**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 dataset을 생성했다. 해당 dataset은 **표 1**과 같이 구성된다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 |

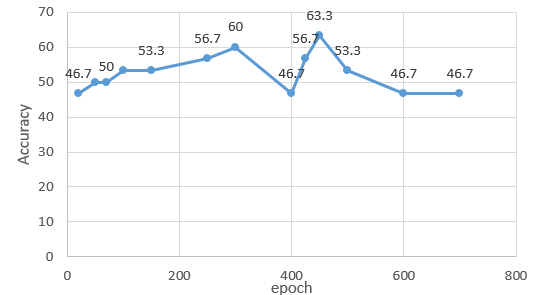
**표1**: Dataset은 각 label당 50장의 데이터를 가진다. 데이터는 최대한 균일한 피부색/성별 분포를 가지도록 수집했다.

**사진2:** train data 예시

**3. Experimental result**

Teachable Machine은 Pre-trained model로MobileNet을 사용한다. MobileNet이라고 하는 VGGNet과 유사한 성능을 가지지만 연산량이 27분의 1 수준인 네트워크로 모델 자체의 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 방식은 우리가 test data가 정확히 어떤 방식으로 결정되는지 알지 못하게 하며 모델을 새로 돌릴 때 마다 학습되는 train data가 다를 수 있으므로 우리는 각 class마다 직접 이미지를 한 장씩 모아 총 30장의 test dataset을 생성했다. 또한 Keras version으로 해당 모델을 다운로드 한 뒤 코드를 약간 개조해 test dataset에 대해 전체 정확도를 계산할 수 있게 했다. 우리는 이 test dataset으로 test한 결과와 Teachable Machine이 표시한 결과를 함께 분석할 것이다. 해당 코드와 사용한 데이터는 다음 Github 링크에서 확인할 수 있다 <https://github.com/CHAT-UOS/ml-team_project>

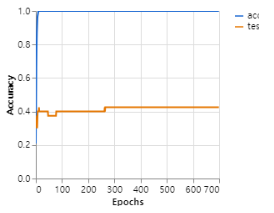
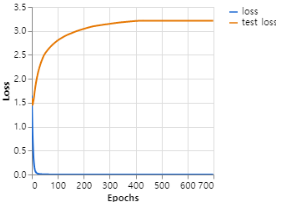
**3-1. epoch에 따른 모델의 성능 변화**

**사진3**: batch\_size/lr 는 Default 값으로 그대로 사용하되 epoch을 조정하며 모델의 테스트 성능을 평가한 지표이다. 항상 같은 test data 를 사용하므로 신뢰할 수 있다.

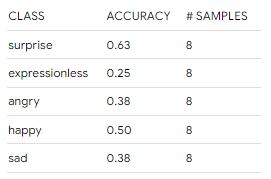
**그림 3**의 지표는 자체적으로 제작한 test dataset을 기반으로 한 성능 평가이다. **그림 4**에서 보여주는 Teachable Machine의 고급 설정에서 분석해준 결과와는 조금 다른 지표가 작성되었다. Epoch가 450일 때 63.3%의 accuracy를 기록하며 최고 성능을 보여주었다.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| epoch | 20 | 50 | 70 | 100 | 150 | 250 | 300 | 400 | 425 | 450 | 500 | 600 | 700 |
| Acc(%) | 46.7 | 50 | 50 | 53.3 | 53.3 | 56.7 | 60 | 46.7 | 56.7 | 63.3 | 53.3 | 46.7 | 46.7 |

**표 2**: 각 epoch마다 model이 자체 제작한 test data에 대해 test를 진행했을 때 각 Acc를 표시

****사진4: Teachable Machine 분석

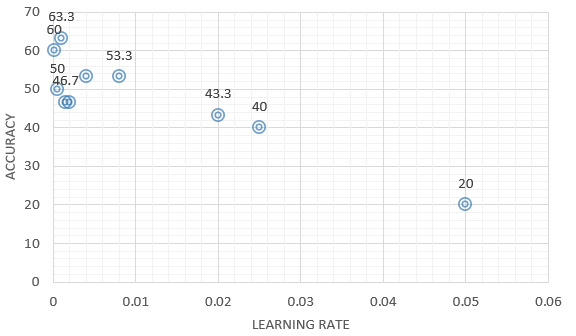
**그림 4**에서 Train data에서 15%를 무작위로 추출한 경우 epoch이 증가하더라도 test 성능이 50%를 넘지 못하는 경향을 보인다. 하지만 **그림 3**에서 알 수 있듯이 별도의 test data로 테스트한 결과는 60%의 비교적 높은 정확도를 보인다. 우리는 이런 차이가 벌어지는 원인을 class별 정확도에서 찾을 수 있었다.

**사진 5**: teachable machine이 각 class에서 8개의 data를 무작위로 추출해 test한 뒤 각 class별 test 결과를 분석한 결과이다. Surprise와 happy는 비교적 높은 Accuracy를 달성한 반면 expressionless, angry, sad는 비교적 낮은 Accuracy를 보였다.

**그림5**를 보면 expressionless의 accuracy 가 다른 감정 클래스에 비해 극단적으로 낮은 것을 볼 수 있다. 이 말은 expressionless의 test data를 다른 감정들로 많이 착각했다는 뜻이다. Expressionless는 한국어로 표현하면 “무표정”, “평소 표정”이 된다. 하지만 이러한 표현은 angry의 “정색”과 의미가 겹치게 된다. 실제로 angry/expressionless의 train data를 보면 화난 표정과 정색한 표정 사이에서 혼란을 유발할 만 할 수 있는 표정들이 몇 가지 존재한다. 이러한 표정들이 Teachable machine에서 분석한 test 성능을 떨어뜨리는 원인 중 하나라고 생각 가능하다. 또한 sad 클래스는 표정 특성상 이미지가 expressionless + “눈물”인 이미지가 많다. 이 또한 test를 어렵게 하는 요소라고 볼 수 있다.

반면에 자체적으로 작성한 test data를 보면 눈물, 눈매 등의 어려운 판단 요소 말고도 순수 표정만으로 감정을 명확하게 분류할 수 있는 데이터들이 주를 이룬다. 쉽게 말해 test dataset이 train dataset에 비해 판단하기 쉽다는 것이다. 이런 원인들이 겹치고 겹쳐 test loss의 차이를 만들어낸 것으로 예상된다.

**3-2. Learning rate에 따른 모델의 성능 변화**

이번 챕터에서는 가장 좋은 성능을 기록했던 700 epoch 16 batch size model을 기반으로 성능을 더 높일 수 있는지, 어느 learning rate를 적용할 때 local minimum으로 수렴하지 못하고 발산하는지 분석해볼 것이다.

**사진 6**: learning rate의 변화에 따른 Accuracy의 변화를 측정한 것이다. 일정 이하의 learning rate는 model이 경량화 되어서인지 인식하지 못하는 문제가 있어 0.000005이하의 learning rate는 실험해보지 못하였다.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LR | 0.00005 | 0.0001 | 0.0005 | 0.001 | 0.0015 | 0.002 | 0.004 | 0.008 | 0.02 | 0.025 | 0.05 |
| Acc | x | 60 | 50 | 46.7 | 63.3 | 46.7 | 53.3 | 53.3 | 43.3 | 40 | 20 |

**표 2.** 각 learning rate에 따른 Acc 값들을 표로 나타낸 것이다.   
**표 2**에서 보이는 대로 다양한 lr값들을 시도해 보았지만 Base lr인 0.0001의 성능을 넘어서는 lr은 찾지 못하였다. 다만 lr이 일정 이상 낮아지면 모델이 vsc환경에서 돌아가지 않는 문제가 있었는데 이 문제가 발생한 이유를 명확히 규명하지 못하였다. 개인적인 추측으로는 모델을 경량화하는 과정에서 양자화를 통해 parameter bite수를 제한할 때 일정 이하의 lr은 돌아가지 못하게 한 것인지, 아니면 코드 구현 상에서 문제가 있었던 것인지 파악하지 못하였다.

**4. Conclusion & Discussion**