



KDT 전이 학습

세계 동전 분류

2건 얼마조? 신민지 신승준 최수창 한아름

22 April, 2025



▶ 목차



AlexNet

신민지



VGG16

한아름



EfficientNet

신승준



ResNet

최수창



▶ 선정 배경

Coin Sorter Market To Reach USD 9.0 Billion By 2032

The coin sorter market size was valued at USD 4.4 Billion in 2023 and is expected to reach USD 9.0 Billion by 2032 at a CAGR of 8.2%. Any delay or error in coin handling can disrupt the customer experience and potentially impact revenue streams. Coin sorters are becoming a strategic investment for businesses aiming to optimize cash management and improve operational efficiency. Integrating machine learning and computer vision into coin sorting significantly improves detection and accuracy across multiple currencies. Machine learning enables coin sorters to continuously adapt and improve as currency designs evolve.

출처: Yahoo Finance, May 13 2024, <https://finance.yahoo.com/news/coin-sorter-market-reach-usd-084000736>.

동전 분류기는 기업의 현금 처리 효율을 높이는 전략적 기술로 주목받고 있으며,
머신러닝과 컴퓨터 비전 기술을 활용한 고도화가 활발히 진행 중

▶ 선정 배경



\$1 Million in Coins Left at Airports

There has been a spate of news articles favoring electronic payment transfers as opposed to physical coins due to the environmental damage caused by mining metals and then minting them into currency. Coins involve resource-intensive processes, including mining for zinc and copper. Operating digital currencies depends on electricity generated from fossil fuels, which contributes to greenhouse gas emissions and other environmental impacts. At the end of 2020, the approximately 50.3 billion bank notes in circulation had an estimated value of \$2.04 trillion. The TSA reported air travelers leaving nearly \$1 million in small change coins behind at checkpoints in 2023. Physical and electronic money are likely to continue to be strange bedfellows.

출처: Numismatic News, May 30 2024, <https://www.numismaticnews.net/coin-market/1-million-in-coins-left-at-airports>

2023년 미국 공항 보안 검색대에서 여행객들이 남긴 동전은 약 100만 달러에 달했으며,
이는 동전의 관리와 분류 시스템의 필요성을 보여줌

▶ 데이터셋



전 세계 여러 국가의 동전 이미지 set

- 출처: [kaggle](#)
- 데이터

총 211개의 클래스	train 데이터	6413장
	valid 데이터	844장
	test 데이터	817장

- cat_to_name.json

	classID	동전 명	통화 단위	국가
"1":	"1	Cent,	Australian dollar,	australia",
"2":	"2	Cents,	Australian dollar,	australia",
"3":	"5	Cents,	Australian dollar,	australia",
"4":	"10	Cents,	Australian dollar,	australia",
"5":	"20	Cents,	Australian dollar,	australia",
"6":	"50	Cents,	Australian dollar,	australia",



1. AlexNet

신민지



▶ 데이터 전처리

Resize

AlexNet 입력 크기에 맞추기 위해
224 x 224로 크기로 조정

Normalize

ImageNet 사전 모델 기준으로 정규화
평균 [0.485, 0.456, 0.406],
표준편차 [0.229, 0.224, 0.225]



ToTensor

PIL 이미지를 PyTorch 텐서로 변환
0~1 범위로 정규화 됨

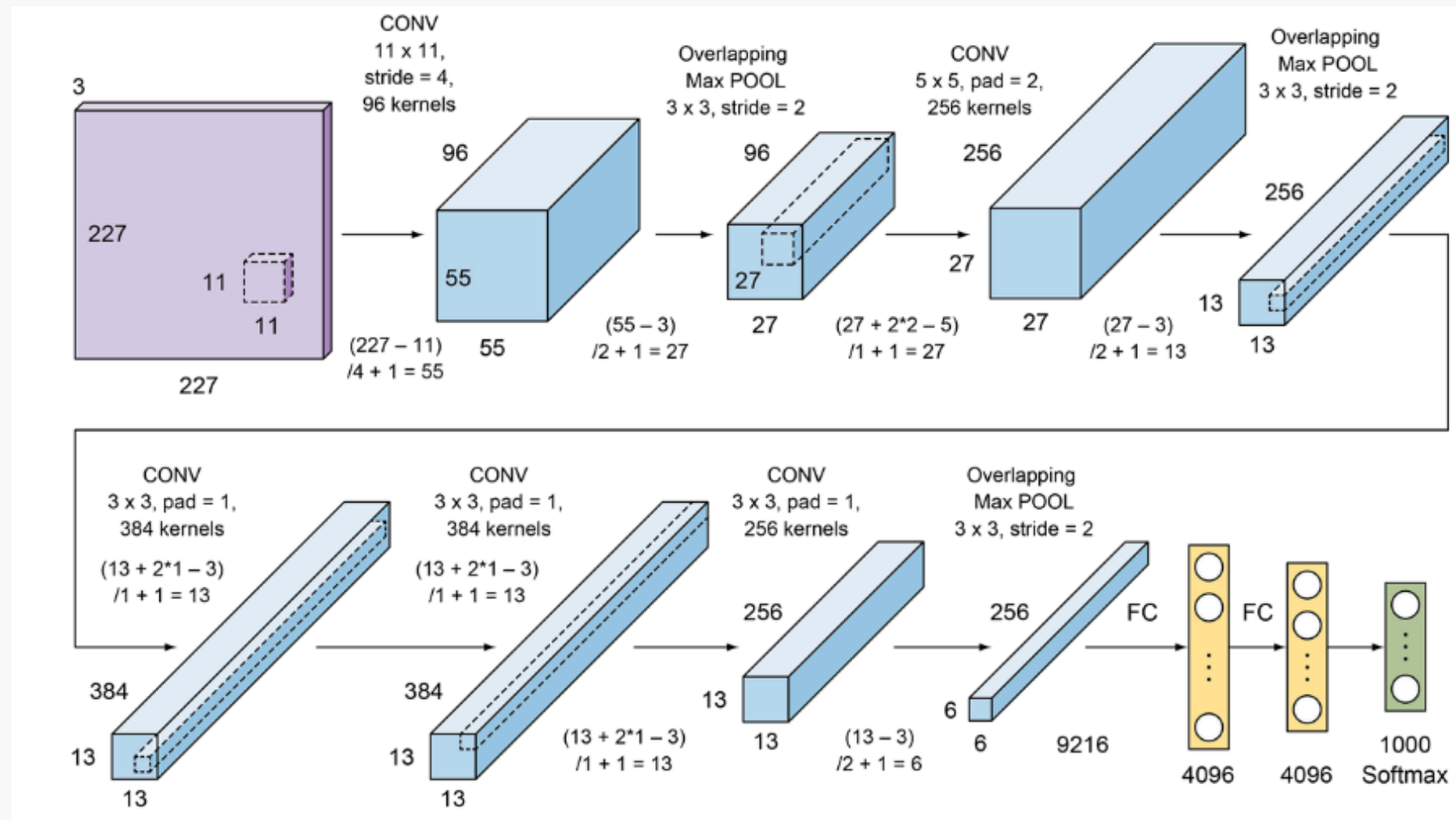
데이터 통합

데이터 불균형 문제로
valid 데이터와 test 데이터를 합쳐 테스트

AlexNet

전이학습이란?

기존에 대규모 데이터셋에서 학습된 모델을 **우리 문제에 맞게 재사용** 하는 것 → 적은 데이터로 높은 성능 달성 가능



> AlexNet

- 224 224 RGB 이미지에 최적화
- ImageNet으로 사전 학습된 pretrained=True 지원
- 상대적으로 작은 구조 → 빠르고 효율적
- 다양한 이미지에 적용 가능한 기본적인 시각 특징 추출기



사전 학습된 특징 추출 부분 중 **일부만 fine-tuning** 하고,
마지막 분류기는 **충만 데이터셋에 맞게 변경**

▶ 학습 방식

손실함수

- CrossEntropyLoss
- 클래스 불균형 보정을 위한 가중치 적용

가중치 계산방식

- 클래스별 이미지 수
→ Counter로 카운트
- 클래스 적을수록
weight 높게 부여

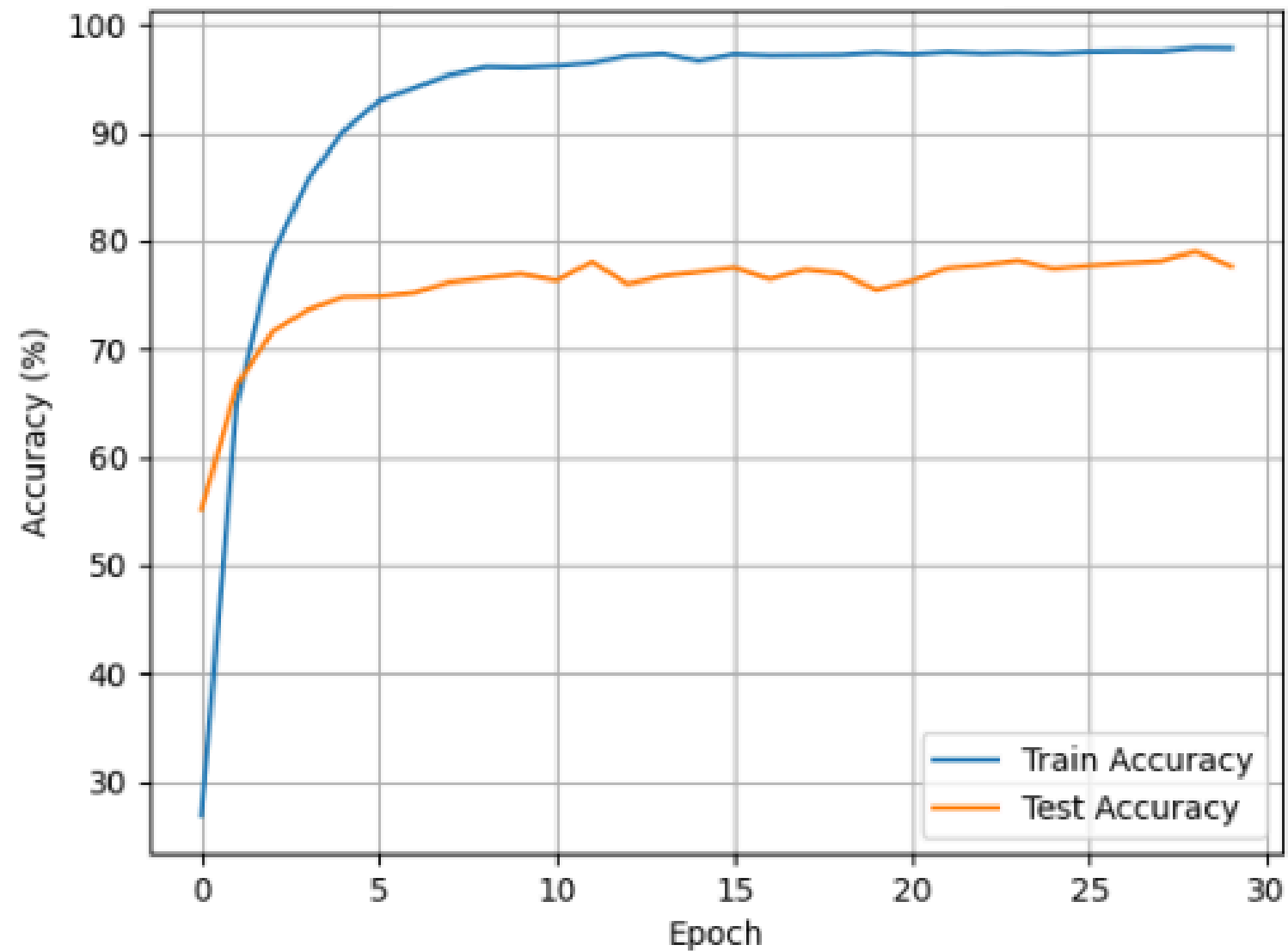
옵티마이저 및 하이퍼파라미터

- Optimizer: Adam
- Learning Rate: 0.0001
- Batch Size: 32
- Epoch 수: 30

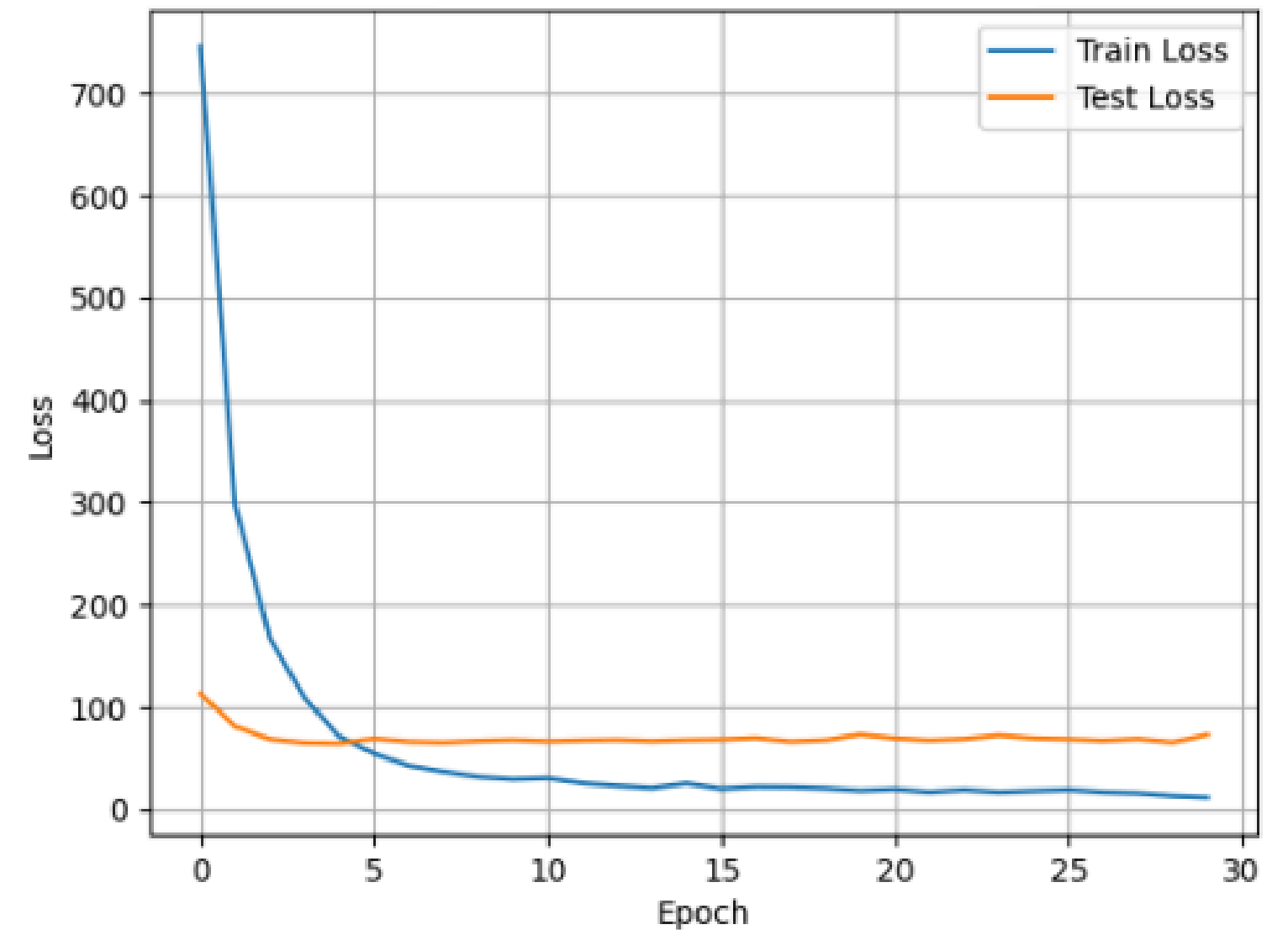
▶ 성능 평가

정확도 및 손실

정확도 그래프



손실 그래프

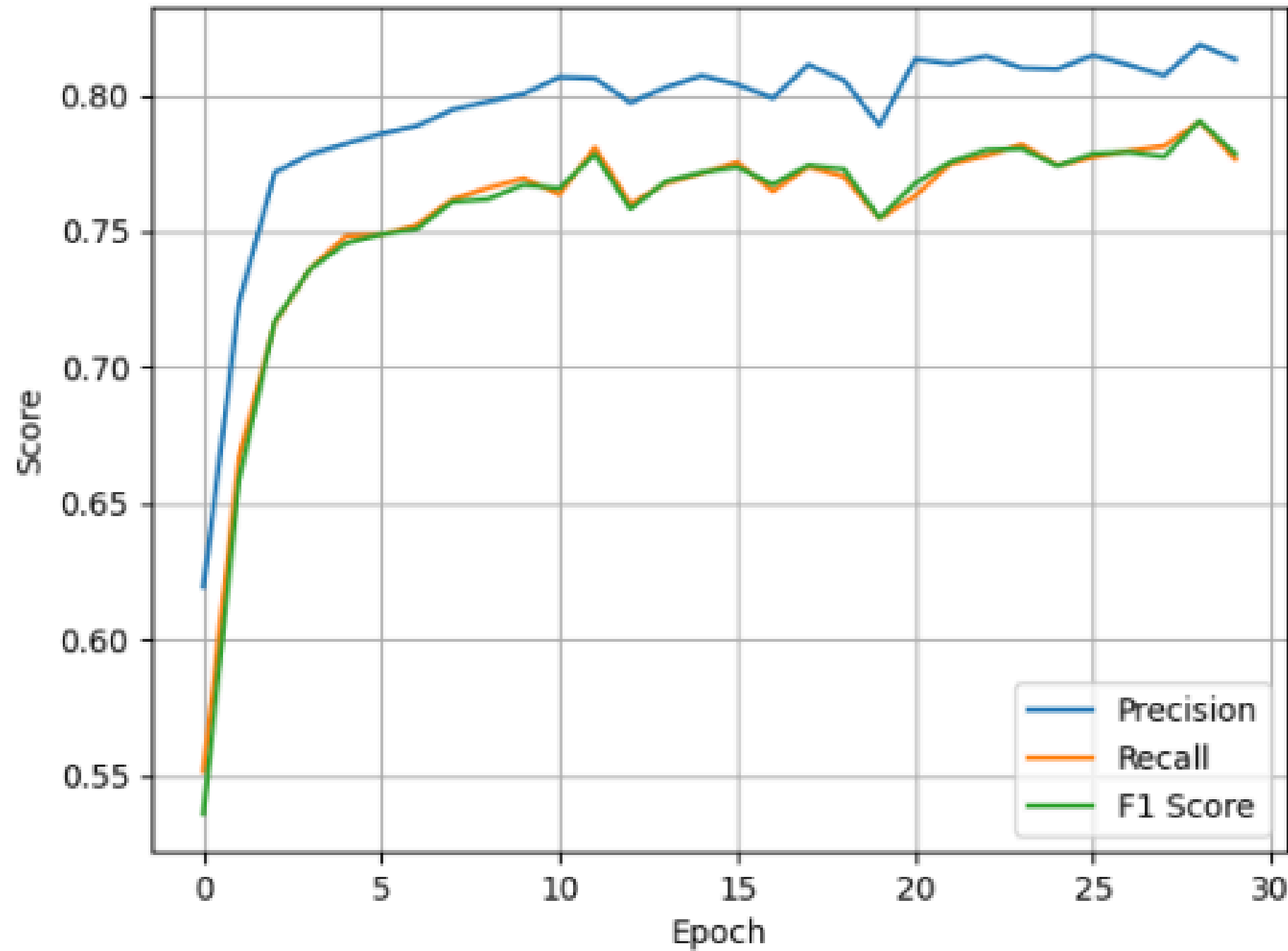


> 29 Epoch 기준 정확도: 79.03%
손실도: 65.79%

▶ 성능 평가

F1 / Precision / Recall

정밀도 / 재현율 / F1 그래프



> 29 Epoch 기준

- 정밀도: 81.87%
- 재현율: 79.03%
- F1 Score: 79.05%

▶ 결과



002_2
Cents_australia

💰 동전 이름: 2 Cents
🇦🇺 통화 단위: Australian dollar
🌐 국가: australia



002_1
Won_south_korea

💰 동전 이름: 1 Won
🇰🇷 통화 단위: Korean Won
🌐 국가: south_korea

▶ 결론

데이터 불균형 문제

일부 클래스는 이미지 수가 너무 적었고, 특히 테스트셋 데이터가 매우 부족하여
결국 테스트셋과 검증셋을 합쳐서 평가에 사용

Loss 값이 과도하게 높음

이미지불균형 + 적은 데이터로 인해 Loss 값이 높게 나오는 현상 발생 → 예측에 큰 영향

모델 성능의 중요성 체감

모델 구조, 전이학습 전략, 데이터 구성이 얼마나 중요한지를 직접 체감



2. VGG16

한아름



▶ VGG16 모델

- VGG16은 이미지 분류에 많이 사용되는 CNN 모델 중 하나
- 적은 양의 데이터로도 높은 성능을 낼 수 있어 효율적
- 앞부분은 이미지에서 특징을 추출하는 역할을 하고,
뒷부분은 그 특징을 바탕으로 어떤 클래스인지 분류하는 역할
- VGG16은 정확도는 높지만 구조가 복잡하고 계산량이 많아,
학습에 시간이 오래 걸림

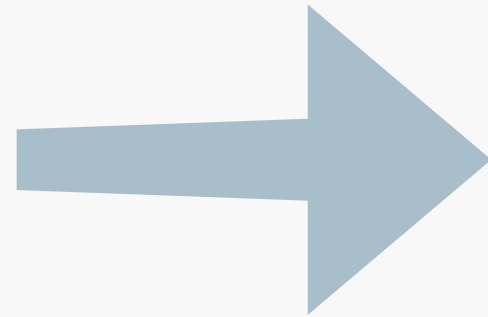
→ 모델의 구조적 특성을 고려해 일부 데이터만 선별



▶ 데이터 전처리



- 211개 클래스



- 68개 클래스

학습 시간이 오래 걸리기 때문에
전체 국가의 화폐 중에서 유럽 국가 화폐만 선택해서 학습을 진행

▶ 데이터 전처리



Resize

- (224x224)
: 모델 입력 크기에 맞게
이미지 크기 통일



ToTensor

- 이미지를 텐서로 변환
(0~1 범위)



Normalize

- 각 채널을 평균 0.5,
표준편차 0.5로 정규화
→ 학습 안정성 향상




RandomHorizontalFlip (train만)

- 좌우 반전을 통해 데이
터 다양성 확보

▶ 모델 학습

✓ [Epoch 10] Test Accuracy: 69.85% | Test Loss: 1.2759

✓ [Epoch 30] Test Accuracy: 75.00% | Test Loss: 1.3213



✓ [Epoch 50] Test Accuracy: 72.79% | Test Loss: 1.6006

75%의 정확도가 엄청 잘 나온 것은 아니지만
적은 데이터로도 전이학습의 효과를 분명히 보여준 결과

▶ 결론

- 데이터가 많지 않은 상황에서 전이학습을 사용해도 성능 향상과 시간 절약을 경험
 - 전이학습을 활용하면 충분히 실용적인 수준의 성능을 낼 수 있다는 가능성을 확인
-

▶ 예측

VGG16 이미지 분류기

이미지를 업로드하면 클래스 예측 결과를 보여줍니다.

이미지를 업로드하세요



Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files



010__10 Centavos_brazil.jpg 39.3KB



The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the use_container_width parameter instead.



업로드한 이미지

예측 클래스: 11

예측 클래스: 11

"11": "10 Centavos,Brazilian Real,brazil",



3. EfficientNet

신승준



▶ 선정 배경

EfficientNet:

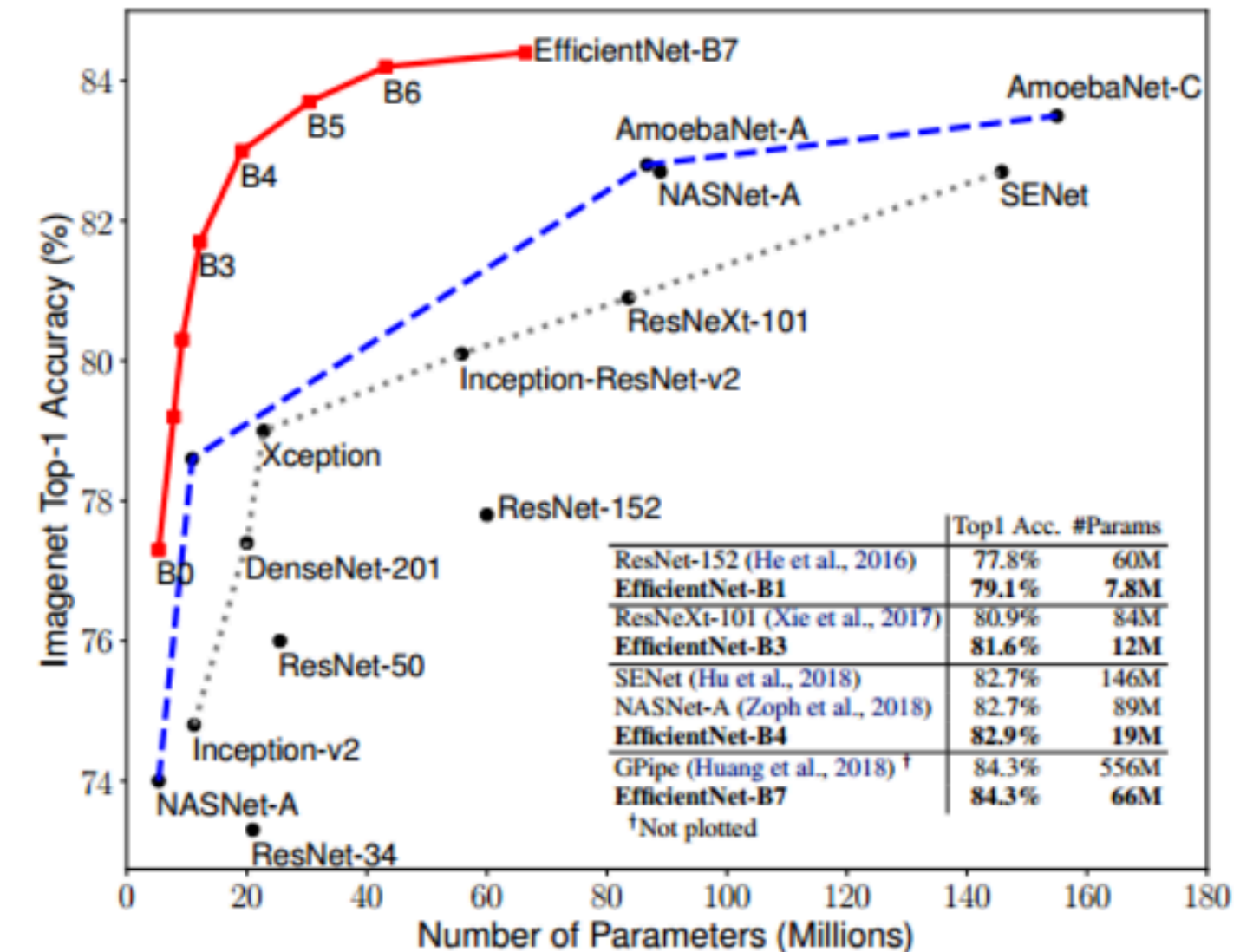
- Google AI에서 개발한 ConvNet 기반 딥러닝 모델
- 기존 모델들은 2가지 요소에 집중하여 무작정 깊고 넓게 만들었던 반면, EfficientNet 모델은 깊이, 너비, 해상도를 균형 있게, 최적의 비율로 컨트롤 하는 복합 스케일링 전략을 사용

장점:

- 비슷한 크기의 타 모델들에 비해 파라미터나 계산량이 훨씬 적음에도 정확도가 더 높아 연산 효율이 더 좋음.
- 핸드폰과 같은 저사양 기기에서도 딥러닝 훈련이 가능 할 정도로 가벼움
- B0-B7까지 다양한 모델이 있어 데이터셋에 따라 적절하게 모델 사이즈를 선택 하여 사용가능
- 현재보다 성능이 우수한 최신모델 EfficientNetV2 버전도 존재

한줄 요약:

- 성능은 좋으면서도, 자원은 적게 사용하는 효율적인 모델



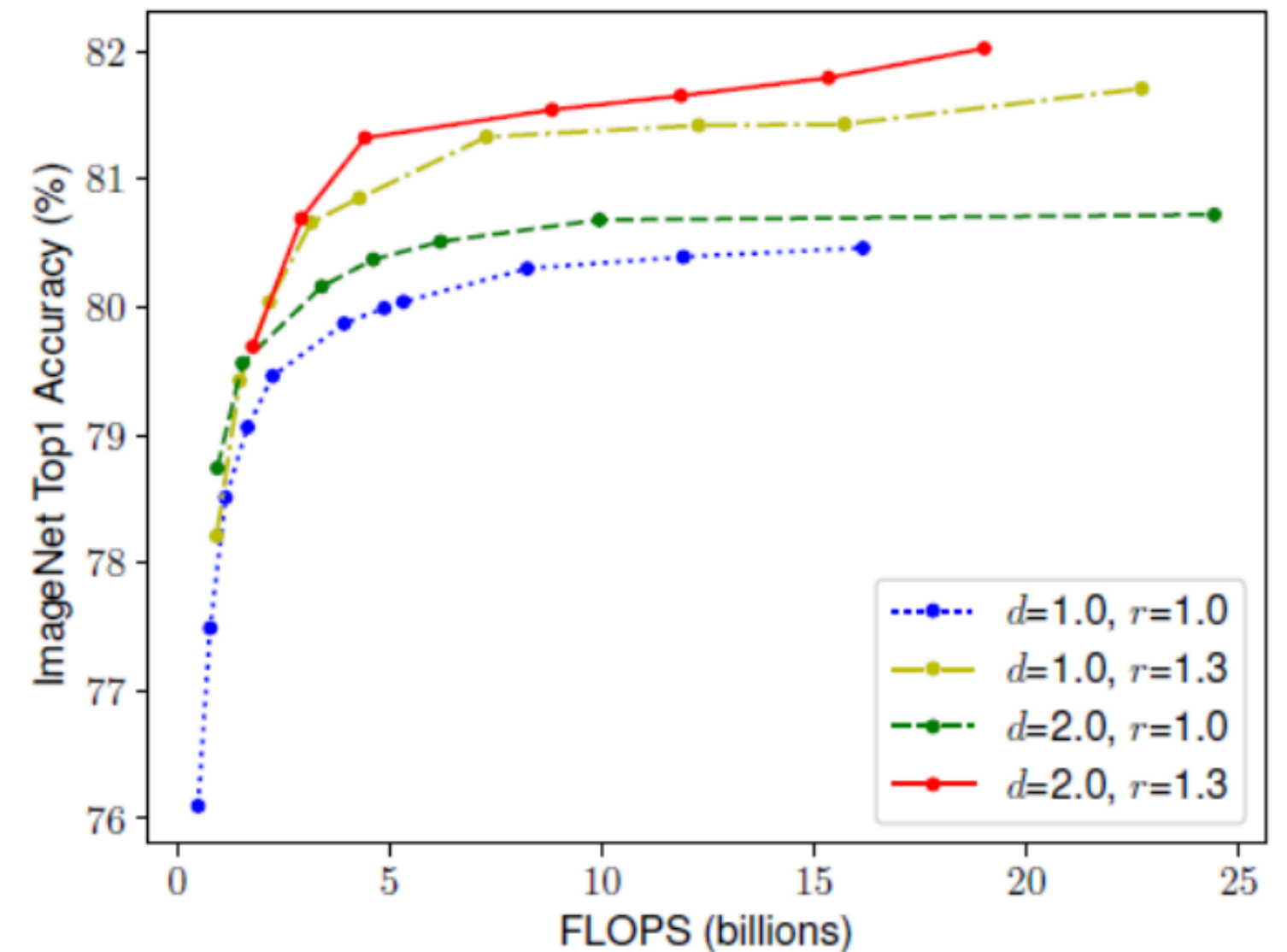
▶ 그러나!

EfficientNet:

- 깊이, 너비, 해상도를 다 늘린다고 반드시 좋아지는 것은 아님. 단지 개발자가 3가지 요소를 균형 있게 증가 시키는 방법을 찾아 낸 것.
- 무작정 키우기만 한다면 OverFitting이 발생함.

단점:

- 구조가 매우 복잡하여 커스터 마이징이나 디버깅이 어려움
- 하이퍼 파라미터 튜닝에 매우 민감하여 잘못 설정시 성능이 급 하락함.
- 입력 이미지가 작을 시 성능이 급격히 떨어지기에 이미지 사이즈를 224 * 224로 맞춰하기에 유연성이 다소 부족함.



▶ 훈련 환경

CNN 모델:

- CNN 모델 사용 하여 훈련 진행.
- 데이터 전처리 - Train, Valid, Test로 분류 후 모델 학습

CNN 모델: 사용 불가능 할 정도의 성능을 자랑함.

```
EPOCH [1/10] -----  
- 학습 손실: 5.19881, 학습 정확도: 0.01925  
- 검증 손실: 4.92651, 검증 정확도: 0.03021  
에폭 진행: 10%|  
새로운 최고 성능 모델 저장: 0.025
```

```
EPOCH [2/10] -----  
- 학습 손실: 4.95901, 학습 정확도: 0.02510  
- 검증 손실: 4.79627, 검증 정확도: 0.04776  
에폭 진행: 20%|  
새로운 최고 성능 모델 저장: 0.037
```

```
EPOCH [8/10] -----  
- 학습 손실: 4.46767, 학습 정확도: 0.07066  
- 검증 손실: 4.48519, 검증 정확도: 0.08869  
에폭 진행: 80%|  
새로운 최고 성능 모델 저장: 0.075
```

```
EPOCH [9/10] -----  
- 학습 손실: 4.40677, 학습 정확도: 0.07529  
- 검증 손실: 4.35560, 검증 정확도: 0.09552  
에폭 진행: 90%|
```

```
EPOCH [10/10] -----  
- 학습 손실: 4.36434, 학습 정확도: 0.07456  
- 검증 손실: 4.28018, 검증 정확도: 0.11598  
에폭 진행: 100%|
```

```
=====
```


▶ 훈련 환경

CNN 모델:

- CNN 모델 사용 하여 훈련 진행.
- 데이터 전처리 - Train, Test로 분류 후 모델 학습

데이터 통합 후: 성능 확인 후 미련없이 학습 종료 시켜버림

```
EPOCH [1/20] -----  
- 학습 손실: 5.15976, 학습 정확도: 0.02144  
- 검증 손실: 4.95314, 검증 정확도: 0.03040  
에폭 진행: 5%|  
새로운 최고 성능 모델 저장: 0.041
```

```
EPOCH [2/20] -----  
- 학습 손실: 4.83323, 학습 정확도: 0.04055  
- 검증 손실: 4.73099, 검증 정확도: 0.04832  
에폭 진행: 10%|  
새로운 최고 성능 모델 저장: 0.046
```

▶ 훈련 환경

EfficientNet 모델:

- EfficientNet 모델 사용 하여 훈련 진행.
- 데이터 전처리 - Train, Valid, Test로 분류 후 모델 학습

EfficientNet 모델: 과대 적합 발생하나, cnn모델에 비해 압도적인 학습력 보여줌.

```
EPOCH [1/20] -----
- TRAIN_LOSS: 3.77675, TRAIN_ACC: 0.18234
- VALID_LOSS: 1.79924, VALID_ACC: 0.43978
Epochs: 5%|
EPOCH [2/20] -----
- TRAIN_LOSS: 1.32699, TRAIN_ACC: 0.54905
- VALID_LOSS: 0.98306, VALID_ACC: 0.63092
Epochs: 10%|
EPOCH [3/20] -----
- TRAIN_LOSS: 0.65231, TRAIN_ACC: 0.73271
- VALID_LOSS: 0.88644, VALID_ACC: 0.65393
Epochs: 15%|
```

```
EPOCH [15/20] -----
- TRAIN_LOSS: 0.02302, TRAIN_ACC: 0.98415
- VALID_LOSS: 0.80472, VALID_ACC: 0.76820
Epochs: 75%|
EPOCH [16/20] -----
- TRAIN_LOSS: 0.02412, TRAIN_ACC: 0.97826
- VALID_LOSS: 0.80458, VALID_ACC: 0.77101
Epochs: 80%|
EPOCH [17/20] -----
- TRAIN_LOSS: 0.02266, TRAIN_ACC: 0.97874
- VALID_LOSS: 0.81186, VALID_ACC: 0.77147
성능 개선이 없어 17에 조기 종료합니다.
Epochs: 80%|
```

▶ 훈련 환경

EfficientNet 모델:

- EfficientNet 모델 사용 하여 훈련 진행.
- 데이터 전처리 - Train, Test로 분류 후 모델 학습

데이터 통합 후: 처음부터 불균형이 심한 관계로 전처리 진행해도 크게 차이 없음.

```
EPOCH [1/10] -----
- TRAIN_LOSS: 3.38570, TRAIN_ACC: 0.23026
- VALID_LOSS: 1.44232, VALID_ACC: 0.50140
Epochs: 10%|
EPOCH [2/10] -----
- TRAIN_LOSS: 1.10765, TRAIN_ACC: 0.59255
- VALID_LOSS: 0.96595, VALID_ACC: 0.63805
Epochs: 20%|
EPOCH [3/10] -----
- TRAIN_LOSS: 0.56090, TRAIN_ACC: 0.74737
- VALID_LOSS: 0.94026, VALID_ACC: 0.66962
Epochs: 30%|
EPOCH [4/10] -----
- TRAIN_LOSS: 0.36046, TRAIN_ACC: 0.83050
- VALID_LOSS: 0.80712, VALID_ACC: 0.70597
Epochs: 40%|
EPOCH [5/10] -----
- TRAIN_LOSS: 0.25453, TRAIN_ACC: 0.87284
```

```
EPOCH [8/10] -----
- TRAIN_LOSS: 0.15638, TRAIN_ACC: 0.92337
- VALID_LOSS: 0.85772, VALID_ACC: 0.73667
Epochs: 80%|
EPOCH [9/10] -----
- TRAIN_LOSS: 0.08833, TRAIN_ACC: 0.95162
- VALID_LOSS: 0.74569, VALID_ACC: 0.77262
Epochs: 90%|
EPOCH [10/10] -----
- TRAIN_LOSS: 0.05356, TRAIN_ACC: 0.96529
- VALID_LOSS: 0.75533, VALID_ACC: 0.77200
Epochs: 100%|
```

▶ 마치며..

장비나 사람이나 비싸면 값을 한다.

- 노트북에서는 1시간 걸리던 훈련이 컴퓨터에서는 5분만에 끝남
- 구글의 뛰어난 인재들이 개발한 EfficientNet 모델의 성능 또한 과대적합 발생 하기는 했으나, 211진 분류임에도 학습 성능만큼은 좋았기에, 데이터셋의 불균형으로 인한 문제라 보여 모델은 손색이 없다고 생각함.

EfficientNetV2 조금만 빨리 알았더라면..

- 기존의 EfficientNet모델도 성능이 매우 좋다고 느꼈는데, 그보다 더 업그레이드 된 EfficientNetV2 모델을 사용해보지 못해서 아쉬움, 다음 프로젝트 때 사용 할 것.
- 이외에도 모델 사이즈 (B0-B7)별로 사용해 보지 못한 것이 아쉬움. 이 또한 기회가 된다면 다음 프로젝트때 사용 해보려 함.

결론:

- 전이학습.. 좋다...!!





4-1.ResNet

ver. 전처리 Before

최수창



▶ 모델 설명

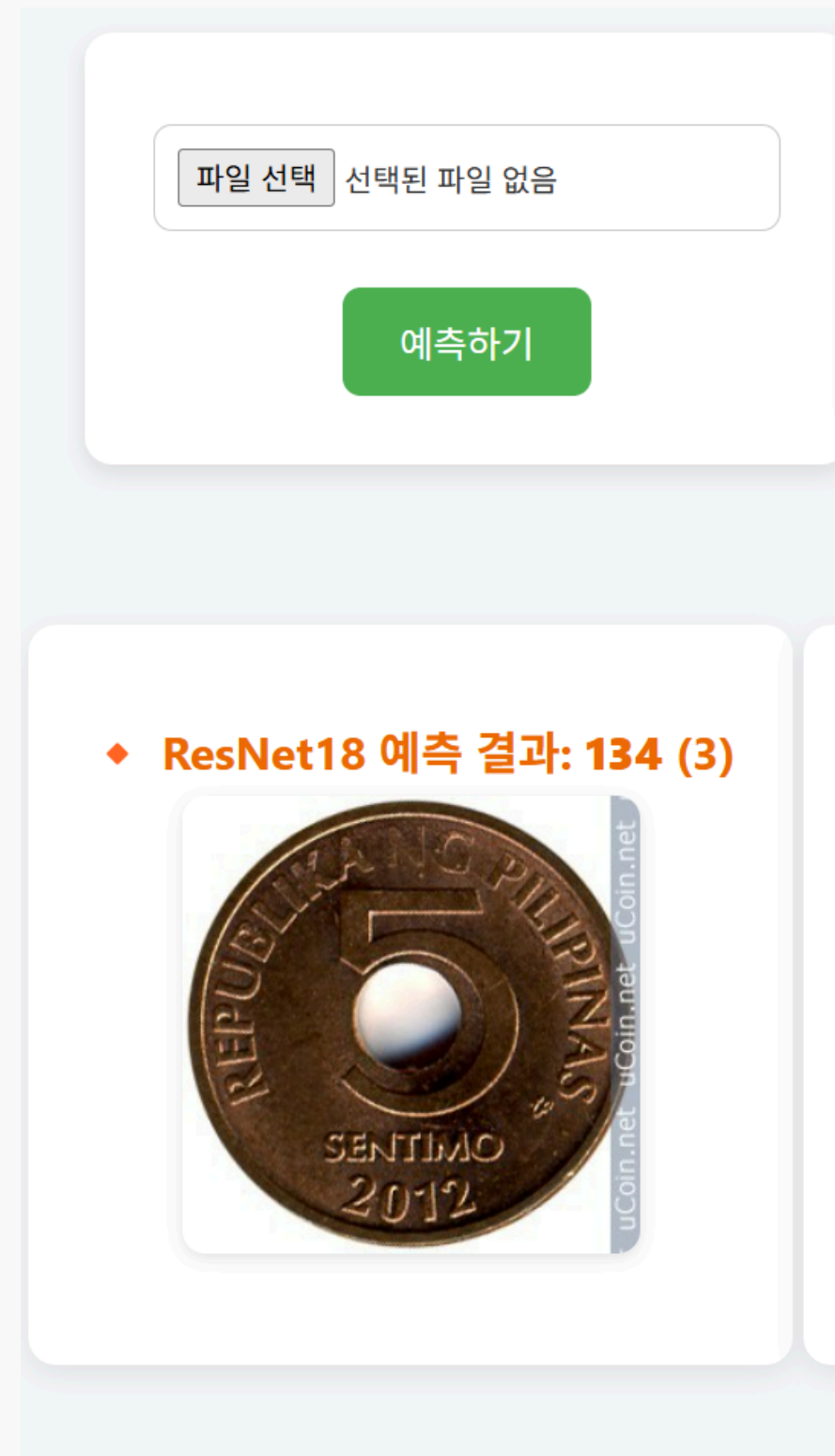
- ResNet (Residual Network)이란?
 - → 잔차 연결(Residual Connection)을 사용하는 딥러닝 모델
 - → 깊은 네트워크에서 발생하는 기울기 소실 문제를 해결
 - 잔차 연결이란?
 - → 이전 층의 입력을 다음 층 출력에 직접 더해주는(skip connection)
- 구조
- ResNet-18 특징
 - 총 18개 레이어 (8개의 BasicBlock으로 구성)



▶ 전처리 X

바로 이미지를 불러와서 모델을 돌렸을때

- 총 train이미지: 6413
- 총 valid이미지: 844
- 총 test이미지: 844
- ResNet 요구 사이즈 244*244



```
"134": "5 Sentimos, Philippine peso, philippines",
```





4-2.ResNet

ver. 전처리 After

최수창



▶ 전처리 과정



- 처음 이미지: 6413
- 같은 이미지 중복
- 제거 후 이미지: 5954



▶ 전처리 과정

- 처음 이미지: 6413
- 같은 이미지 중복
- 제거 후 이미지: 5954
- valid랑 test 개수가 train에 비해 적어서 병합 → 1688

```
valid 갯수: 844  
test 갯수: 844
```



▶ 전처리 0

중복된 이미지 제거후 돌렸을때,

- Test Accuracy = 95%
- Train Accuracy = 88%


🖼️ 이미지로 동전을 분류해보세요!

파일 선택 선택된 파일 없음

예측하기


◆ AlexNet 예측

🏷️ 동전: 1 Cent
🌐 통화: Australian dollar
🇦🇺 국가: australia




◆ ResNet18 예측

🏷️ 동전: 1 Cent
🌐 통화: Australian dollar
🇦🇺 국가: australia



● EfficientNet 예측

🏷️ 동전: 50 Korun
🌐 통화: Czech Koruna
🇨🇪 국가: czech_republic



결과값 잘나옴



▶ 결론

데이터를 다운 받은 후 이미지 파일을 확인하자

- Train, Valid, Test 간 데이터 양의 불균형은 모델 성능에 영향을 미치므로 사전 확인이 필수
- 데이터 폴더 내 비정상 이미지가 포함될 수 있어 모든 이미지를 열어 확인하는 과정이 필요

전이학습모델 선택의 중요성

- 클래스 수가 211인 만큼 ResNet에도 여러 종류가 있는데 가장 작은 ResNet18을 사용하여서 다음엔 ResNet152등을 사용하고 싶음

모델 학습 전 중요성

- 전이학습 모델을 사용하더라도, 내가 원하는 결과를 얻기 위해서 계획적인 설계의 필요성을 느낌



최종 결론

동전의 실질적 수요와 관리 문제에 대응하는 데 있어,
전이학습 기반 분류 모델은
정확도와 효율성을 갖춘 실용적인 대안임을 입증



감사합니다

