



2023 데이터 크리에이터 캠프 DATA CREATOR CAMP

대학부

(공주대인공)

팀장 (손건희)

(팀원 강정선)

(팀원 손재현)

(팀원 노민주)

목차

1. 미션_1

1. 데이터 분석하기(사진 분석)
2. 데이터 전처리(사진 자르기, 사진 삭제)
3. 데이터 모델링(모델성능 개선점)
4. 최종 정확도, 배운점
5. 미션별 답변

3. 미션_3

1. 데이터 분석하기(사진 분석)
2. 데이터 전처리(사진 자르기, 사진 삭제)
3. 데이터 모델링(모델성능 개선점)
4. 최종 정확도, 배운점
5. 미션별 답변

2. 미션_2

1. 데이터 분석하기(사진 분석)
2. 데이터 전처리(사진 자르기, 사진 삭제)
3. 데이터 모델링(모델성능 개선점)
4. 최종 정확도, 배운점
5. 미션별 답변

조건

Classifier의 아키텍처(architecture)로는 ResNet계열을 사용해야 한다.

ResNet18, 34, 50 등

제출파일(ipynb)에서 최종결과와 관련없는 cell은 삭제하고 마크다운 및 주석으로 해당 cell의 설명을 정리한다.

Pretrained weights를 사용할 수 없다.

Validation accuracy 측정 시 제공된 코드를 사용한다.

성능 평가지표는 Top-1 accuracy를 사용한다.

역할 분담

손건희 : **데이터 모델링**(모델 하이퍼 파라미터 개선, 모델 구조 개선)

강정선 : **데이터 모델링**(모델 구조 개선, 모델 하이퍼 파라미터 개선)

손재현 : **데이터 전처리**(이미지 편집, 이미지 삭제)

노민주 : **데이터 전처리**(이미지 편집, 이미지 삭제)

미션_1_데이터 분석하기(사진 분석)

- 먼저 데이터는 train_set과 val_set으로 2개의 폴더를 받았고 미션 2에서 사용되는 데이터의 경우에는 42개의 클래스로 이루어진 데이터였다.
- 클래스 당 **약 900개** 정도가 있고 **약 33000개** 정도가 있었습니다.
- 본격적으로 사진을 분석하면서 사진들에 **많은 이상치**들이 존재했습니다. 예를 들자면 갈치 구이 사진이지만 갈치가 있는것이 아니고 다른 이상한 음식들과 섞여있거나 평범한 사람이 봐도 갈치라고 여기지 못할 만한 사진들을 사진을 자르던가 또는 삭제하는 방식으로 데이터를 전처리를 진행하고자하는 계획을 세우게 되었습니다.

미션_1_데이터 전처리(사진 자르기, 사진 삭제)

미션_1과 같은 경우는 주어진 조건에 맞게 기본 모델을 사용하여 일반적인 성능을 뽑아보았기 때문에 전처리는 진행하지 않았습니다.

미션_1_데이터 모델링(모델성능 개선점)

- 미션_1 같은 경우는 어떤 클래스가 있는지 확인하고, 데이터는 어떤 형식으로 이루어져 있는지, 이미지는 어떤 이미지가 있는 것을 이해하는 시간으로 사용했고 모델 향상을 시작하기 전 저희는 초기의 기본적인 하이퍼파라미터를 아래와 같이 설정하게 되었습니다.

- learning_rate : 0.01
- model : resnet_18
- batch_size : 32
- optimazition : RAdam

MISSION 1. 한국 이미지(음식) 데이터셋을 분류하시오.

(train/val 폴더 내에 메뉴별 폴더가 들어있다. = 갈비구이, 갈치구이 등 총 42종)

1-1. 각 메뉴를 클래스로 하는 분류 데이터셋과 데이터로더를 준비하고
예시 이미지를 클래스별로 하나씩 총 42장을
한번에 시각화(plotting)하여 확인한다.

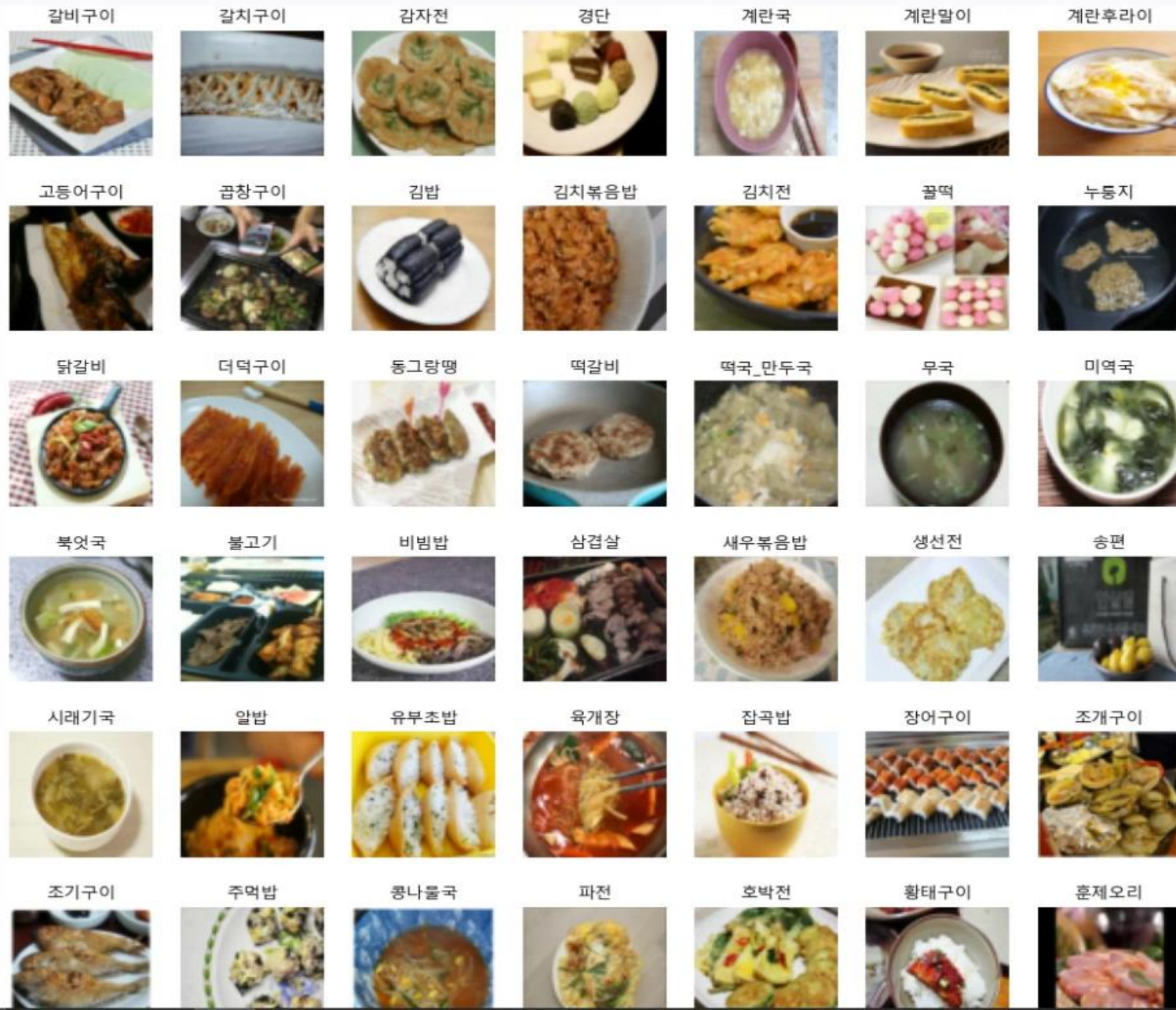
1-2. ResNet18를 활용하여 42종의 클래스 분류를 수행하고
Validation 데이터에 대한 정확도를 제시한다.

- ResNet18의 parameters는 무작위로 초기화하여 사용한다.
(즉, pretrained weights는 사용할 수 없다.)
- 학습 길이는 50 epoch를 기본으로 하되 추가해도 가능하다.

42장 중
예시 9장



1-1. 각 메뉴를 클래스로 하는 분류 데이터 색과 데이터를 준비하고 예시 이미지를 클래스별로 하나씩 42장을 한번에 시각화(Plotting)하여 확인하다.



미션별 답변

1-2. ResNet18를 활용하여 42종의 클래스 분류를 수행하고 Validation 데이터에 대한 정확도를 제시한다.

- 미션_1_정확도 : train_acc : 98.84% / val_acc : 63.77%

<MISSION_2>

[심화] 한국 이미지(음식) 데이터셋을 분류하시오

미션_2_데이터 분석하기(사진 분석)

- 먼저 데이터는 train_set과 val_set으로 2개의 폴더를 받았고 미션 2에서 사용되는 데이터의 경우에는 42개의 클래스로 이루어진 데이터였다. 이 데이터의 경우는 주로 워터마크와 많은 글씨들이 있는 것이 특징이었다. 더 나아가 학습에 관련없는것과 같은 술과 다른 음식들이 섞여있는 train_set이었다.
- 클래스 당 **약 900개** 정도가 있고 **약 33000개** 정도가 있었습니다.
- 본격적으로 사진을 분석하면서 사진들에 **많은 이상치**들이 존재했습니다. 예를 들자면 갈치 구이 사진이지만 갈치가 있는것이 아니고 다른 이상한 음식들과 섞여있거나 평범한 사람이 봐도 갈치라고 여기지 못할 만한 사진들을 사진을 자르던가 또는 삭제하는 방식으로 데이터를 전처리를 진행하고자하는 계획을 세우게 되었습니다.

미션_2_데이터 전처리(전처리 필요성)

사진에 하나의 음식뿐만 아니라 다양한 음식이 존재하는 경우



미션_2_데이터 전처리(전처리 필요성)

전 음식이 많았는데 모든 전에는
간장 종지가 있음, 공통된 특징
은 핫가리는 요소라는 가설



미션_2_데이터 전처리(전처리 필요성)

우리가 분류하는 모델 중에 계란 후라이와 시레기 국이 있는데 두 개의 음식이 같이 존재하는 경우 확인



미션_3_데이터 전처리(사진 자르기, 사진 삭제)

전처리 전



전처리 후



미션_3_데이터 전처리(사진 자르기, 사진 삭제)

전처리 전



전처리 후



미션_3_데이터 전처리(사진 자르기, 사진 삭제)

전처리 전



전처리 후



미션_2_데이터 모델링(하이퍼파라미터)

하이퍼파라미터

미션_1과 다른 점은 하이퍼 파라미터 튜닝도 진행했습니다. 학습률을 조금 더 낮추니까 성능의 향상을 가져올 수 있었습니다.

- learning_rate : 0.001
- model : resnet_18
- batch_size : 32
- optimazition : RAdam

미션_2_데이터 모델링(모델성능 개선점)

L_2 규제 적용

데이터 모델링은 처음에는 미션_1의 성능 개선을 위해서 과적합 현상을 없애기 위해서 규제를 사용했다. 규제는 처음은 강하게 먹히면서 점차 줄여가는 식으로 주었는데 결과는 학습이 안되는 현상만 발생할 뿐 val_acc의 성능이 좋아지는 것은 확인 하지 못했다.

미션_2_데이터 모델링(모델성능 개선점)

데이터 증강 사용

데이터 모델링은 처음에는 미션_1의 성능 개선을 위해서 과적합 현상을 없애기 다른 접근 방법으로는 모델에게 다양한 학습이미지를 제공해주는 것이 또 다른 방법이 될 수 있다고 생각하고 증강 기법으로 수직으로 수평으로 랜덤으로 섞어주는 기능을 사용해 보았다. 결과는 val_acc 가 62 %-> 76% 향상되는 것을 확인할 수 있었다.

미션_2_데이터 모델링(모델성능 개선점)

데이터 전처리

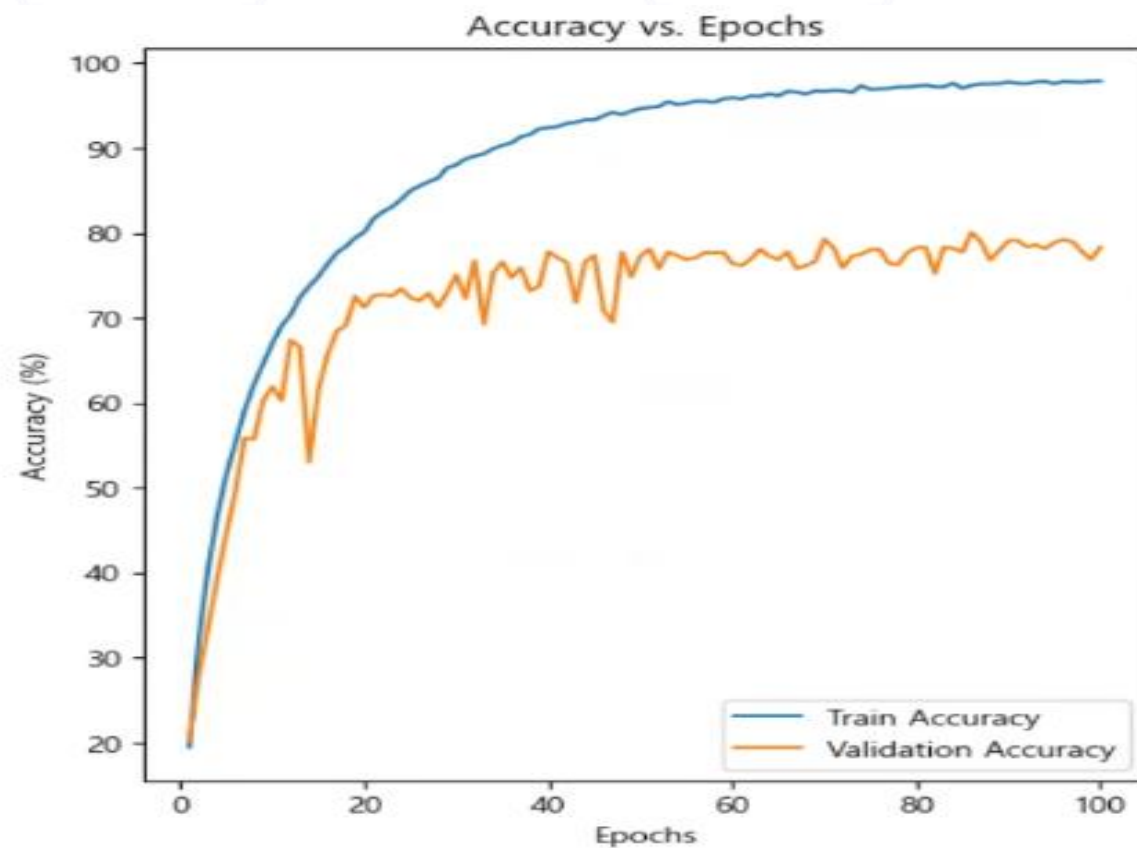
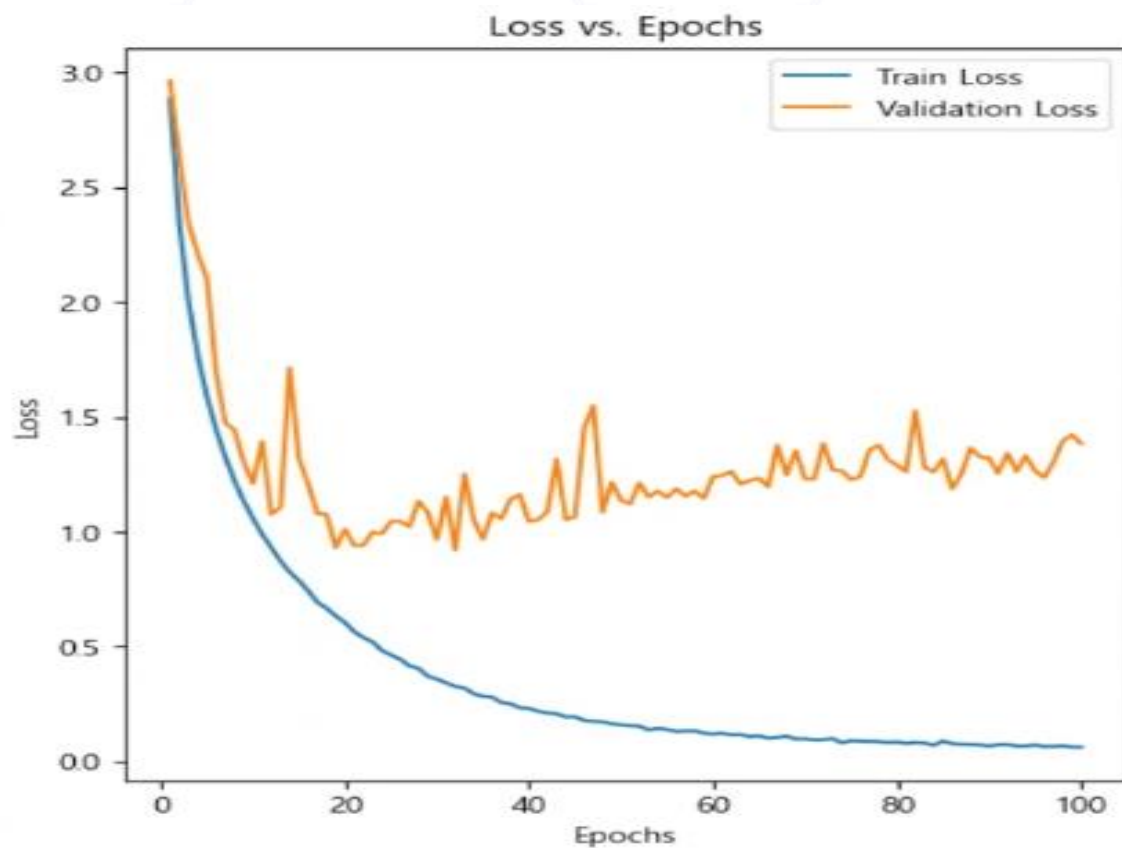
데이터 모델링은 처음에는 미션_1의 성능 개선을 위해서 과적합 현상을 없애기 다른 접근 방법으로는 앞에서 2가지의 시도를 해보고 늦게나마 느꼈다. 이걸 학습이미지가 val와 많이 다른 것이 즉 이상치가 존재하고 그것을 학습하면서 발생하는 문제라는 것을 알게 되었고 우리는 모든 이미지를 하나씩 필요없는 부분을 삭제하는 방향으로 진행했다. 성능은 76% -> 80.06% 으로 향상되었습니다.

최종 정확도, 배운점

- 미션_2_정확도 : train_acc : 97.39% val_acc : 80.06%

- 미션_2_배운점 : 우리팀은 맨 처음에는 데이터의 과적합을 잡기 위해서 하이퍼 파라미터 조정을 먼저 했다. 하지만 미션을 진행하면서 느낀 것을 앞으로 우리가 일할 현업에서도 그렇고 대회에서 가장 중요한 것은 데이터 전처리 과정이라는 것을 배웠다.

최종 정확도, loss 시각화



MISSION 2. [심화] 한국 이미지(음식) 데이터셋을 분류하시오.

2-1. Mission 1의 결과를 분석한다.

- Mission 1에서 맞춘 샘플의 공통점이 있는가?
- 틀린 샘플들의 공통점이 있는가?
- 맞춘/틀린 샘플들의 차이점이 있는가?

2-2. 2-1의 분석을 기반으로 성능 향상을 위한 작업을 수행한다.

선택지 예시

- 아키텍처(architecture) 변경: ResNet18,34,50 등
- LR scheduling, augmentation, regularization 등
(단, ResNet 아키텍처만 사용)

2-3. 선택지들의 성능 경향을 파악한다.

- 어떤 논리/기대를 가지고 어떤 수행을 했나?
- 그에 따라 정확도는 어떻게 변했는가?
- 논리/기대와 결과가 같은가? 다른가? 왜 그런 결과가 나왔는가?

미션별 답변

2-1. Mission_1의 결과를 분석한다.

[Mission_1 에서 맞춘 샘플의 공통점이 있는가?]

[틀린 샘플들을 공통점이 있는가?]

[맞춘/틀린 샘플들의 차이점이 있는가?]

다음 페이지에 이미지 제공하고 설명드리겠습니다.

Class 송편 - Train Accuracy: 89.61%, Validation Accuracy: 71.23%

Class 시래기국 - Train Accuracy: 89.97%, Validation Accuracy: 66.15%

Class 알밥 - Train Accuracy: 92.22%, Validation Accuracy: 70.54%

Class 유부초밥 - Train Accuracy: 87.90%, Validation Accuracy: 41.30%

Class 육개장 - Train Accuracy: 92.36%, Validation Accuracy: 69.79%

Class 잡곡밥 - Train Accuracy: 91.46%, Validation Accuracy: 66.27%

Class 장어구이 - Train Accuracy: 86.29%, Validation Accuracy: 43.49%

Class 조개구이 - Train Accuracy: 91.95%, Validation Accuracy: 64.46%

Class 조기구이 - Train Accuracy: 88.38%, Validation Accuracy: 58.56%

Class 주먹밥 - Train Accuracy: 86.89%, Validation Accuracy: 44.88%

Class 콩나물국 - Train Accuracy: 86.25%, Validation Accuracy: 58.70%

Class 파전 - Train Accuracy: 89.96%, Validation Accuracy: 59.03%

Class 호박전 - Train Accuracy: 89.12%, Validation Accuracy: 50.28%

Class 황태구이 - Train Accuracy: 89.61%, Validation Accuracy: 61.38%

Class 훈제오리 - Train Accuracy: 89.59%, Validation Accuracy: 62.53%

Class 갈비구이 - Train Accuracy: 86.64%, Validation Accuracy: 41.72%

Class 갈치구이 - Train Accuracy: 88.52%, Validation Accuracy: 51.47%

Class 감자전 - Train Accuracy: 84.91%, Validation Accuracy: 40.85%

Class 경단 - Train Accuracy: 86.60%, Validation Accuracy: 48.13%

Class 계란국 - Train Accuracy: 91.33%, Validation Accuracy: 62.00%

Class 계란말이 - Train Accuracy: 86.51%, Validation Accuracy: 43.93%

Class 계란후라이 - Train Accuracy: 91.64%, Validation Accuracy: 61.49%

Class 고등어구이 - Train Accuracy: 86.61%, Validation Accuracy: 48.26%

Class 곱창구이 - Train Accuracy: 88.43%, Validation Accuracy: 45.36%

Class 김밥 - Train Accuracy: 91.13%, Validation Accuracy: 73.06%

Class 김치볶음밥 - Train Accuracy: 90.18%, Validation Accuracy: 61.69%

Class 김치전 - Train Accuracy: 90.14%, Validation Accuracy: 57.55%

Class 꿀떡 - Train Accuracy: 91.99%, Validation Accuracy: 66.04%

Class 누룽지 - Train Accuracy: 88.25%, Validation Accuracy: 64.94%

Class 닭갈비 - Train Accuracy: 90.51%, Validation Accuracy: 58.59%

Class 더덕구이 - Train Accuracy: 89.81%, Validation Accuracy: 56.44%

Class 동그랑땡 - Train Accuracy: 86.05%, Validation Accuracy: 36.36%

Class 떡갈비 - Train Accuracy: 87.19%, Validation Accuracy: 47.75%

Class 떡국_만두국 - Train Accuracy: 89.33%, Validation Accuracy: 58.67%

Class 무국 - Train Accuracy: 90.11%, Validation Accuracy: 60.02%

Class 미역국 - Train Accuracy: 94.36%, Validation Accuracy: 80.76%

Class 북엇국 - Train Accuracy: 87.47%, Validation Accuracy: 49.81%

Class 불고기 - Train Accuracy: 86.77%, Validation Accuracy: 50.26%

Class 비빔밥 - Train Accuracy: 90.47%, Validation Accuracy: 64.24%

Class 삼겹살 - Train Accuracy: 86.40%, Validation Accuracy: 45.08%

Class 새우볶음밥 - Train Accuracy: 91.23%, Validation Accuracy: 61.75%

Class 생선전 - Train Accuracy: 86.77%, Validation Accuracy: 40.57%

미션별 답변

[틀린 샘플들을 공통점이 있는가?]

Class 동그랑땡 - Train Accuracy: 86.05%, Validation Accuracy: 36.36%

Class 생선전 - Train Accuracy: 86.77%, Validation Accuracy: 40.57%

Class 감자전 - Train Accuracy: 84.91%, Validation Accuracy: 40.85%

틀린 샘플들의 공통점으로는 간장종지에 찍어먹는데 서로의 음식 색도 비스듬한 경우를 모델이 헛갈려한다고 생각할 수 있었습니다. 즉 우리는 이것을 개선하기 위해서 전처리를 진행할때 해당 부분에 초점을 맞추어서 진행을 했습니다.

미션별 답변

2-1. Mission_1의 결과를 분석한다.

[Mission_1 에서 맞춘 샘플의 공통점이 있는가?]

[틀린 샘플들을 공통점이 있는가?]

[맞춘/틀린 샘플들의 차이점이 있는가?]

Class 미역국 - Train Accuracy: 94.36%, Validation Accuracy: 80.76%

Class 김밥 - Train Accuracy: 91.13%, Validation Accuracy: 73.06%

특히 성능이 월등하게 잘 나온 class는 이미지를 살펴보면(전처리 전) 다른 음식들과 다르게 먹는 음식에서 같이 먹는 음식이 한창되어있거나 예를 들면 김치, 밥 또는 주위 사진들에 에러를 이르기면서 학습이 될 만한 사물과 음식이 상대적으로 적다라는 것을 알 수 있었습니다. 즉 저희는 전처리 하는 과정에서 주위 사물도 영향을 줄 수 있겠다라는 생각하고 전처리를 진행하게 되었습니다.

미션별 답변

2-2. 2-1의 분석을 기반으로 성능 향상을 작업을 수행한다.

먼저 2-1의 분석을 기반으로 결국은 데이터 전처리를 진행하는데 어떤 부분이 데이터 전처리가 필요하고 어떤 부분을 중요시하면서 데이터를 만들어가야하는지에 대해서 이해할 수 있었다. 예를 들어 간장종이를 제거 전의 특유의 특징을 잘 지키면서 진행했다. 그리고 주위 음식들이 없고 한정되어 있다면 해당 특징을 파악하는데 도움이 된다는 것입니다.

미션별 답변

2-3. 선택지들의 성능 경향을 파악한다.

다음 페이지에 설명하겠습니다.

2-3. 선택지들의 성능 경향을 파악한다.

데이터 증강 사용

데이터 모델링은 처음에는 미션_1의 성능 개선을 위해서 과적합 현상을 없애기 다른 접근 방법으로는 모델에게 다양한 학습이미지를 제공해주는 것이 또 다른 방법이 될 수 있다고 생각하고 증강 기법으로 수직으로 수평으로 랜덤으로 섞어주는 기능을 사용해 보았다. 결과는 val_acc 가 62 %-> 76% 향상되는 것을 확인할 수 있었다.

2-3. 선택지들의 성능 경향을 파악한다.

L_2 규제 적용

데이터 모델링은 처음에는 미션_1의 성능 개선을 위해서 과적합 현상을 없애기 위해서 규제를 사용했다. 규제는 처음은 강하게 먹히면서 점차 줄여가는 식으로 주었는데 결과는 학습이 안되는 현상만 발생할 뿐 val_acc의 성능이 좋아지는 것은 확인 하지 못했다.

2-3. 선택지들의 성능 경향을 파악한다.

데이터 전처리

데이터 모델링은 처음에는 미션_1의 성능 개선을 위해서 과적합 현상을 없애기 위해서 규제를 사용했다. 그리고 데이터 증강도 사용했지만 많은 성능 향상을 가져오지는 못했다. 이후 시도한 것은 미션_1의 클래스 분류 결과를 보고 이미지와 대조해 보면서 차이점과 공통점을 찾아가기 시작했다. 이후 전처리 성능은 76% -> 80.06% 으로 향상되었습니다.

<MISSION3>

MISSION2 에서 학습시킨 모델로 건강관리를 위한 음식이미지 데이터셋 분류하시오.

미션_3_데이터 분석하기(사진 분석)

- 먼저 데이터는 train_set과 val_set으로 2개의 폴더를 받았고 미션 3에서 사용되는 데이터의 경우에는 13개의 클래스로 이루어진 데이터였습니다.
- 클래스 당 **약 1000개** 정도가 있고 **약 14000개** 정도가 있었습니다.
- 적은 것은 약 900 개 많은 것은 약 1500 개 정도 존재하는 클래스 간의 차이를 발견했습니다.
- 본격적으로 사진을 분석하면서 사진들에 **많은 이상치**들이 존재했습니다. 예를 들자면 주위에 많은 반찬이나, 많은 사물들이 존재했습니다. 그리고 공백의 여부가 많이 존재하는 사진이나, 중앙이 뚫여있고 사이드에 음식이 배치되어있는 사진들도 상당히 분포하는 것을 확인 할 수 있었습니다.

미션_3_데이터 모델링(하이퍼파라미터)

하이퍼파라미터

미션_2과 다른 점은 하이퍼 파라미터 튜닝도 진행했습니다.

- learning_rate : 0.001
- model : resnet_50
- batch_size : 32
- optimazition : RAdam

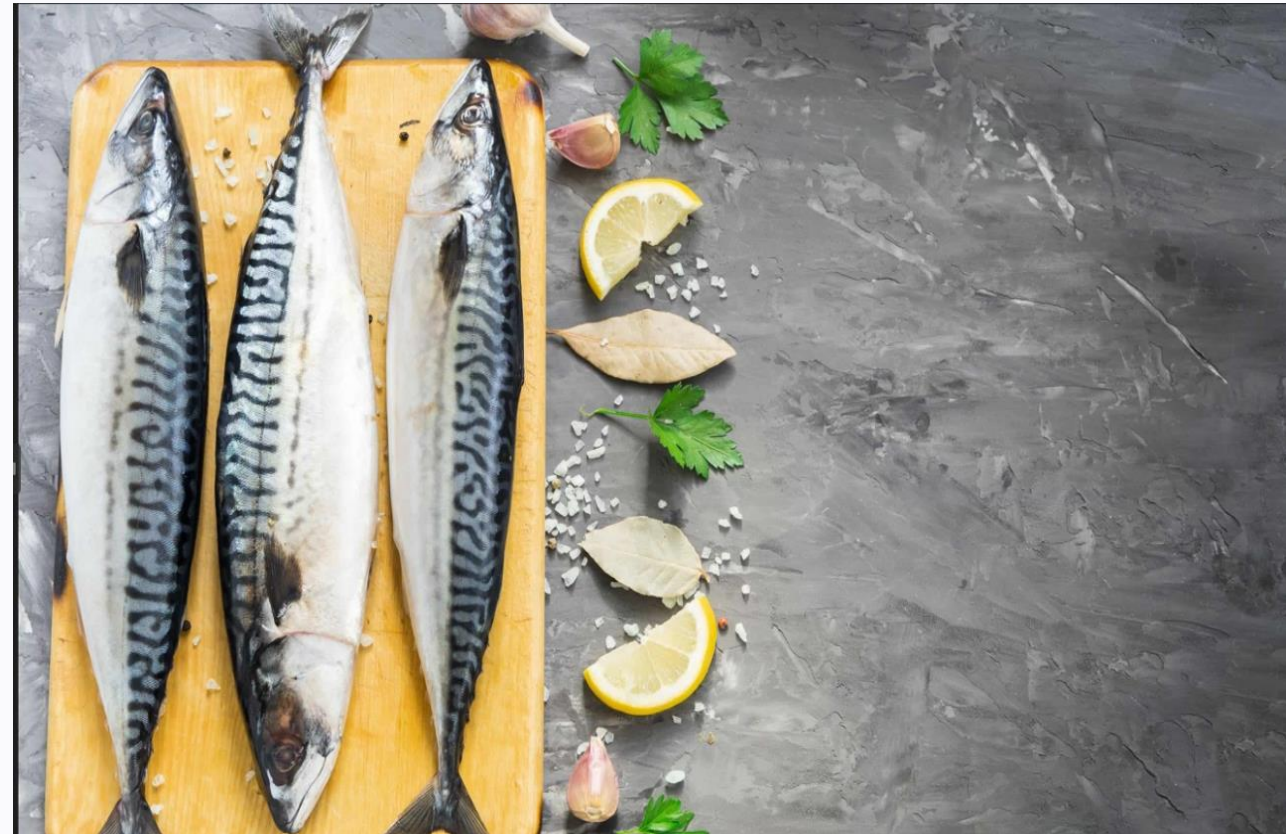
미션_3_데이터 분석하기(전처리 필요성)

연관성 없는 음식과 장식품이 같이 있는 경우
-> 연관성 없는 사물이나 음식제거



미션_3_데이터 분석하기(전처리 필요성)

음식 사진이 많은 공백을 가지고 한쪽으로 치우쳐 있는 경우 -> 공백을 없애는 쪽으로 전처리 진행



미션_3_데이터 전처리(사진 자르기, 사진 삭제)

전처리 전

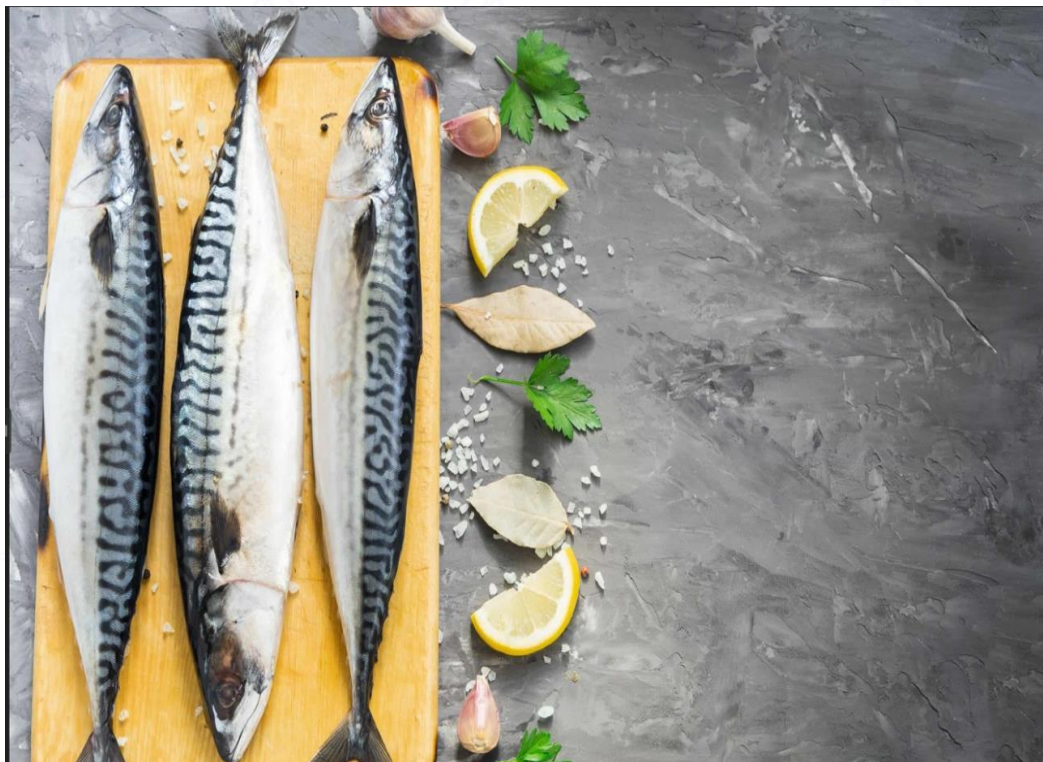


전처리 후



미션_3_데이터 전처리(사진 자르기, 사진 삭제)

전처리 전



전처리 후



미션_3_데이터 모델링(모델성능 개선점)

먼저 모델은 미션_2의 모델을 이용해서 진행했었는데 성능은 90%까지 나왔지만 모델이 안정적이지 않다는 단점이 존재했고 이미지를 보면서 이상한 이미지들이 존재하는 것을 확인할 수 있었다.

우리팀은 이 이미지 전처리 작업을 진행했습니다. 미션_3의 이미지는 미션_2와 다르게 마치 전문 사진관이 찍은 것과 같은 정갈한 이미지 였고 처리할 부분은 공백이 많거나 다른 음식이나 사진이 있는경우를 제거해 나갔습니다. 그렇게 하고 다른 부분은 동일하게 유지했음에도 불구하고 모델이 더욱 안정적이게 학습되는 경향성을 느낄 수 있었습니다.

최종 정확도, 배운점

- 미션_3_정확도 : train_acc : 97.26% val_acc : 96.88

- 미션_3_배운점 : 미션_3에서 배운 점은 2와 동일하게 데이터 전처리가 정말 중요하다는 것이다. 데이터를 전처리를 진행하으로써 더욱 안정적인 모델을 얻을 수 있었다.

MISSION 3. Mission2에서 학습시킨 모델로 건강관리를 위한 음식 이미지 데이터셋을 분류하시오.

3-1. Mission2에서 학습시킨 모델을 활용하여 건강관리를 위한 음식 이미지 데이터를 13개의 클래스로 분류하는 모델을 만들고, 학습시킨다.

HINT : transfer learning, linear probing, fine-tuning, few-shot learning 등

* 가장 우수한 성능을 달성한 세팅으로 제출한다.

3-2. 성능 경향을 파악한다.

- 어떤 논리/기대를 가지고 어떤 수행을 했나?
- 그에 따라 정확도는 어떻게 변했는가?
- 논리/기대와 결과가 같은가? 다른가? 왜 그런 결과가 나왔는가?

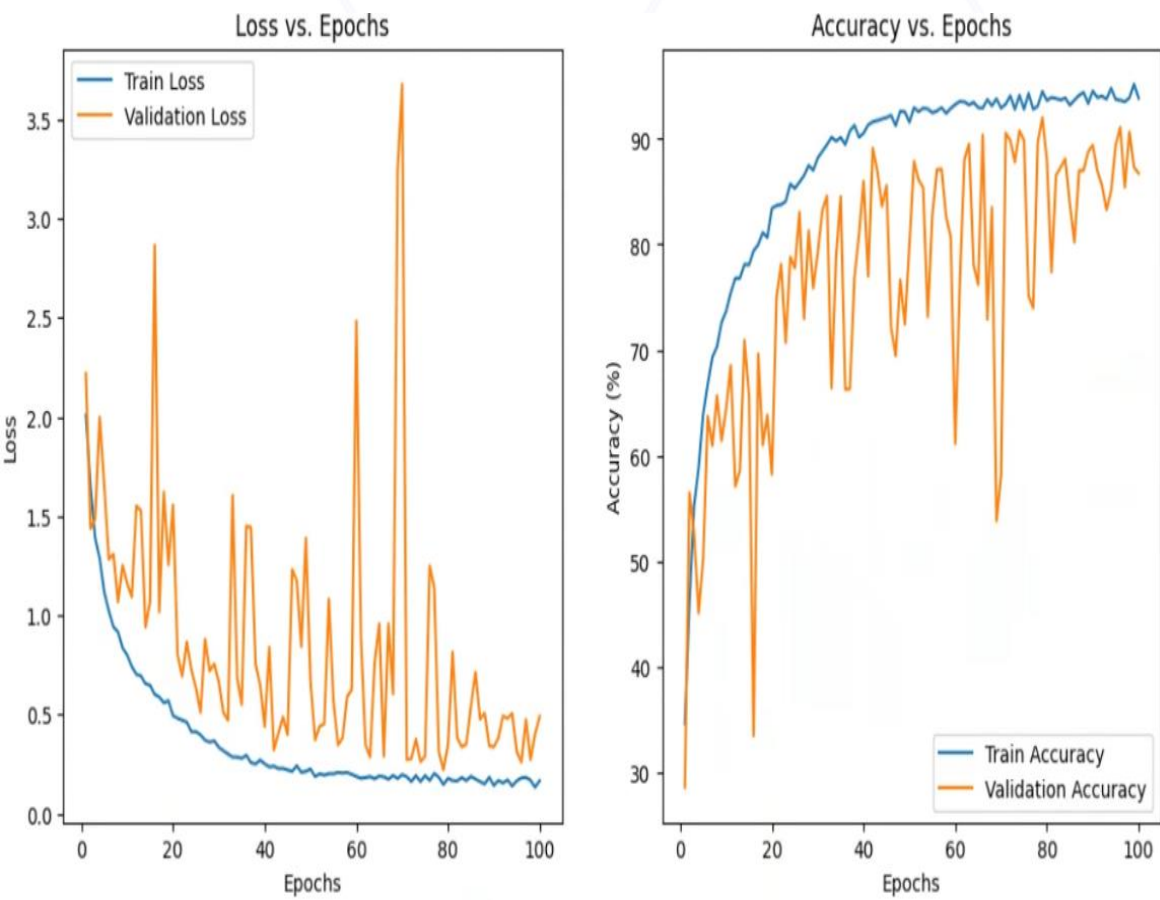
미션별 답변

3-1. Mission_2에서 학습시킨 모델을 활용하여 건강관리를 위한 음식 이미지 데이터를 13개의 클래스로 분류하는 모델을 만들고, 학습시킨다.

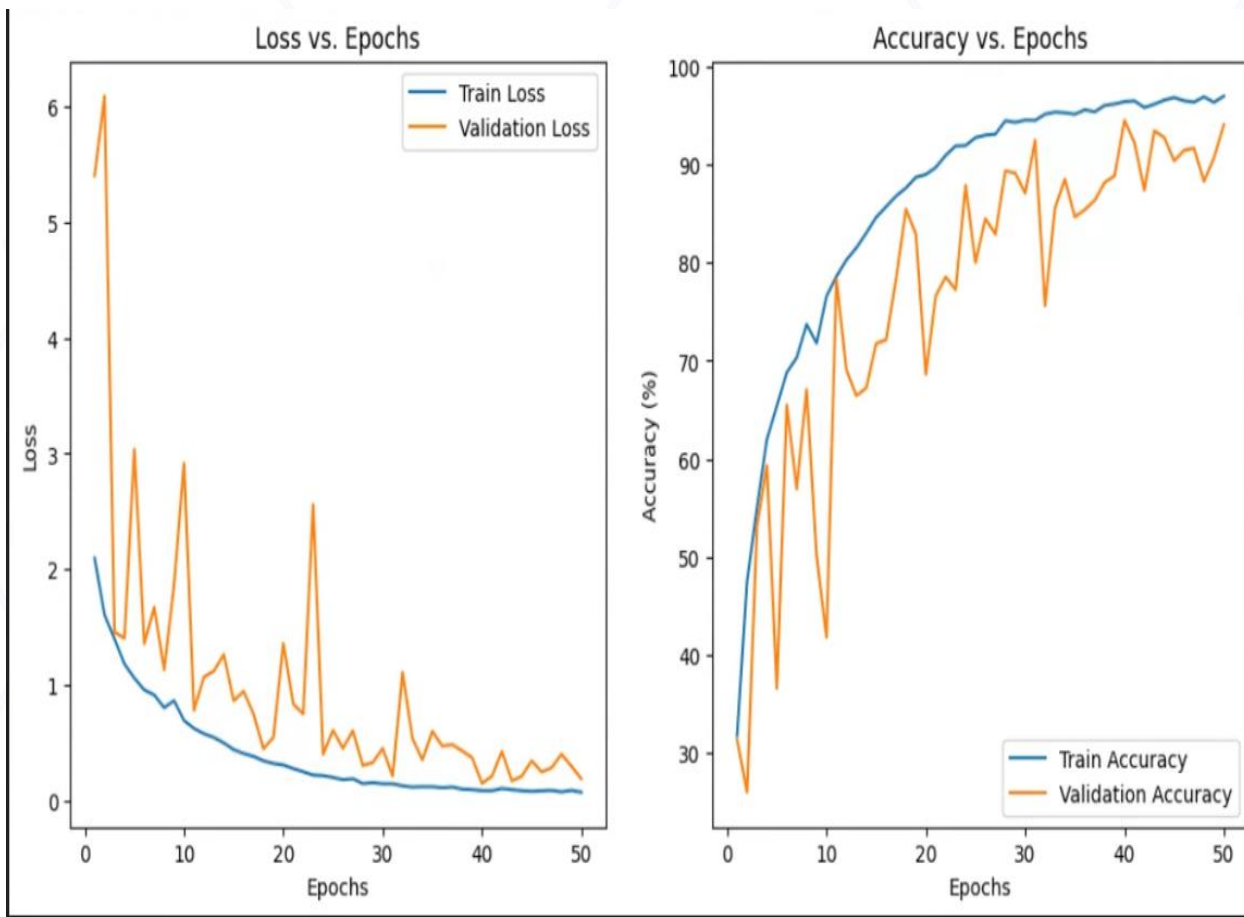
- 기본적으로 미션_2에서 넣은 모델을 이용해서 성능을 평가해본 결과는 상당히 좋았습니다. val_acc 가 90% 정도가 나왔고 하지만 모델이 안정적으로 학습하지는 못했습니다. 그래서 저희팀은 전처리를 진행했고 결과는 더욱 안정적인 모델을 얻을 수 있었습니다.
- 사진을 다음장에 제시하겠습니다.

미션별 답변

전처리 전



전처리 후



미션별 답변

3-2. 성능 경향을 파악한다.

성능은 원래 기존의 모델을 사용했을때 부터 좋았지만, 학습 중에 한 번씩 val_acc가 많이 감소하는 부분이 종종 존재하는 것을 확인할 수 있었습니다. 그래서 그러한 불안정한 학습을 개선하기 위해서 데이터 전처리를 통해서 성능을 향상시킬 수 있었습니다. 결국은 가장 중요한것은 데이터를 어떻게 전처리할지에 대한 고민과 시도가 중요하다는 것을 배울 수 있었습니다

- 초기에 확인하였던 클래스 간의 차이점을 확인하고 클래스 갯수를 맞추어주어보았지만 그렇게 뛰어난 차이변화를 느끼지 못해서 클래스의 파일 갯수를 맞추는 것은 시도하지 않았습니다. 클래스 간의 차이가 이진 분류와 같은 것은 많은 영향을 줄 수 있지만 또는 클래스가 극단적으로 많이 차이가 난다면 그럴 수 있지만 저희 그러한 상황이 아니기 때문에 학습을 진행하면서 성능의 차이가 많이나는 현상이 상대적으로 적었다고 생각합니다.

감사합니다.

대학부

(공주대인공)

팀장 (손건희)

(팀원 강정선)

(팀원 손재현)

(팀원 노민주)

2023 데이터 크리에이터 캠프

DATA CREATOR CAMP