Final Project Wrap-up Report

CV-03

김승기 김준영 신우진 전형우 천지은

---

1. 프로젝트 개요

앨범을 만드는 것은 많은 비용과 시간이 소요됩니다. 재능 기부와 같은 형식으로 돈을 지불하고 앨범 표지를 그나마 싸게 인터넷을 통해 구매할 수 있지만 짧게는 하루 길게는 몇 주까지도 시간 이 소요됩니다. 이러한 과정을 단순화해서 빠르면 몇 분 이내 느리면 몇 시간 이내에 만들어주도록 하고자 다음과 같은 주제를 선정했습니다. 저희 프로젝트에서는 다음의 두 가지 기능을 구현했습니다.

1-1. 앨범 이미지 생성

Text-to-Image 생성 모델을 사용해서 노래의 제목, 가수 이름, 장르, 가사 등의 정보를 넣으면 해당 정보를 활용해 곡의 분위기와 어울리는 앨범 이미지를 생성합니다.

1-2. 개인 맞춤 앨범 생성

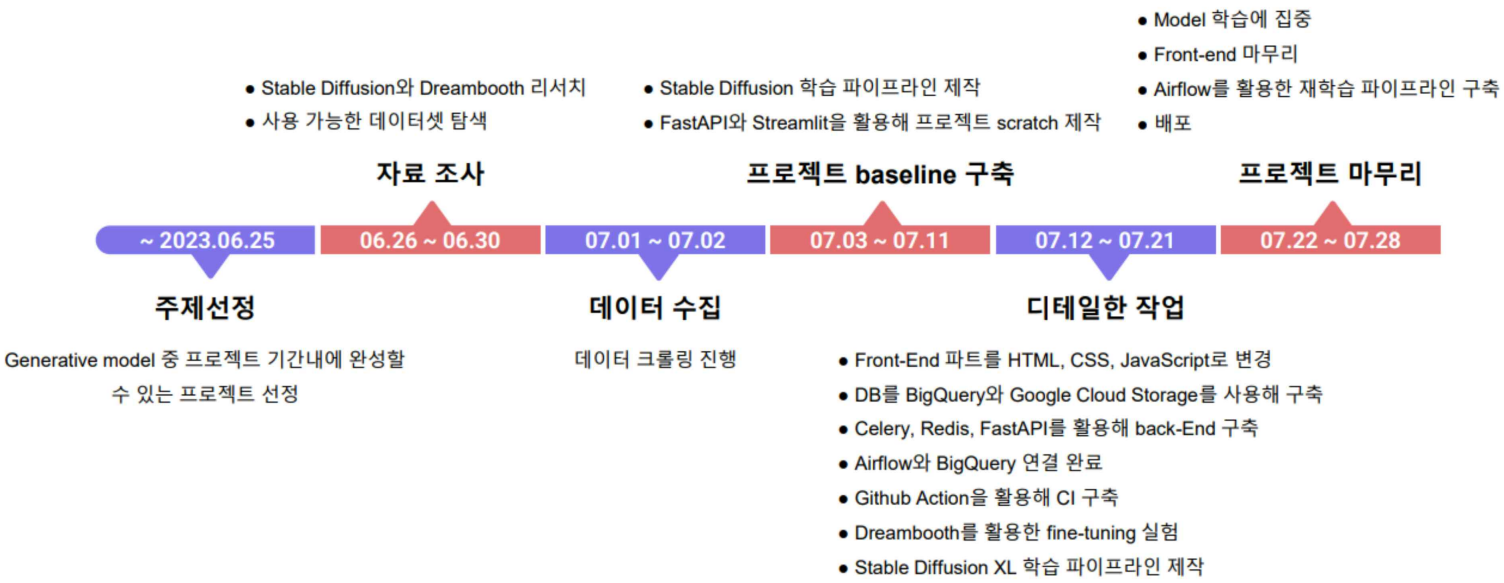
사용자의 사진을 입력 받고 해당 이미지를 식별할 수 있도록 Text-to-Image 모델을 Fine-Tuning 하여, 위와 같이 노래 정보를 활용해 사용자의 얼굴 이미지가 들어간 앨범 이미지를 생성합니다.

---

2. 팀 구성 및 역할

팀원	역할
전체	- 데이터셋 탐색 및 수집, 프로젝트 기획
김승기	- FastAPI, Redis, Celery, BigQuery, GCS를 이용한 Back-end 구축 - 서비스 아키텍쳐 설계, 모델 실험, 배포 준비
김준영	- Stable Diffusion 파인튜닝 파이프라인 구축 및 실험 - Github Action 활용한 CI 파이프라인 구축
신우진	- 주제 아이디어 제안 - Airflow를 사용한 재학습 파이프라인 구축, BigQuery와 연결
전형우	- Dreambooth 파인튜닝 파이프라인 구축 및 실험 - Error Reporting
천지은	- HTML/CSS, JavaScrip, Bootstrap를 활용한 Front-end 구축 - 데이터베이스 설계 및 구축, 모델 실험

3. 프로젝트 타임라인



---

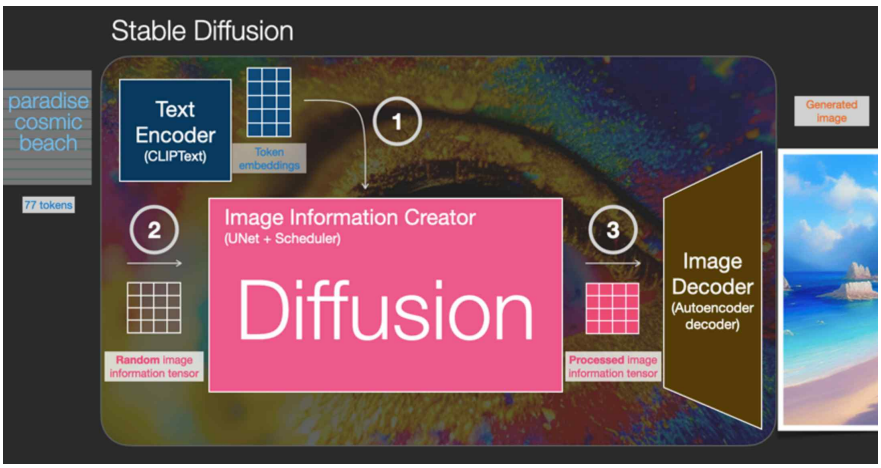
4. 프로젝트 수행 절차 및 방법

4-1. 데이터 수집

여러 음원사이트 중 국내뿐 아니라 국외 데이터도 충분히 확보되어 있고, 데이터 수집이 편리한 멜론 음원사이트를 최종 선택하였습니다. 해당 음원사이트로부터 노래 제목, 가수 이름, 앨범 이름, 발매일, 노래 장르, 노래 가사, 앨범 표지 URL을 Crawling하여 총 3,851개의 데이터를 수집하였습니다.

4-2. 모델 선정 및 학습

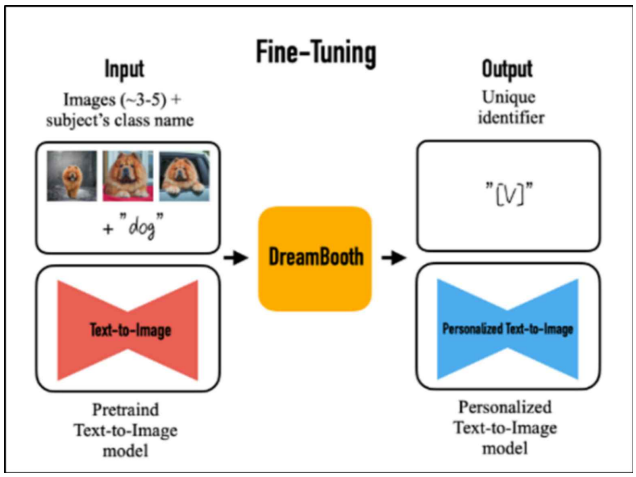
4-2-1. Stable Diffusion XL (SDXL)



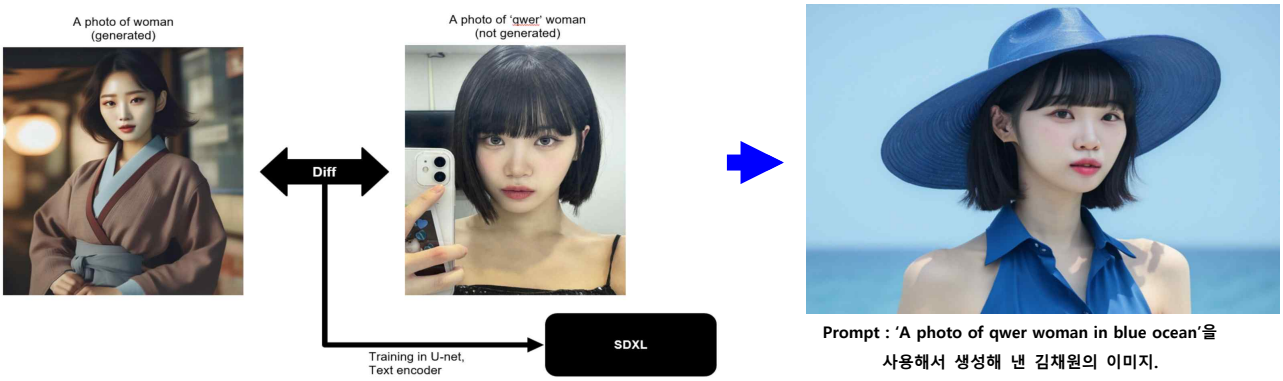
저희 프로젝트에서 사용한 모델은 StabilityAI에서 낸 Stable Diffusion의 가장 최근 모델인 XL 모델입니다. 텍스트 인코더 부분에서 CLIPText 모델을 활용해 입력된 텍스트를 Tokenizer로 만들고, 이를 UNet+Scheduler로 이루어진 Diffusion에서 학습을 거친 뒤 Image Decoder로 통해 이미지를 만들어 냅니다. 전 버전인 V1, V2에 비해 UNet의 크기가 3배 이상 크며 다양한 종횡비를 사용한 학습을 마쳤기에 어떤 상황에서도 고품질의 이미지를 생성할 수 있고, 성능이 크게 개선되었기에 해당 모델을 사용하기로 했습니다.

4-2-2. Train / Fine-Tuning

Crawling을 통해 수집한 텍스트 정보는 OpenAI의 GPT3.5 API를 통해 Prompt를 생성하고, 이미지 URL과 결합하여 CSV 파일로 학습을 진행하였습니다. 또한, 해당 Prompt는 텍스트 정보 중 일부를 활용하여 어떤 Prompt가 이미지의 특성을 더 잘 반영하고 앨범 커버를 생성하는데 도움이 되는지 실험을 진행하였습니다. 하지만, 생성 모델의 특성상 평가 기준이 명확하지 않아 해당 프로젝트에서는 주관적인 판단을 통해 평가를 진행했습니다.

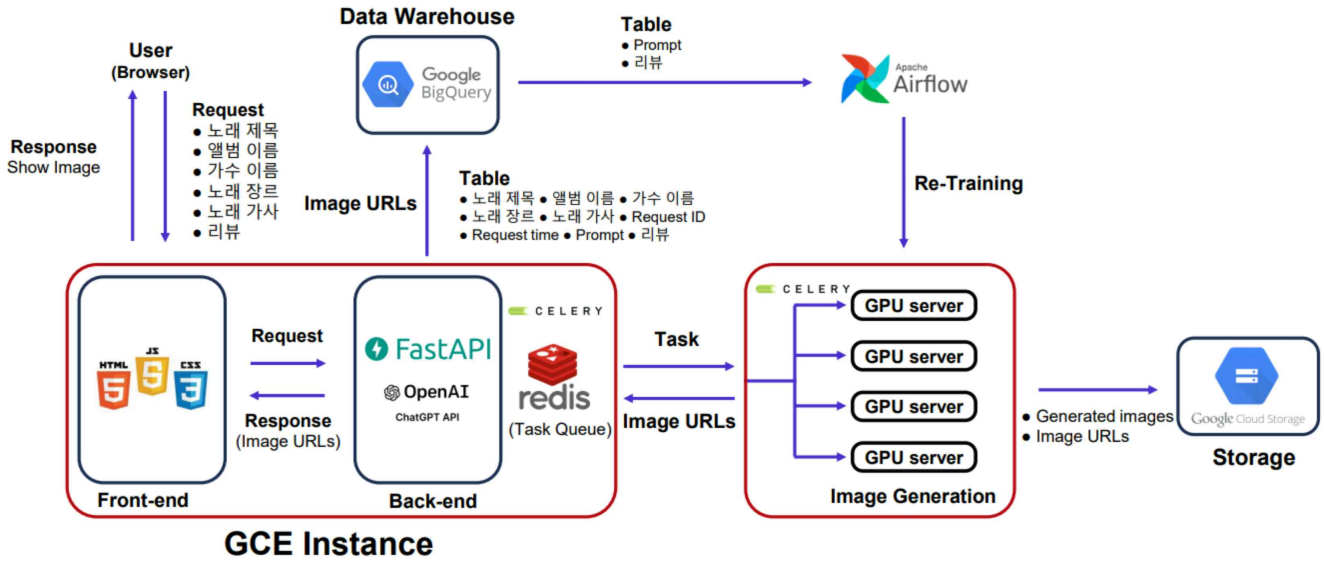


Dreambooth는 이미지 생성 모델에서 특별한 객체를 해당 객체에 대한 Unique Identifier를 주어 따로 학습을 진행해 Fine-Tuning을 진행하는 기법으로 이 방법을 통해 모델이 특정 사용자의 얼굴을 따로 인식할 수 있습니다. 예를 들어 모델에게 학습시키고자 하는 특정 인물이 연예인 '김채원' 인 경우 이 인물에 대해 존재하지 않는 영단어인 'qwer'을 Unique Identifier로 줍니다. 그리고 모델이 생성해내는 일반적인 'woman' 클래스의 이미지와 'qwer' woman을 입력으로 모델이 학습하고, 이렇게 Fine-Tuning 된 모델을 사용해 존재하지 않는 김채원의 사진을 생성할 수 있습니다.



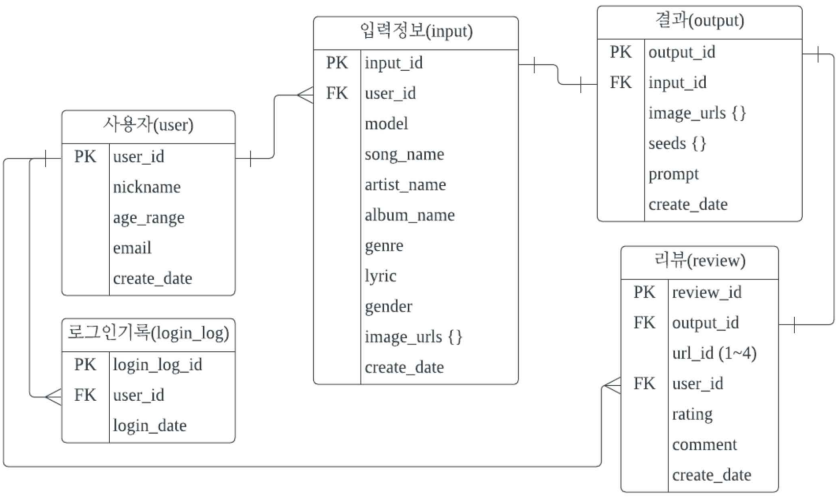
Prompt : 'A photo of qwer woman in blue ocean'을 사용해서 생성해 낸 김채원의 이미지.

4-3. 서비스 아키텍쳐



Front-end, Back-end(FastAPI), Redis 서버를 하나의 VM 위에 올려서 구축했습니다. 그리고 높은 비용의 계산이 드는 작업을 Celery를 이용해 GPU 서버로 Worker를 구성해서 처리하도록 했습니다. 전체적인 흐름은 사용자가 이미지 생성에 필요한 정보를 입력하면 Back-end 서버에서 해당 정보를 가지고 OpenAI의 GPT3.5 API를 통해 필요한 내용을 몇 가지의 특징을 나타내는 단어로 요약 및 추출합니다. 특징 단어들을 통해서 설정한 Prompt를 Task과 함께 Redis의 메세지 큐에 저장했다가, Celery를 통해서 각 GPU 서버의 Worker로 작업을 분산시켰습니다. 유저의 입력 정보 같은 경우 Back-end 서버에서 BigQuery로 로깅을 했고, 생성된 이미지는 GPU 서버에서 Google Cloud Storage로 업로드 하면서 해당 이미지의 URL을 다시 Back-end 서버로 전달됩니다. Back-end 서버에서는 해당 이미지 URL들을 Response로 사용자에게 전달합니다.

4-4. 데이터베이스 설계



BigQuery에 저장된 데이터베이스의 구조는 총 5개의 테이블로 이루어져 있고, 구조를 단순화하기 위해 정규화를 고려하지않고, 생성된 4개의 이미지를 하나로 함께 저장하도록 설계하였습니다. 또한, 보안성을 고려하여 PK는 UUID, 사용자 테이블의 email은 도메인 주소를 마스킹하여 저장합니다.

4-5. Re-Training / Reporting

모델 재학습은 일주일 단위로 이루어집니다. 일주일간 사용자가 남긴 리뷰와 입력값을 BigQuery에서 가져오고 저장하여 학습이 가능한 형태로 데이터를 가공합니다. 이후 이전 모델을 Fine-Tunning합니다. 또한 주간 사용자 레포트를 작성하고 결과를 Slack으로 전송시켜서 주간 이용 현황을 확인할 수 있도록 만들었습니다. Re-Training, Reporting은 모두 Airflow로 구성하였습니다.

4-6. UI / UX

해당 프로젝트의 Front-end는 HTML/CSS, JavaScript 그리고 Bootstrap을 활용하여 웹 페이지를 구성하였습니다.

4-6-1. 메인 화면

메인 화면은 사용자들이 프로젝트의 기능을 쉽게 이해하고 흥미를 높일 수 있도록 디자인되었습니다. 사용자들은 메인 화면에서 예시 이미지들을 살펴볼 수 있으며, 이러한 이미지들은 BigQuery에 저장된 테이블 정보를 기반으로 가장 최신순으로 별점이 높은 이미지 12개를 표시합니다.

4-6-2. 앨범커버 생성 화면

앨범커버 생성 화면은 사용자가 앨범 커버를 손쉽게 생성할 수 있도록 직관적으로 구성되었습니다. 'Create'버튼과 이미지가 표시될 위치가 사용자에게 잘 보이도록 배치되었습니다. 이미지는 한 번에 4개씩 보여지며, 이미지가 생성되면 각 이미지 위에 다운로드 버튼이 나타납니다. 사용자는 해당 버튼을 클릭하여 리뷰를 작성한 후 이미지를 다운로드할 수 있습니다.

4-6-3. 마이페이지 화면

로그인한 사용자는 마이페이지에서 자신이 리뷰를 작성한 이미지를 최신순으로 확인할 수 있습니다. 이미지를 가져오는 시간을 고려하여 20개로 제한하였고, 각 이미지를 클릭하여 해당 이미지에 입력했던 정보를 확인할 수 있습니다.

5. 프로젝트 수행 결과

5-1. 앨범 이미지 생성 결과

Song title is 'Fighting', singer name is 'KJY' keyword is clock

Song title is 'Super shy', singer name is 'New Jeans'

A photo of qwer woman in blue ocean

5-2. 달성도 / 완성도

프로젝트 달성도	프로젝트 완성도
■ Front-end / Back-end 구축 ■ Github Action을 활용한 CI 구축 ■ 모델 학습 파이프라인 구축 ■ Airflow 재학습 파이프라인 구축	■ 서비스 배포 준비 완료 △ 생성된 앨범 표지가 실제 사용

5-3. 결과 및 고찰 / 후속 개발

데이터셋	결과 및 고찰	- Crawling에 대한 저작권 문제 - 기존 앨범 커버에 대한 사용자 평가가 없음
	후속 개발	- 데이터 사용에 대한 적절한 권한과 라이선스 확보 필요 - 더 나은 이미지를 만들도록 Prompt를 RL 기반으로 학습 - 데이터의 종류를 더욱 세분화 해서 학습에 사용
모델	결과 및 고찰	- 앨범 커버 정보가 아닌 앨범 정보 사용 - 생성모델 특성상 재학습을 할 때 어떤 기준으로 재학습을 진행할지 애매함
	후속 개발	- OCR task를 후처리 작업에 사용 - 생성시 사용자가 원하는 스타일, 인물 포함 여부, 색감 등 각종 Variation을 조절할 수 있도록 개발 - 한국의 앨범 커버를 만들기 위해 Tokenizer, Text Encoder 모두 한국어를 학습

Back-end	결과 및 고찰	- 개인 사용자가 GCE instance를 사용하기에는 비용이 너무 부담됨
	후속 개발	- Scalability를 고려해 Front-end, Back-end 서버를 분리
Front-end	결과 및 고찰	- 개발자 입장에서 사용자의 입장을 이해하는 것이 굉장히 어려움
	후속 개발	- 이미지를 사용자가 생성하고 싶은 개수를 지정할 수 있도록 변경 - 광고를 넣을 수 있도록 UI를 변경 - 사용자 경험에 따른 인터페이스 디자인 반영

6. 자체 평가 의견

잘했던 점

→ 꾸준한 커뮤니케이션과 협업을 통해서 프로젝트를 진행했습니다.

→ 초기 계획했던 작업들은 모두 마무리 지었습니다.

시도했으나 잘 되지 않았던 점

→ 모델 학습의 결과가 예상만큼 나오지가 않았습니다.

→ 후반으로 갈수록 가상환경 관리가 잘 안됐던 것 같습니다.

아쉬웠던 점

→ Github 관리가 후반으로 갈수록 잘 안됐던 점이 아쉽습니다.

→ 배포 브랜치 관리가 잘 안됐던 점이 아쉽습니다.

프로젝트를 통해 배운 점 또는 시사점

→ 서비스의 유지보수가 어렵다는 것을 느꼈습니다.

→ Github Action과 Poetry 가상 환경으로 확실한 의존성 관리와 깃 관리에 대해 배웠습니다.

다음 프로젝트에서 시도할 것

→ CI를 좀 더 확실히 해서 코드가 엉키는 일이 없도록 해야할 것 같습니다.

개인 회고 : 김승기

---

1. 이번 프로젝트의 개인 목표

・재학습을 고려한 머신러닝 프로젝트의 서비스 아키텍처 구성해보기

・확장성을 고려한 설계를 고민해보기

2. 이번 프로젝트를 통해 배운 점

・모듈화의 중요성을 느꼈다. 용이한 유닛 테스트와 확장성을 고려한 설계를 하기 위해서 모듈화를 잘해야 한다

・대부분 머신러닝 프로젝트는 대용량 트래픽을 감당을 해야 하고, 이러한 대용량 트래픽을 감당하기 위해 고려할 점이 너무나 많다는 것을 느꼈다. 단순히 로드 밸런싱이나 확장성을 넘어서 실시간 전처리, 캐싱, 대용량 데이터의 저장 등에 대해서 공부할 필요성을 인지했다.

・에자일한 개발을 위해서는 CI/CD 파이프라인이 효율적으로 설계 되어야 한다.

・새로운 영역에 대한 재학습 기준을 정하기가 어렵다. 생성모델로 이미지를 생성하는 경우 FID, CLIP score, 등 여러가지 평가방법이 존재하지만 이 metric들은 결국에 하나의 참고용 지표가 될 수 밖에 없다는 것을 느꼈다. 가령 CLIP score가 기존의 pretrained StableDiffusion 보다 높아도 그것이 더 높은 퀄리티의 이미지를 보장하지 않다는 것을 알게되었다. 이번 프로젝트 같은 경우 재학습 기준을 결국에는 유저 피드백을 통해 정했지만, 어떤 정량적인 평가와 유저 피드백을 동시에 참고한 기준을 정하는 것이 어려웠다.

3. 앞으로의 목표 및 학습 방향

・백엔드 더 깊게 공부하기

・서버에 대한 이해 그리고 전체적인 서비스 아키텍처 설계에 대한 내용을 공부하고 싶다

・머신러닝 프로젝트 디자인 패턴에 대해 공부하고 사이드 프로젝트에 적용해보기

・MLops 를 프로젝트에 적용하기

4. 추후에 프로젝트에 적용해보고 싶은 내용

・생성 모델이 발전하면서 입력과 버튼 하나만으로 퀄리티가 있는 이미지를 생성할 수 있는 시대가 열렸다. 하지만 아직까지도 세세한 특징의 조작이 어렵고, prompt 엔지니어링 만으로 모든 이미지를 원하는 데로 생성하는 것을 보장하지 못한다. 소위 AI 아티스트라고 자기를 칭하는 사람들을 보면 단순히 prompt 만으로 이미지 생성하는 것이 아니라, prompt로 이미지 생성 -> 작업이 필요한 영역 삭제 후 Inpainting -> 다시 prompt로 조정하는 workflow를 가지고 이미지를 만들어낸다. 이러한 workflow를 자동화 해서 생성모델을 이용한 서비스에 적용해보면 재밌을 것 같다는 생각이 들었다.

개인 회고 : 김준영

---

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

- Stable diffusion을 사용하기 위해 리서치를 진행했습니다.

- Github action을 활용해 CI를 구축하기 위해 많이 사용되는 여러 github들을 살펴보며 프로젝트 구조를 익혔습니다.

2. 나는 어떤 방식으로 모델을 개선했는가?

- Prompt를 최대한 원하는 결과를 얻을 수 있도록 변경해가며 실험을 진행했습니다.

- LoRA를 활용해 좀 더 낮은 코스트로 좋은 결과를 얻을 수 있도록 했습니다.

3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

- Github action을 활용해 CI를 구축하면서 협업에 있어서 CI가 얼마나 중요한지를 깨달았고 프로젝트의 구조가 어떻게 구성되는지를 알 수 있어서 앞으로 다른 github을 참고할 때 이전보다 편할 것 같습니다.

4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

- 패키지 관리를 poetry로 진행하면서 팀원들과 같은 환경속에서 실험을 진행할 수 있었습니다. 하지만 이를 처음부터 진행했다면 어땠을까 하는 아쉬움이 남습니다. 이렇게 poetry를 사용함과 동시에 CI를 진행하면서 코드 자체에 대한 오류는 줄일 수 있었던 것 같습니다.

- 프로젝트를 end-to-end로 진행하면서 프로젝트를 어떻게 바라봐야 하는지 알 수 있었던 것 같습니다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 한 달이라는 시간이 생각보다 너무 짧았던 것 같습니다. 서비스 배포라는 것을 알 때쯤 끝나는 것 같아서 제일 아쉬움이 남는 프로젝트인 것 같습니다.

- Text-to-image를 처음 접했을 때는 넣어주면 나오겠지라고 생각했던 것 같습니다. 그러나 생각과 달리 text에 맞는 이미지를 뽑기 위해서 text는 image의 시각적인 부분에 대한 설명이 포함되어 있어야 함을 깨달았습니다. 그러나 이 점을 조금 늦게 깨달았던 점이 아쉽습니다.

6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- CI를 좀 더 자세히 작성해볼 것 같습니다.

- 백엔드에 대해서 좀 더 자세히 공부해볼 것 같습니다.

개인 회고 : 신우진

---

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

이번 저의 학습 목표는 MLops를 전반적으로 경험해 보는 것 이었습니다. MLops에서 자주 사용되는 라이브러리부터 공부하고 프로젝트에 적용했습니다.

2. 나는 어떤 방식으로 모델을 개선했는가?

저희 팀이 사용한 모델은 stable diffusion으로 생성 모델 특성상 Metric이 애매한 문제점이 있기 때문에 최대한 사용자의 review를 활용하여 fine tunning하는 것을 목표로 삼았습니다. 높은 점수를 받은 이미지는 다시 재학습에 활용하고 아닌 사진은 뺌으로써 모델을 고도화 하도록 만들었습니다. 또한 이 과정을 자동화 하기 위해 airflow를 사용하였고 주기적으로 데이터를 가져오고 가공하여 모델 학습까지 시키도록 만들었습니다.

3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

Airflow를 사용하여 재학습 pipeline을 구축하면서 어떻게 해야 사용자로부터 받은 review를 통해 모델을 더 고도화 시켜야 하는지를 고민했습니다. 실질적으로 사용자로부터 데이터가 들어오지는 않았기 때문에 실제 데이터로 고민하지는 못했지만 여러 기업들의 사례를 보며 코드를 작성했습니다.

4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

모델을 만들고 성능을 평가하고 다시 직접 모델링을 했다면 이번 프로젝트에서는 어떻게 모델링을 자동화할지 고민하는 시간이었습니다. 사람이 직접 개입하지 않더라도 계속해서 모델이 계속해서 데이터변화를 따라가며 성능이 발전할 수 있도록 만들 수 있을지 고민했습니다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

사용자로부터 직접 받은 데이터가 적은 것이 한계점이었습니다. 그래서 재학습 파이프라인의 성능을 제대로 확인할 순 없었습니다.

6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

만일 사용자로부터 직접 피드백을 받을 수 있는 프로젝트라고 한다면 여러가지 재학습 파이프라인을 만들고 어떤 모델의 성능이 가장 향상했는지 확인해 보고 싶습니다.

개인 회고 : 전형우

---

1. 이번 프로젝트의 개인 목표

・AI 프로덕트 서빙을 end-to-end로 진행을 하면서 관심 있는 주제의 AI 모델을 직접 사용해보고 서빙까지 연결을 하면서 전반적인 MLOps와 모델 라이브러리에 대해 익숙해지는 것을 목표로 했습니다.

・또한 최종 프로젝트인만큼 깃을 더 많이 사용하면서 팀원들과 협업을 진행하고, 이제 부스트캠프를 수료하고 밖에 나가서 더 많은 협업을 하게 될 것을 대비해 깃을 사용한 협업에 더욱 익숙해지는 것을 목표로 했습니다.

2. 이번 프로젝트를 통해 배운 점

・HuggingFace의 diffuser 라이브러리를 사용하면서 최근 나온 생성 모델인 Stable Diffusion XL 모델에 대해 사용 방법을 익히고, 아직 구현되지 않은 기능을 issue와 PR에 들어가 찾아서 코드를 참고하며 새 코드를 짜면서 라이브러리에 대한 이해와 현재 라이브러리 멤버들이 어떻게 작업을 진행하는지에 대해 배웠습니다.

・깃허브 액션을 통한 CI/CD를 사용하면서 가상환경 관리와 깃 관리에 대해 더 배울 수 있었습니다.

・ BigQuery와 FastAPI를 사용하면서 DB와 백엔드 구축에 대한 지식을 조금이지만 알 수 있었고 이를 직접 구현하며 Python으로 백엔드를 구축하는 방법에 대해 조금 알게 되었습니다.

3. 앞으로의 목표 및 학습 방향

・데이터 엔지니어링이 중요하다는 것을 아는만큼 AI 분야로 더 모델링을 공부해서 나아가면서도 CS지식에 대한 공부를 꾸준히 해야할 것 같습니다.

・AI엔지니어 역시 개발자라는 것을 뼈저리게 깨달았기 때문에 취직을 준비하면서 대학원도 생각하고 있기 때문에 관심 분야에 대한 주제를 정하고 해당 분야에 대한 학습을 계속 하면서도 알고리즘, CS등 개발자로서 알아야 할 지식들을 꾸준히 공부해야겠습니다.

4. 추후에 프로젝트에 적용해보고 싶은 내용

・재학습 기능을 사용하면서 드림부스를 통한 파인튜닝 기법에 모델이 토큰을 사용해서 점차 더 많은 사람들의 이미지를 학습하고 기억할 수 있게 만들고, 이를 이용해서 어느 정도 닮은 사람끼리는 학습하는 속도를 개선하는 것을 적용해보고 싶습니다.

・30분 이상 기다려서 결과를 뽑는 것이 사실상 실제에 적용하기엔 많은 부담이 된다고 생각하기 때문에 이를 최대한으로 줄이면서도 퀄리티를 유지하는 방법을 적용해보고 싶습니다.

개인 회고 : 천지은

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

저의 학습목표는 팀원들과 지속적인 소통으로 협업을 강화하여 프로젝트를 성공적으로 마무리하는 것이었습니다. 프로젝트를 효율적으로 진행하기 위해 협업 도구를 적극 활용하고 이를 통해 작업 상태를 공유하였습니다. 또한, 제가 맡은 업무에 대해 팀원에게 지속적으로 피드백을 받아 더 나은 방향으로 발전시키는 데 주력하였습니다.

2. 어떤 역할을 수행했고, 이를 통해 어떤 깨달음을 얻었는가?

팀 내에서 전반적으로 Front-end 구축하면서 사용자 중심의 서비스를 제공하고자 노력하였고, 팀원들의 의견을 적극적으로 반영하여 UI/UX를 지속적으로 개선해 나갔습니다. 이를 통해 개발자가 아닌 사용자의 피드백을 통해 원하는 기능과 디자인을 파악하고, 그에 맞춰 서비스를 개선하는 것의 중요성을 느낄 수 있었습니다.

또한, 모델 학습 측면에서 Stable Diffusion 모델을 통해 가사의 인사이트, 감정 키워드 등 어떤 Prompt가 이미지 생성에 더 효과적인지 최적의 Prompt를 찾기 위한 실험을 진행하였습니다. 이를 통해 Prompt의 내용에 따라 이미지 생성 결과에 큰 영향을 미친다는 것을 깨달았습니다. 그리고 데이터베이스 설계 및 구축을 진행하면서 데이터베이스의 성능을 최적화 하기 위해 정규화와 반정규화를 적절히 활용하여 프로젝트에 더 효율적인 방향으로 설계하는 것이 중요하다는 것을 알게 되었습니다.

3. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

제한된 시간으로 기능 구현에 우선순위를 두어 디자인 부분을 더 개선하지 못한 점이 아쉽습니다 또한, 프로젝트 구축 과정에서 실제 사용자들의 피드백을 직접 받지 못한 점이 아쉽습니다. 팀원 외에 실제 사용자들의 요구사항은 파악하지 못한 채로 기능과 디자인이 구현되었습니다.

또한, 모델 학습 과정에서 이미지의 퀄리티를 높이는데 어려움을 겪었습니다. 또한, 학습 과정에서 Prompt 외에 정제한 데이터, 파라미터 조정 등 실험을 진행하지 못한 점이 아쉽습니다. 그리고 데이터베이스를 설계할 때 Google Cloud의 BigQuery의 특성상 수정/삭제가 불가능하다는 점을 사전에 고려하지 못했습니다. 처음부터 사전조사를 하고 데이터베이스 설계를 더 신중하게 했다면 더 나은 데이터베이스를 구축할 수 있었을 것으로 생각됩니다.

4. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

모델 학습에 있어서 더욱 깊이 있는 연구와 실험이 필요하다는 것을 깨달았습니다. 또한, 데이터베이스를 설계할 때, 다음에는 저장 공간의 특성과 제약공간을 고려하여 데이터 구조를 선택하고, 데이터 수정/삭제가 필요한 경우 대비하여 대안을 마련해야한다는 깨달음을 얻었습니다. 마지막으로 팀원과의 소통과 협업을 강화하여 효율적인 협업을 이루어내는 것도 앞으로의 프로젝트에서 계속해서 발전시켜야 할 숙제입니다!