



1. 인공지능 활용 설문 사례 고찰 (1/7)

✓ “머신러닝 탐구생활”

1장 파이썬과 머신러닝 그리고 캐글

2장 산탄데르 제품 추천 경진대회
(고객/제품 텍스트 정보, 상품 추천)

3장 텐서플로 음성 인식 경진대회
(음성인식 인터페이스 정확도 개선/대중화)

4장 포르토 세구로 안전 운전자 예측 경진대회
(고객에게 합리적인 보험금 청구)

5장 스테이트 팜 산만한 운전자 감지 경진대회
(운전자 행동 패턴)

5.1 경진대회 소개

5.2 경진대회 주최자의 동기

5.3 평가 척도

5.4 주요 접근

5.5 데이터 준비하기

5.6 탐색적 데이터 분석

5.7 BASELINE 모델

5.8 성능 개선 실험

5.9 승자의 지혜

※ 소스코드 :
<https://github.com/bjpublic/kaggleml>

2022-09-06

2

5장. 머신러닝 탐구생활 (1/23)

✓ "스테이트 팜 산만한 운전자 감지 경진대회" (2016, 1,440팀 참가)

- 이미지 기반
- 문제 유형 : MULTI-CLASS CLASSIFICATION (다중 클래스 분류)
 - 10가지 운전상태 : 안전운전, 오른손으로 문자, 오른손으로 전화, 왼손으로 문자, 왼손으로 전화, 라디오 조작, 음료수 섭취, 뒷좌석 손뼉기, 얼굴/머리 만지기, 조수석과 대화
- 시나리오 : 카메라 이미지 분석, 운전자들의 산만한 행동 분류
- 제공 데이터 : 학습 데이터 22,424개, 테스트 데이터 79,726개

2022-09-06

5

5장. 머신러닝 탐구생활 (2/23)

✓ "스테이트 팜 산만한 운전자 감지 경진대회" (2016, 1,440팀 참가)

- 결과 제출 : 테스트 이미지에 대한 분류별 확률값
 - 평가 척도 : 다중 클래스 로그 손실 (MULTI-CLASS LOGARITHM LOSS)
 - 모든 클래스에 대한 이진 로그 손실 평균값
 - 0~1 사이 값을 갖는 확률을 로그 함수에 넣은 값
 - 운전자 감지 서비스의 "높은 정확도보다 높은 확률로 틀리는 경우를 최소화"
 - 1등 팀 기록 : 0.08739점 (PUBLIC 리더보드)
 - ※ 파이선 함수 : p. 309 (이진분류 로그 손실 값을 구하는 함수)

2022-09-06

6

5장. 머신러닝 탐구생활 (3/23)

✓ 머신러닝 과정

- 케라스 프레임워크를 통하여 이미지 분류 모델을 구축, 학습하는 일련의 과정
- BASELINE 모델 구축 과정
 - ① CNN 모델 정의
 - ② 데이터 전처리
 - ③ 교차검증 평가 준비
 - ④ 모델 학습
 - ⑤ 테스트 데이터 예측 및 캐글 업로드
 - ⑥ 성능개선

2022-09-06

7

5장. 머신러닝 탐구생활 (4/23)

✓ ① CNN 모델 정의 (1/1)

- 이미지 분류 대표적인 모델 VGG16 정의
 - 케라스 함수, 파이선 코드 : (P. 350)
 - 3가지 중요한 설정값
 - INCLUDE_TOP=FALSE : 모델 최상위 전결층 을 제외한 컨볼루션 블록들만 정의
 - WEIGHT=None : 기학습 파라미터 미 사용
 - INPUT_SHAPE=(224,224,3) : 모델 입력값의 크기 (이미지 크기480x640x3을 리사이징)

2022-09-06

8

5장. 머신러닝 탐구생활 (5/23)

✓ ② 데이터 전처리 (1/1)

- 역전파 알고리즘 적용을 위해 훈련 데이터 정답 값을 사전에 파악
 - 배치 학습 : 한번의 역전파 알고리즘 과정에서 모든 훈련 데이터를 사용
 - 미니배치 학습 : 한번의 역전파 알고리즘 과정에서 훈련 데이터의 일부만 학습 (○)
 - 미니배치 크기 : 8
 - 훈련 데이터 중 임의로 8개 선정
 - GPU에서 연산 및 역전파 알고리즘을 통해 모델 파라미터 갱신
 - 훈련 데이터 모두 학습(1 EPOCH 학습) 위해 2,803번 반복
 - 케라스 함수, 파이선 코드 : `IMAGEDATAGENERATOR()`, (P. 353)

2022-09-06

9

5장. 머신러닝 탐구생활 (6/23)

✓ ③ 교차검증 평가 준비 (1/1)

- 5-FOLD STRATIFIEDKFOLD 기법 사용
 - 훈련 데이터 22,424개를 랜덤하게 5등분
 - 4/5 데이터를 학습용, 1/5 데이터를 검증용(모델 평가)으로 활용
 - 교차 검증을 위해 데이터를 분리(학습용, 검증용)하는 파이선 코드 : (P. 355)
 - 학습 함수 : `MODEL.FIT_GENERATOR()`

2022-09-06

10

5장. 머신러닝 탐구생활 (7/23)

✓ ④ 모델 학습 (1/3)

■ 과적합 방지를 위해 EARLY-STOPPING 기법 도입

- 훈련 데이터를 1 EPOCH 학습할 때마다 검증 데이터에 대한 다중 클래스 손실 값 계산
- 3 EPOCH 연속 로그손실값이 개악될 때 학습 멈추고 해당 모델 파라미터 저장
 - ※ 한번도 학습에 사용하지 않은 검증 데이터의 평가 점수를 기반으로 일반화 성능이 가장 좋은 지점에서의 모델 파라미터
- 파이선 코드 : (p. 357)
- 모델 학습 정상 진행 로그 (케라스가 학습 과정을 실시간으로 로깅, p. 358)
 - 현재 진행 미니배치 및 전체 미니배치 수, 진도(시각화), 1 EPOCH 완료 예상 소요 시간
 - **Loss** : 훈련 데이터에 대한 다중 클래스 로그 손실값, 낮을수록 좋음
 - **Acc** : 훈련 데이터에 대한 정확률, 1에 가까울수록 좋음

2022-09-06

11

5장. 머신러닝 탐구생활 (8/23)

✓ ④ 모델 학습 (2/3)

■ 과적합 방지를 위해 EARLY-STOPPING 기법 도입

- 학습 결과 로그
 - EPOCH을 거듭할수록 훈련 데이터 및 검증 데이터의 LOG-LOSS가 줄어들면 훈련이 정상적으로 진행됨을 의미
 - 훈련 데이터의 LOG-LOSS가 너무 큰 값 또는 장기간 감소하지 않으면 머신러닝 파이프라인에 문제 있음
 - **VAL_LOSS** : 15째 EPOCH에서 가장 낮은 값, 그 이후 3번 연속 개선 안되었음
→ 15번째 EPOCH의 VAL_LOSS가 모델 파라미터 저장

2022-09-06

12

5장. 머신러닝 탐구생활 (9/23)

④ 모델 학습 (3/3)

```

Using TensorFlow backend.
# Train Model
Found 79726 images belonging to 10 classes.
Found 1784 images belonging to 10 classes.
Found 4590 images belonging to 10 classes.

Epoch 1/500
2229/2229 [=====] - 383s 172ms/step - loss: 2.0548 - acc: 0.2431 - val_loss: 1.1978 - val_acc: 0.5711
Epoch 2/500
2229/2229 [=====] - 377s 169ms/step - loss: 0.9761 - acc: 0.6608 - val_loss: 0.4253 - val_acc: 0.8698
Epoch 3/500
2229/2229 [=====] - 377s 169ms/step - loss: 0.5039 - acc: 0.8368 - val_loss: 0.2443 - val_acc: 0.9291
Epoch 4/500
2229/2229 [=====] - 376s 169ms/step - loss: 0.3040 - acc: 0.9037 - val_loss: 0.2147 - val_acc: 0.9494
Epoch 5/500
2229/2229 [=====] - 377s 169ms/step - loss: 0.2085 - acc: 0.9348 - val_loss: 0.1338 - val_acc: 0.9583
Epoch 6/500
2229/2229 [=====] - 379s 170ms/step - loss: 0.1516 - acc: 0.9531 - val_loss: 0.0938 - val_acc: 0.9751
Epoch 7/500
2229/2229 [=====] - 378s 170ms/step - loss: 0.1124 - acc: 0.9641 - val_loss: 0.0951 - val_acc: 0.9743
Epoch 8/500
2229/2229 [=====] - 377s 169ms/step - loss: 0.0904 - acc: 0.9727 - val_loss: 0.0759 - val_acc: 0.9784
Epoch 9/500
2229/2229 [=====] - 377s 169ms/step - loss: 0.0754 - acc: 0.9755 - val_loss: 0.0738 - val_acc: 0.9804
Epoch 10/500
2229/2229 [=====] - 377s 169ms/step - loss: 0.0620 - acc: 0.9805 - val_loss: 0.0493 - val_acc: 0.9860
Epoch 11/500
2229/2229 [=====] - 377s 169ms/step - loss: 0.0489 - acc: 0.9845 - val_loss: 0.0513 - val_acc: 0.9858
Epoch 12/500
2229/2229 [=====] - 377s 169ms/step - loss: 0.0427 - acc: 0.9868 - val_loss: 0.0551 - val_acc: 0.9863
Epoch 13/500
2229/2229 [=====] - 378s 170ms/step - loss: 0.0359 - acc: 0.9898 - val_loss: 0.0403 - val_acc: 0.9897
Epoch 14/500
2229/2229 [=====] - 377s 169ms/step - loss: 0.0324 - acc: 0.9903 - val_loss: 0.0442 - val_acc: 0.9889
Epoch 15/500
2229/2229 [=====] - 377s 169ms/step - loss: 0.0282 - acc: 0.9913 - val_loss: 0.0370 - val_acc: 0.9904
Epoch 16/500
2229/2229 [=====] - 377s 169ms/step - loss: 0.0269 - acc: 0.9911 - val_loss: 0.0468 - val_acc: 0.9884
Epoch 17/500
2229/2229 [=====] - 376s 169ms/step - loss: 0.0186 - acc: 0.9938 - val_loss: 0.0418 - val_acc: 0.9893
Epoch 18/500
2229/2229 [=====] - 376s 169ms/step - loss: 0.0179 - acc: 0.9940 - val_loss: 0.0440 - val_acc: 0.9895

79726/79726 [=====] - 886s 11ms/step
Warning: Looks like you're using an outdated API Version, please consider updating (server 1.4.2 / client 1.3.12)
Successfully submitted to State Farm Distracted Driver Detection
  
```

2022-09-06

13

5장. 머신러닝 탐구생활 (10/23)

⑤ 테스트 데이터 예측 및 캐글 업로드 (1/2)

- BASELINE 모델은 5-FOLD 교차검증의 매 FOLD마다 테스트 데이터에 대한 예측 및 예측 결과물을 캐글에 업로드
- 파이선 코드 : (p. 360)
 - 테스트 데이터에 대한 확률 값 예측 함수 : MODEL.PREDICT_GENERATOR()
 - 캐글에 업로드하는 코드
- 캐글에 업로드 되는 값은 리더보드에서 확인

2022-09-06

14

5장. 머신러닝 탐구생활 (11/23)

✓ ⑤ 테스트 데이터 예측 및 캐글 업로드 (2/2)

- 교차 검증에서 저장된 5-Fold별 모델 예측값을 기반으로 앙상블
 - 서로 다른 모델의 다양성을 활용하여 일반화 성능을 더욱 높이는 방법
 - 평균 앙상블하는 파이선 코드 : (p. 362)
 - 해석 : (p. 363)

2022-09-06

15

5장. 머신러닝 탐구생활 (12/23)

✓ ⑥ 성능개선 (1/23)

- BASELINE 모델의 과적합 원인은 잘못된 교차 검증 전략 때문
 - 랜덤으로 훈련 데이터를 훈련/검증 데이터로 분리하면 훈련/검증 데이터에 동일한 운전자의 이미지 데이터가 공통으로 포함
 - 이런 교차 검증 전략은 모델이 새로운 운전자의 이미지를 입력 값으로 받았을 때 성능을 올바르게 반영하지 못함

2022-09-06

16

5장. 머신러닝 탐구생활 (13/23)

✓ ⑥ 성능개선 (2/23)

■ 성능개선 1 : 훈련/검증 데이터 분리 기준을 운전자 ID로 선정

- 운전자 ID는 메타데이터로 제공
- 이미지별 운전자 ID를 구하는 파이선 코드 : (p. 365)
- 총 26명의 운전자를 5개 FOLD별로 분리, 훈련 데이터에 20명, 검증 데이터에 6명 배정으로 분리 : 코드 (p. 366~367)
- 코드를 수정하고 교차 검증을 진행 : 로그 (p. 368)

2022-09-06

17

5장. 머신러닝 탐구생활 (14/23)

✓ ⑥ 성능개선 (3/23)

■ 성능개선 2 : IMAGENET 경진대회에서 학습된 VGG16 모델 파라미터 사용

- 경진대회에 최적화된 새로운 모델 생성 : (p. 370)

2022-09-06

18

5장. 머신러닝 탐구생활 (15/23)

✓ ⑥ 성능개선 (4/23)

■ 성능개선 3 : 실시간 데이터 AUGMENTATION (★)

- 훈련 데이터의 입력 값에 약간의 노이즈를 추가, 모델이 학습할 데이터 양 증가, 과적합 피하고 높은 일반화 성능 기대
- 이미지 데이터 본래의 클래스 정보를 해치지 않는 수준, 랜덤으로 이미지 회전, 이동, 줌인, 줌아웃 등을 실시간 수행
- 데이터 어그멘테이션을 통해 다양한 형태의 노이즈 학습
- 실시간으로 미니배치 데이터를 생성하는 이미지 데이터 생성기 함수에 어그멘테이션 함수 추가 : 파이선 코드 (p. 372)
 - 노이즈 추가 : 이미지 회전은 -20~20도, 이동은 -30~30, 줌은 0.8~1.0
 - 매 EPOCH 마다 모델은 조금씩 다른 이미지를 입력으로 받음
- 학습 로그 : (p. 374)

2022-09-06

19

5장. 머신러닝 탐구생활 (16/23)

✓ ⑥ 성능개선 (5/23)

■ 성능개선 4 : 실시간 데이터 AUGMENTATION을 적용한 랜덤 교차 검증

- (p. 376)

2022-09-06

20



5장. 머신러닝 탐구생활 (17/23)

✓ ⑥ 성능개선 (6/23)

■ 성능개선 5 : 다양한 CNN 모델로 학습 (p. 378)

- VGG19 + 랜덤, VGG19 + 운전자별, RESNET50 + 랜덤, RESNET50 + 운전자별

2022-09-06

21



5장. 머신러닝 탐구생활 (18/23)

✓ ⑥ 성능개선 (7/23)

■ 성능개선 6 : 앙상블 (p. 380)

- VGG16, VGG19, RESNET50 결과물을 앙상블

2022-09-06

22



5장. 머신러닝 탐구생활 (19/23)

✓ ⑥ 성능개선 (8/23)

■ 성능개선 7 : 준지도 학습 (p. 381)

- 훈련 데이터 양을 보완하기 위하여 클래스 정보가 없는 테스트 데이터를 학습에 활용
- 모델 적용 : VGG16, VGG19, ResNet50

2022-09-06

23



5장. 머신러닝 탐구생활 (20/23)

✓ ⑥ 성능개선 (9/23)

■ 성능개선 8 : 최종 앙상블 (p. 383)

- 준지도 학습으로 얻은 3개 모델(VGG16, VGG19, ResNet50) 결과물을 가중 평균으로 앙상블

2022-09-06

24

5장. 머신러닝 탐구생활 (21/23)

✓ 상위 20위 캐글러들의 지혜 (p. 387)

- 부족한 훈련 데이터 양을 어떻게 보완/해결
- 앙상블 기법 (모델별 클래스마다 가중치를 다르게 줌)

2022-09-06

25

5장. 머신러닝 탐구생활 (22/23)

✓ 데이터 준비 (p. 318)

- 회원 가입 ~ 데이터 다운로드
 - 훈련 데이터와 테스트 데이터 파일
 - 훈련용 데이터에 대한 운전자 메타 데이터 파일
 - 캐글 제출용 샘플 파일

✓ 탐색적 데이터 분석 (p. 321)

- 폴더 구조
- 훈련용 데이터 클래스 (10개 클래스)
- 시각화로 데이터 살펴보기 : 훈련 데이터, 테스트 데이터, 운전자 메타 데이터, 특이한 훈련 데이터(OUTLIER, WRONG LABEL,)
- 데이터 AUGMENTATION

2022-09-06

26

5장. 머신러닝 탐구생활 (23/23)

✓ 기타

- 케라스에서 재현 가능

2022-09-06

27

이미지에서 노이즈 제거, 객체 식별 (1/2)

✓ 사례 1 : AI CAN NOW FIX YOUR GRAINY PHOTOS BY ONLY LOOKING AT GRAINY PHOTOS (NVIDIA)

- 노이즈가 많은 이미지와 깨끗한 이미지를 이용하여 딥러닝 네트워크를 훈련하는 대신, 노이즈 패턴이 다른 두 개의 이미지를 사용하여 훈련
- 노이즈가 없는 이미지를 실제로 보지 않고도 깨끗한 이미지 생성

✓ 사례 2 : GOOGLE'S NEW AI IMAGE NOISE REDUCTION COULD BE A GAME CHANGER

- 어둠 속에서 찍은 사진의 이미지를 손실시키지 않고 노이즈만 제거, 선명하게 만들어주는 AI 기술 'RAWNeRF' 개발
- 재구성한 이미지는 후처리된 JPEG 이미지가 아니라 원시(RAW) 이미지 데이터로 훈련 (훨씬 선명)
- 다양한 각도에서 촬영한 이미지를 결합하고 노이즈를 제거하는 방식으로 사진에서 어둠을 걷어내는 효과를 볼 수 있음

2022-09-06

28

이미지에서 노이즈 제거, 객체 식별 (2/2)

✓ 논문/게시글

- [DoPAMINE](#) (성균관대학교)
- [SUBTASK GATED NETWORKS FOR NON-INTRUSIVE LOAD MONITORING](#) (성균관대학교)
- [원자로 초소형 내시경의 실시간 영상 잡음 제거 기법](#) (목포대학교)
- [영상 분석에서의 AI - 머신 러닝 및 딥 러닝 기반의 분석을 위한 고려사항](#) (AXIS)
- [인공지능 이미지 인식 기술 동향](#) (TTA)
- [촬영영상과 인공지능 알고리즘 기반의 균열검출 자동화 프로그램을 이용한 터널 라이닝 표면의 균열 검출 방법 및 시스템](#) (특허자료)
- [OPENCV를 사용한 파이썬 이미지 처리\(이미지에서 노이즈 제거\)](#) (게시글)
- [MOBILE-UNET을 활용한 실시간 배경 제거 모델](#) (게시글)
- [실시간 배경 제거, REAL-TIME BACKGROUND REMOVAL](#) (게시글)
- [얼굴 감지를 구현하기 위한 OPENCV 로드 딥 러닝 모델](#) (게시글)
- [딥러닝과 OPENCV를 활용해 사진 속 글자 검출하기](#) (게시글)
- [OPENCV, 이미지에서 특정 이미지 찾아내기](#) (게시글)

2022-09-06

29

실습 데이터 확보 (1/2)

✓ 실습 데이터로 활용 가능 동영상

- [HTTPS://YOUTU.BE/k5ABNeK0k6w](https://youtu.be/k5ABNeK0k6w) (작전)
- [HTTPS://YOUTU.BE/BWY_HMEDWSs](https://youtu.be/BWY_HMEDWSs) (작전)
- [HTTPS://YOUTU.BE/dvT2TptuQfw](https://youtu.be/dvT2TptuQfw) (작전)
- [HTTPS://YOUTU.BE/HYDaJccliBg](https://youtu.be/HYDaJccliBg) (작전)
- [HTTPS://YOUTU.BE/U1YfO2BQno0](https://youtu.be/U1YfO2BQno0) (작전)
- [HTTPS://YOUTU.BE/hVHdJt2GzNU](https://youtu.be/hVHdJt2GzNU) (작전)
- [HTTPS://YOUTU.BE/w9p8OiqARNA](https://youtu.be/w9p8OiqARNA) (작전)
- [HTTPS://YOUTU.BE/1YuS7M4kVZg](https://youtu.be/1YuS7M4kVZg) (작전)
- [HTTPS://YOUTU.BE/VRD1ZHovAVc](https://youtu.be/VRD1ZHovAVc) (작전)
- [HTTPS://YOUTU.BE/qX5Hb_uVcTI](https://youtu.be/qX5Hb_uVcTI) (작전)
- [HTTPS://YOUTU.BE/RscSLag5WAI](https://youtu.be/RscSLag5WAI) (작전)
- [HTTPS://YOUTU.BE/NYKJrWMKFng](https://youtu.be/NYKJrWMKFng) (작전)
- [HTTPS://YOUTU.BE/FW3k80VUq34](https://youtu.be/FW3k80VUq34) (동물)
- [HTTPS://YOUTU.BE/EMdAm1JQuNs](https://youtu.be/EMdAm1JQuNs) (취)
- [HTTPS://YOUTU.BE/LDSBB8dQw6w](https://youtu.be/LDSBB8dQw6w) (동물)
- [HTTPS://YOUTU.BE/OqZvXWqMKQ0](https://youtu.be/OqZvXWqMKQ0) (동물)

2022-09-06

30

실습 데이터 확보 (2/2)

✓ 동영상을 이미지로 확보

- [FREE VIDEO TO JPG CONVERTER](#) (무료) (방법 설명)
- [HTTPS://ANYCONV.COM/KO/VIDEO-TO-JPG-BYEONHWANGI/](https://anyconv.com/ko/video-to-jpg-byeonhwangi/) (온라인)
- 방법 설명
- OPENCV로 구현

2022-09-06

31

실습 계획

구 분	월	화	수	목	금
1주차	머신러닝 탐구생활 5장 따라하기 (감잡기)			실습용 이미지 획득 (변환)	
2주차	2022 국방 데이터 활용 경진대회 관계자 초청 토론 (가용 시)	(→) 실습 (환경 이해)	숙달 실습1 (🔗)		토의
3주차	숙달 실습2 (🔗)		숙달 실습3 (🔗)		토의

2022-09-06

32

