Sketch Image Data Classification Wrap-up Report

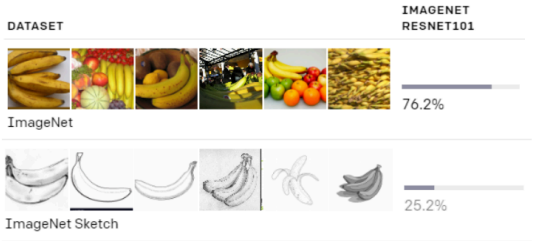
Efficient Learning Using Data Augmentation

Hyeonwoo Jung [1] Jin-Suk Kim [1] Min-Seok Choi [1]
Jae Hoon Choi [1] Dong Yeong Kim [1] Jung Woo Yoon [1]

1. 프로젝트 개요

1.1. 개요

스케치 데이터는 사진 이미지와 달리, 객체의 기본적인 형태나 윤곽선만을 강조하며 인간의 상상력과 개념적 이해를 반영하는 추상적 표현을 담고 있다. 색상, 질감, 세부적인 형태가 결여된 데이터는 사진 이미지와 비교하면 분류 성능이 매우 낮다는 것을 아래 그림을 통해 확인할 수 있다[1].



이번 대회의 목표는 스케치 데이터의 특성을 심층적으로 분석하고, 이를 바탕으로 우수한 분류 성능을 가진 모델을 제작하는 것이다. 이를 달성하기 위해 다양한 데이터 증강 기법을 적용하여 모델의 일반화 성능을 향상한 뒤, 스케치 이미지의 특성을 효과적으로 포착할 수 있는 특화된 여러 모델 아키텍처를 도입하고 적합한 학습률, 에폭, 옵티마이저, 손실 함수 등 하이퍼파라미터를 여러 실험을 통해 최적화하는 학습 전략을 모색한다.

1.2. 개발 환경

1. (팀 구성 및 컴퓨팅 환경) 6인 1팀, 4개의 V100 서버를 VSCode와 SSH로 연결하여 사용
2. (협업 환경) Notion, Github, Coggle
3. (의사소통) ZOOM, Slack, Discord

2. 프로젝트 팀 구성 및 역할

전체	모델별 성능 table 작성, 모델 성능 실험, Wrap up report 작성
정현우	팀의 목표 설정 및 실험 설계, 모델 최적화, 디퓨전 모델을 활용한 데이터 증강
김진석	Git 사용 관리 및 도움, Scheduler, Advanced Data Augmentation
최민석	Notion template 구축, Basic Augmentation 시각화
최재훈	Code Refactoring, EDA (Streamlit을 활용한 시각화), Data Augmentation (Customizing, AugMix)
김동영	앙상블, Optuna 시도, 하이퍼파라미터 조정 전략, Advanced Data Augmentation (Random Aug)
윤정우	학습률 조정, Basic Data Augmentation, L2 정규화 적용

3. 프로젝트 수행 절차 및 방법

A. 팀 목표 설정

(전체) Public Accuracy 0.9 달성,
Git 활용 협업 경험 축적 및 체화

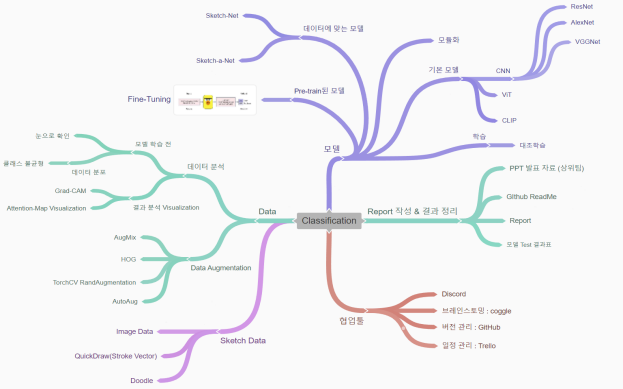
(1주차) AI 기초 및 CV 기초 프로젝트 강의 수강,
베이스라인 코드 이해, 코드 리팩토링,
모델 리서치 및 모델 선정

(2주차) Train / test data 확인, EDA,
Learning Rate 조정, L2 정규화 실험

(3주차) Data Augmentation 비교 실험,
Diffusion Augmentation 적용 실험,
Wrap up report 및 회고 작성

B. 프로젝트 사전기획

(1) Mind Map 정리



(2) 협업 문화

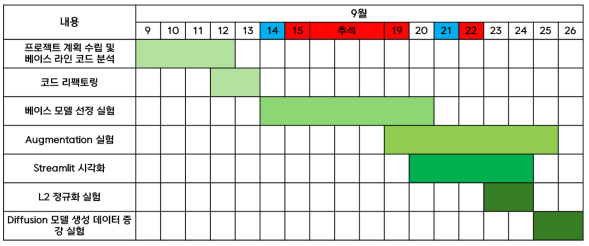
[1] Core 시간에 지속적인 소통으로 무엇을 해야 하는지 토론하고 각자에게 역할 분배

[2] Notion을 통해 To-Do list를 시작 전, 진행 중, 완료로 구분해 관리

[3] 서버 사용자 현황을 작성해 서버 현황 관리

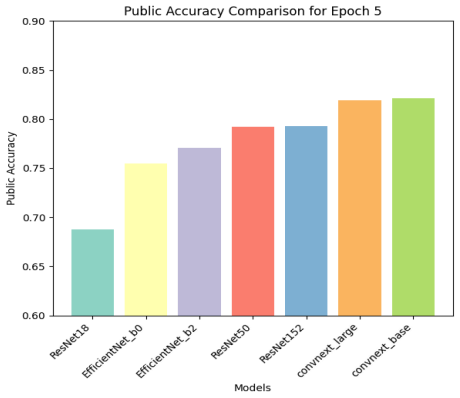
[4] 모델 훈련 시 사용한 모델, Data Augmentation, Configuration 들을 Notion 기록 테이블에 기록

[5] 새 기능 추가 후 검증이 완료되면 Github Pull Request를 통해 팀원들의 Review를 받고 Main에 Merge하는 과정 수행



4. 프로젝트 수행 결과

4.1. 모델 선정

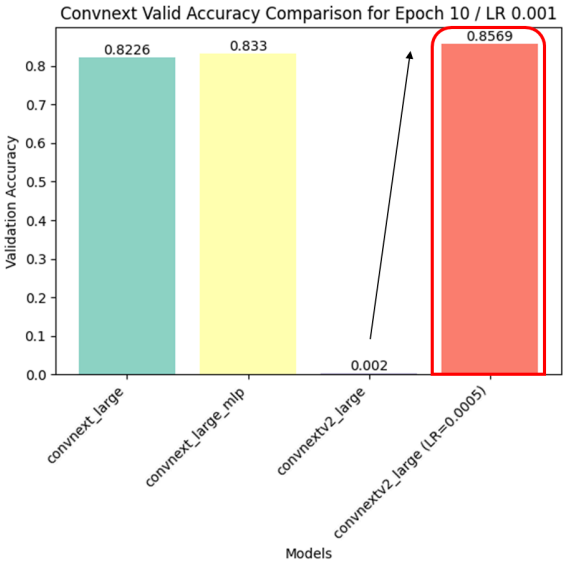


Sketch 이미지 데이터셋의 특성을 고려하여, 엣지와 국소적 정보를 효과적으로 추출할 수 있는 모델들을 후보로 선정하고 성능을 평가했다. 이를 위해, **Timm** 라이브러리에서 제공하는 다양한 모델들을 Learning Rate 0.001, Epoch 5로 설정하여 비교 실험을 수행했다 .

실험 결과, ConvNeXt 계열 모델이 다른 모델에 비해 가장 낮은 Train Loss 값을 기록하였으며, Validation Accuracy와 Public Accuracy에서도 가장 우수한 성능을 나타냈다. ConvNeXt 모델의 구조적 특징을 분석한 결과, 해당 모델이 기존 CNN 구조를 기반으로 저차원에서 고차원까지의 특징을 순차적으로 학습한다. 이러한 구조적 특성으로 인해 ConvNeXt 모델이 Sketch 이미지 분류에서 높은 성능을 발휘한 것으로 판단된다[2].

따라서 ConvNeXt 모델은 Sketch 이미지 분류 작업에 적합한 모델로 결론지을 수 있다.

4.2. 하이퍼파라미터 조정



초기 실험에서 timm 라이브러리가 제공하는 convnext_large의 다음 버전인 convnextv2_large 모델을 사용하였으나, 학습 과정에서 수렴하지 않고 정확도가 2% 아래에서 머무르는 문제가 발생했다.

이는 학습률 (Learning Rate)의 설정과 모델 최적화 방식에서 기인한 것으로 판단되었다. 이를 해결하기 위해 학습률을 조정하고 다양한 스케줄러와 최적화 방법을 적용하였다.

우선, 학습률(LR)을 단계적으로 조정하면서 모델이 안정적으로 수렴할 수 있는 최적의 값을 찾기 위한 실험을 진행하였다. 기본 학습률을 줄이고, StepLR 스케줄러를 적용하여 학습이 안정적으로 진행되도록 했다.

L2 Penalty(가중치 제한)를 적용하여 모델의 일반화 성능을 더욱 향상시키고자 했다. 본 실험에서는 L2 Penalty를 0.000005로 설정했을 때 모델의 일반화 능력이 개선되는 것을 확인했다.

이를 통해 모델이 보다 효과적으로 학습될 수 있었으며, 초기의 수렴 문제를 해결하고, 보다 높은 정확도를 달성할 수 있는 토대를 마련했다.

4.3 모델 최적화

모델 성능 최적화를 위해 여러 기법을 적용하여 학습 효율성과 결과 향상에 기여했다. 다음은 적용한 주요 기법들이다.

4.3.1. 그래디언트 축적

메모리 한계를 극복하기 위해 그래디언트 축적 기법을 도입했다. 이 방법은 작은 배치 사이즈로 모델을 학습한 후, 일정 에폭마다 그래디언트를 누적하여 큰 배치 사이즈의 효과를 달성하는 방식이다[3].

4.3.2 데이터 로딩 최적화

데이터 로딩 성능을 높이기 위해 num_worker를 8로 설정했다. 이는 배치 사이즈가 64인 상황에서 CPU 코어 수를 고려한 설정으로, 데이터 로딩 속도를 극대화하여 GPU가 대기하는 시간을 최소화했다. 또한 pin_memory를 True로 설정하여 GPU로의 데이터 전송 속도를 향상시켰다.

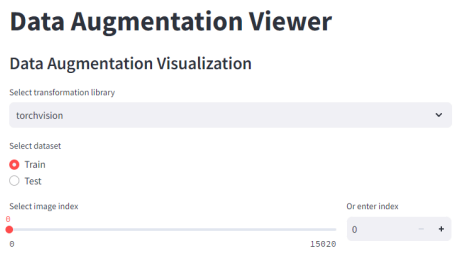
4.3.3 Mixed Precision Training

혼합 정밀도 훈련(Mixed Precision Training) 기법을 도입하여 float32와 float16을 혼합하여 사용했다. 이 방식은 메모리 사용량을 줄이는 동시에 연산 속도를 향상시키는 효과가 있다[4]. 이러한 기법들의 조합을 통해 모델의 성능과 학습 속도를 크게 개선하였으며, 최종적으로 더 높은 수준의 이미지 분류 성능을 달성할 수 있다.

이러한 결과적으로 학습 속도를 높여 더 다양한 실험을 용이하게 하였다.

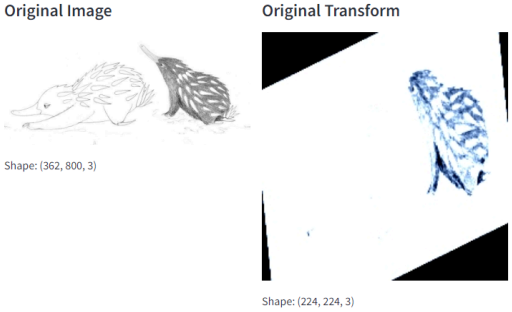
4.4. Streamlit 시각화

모델 학습 과정에서 중요한 부분은 데이터가 어떻게 증강되어 모델에 입력되고, 그 결과로 어떤 학습 성능을 발휘하는지 이해하는 것이다. 이를 효율적으로 파악하기 위해 **Streamlit**을 활용하여 시각화를 진행했다. 기존의 Python 파일이나 Jupyter Notebook으로 일일이 확인하는 것보다 Streamlit을 사용해 간단하고 직관적으로 시각화할 수 있는 웹 인터페이스를 제작했다.



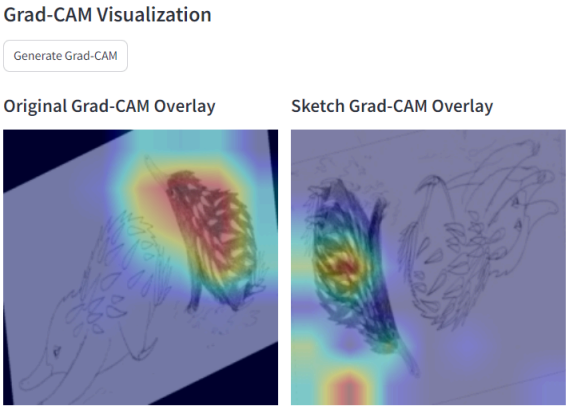
4.4.1. Augmentation Visualization

Augmentation 시각화 기능은 증강된 이미지를 쉽게 방식을 바꿔가며 확인할 수 있는 기능이다. 이를 통해 다양한 증강 기법이 스케치 데이터에 어떻게 적용되는지 시각적으로 확인할 수 있으며, 모델 학습에 적합한 증강 방식을 선택하는 데 도움이 된다.



4.4.2 Grad-CAM Visualization

Grad-CAM (Gradient-weighted Class Activation Mapping)은 CNN 기반 모델에서 특정 클래스에 대해 시각적 설명을 제공하는 기법이다. 이 기능을 추가하여, 선택한 이미지 분류 모델이 증강된 이미지에서 어느 부분에 주목하고 있는지를 시각화했다. 이를 통해 모델이 이미지를 어떻게 해석하고 있는지 직관적으로 파악할 수 있다 [5].



4.4.3 Misclassified Image Visualization

마지막으로, 모델이 어떤 이미지에서 분류에 실패했는지 직접 확인하고자, 라벨이 있는 train 데이터 기준으로 오분류된 이미지를 시각화하는 기능을 추가했다. 각 이미지 아래에는 모델이 예측한 라벨과 실제 정답 라벨을 나란히 보여주어, 모델이 어떤 라벨로 잘못 예측했는지 쉽게 확인할 수 있다.

4.5. Data Augmentation

모델 학습을 진행하는 동안, **Train Accuracy**는 지속해서 향상되었으나, **Validation Accuracy**는 일정하게 유지되거나 오히려 감소하는 경향을 보였다. 이러한 현상의 원인으로 훈련 데이터의 다양성이 부족하여 모델의 일반화 성능이 저하되었고, 이를 해결하기 위해 다음과 같은 **데이터 증강(Data Augmentation)** 기법을 도입했다.

우선, **AugMix**를 통해 이미지의 다양한 변형을 생성하여 모델이 여러 상황에 잘 적응할 수 있도록 했다. 그리고 **Diffusion Model**을 활용한 데이터 증강 기법도 도입하여, 훈련 데이터의 다양성을 한층 더 향상하는데 기여했다. 이러한 증강 기법들을 통해 모델의 일반화 성능을 개선하고, 최종적으로 Validation Accuracy의 향상을 도모했다.

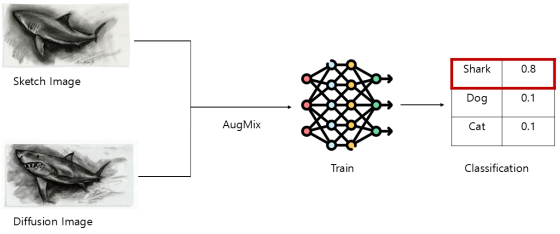
4.5.1. AugMix

기본적인 데이터 증강은 성능 향상이 미미하여, 모델의 일반화 성능을 높이기 위해 자동화된 데이터 증강 기법인 AugMix를 도입했다. AugMix는 데이터의 다양성을 극대화하여 모델이 다양한 상황에 잘 적응하도록 돕고, 원본 라벨을 유지하면서 학습 데이터의 품질을 보존한다. 이 기법은 여러 증강 방법을 조합해 모델의 강건성을 향상하고, 다양한 입력에 대한 대응 능력과 노이즈 저항력을 증가시킨다[6]. 이를 통해 모델의 Accuracy를 89.3%에서 90.0%로 향상할수 있었다.

4.5.2. Diffusion Data Augmentation

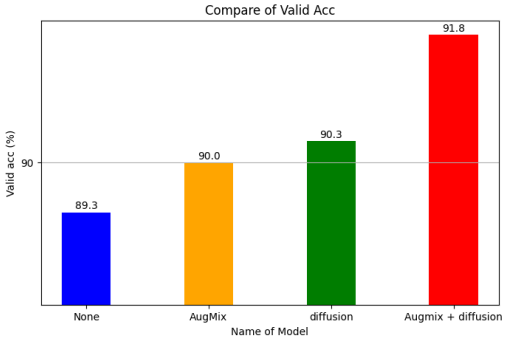
스케치넷 데이터셋의 크기가 모델 학습에 충분하지 않다고 판단되어, 데이터 증강을 위해 딥러닝 기반의 디퓨전 모델을 활용했다. 구체적으로, image-to-image 변환 방식을 사용하여 스케치 데이터를 증강 했고, 이를 통해 스케치 이미지의 다양성과 품질을 향상했다 [7].

각 이미지에 "black and white sketch of " + mapping[class_name] + "A refined sketch with subtle line enhancements..."라는 프롬프트를 사용해, 스케치의 원래 구성을 유지하면서 선의 명확성과 정의를 개선했다.



이 방법을 통해 총 15,000장의 추가 스케치 이미지를 생성하여 데이터셋을 확장하고, 위와 같은 파이프라인으로 모델을 학습하여 성능을 91.8%로 향상시켰다.

4.6. 실험 결과



결과적으로, Augmix와 디퓨전 모델을 활용한

데이터 증강을 통해 모델의 성능을 89.3%에서 91.8%로 향상할 수 있었다. 기존 스케치넷 데이터셋의 한계를 보완하기 위해 총 15,000장의 추가 스케치 이미지를 생성하였으며, 이 증강 데이터는 모델의 일반화 능력을 개선하는 데 기여했다. 특히, image-to-image 변환 방식으로 원본 스케치의 구성은 유지하면서도 선의 명확성을 높이는 방향으로 증강을 진행함으로써, 모델의 학습에 필요한 다양한 특성을 더 효과적으로 반영할 수 있었다.

5. 자체 평가 의견

A. 잘했던 점

- 프로젝트 목표 및 계획을 마인드맵으로 가독성 있게 정리했다.
- 베이스라인 코드 구조를 유지보수가 쉽게 스크립트로 리팩토링했다.
- 강의에서 배웠던 Streamlit 웹 서비스를 활용하여 데이터 시각화를 했다.
- 기본적인 Data Augmentation부터 Advanced Data Augmentation까지 다양하게 사용했다.

B. 시도 했으나 잘 되지 않았던 것들

- Noise and Blur, Geometric / Morphological transformation 등 다양한 데이터 증강 기법을 활용했으나, 유의미한 성능 향상이 이뤄지지 않았다.
- ChainedScheduler 등 다양한 스케줄러를 활용했으나 적절한 조합 및 파라미터를 찾지 못했다.
- TensorBoard, WandB 등 모델 학습 과정 시각화 도구를 활용하고 싶었으나 적용하지 못했다.
- 앙상블 기법, 하이퍼 파라미터를 시간이 없어서 오래 시도 하지 못했다.

C. 아쉬웠던 점들

- ViT 기반 다양한 변형 모델에 대한 충분한 실험이 미흡했다.
- 모델 선정 및 하이퍼파라미터 튜닝 과정에서 AutoML 기술을 사용해 보지 못했다.
- 체계적인 실험 계획 수립 및 실험 테이블 작성 미흡으로 인해 비효율적으로 진행됐다.
- Task에 적합한 다양한 모델 탐색 및 선정 과정이 미흡했다.
- GitHub의 Issue, Pull Request의 Reviewer 기능 등 고급 협업 도구 활용을 못 했다.

D. 프로젝트를 통해 배운 점 또는 시사점

- **Git**을 포함한 다양한 툴을 활용한 협업 경험을 쌓았다.
- AI 라이브러리를 사용하여 머신러닝 프로젝트의 구조를 설계하고 **훈련 프로세스**를 구현하는 방법을 익혔다.
- Config 파일을 통해 실험 설정을 편리하게 변경하며 다양한 실험을 수행하는 방법을 배웠다.
- **Streamlit** 라이브러리 사용법을 학습했다.
- 자동화된 **AugMix** 기법을 통해 모델 성능을 향상할수 있음을 경험했다.
- **디퓨전 기반 데이터 증강**을 통해 데이터셋 크기를 확장하고 성능을 개선할 수 있음을 확인했다.

[Reference]

[1] Haohan Wang, et al., “Learning Robust Global Representations by Penalizing Local Predictive Power”, *NeurIPS*, 2019.
https://arxiv.org/pdf/1905.13549v2

[2] Zhuang Liu, et al.,"A ConvNet for the 2020s", *Computer Vision and Pattern Recognition*, 2022.
https://arxiv.org/pdf/2201.03545

[3] Hermans J, et al., "Accumulated Gradient Normalization", *Asian Conference on Machine Learning,* 2017.
https://arxiv.org/abs/1710.02368

[4] Micikevicius P, et al., "Mixed Precision Training", International Conference on Learning Representations, 2017.
https://arxiv.org/abs/2302.07944

[5] Selvaraju, et al., "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization", *Int J Comput Vis*, 2020.
https://link.springer.com/article/10.1007/s11263-019-01228-7

[6] Hendrycks, et al., "Augmix: A simple data processing method to improve robustness and uncertainty.", *International Conference on Learning Representations,* 2019.
https://doi.org/10.48550/arXiv.1912.02781

[7] Brandon Trabucco, et al., "Effective Data Augmentation With Diffusion Models", *International Conference on Learning Representations,* 2023.
https://arxiv.org/abs/2302.07944

Report Icon

https://kr.freepik.com

Sketch 이미지 데이터 분류 프로젝트 개인 회고

정현우 T7251

1. 학습 목표 달성을 위한 노력

1.1. Learning Rate 조정

- ✓ **문제 발견**: 모델의 정확도가 2% 이하로 수렴하지 않는 문제를 확인.
- ✓ **해결 방안**: 학습률(lr)을 조정하여 학습 과정을 최적화.
- ✓ **결과**: 조정 후 성능이 89%까지 향상됨.

1.2. 학습 최적화

- ✓ **문제 발견**: 모델이 커짐에 따라 학습 시간이 증가.
- ✓ **해결 방안**: num_workers 값을 늘리고, 기울기 축적(gradient accumulation) 및 mixed precision 기법을 도입하여 학습 속도 최적화.
- ✓ **결과**: 학습 속도가 약 5 배 향상됨.

1.3. Diffusion 모델을 통한 데이터 증강

- ✓ **문제 발견**: 학습 데이터셋의 다양성이 부족하여 valid 정확도가 오르지 않음.
- ✓ **해결 방안**: Diffusion model 을 사용하여 이미지 변형(예: 선 강조) 및 데이터 양을 2 배로 늘림.
- ✓ **결과**: 모델 성능이 최종적으로 91.8%까지 향상됨.

그 결과, 모델 성능을 **최종적으로 91.8%**까지 끌어올릴 수 있었습니다.

2. 마주한 한계와 아쉬웠던 점

이번 프로젝트에서 몇 가지 한계와 아쉬움이 있었습니다:

- **A.** **베이스라인 모델 탐색**에 충분한 시간을 투자하지 못한 점이 아쉬웠습니다. 다양한 모델을 실험하지 못한 것이 성능 향상에 방해가 되었다고 생각됩니다.
- **B.** **일정 관리에 대해 아쉬움이** 있었습니다. 팀원들은 열심히 참여했지만, **일정을 체계적으로 짜지 못한** 탓에 협업의 효율성이 떨어졌습니다. 프로젝트 일정 관리에 더 많은 노력을 기울이지 못한 것이 아쉬움으로 남습니다.

3. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점

- **A.** **더 다양한 베이스라인 모델을 탐색**하여 성능을 비교 분석할 계획입니다. 이를 통해 최적의 모델을 찾고, 빠르게 성능을 높이는 데 주력할 것입니다.
- **B.** **일정 관리의 중요성**을 인식했기 때문에, 다음 프로젝트에서는 **더 체계적으로 일정을 수립**하여 협업의 효율성을 높이고자 합니다. 팀원들과의 **코어 타임을 정하고, 주도적으로 일정 관리**를 하는 것이 저의 역할이라 생각합니다. 앞으로는 일정 관리에 더 많은 시간을 투자하여, **빈틈없는 협업 환경**을 구축할 것입니다.

이러한 경험을 바탕으로, 다음 프로젝트에서는 더 나은 결과를 도출할 수 있을 것이라 기대합니다.

Sketch 이미지 데이터 분류 프로젝트 개인 회고

김진석 T7131

1. 학습 목표 달성을 위한 노력

본 프로젝트를 통해 AI 모델 개발의 전체 프로세스를 경험함으로써, 기본 역량을 강화할 수 있었다.

1.1 데이터 증강 기법 탐구

스케치 이미지의 특성을 고려한 데이터 증강 기법 적용에 집중했다.

- GrayScale 변환과 RandomResizeAndCrop 등 기본적인 Data Augmentation 적용 시 성능 저하 발견
- TrivialAugment 적용과 같은 Advanced Data Augmentation으로 성능 향상 확인

이 과정을 통해 데이터 특성에 맞는 증강 기법 선택의 중요성을 깊이 이해하게 되었습니다.

1.2 모델 최적화 및 학습률 조정

다양한 모델과 학습률 스케줄러를 실험하여 최적의 조합을 탐색했다.

- CNN과 ViT 모델 비교 실험을 통해 ViT의 낮은 학습률 요구 특성 발견
- ViT: StepLR, LinearLR + CosineLR, CosineAnnealing 적용
- CNN: StepLR, OneCycleLR, ChainedScheduler(StepLR + ReduceOnLRPlateau) 적용

이를 통해 모델별 최적의 학습률 조정 방법과 세밀한 튜닝의 중요성을 이해했다. 다만, 일부 경우에는 다양한 스케줄러를 적용해도 성능 향상이 제한적이었는데, 이는 모델 구조나 데이터셋의 특성 등 다른 요인들도 함께 고려해야 함을 배웠다.

2. 마주한 한계와 아쉬웠던 점

- Competition과 AI 프로젝트 경험 부족으로 효과적인 계획 수립에 어려움
- 체계적인 기록 규칙 부재로 초기 모델 개선의 효율성 저하
- GitHub의 Issue 기능, Reviewer, 정기적 commit 등의 협업 기능 미활용
- 휴식 기간 동안 프로젝트 집중도 유지의 어려움
- 모델 선택 시 깊이 있는 조사 부족 및 Task에 최적화된 모델 선정의 어려움

3. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점

- 프로젝트 시작 전 체계적인 실험 계획 수립 및 실험 테이블 작성 규칙 정립
- GitHub의 Issue, Pull Request, Reviewer 기능 적극 활용
- Task에 적합한 최신 논문 및 모델 검색 및 적용
- 앙상블, k-fold 교차 검증 등 고급 기법 실제 활용
- Attention Map 시각화, WandB 등을 활용한 모델 분석 및 시각화 강화

Sketch 이미지 데이터 분류 프로젝트 개인 회고

최재훈 T7259

1. 학습 목표 달성을 위한 노력

1.1 베이스라인 코드 리팩토링

- **이유**: 코드의 유지보수를 용이하게 하고 협업하기 쉽게 만들기 위해서 진행했습니다.
- **내용**: 기존의 베이스라인 코드를 모듈화했고 Json파일 형식의 config를 활용해서 hyperparameter를 쉽게 바꾸면서 실험을 진행할 수 있도록 만들었습니다.

1.2 Streamlit을 활용한 데이터 시각화

- **이유**: 이미지가 어떤 형태로 증강되어서 모델에 입력되고 학습이 되는지 확인하기 위해 진행했습니다. Streamlit이라는 웹 서비스 개발 Python 라이브러리를 활용한 이유는 기존 코드에서 간단한 수정을 통해 웹 프로토타입을 만들 수 있고 코드에서 확인하는 것이 아닌 GUI로 편하게 확인하려는 목적이었습니다.
- **내용**: 다음과 같이 총 세 가지 기능을 만들었습니다.
  - **Augmentation 시각화 기능**: 다양한 증강 기법이 적용된 데이터를 원본 데이터와 함께 비교하여 시각적으로 확인이 가능합니다.
  - **Grad-CAM 시각화 기능**: 학습된 모델이 이미지의 어느 부분에 주목하고 있는지 시각화하여 어떻게 해석하고 있는지 파악할 수 있습니다.
  - **오분류 이미지 시각화 기능**: 학습된 모델이 분류를 실패한 이미지들을 시각화하고 모델이 예측한 라벨과 정답 라벨을 보여줍니다.

1.3 Augmentation 실험

1) AugMix 라이브러리 활용

- **가설**: 스케치 데이터 특성을 생각해 직접 증강 방법을 설정해 봐도 큰 변화가 없었고 많은 실험이 필요해 비효율적이라서 자동화된 데이터 증강 기법인 AugMix를 사용해서 데이터의 다양성을 늘리고 일반화 성능을 높이는 동시에 효율적으로 실험을 할 수 있다고 생각했습니다.
- **결과**: 간단한 적용만으로 베이스라인 코드 증강 방식보다 모델 성능이 0.3% 더 향상했습니다.

2) 증강 기법 Customizing

- **가설**: Streamlit에서 이미지 시각화 결과 ImageNet 데이터 기반으로 Normalize 된 방법이 스케치 이미지의 형태가 옅어지고 지워지는 현상이 생겨서 Normalize 분포를 변경하고, 위의 AugMix 방식에는 이미지를 Flip 하는 증강이 없어서 따로 추가했습니다.
- **결과**: 기존의 베이스라인 코드 증강 방식보다 모델 성능이 0.8% 더 향상했습니다.

2. 마주한 한계와 아쉬웠던 점

- 시간이 부족해 Streamlit 코드 정리와 오류 수정 등 아직 디버깅을 다 못한 점이 아쉬웠습니다.
- 모델 학습 시각화 도구를 사용하지 않고 실험을 진행해서 실험 관리의 한계가 있었습니다.

3. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점

- 이번에 개발한 Streamlit 기능을 다시 활용하기 위해 성능 개선과 기능 추가를 할 예정입니다.
- TensorBoard, WandB 등 모델 학습 시각화 도구를 활용해서 체계적, 분석적으로 실험을 해보고 싶습니다.

Sketch 이미지 데이터 분류 프로젝트 개인 회고

윤정우 T7214

1. 학습 목표 달성을 위한 노력

1.1 모델 선정 및 학습률 조정

- convNeXt 계열 모델 간 비교 실험 및 동일한 convNeXt 모델 간 학습률 비교 실험을 통해 convNeXtv2_large 모델의 최적의 학습률 발견
- L2 정규화 적용으로 일반화 성능의 유의미한 향상 확인
- 이 과정을 통해 모델별 학습률 조정의 중요성을 이해함
- 최종적인 일반화 성능 개선을 위한 정규화의 필요성을 확인함

1.2 데이터 증강 기법 탐구

- Geometric transformations, Morphological transformations 등을 적용했을 때, 성능 저하 확인
- Sobel Filter Augmentation(=이미지에서 경계를 검출하기 위해 사용하는 필터)를 사용하였을 때, 성능의 미세한 상승 확인
- 이 과정을 통해 데이터 특징에 적합한 증강 기법 선택의 중요성을 이해함

2. 마주한 한계와 아쉬운 점

1. 체계적인 실험 설계 및 기록 시스템, 의사결정 시스템의 부족

- AI 프로젝트 경험 부족에 기인한 전체적인 프로젝트 및 실험 설계 난항
- 체계적인 실험, 기록 시스템의 부재에 기인한 초기 모델 조기 선정, 그로 인한 프로젝트 후기 모델 성능 개선의 어려움

2. GitHub 협업 기능 미활용

- GitHub의 branch 생성 및 commit, Issue, Pull Request 등의 협업 기능 미활용

3. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점

1. 프로젝트 및 실험 설계/기록 시스템 체계화

- 프로젝트 시작 전 프로젝트 타임라인을 고려한 전체 계획 수립
- 프로젝트 시, 명확한 기록에 근거한 체계적인 의사결정 시스템 설정

2. GitHub 협업 기능 사용

- Branch 생성 및 Pull Request, Issue 기능 사용을 통한 효율적인 협업 기여

3. 다양한 기법 시도

- 앙상블 등 시도해보지 않은 기법 시도

Sketch 이미지 데이터 분류 프로젝트 개인 회고

최민석_T7258

1. 학습 목표 달성을 위한 노력

이번 classification Task가 첫 competition이라 협업을 중점적으로 고려하여 진행했고, 이번 기회를 통하여 모델의 Inductive Bias 관점으로 데이터를 분석하고자 노력하였다. 또한 성능이 좋은 모델보다 Base Model을 통하여 최적화 방법을 모색하였다.

가. 데이터 분석

- 데이터 중 같은 과의 동물 중 다양한 종에 대한 데이터가 존재
- 데이터에 노이즈, 워터마크 등 데이터 관점에서 관찰

나. 데이터 증강

- Base Line code에 존재한 Augmentation을 기반으로 시각화 코드를 작성 후 분석
- One Of 메소드를 통하여 메소드 내 증강 중 하나를 택하여 적용하는 방식을 적용

다. 모델 최적화

큰 모델은 학습하는 환경에 따라 학습 성능이 달라진다는 점을 알게 되었고, 이미지는 많은 메모리를 차지, CPU에서 GPU로 데이터를 올릴 때의 속도를 향상하는 모델 최적화 기법을 이용하였다.

- Data Loader의 매개변수 Pin_memory 활성화
- num_ worker의 수를 늘려 병렬처리 이용
- 모델을 미리 컴파일하여 모델을 불러오는 pytorch 2.0 API torch.compile()이용

2. 마주한 한계와 아쉬운 점

스케치 데이터는 주로 선으로 구성되어 있어 CNN의 Inductive Bias인 지역적인 정보 즉, 엣지 정보가 중요하다 판단했고, 전역적인 정보도 중요할 것이라고 판단했다. 따라서 그 두개의 장점과 ViT와 비슷한 성능을 지닌 ConvNext 모델을 사용했다. 하지만 큰 사이즈의 커널을 사용한다고 하여 전역적인 정보를 내포하는 것이 아님을 알게 되었다. 대회 경험이 부족하여 어떤 관점으로 모델을 선택해야 하는지 신중하게 생각하지 못한 점이 조금 아쉬웠다.

3. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점

다음 프로젝트에서는 좀 더 신중하게 데이터관점에서 Inductive Bias를 고려하여 모델을 선택하고 여러 평가지표를 사용하여 모델이 어떤 식으로 학습이 되는지 시도해 볼 예정이다. 또한 Paper with code 및 Hugging face의 State Of The Art 상위 모델에 대해 파악해보고 여러 모델을 테스트하는 과정을 통해 이번 대회보다 좀 더 많은 것을 시도해 볼 예정이다.

Sketch 이미지 데이터 분류 프로젝트 개인 회고

김동영 T7107

1. 학습 목표 달성을 위한 노력

팀원끼리 목표로는 정확도 90% 달성이 있었고,

개인적인 목표로는 꾸준히 프로젝트에 참여하는 것이 있었다.

프로젝트 초반 부에는 여러 모델을 돌려 보면서 나만의 가설을 세워보는 시간이었다. VIT, RESNET, Efficientnet, Convnext, 등을 돌려보면서 각각의 이미지도 살펴 보았을 때, 내가 발견한 특징은 세 가지였다.

- 이미지의들의 전체 카테고리 사이에는 전체적인 모양으로 구분된다.

- 이미지들의 상세 카테고리 사이에는 지역적인 특징으로 구분된다.

- 모델에 Residual 개념이 들어간 모델이 성능이 높다.

- 이미지가 대부분 흑백 이미지이고 일부 색상이 들어간 이미지가 존재한다.

그래서 이때 나는 두가지 가설을 세웠다.

가설1. 이미지에 지역적인 정보를 보는 CONV 모델과 , 전체적인 정보를 보는 VIT 모델, Residual 개념이 들어간 Resnet 모델 이 세 개를 참고한 모델이 예측을 더 잘 하지 않을까?

가설2. 이미지 전체를 Grayscale 이미지로 바꾸면 성능이 잘 나오지 않을까?

이때 색상이 들어간 이미지는 오히려 이미지의 분산을 증가시켜 학습에 악영향을 줄 것이라고 생각했었다.

중반부에는 강의를 들으며, 가설을 증명해 보았다.

앙상블 모델을 적용해보고, Convnext 모델, CoatNet 모델을 적용해 보았다. 이때 CoatNet, ConvNext 모델, 앙상블 모델이 전부 성능이 타 모델대비 성능이 좋아지지 않았다.

또한 이미지에 Grayscale 을 적용해 모델을 학습했을 때, 일부 모델은 성능이 오르나, 일부 모델에서는 떨어지는 것을 발견했다.

후반부에는 이미지 증강을 시도하고 하이퍼 파라미터 튜닝을 시도했다.

다른 팀원 분들이 발견한 ConvNext 모델에 RandomAug를 적용했을 때 실제로 성능이 올랐었다.

그리고, Optuna를 통한 하이퍼 파라미터 튜닝을 시도했지만, 시간이 촉박하여, 중단에 중단하고 중간 결과값을 모델에 적용해 보았지만, 성능은 오히려 떨어졌다. 이때 아직 수행을 다하지 못해 떨어졌다고 판단했다.

2. 마주한 한계와 아쉬운 점

①. 앙상블

내가 수행했던 앙상블은 스태킹이라는 방식이었다. 이때 당시에, 시간이 부족하다 판단하여 스태킹이라는 방식이 간단하게 할 수 있는 방식이기에 이 방식을 택했다. 추후 다른 팀의 얘기를 들어보니, Soft Voting, Hard Voting 방식으로 했을 때 성능이 올랐다는 얘기를 듣고 "**앙상블에 시간을 더 투자할껄"** 이라는 아쉬움이 남았다.

②. 모델 서치 부족

처음에 초기 모델을 팀원 분 중 한 분이 공유해주신 모델 리스트 중 하나로 정했다. 경진대회 종료 후 VIT 기반 eva 모델이 성능이 잘 나옴을 다른 팀을 통해 듣게 되었다.

간단하게 인터넷에서 찾을 수 있는 모델이 었다. 그래서 조금더 모델을 서치를 많이 해볼 걸 이라는 아쉬움이 남았다.

3. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점

①. 소통

제일 아쉬운 부분이다. "개인적으로 나의 의견이 상대방에게 불쾌감을 주지 않을까?"라는 생각에 개인적인 의견을 잘 표출하지 못하는 콤플렉스가 있다. 다음 프로젝트에서는 나의 의견을 잘 표현해서 팀원들과 잘 소통하고 싶다.

②. 모델 서치

결국, 성능을 가르는 것은 정보력이 큰 영향을 줌을 알게 되었다. 초반부에 다 같이 모델을 잘 서치해보는 시간을 가지면 좋겠다는 생각이 든다.

③. 가설 수립

1번 내용과 이어지는 부분이다. 결과적으로 보았을 때, 가설을 세우고 프로젝트에 임한 팀과 안한 팀과 성과 차이가 크게 나는 것 같다. 또한, 개인적으로, 팀적으로 가설을 수립하고 프로젝트에 임하면 성장도 더욱 가속화됨을 타 팀의 프로젝트 발표에서 느꼈다. 다음 프로젝트에는 팀의 가설을 수립해보고 프로젝트에 임하면 좋겠다고 생각한다.