**1. 프로젝트 소개 & 기획 의도**

사람의 감정은 단 하나의 감정으로 분류하는 것에는 많은 어려움이 있다. 예를 들어 ‘100% 행복’한 표정이라는 것은 존재하기 힘들다. 아무리 행복한 표정일지라도 사람의 표정에서는 동시에 슬픔, 분노, 두려움 등 다양한 감정들이 표현되기 마련이다. 하지만, 여러 경험들을 통해 특정한 얼굴 표현이 어떠한 감정을 나타내는지를 짐작하는 것은 가능하다.

연구에 따르면 의사소통에서 말이나 대화 등 언어적 소통이 차지하는 부분은 7% 남짓이고, 비언어적 표현이 90% 이상을 차지한다. (메라비언의 법칙) 이때, 표정 등의 시각적 표현이 차지하는 부분은 55%로, 감정 표현의 파악은 의사소통에서 상당히 중요하다.

따라서, 감정 표현이 드러나는 사진을 감정에 따라 7가지의 카테고리로 분류하고, 많은 학습을 통해 해당 표정이 어떠한 감정을 드러내고 있는지를 예측할 수 있는 예측 모형을 만드는 것이 목표이다.

**2. Data Description**

- 활용 데이터 셋

1. MMA FACIAL EXPRESSION

<https://www.kaggle.com/datasets/mahmoudima/mma-facial-expression/code>

- 오 분류된 데이터의 수가 많음 -> 데이터셋 변경

2. Facial Expression Dataset

<https://www.kaggle.com/datasets/aadityasinghal/facial-expression-dataset>

- train, test dataset으로 이미 분리가 되어있는 상태

- 각각의 데이터는 다시 7개의 감정(angry, disgust, fear, happy, neutral, sad, surprise)으로 분류

- disgust의 경우 다른 데이터에 비해 개수가 매우 적음 -> 이에 대한 대책이 필요

> 해결

- 이미지 추가 -> 한계가 있음.

- One-shot learning(<https://wikidocs.net/150812>)

- Under Sampling, Over Sampling(SMOTE, GAN 등):

<https://shinminyong.tistory.com/34>

<https://dining-developer.tistory.com/27>

<https://www.samsungsds.com/kr/insights/generative-adversarial-network-ai-2.html>

- Data Augmentation: 주로 불균형 ‘이미지’ 데이터 처리에 많이 사용되는 기법

<https://pytorch.org/vision/stable/transforms.html>

<https://keraskorea.github.io/posts/2018-10-24-little_data_powerful_model/>

- 각 사진의 크기는 48 \* 48 size로 고정되어 있음

- 사진들의 이름이 제각각임 -> 전처리

(아니면 폴더에서 데이터를 가져올 수 있도록 다른 방안 생각하기)

- 사람 얼굴이 아닌 것들도 포함되어 있음(캐릭터 등) -> 삭제

- 흑백 사진

**3. 문제 정의 & 활용 기법 정리**

- 해당 프로젝트의 목표는 7개의 카테고리로 분류된 감정들 중 가장 적합한 감정을 예측하는 모델을 만드는 것이기에, **분류** 문제이고 정답은 1개로 **특정**되어야 한다.

- 이미지 데이터에 대한 학습 진행 -> CNN 기술을 활용할 예정

* CNN: NN(FC Layer라고도 한다.) + 컨볼루셔널 계층(for 이미지 처리)
* 컨볼루셔널 계층 -> 추출된 features -> NN -> 최종 분류

- torch 모듈 활용

- 여러 개의 분류 모델을 활용

* 개별 분류 모델을 통해 각각 데이터 셋 학습
* 이후 앙상블 기법을 적용하여 여러 분류기를 합쳐 성능을 향상시킨다.

-> 주로 soft voting(여러 예측값들을 합침 -> 평균)이나 스태킹 활용

- 일단 각각의 단일 모델들을 만들어 이를 학습/평가하고, 마지막에 스태킹을 이용하여 ensemble 하고자 한다.

* 활용할 분류기 모델: VGG19, EfficientNetV2(ImageNet), ResNet50 + YOLO

- parameter tuning(자동화)

1) RandomizedSearchCV

2) GridSearchCV: 교차 검증 + 파라미터 집합의 순차적 적용 -> 최적 파라미터 탐색

3) Optuna: 파라미터의 범위/목록 설정 시 매 trial마다 파라미터를 변경하며 최적 파라미터 탐색

4) AutoML – PyCaret 라이브러리 활용

**📌 고려사항들**

각각의 단일 모델에 대해

**1) 손실함수의 종류(loss function)**

- BinaryCrossEntropyLoss(LogisticLoss): 이진 분류

- CrossEntropyLoss: 다중 분류 – **고정**

**2) 옵티마이저**

- 경사 하강법(Gradient Descent)

- 확률적 경사 하강법(Stochastic Gradient Descent)

- 미니배치 경사 하강법(Mini-Batch Gradient Descent)

- AdaGrad

- RMSProp

- Adam

**3) 활성화 함수**

- ReLU

- Softmax -> CrossEntropyLoss의 활성화 함수 **– 고정**

- Sigmoid

**4) 불균형 데이터에 대한 처리**

- Over Sampling

- Data Augmentation

**5) Parameter Tuning – 어떻게 AutoML 수행할 지**

- AutoML을 통해서 위에서 언급한 것들의 최적 조합을 찾을 수 있을까?

- Pipelining

**4. 참고할 것들**

- 수업 예제 중 Project(1)

- CNN 모델 앙상블

<https://dbstndi6316.tistory.com/287>

<https://github.com/KerasKorea/KEKOxTutorial/blob/master/16_Ensembling%20ConvNets%20using%20Keras.md>

- Optuna 관련 발표 자료 pdf

- PyCaret 예제(Kaggle)