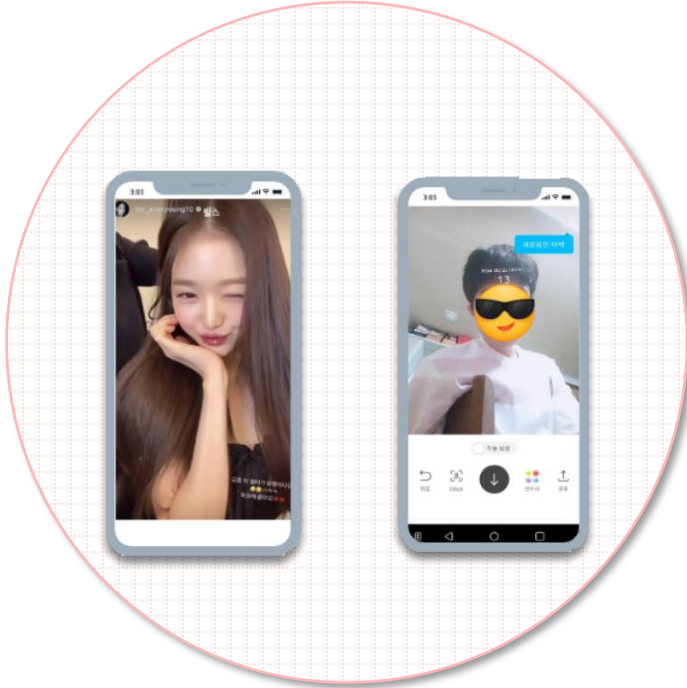
Predict: 25, Actual: 26



Predict: 33, Actual: 33



- 모델 예측 값과 실제 나이를 TITLE로 표시
- 얼굴이 잘 보이는 사진을 사용
- 약간의 오차가 있지만 예측값과 실제값이 비슷함



- 인스타그램, 스노우 어플과 같은 얼굴 필터를 사용할 수 있는 플랫폼에서 나이를 예측해보는 필터



- 얼굴을 인식하여 나이를 예측하고 특정 서비스를 연령에 맞추어 제공


seok830621

문제점

1. 이미지 데이터셋 내 인물사진이 아닌 사진들이 많았다.
2. 데이터의 배경 노이즈가 학습 방해

개선사항

1. 이미지 데이터셋에서 인물사진이 아니거나 관련 없는 이미지는 제거
2. OpenCV를 사용하여 배경 노이즈를 제거하는 블러처리나 인물만을 또렷하게 잡아주는 이미지 프로세싱이 필요

Thank You!

