**Emotion Classification in Images Using Teachable Machine.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**박현우 잘리너바 아이가늠**

**2022480007 2021920046**

**Abstract**

본 연구는 Teachable Machine framework를 활용하여 이미지 내 감정 분석을 수행한다. 우리는 양질의 데이터를 수집한 뒤Teachable Machine 내부의 Hyper-parameter tuning을 통해 모델이 고차원의 감정 분석 능력을 갖추는 것을 목표로 삼는다. Teachable Machine은 모델 자체적으로 데이터를 85%/15% 로 나눠 train/test를 수행하지만 이 방식은 train data에 bias를 발생시킬 우려가 있으므로 따로 test data를 생성해 성능을 측정하였다.

**1. introduction**

매체를 통한 감정 분석 작업은 다양한 현대 기술에 활용되는 분야이다. 감정은 음성, 표정, 상황, 텍스트 등 다양한 형태로 표현될 수 있으며 이 정보들을 종합해 일상적인 상황에서 사람의 감정 상태를 보다 면밀히 파악하고 그에 맞는 대응을 취하는 것은 서비스/마케팅, 문화 사업에서 매우 중요하기 때문에 지금도 다양한 분야의 기술들을 접목시킨 모델이 많이 발표되고 있다. 대표적으로 SOTA에는 computer vision, Natural Language Processing 분야에서 emotion classification, emotion recognition 분류 성능을 극대화한 모델들이 최근에도 업데이트되고 있다. 우리는 음성/텍스트/상황/표정 등 감정을 인식할 수 있는 다양한 요소들 중에서 가장 직관적이고 Teachable Machine으로 해결하기 용이한 표정, 즉 이미지를 이용해 감정을 추론하고 분류하는 모델을 작성해볼 것이다. 분류할 감정 class는 총 5가지로 happy, sad, expressionless, surprise, angry이다.

**2. Methodology**

현재 ML 패러다임은 모델이 만족할만한 성능을 내기 위해서 많은 양의 데이터를 필요로 한다. 이때 주의해야 할 것은 train data는 가능한 한 다양한 상황에 대해 내성이 있도록 수집해야 한다는 것이다. 그렇지 않으면 overfitting이 발생할 우려가 있다. Overfitting은 모델의 과도한 학습능력으로 발생할 수도 있지만 데이터의 수가 필요한 모델 대비 과도하게 적거나 train data의 질이 나빠 발생하기도 한다. 학습하는 데이터가 편향되어 모델이 test data에 대해 성능이 train data 대비 과도하게 낮아지는 overfitting이 나타나는 것이다.

우리가 수행하는 작업은 사람의 감정을 분류하는 것이기에 모델에게 다양한 사람의 정보를 학습시킬 필요가 있다. 예를 들어 백인 데이터만 활용해 학습시킨 모델은 흑인 데이터에 대해서 제대로 된 성능을 내기 어려울 것이다. 이러한 이유로 반드시 우리가 수집해야 하는 Train dataset은 다양한 “인종”, “성별”, “연령”에 대해서 균일한 분포를 가질 수 있어야 한다.

이러한 데이터들을 수집하기 위해 우리는 먼저 Image scrapping 을 시도했다. 이미지를 각 label에 맞게 찾아야 하므로 인터넷에서 일일이 찾아 저장하는 것 보다는 scrapping을 수행하는 tool을 사용해 이미지 데이터를 수집한 뒤 분류하는 과정을 거치는 것이 더 나을 것이라 판단했다.



**사진1:** Image scrapping을 수행하기 위한 기초적인 코드이다. “사람 표정”이라는

검색어로 찾은 이미지 120장을 현재 경로에 emotion 폴더를 생성해 넣는 것이다.

그러나 scrapping을 통해 저장한 이미지는 실망스러운 품질을 가지고 있었다. Human smiley face라는 키워드로 검색한 결과 총 120장의 이미지 중에서 총 38장만 Train에 활용 가능한 이미지였으며 26장은 남자, 12장만 여자로 데이터가 남자 쪽으로 심하게 편향되어 있었다. 인종으로 보았을 때 또한 백인 23명, 황인: 8명, 흑인 7명으로 백인 쪽에 심하게 치우쳐져 있었다.

따라서 구글링을 통해 데이터 셋을 생성했다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Emotion **\** Standard | White-M | White-W | Brown-M | Brown-W | Black-M | Black-W |
| SURPRISE | 7 | 8 | 8 | 10 | 8 | 9 |
| HAPPY | 8 | 8 | 9 | 9 | 7 | 9 |
| SAD | 7 | 9 | 9 | 8 | 7 | 10 |
| EXPRESSIONLESS | 9 | 9 | 8 | 8 | 7 | 8 |
| ANGRY | 9 | 8 | 8 | 9 | 8 | 8 |

**표1**: Dataset은 각 label당 50장의 데이터를 가진다. 데이터는 최대한 균일한 피부색/성별 분포를 가지도록 수집했다.

**사진2:** train data 예시

**3. Experimental result**

Teachable Machine은 Pre-trained model로MobileNet을 사용한다. MobileNet이란 VGGNet과 유사한 성능을 가지지만 연산량이 27분의 1 수준인 모델로 모델의 성능보다는 경량화에서 탁월한 성과를 보인 모델이다. 우린 이 모델을 베이스로 삼아 fine-tuning 을 수행할 것이다. 그런데 Teachable Machine은 사용자가 원하는 데이터를 입력하면 각 class마다 85%의 데이터를 무작위로 추출해 학습을 진행하며 나머지 15%의 data는 성능 표시를 위해 사용한다. 이 test data가 정확히 어떤 방식으로 결정되는지 알지 못하므로 우리는 각 class마다 직접 이미지를 한 장씩 모아 총 30장의 test dataset을 생성했다. 또한 Keras version으로 모델을 다운로드 한 뒤 코드를 약간 개조해 test dataset에 대해 전체 정확도를 계산할 수 있게 했다. 해당 코드는 아래 Github 링크에서 확인할 수 있다.

**그림3**: Batch\_size와 Learning rate는 16과 0.001로 설정한 채 epoch만 8차례에 걸쳐 변경해가며 성능을 측정한 기록이다. 300 epoch에서 최고 성능인 60%를 달성한 뒤 test\_loss가 점차 오름에 따라 overfitting되는 모습을 보여준다.

**4. Conclusion & Discussion**