

오늘의 날씨를 알려주는 AI

(CNN을 활용한 날씨 분류 모델 개발)

Yong jun-Hong

홍 용 준

눈으로 판단하는 날씨와 실제 날씨가 다른 적이 많았기 때문에 날씨를 판단해주는 학습모델을 개발했습니다. 학습모델은 전이 학습 모델과 CNN 모델을 정의하고, 학습 및 평가를 위한 함수를 구현한 후, 데이터셋을 활용하여 모델을 학습 및 평가하였습니다. 학습 및 평가 과정에서는 손실 함수 및 옵티마이저를 정의하여 모델을 최적화하였고, GPU를 활용하여 모델을 가속화하였습니다. 정확도를 올리기 위해 Epoch을 15번 진행했고 결과를 시각화 해서 차트로 도출했습니다. 결과로는 날씨에 따라 학습 모델이 잘 구분 하는 모습을 보였습니다.

1. 서 론

1. 연구 배경 소개

눈으로 날씨를 판단한 뒤, 외출했더니 눈으로 판단한 날씨와 실제 날씨 간 불일치가 발생해서 혼란을 야기한 적이 많았습니다. 이런 혼란을 방지하고자 정확한 날씨 판단을 위한 학습 모델의 개발이 필요하다 생각해 모델을 개발하게 되었습니다.

2. 연구 목적 및 문제 정의

본 연구의 목적은 눈으로 판단하는 날씨와 실제 날씨 간의 불 일치를 해소하기 위한 학습 모델의 개발입니다.

3. 연구 방법 및 접근법

본 연구에서는 전이 학습 모델과 CNN 모델을 결합하여 날씨를 판단하게 하였습니다. 데이터 셋은 눈으로 볼 수 있는 다양한 날씨를 포함하며, 이를 통해 모델을 학습합니다. 최적화 함수로는 Adam을 사용했으며 손실 함수로는 CrossEntropyLoss를 사용해 모델을 최적화 시켰으며, GPU를 사용하게 해서 학습 속도를 향상시켰습니다.

2. 본 론

2.1 데이터 세트

Shine(화창한 날씨), Snow(눈이 오는 날씨), Snowsand(황사) 이렇게 3가지 데이터 세트를 가지고 학습했습니다.

또한 Test 이미지 개수를 78개로 하고 Train 이미지 개수를 132개로 해서 과적합시 훈련 데이터 세트에 대해서는 오류가 감소하지만, 테스트 데이터 세트에 대해서는 오류가 증가하는 것을 방지했습니다.

또한 Train 데이터 셋은 Resize 함수를 통해 크기를 조정하

였으며, RandomResizeCrop 함수를 이용해서 임의로 이미지를 잘라내서 크기를 조정하였습니다.

그 후, RandomHorizontalFlip 함수를 이용해 이미지를 수평으로 무작위로 뒤집고 Tensor 함수를 통해 이미지를 텐서로 변환하였습니다.

그에 비해 Test 데이터 셋은 Train 데이터 셋과 똑같은 이미지 정제 과정을 거쳤지만 RandomHorizontalFlip 함수는 사용하지 않았습니다.

2.2 수행 환경

파이토치를 사용했고, 코랩으로 실험했으며 데이터와 모델을 GPU로 옮겨와서 실험했습니다.

2.3 알고리즘

ResNet-18과 CNN을 사용해서 이미지 분류 작업을 수행했으며, ResNet-18 모델을 통해 미리 학습된 ResNet-18 모델을 불러왔으며, 모든 파라미터를 고정시켰습니다. 파라미터를 기본값으로 했을 때 만족하는 결과가 나왔기 때문에 파라미터를 별도로 변경하지 않았습니다.

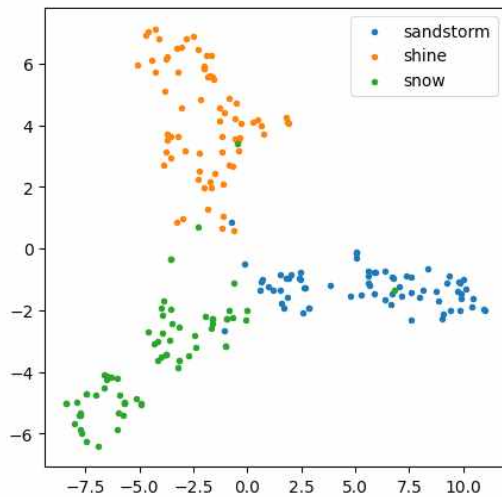
최적화 함수로는 Adam을 사용했으며 손실 함수로는 CrossEntropyLoss를 사용했습니다.

그리고 주어진 코드에서는 데이터 셋의 클래스가 Cat,Dog 두 개였지만 필자의 데이터 셋에는 Shine, Snow, SandStorm 3개 이므로 name 변수에 새로운 클래스의 이름을 작성하고 FashionCnn 클래스의 출력 뉴런 수를 2개에서 3개로 변경했습니다.

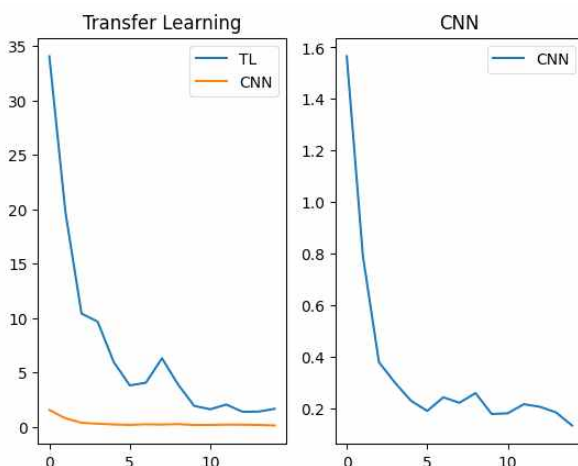
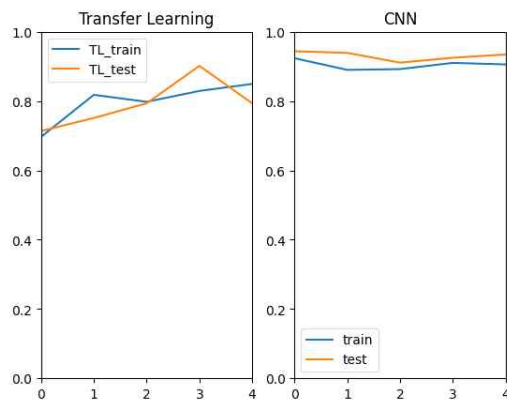
CNN은 FashionCNN 클래스를 정의해 CNN 모델을 만들어 으며, 두 개의 합성곱층과 풀링층을 포함했습니다.

합성곱층과 풀링층을 이용해 특징을 추출하고 분류를 진행했습니다.

3. 결 과



분류한 결과 일부 snow가 Shine으로 인식되는 경우가 있었지만 나머지는 잘 분류되는 모습을 보였습니다.



전이학습과 CNN의 모델 학습 정확도 모두 0.9를 보다 높은 정확도를 보였으며 손실 함수 차트 또한 0에 수렴하는 모습을 보였습니다.

정확도	0.9213
검증 정확도	0.9718

4. 고 찰

1에 수렴하는 높은 정확도를 보였지만 일부 Shine(화창한 날씨)와 Snow(눈 오는 날씨)에서 틀리는 모습을 가끔 보였다. 화창한 날씨와 눈이 오는 날씨의 데이터 셋을 더 추가시켜서 정확도를 높여 보완해 보겠다. 또한 비가 오는 날씨, 안개가 짙은 날씨 등을 추가시켜서 다양한 상황에 대해 학습할 수 있도록 보완하겠다.

5. 결 론

아침에 일어났을 때 오늘 날씨가 황사인지 아닌지 구분이 안 돼서 핸드폰으로 찾아봤었는데 이런 날씨를 구분해 주는 학습모델이 있으면 좋을 것 같다고 생각해서 실험을 진행했습니다. 데이터 셋은 황사와 눈이 오는 날 그리고 맑은 날로 진행했습니다. 가끔 황사와 맑은 날을 헷갈리기는 하지만 높은 정확도를 보였습니다. 이런 데이터 모델을 기상청의 AI에 사용하면 좋을 것 같다는 생각을 했습니다.

참고문헌

데이터 셋 : Kaggle

코드 : 순천향대학교 인공지능 CH10, chat GPT

논문 : 인공지능을 활용한 과거 기상 데이터 분석 기반 날씨 예측 - 신주환, 홍성은, 이현수, 김선욱, 김화중*