QA를 통한 소설 삽화 자동 생성

프로젝트 기간	담당역할	개발 언어	기타 사용 IT Tool
2022.01~2022.03	모델 개발, 파이프라인 구축, Prompt 테스트	Python	Pytorch, Haystack

코드: https://github.com/naem1023/novel-illustration-disco-diffusion
발표자료:
https://docs.google.com/presentation/d/1Eqa9TNg_kxKNZDyDHA2HxYeKfrgrp2dx/edit?usp=sharing&ouid=108867471698138933426&rtpof=true&sd=true

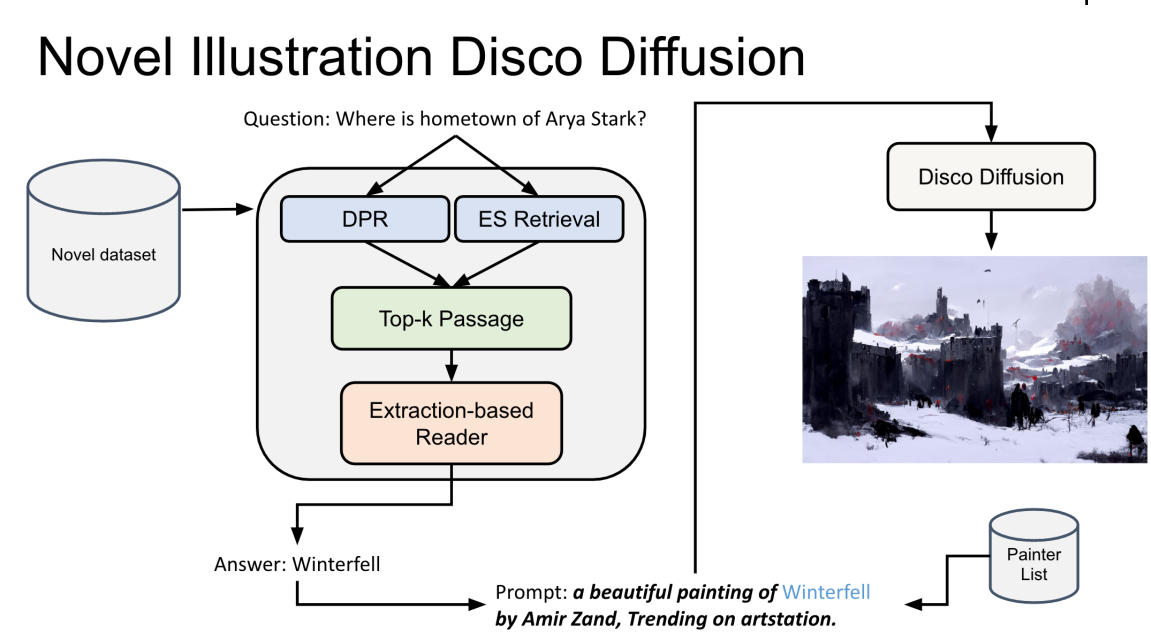
Overview
사용자가 소설에 대한 질문에 답을 하고 이미지를 생성하는 파이프라인입니다.
최상의 시나리오는 사용자가 소설의 모든 내용을 기억하고, 모델에 적합한 prompt를 직접 생성하는 것입니다. 하지만 소설에는 많은 양의 내용이 있고, 이를 모든 사용자가 항상 기억하는 것은 어렵습니다.
따라서 소설에 대해 궁금한 점을 Question-Answering을 통해 해결하고, 이에 대한 이미지를 생성해서 일종의 소설 삽화를 자동으로 생성하는 아이디어를 고안했습니다.

Pipeline
1. 소설에 대한 질문을 QA model에 투입
2. QA model이 질문에 대한 대답을 생성 or 추출.
3. Post-processing을 통해 질문에 대한 답을 prompt에 맞게 변환. (Hard-coding, Few-shot Learning)
4. Disco Diffusion에 prompt를 투입해 이미지 생성.

Question Answering
1. DPR(Dense Passage Retriever)과 ES(Elastic Search)가 질문과 가장 관련 있는 k개의 passage 혹은 page를 탐색.
2. Extraction-based Reader model로 질문에 대한 대답을 추출.

Disco Diffusion
CLIP guidance Diffusion model, pretrained by OpenAI.

Referene
Original Disco Diffusion: https://colab.research.google.com/github/alembics/disco-diffusion/blob/main/Disco_Diffusion.ipynb
Diffusion model: https://github.com/openai/guided-diffusion



논문 스터디, 독학, 대회

Neural Search Study(2022.01 ~ 2022.03)

- Neural Search 관련 개인 프로젝트 피드백, 토의
  Hyperclova를 쓰시는 분들의 조언으로 Prompt programming(Few-shot, P-Tuning)에 대한 간단한 공부도 했습니다.
    - Github: https://github.com/naem1023/novel-illustration-disco-diffusion
    - Presentation: https://docs.google.com/presentation/d/1Eqa9TNg_kxKNZDyDHA2HxYeKfrgrp2dx/edit?usp=sharing&ouid=108867471698138933426&rtpof=true&sd=true
    - Feedback/Discussion: https://docs.google.com/presentation/d/1vypEq9scv_n66ZDBGgKG0z58gyWcV6kqfCqPbEkvTJE/edit?usp=sharing
- 리뷰 논문
    - Retrieving and Reading : A Comprehensive Survey on Open-domain Question Answering https://arxiv.org/abs/2101.00774
- 리뷰 프레임워크
    - Haystack

NLP 논문, 구현체 리뷰 스터디(2022.04 ~)

Reference, target Github: https://github.com/luyug
NLP 연구자 분의 Github repo들(GradCache, Reranker, Condenser, COIL, GC-DPR, Dense)을 코드 레벨에서 분석. 해당 repo의 논문들도 분석.

- 리뷰 논문
    - Scaling Deep Contrastive Learning Batch Size under Memory Limited Setup https://arxiv.org/abs/2101.06983
      Review: https://naem1023.github.io/ml-engineering/nlp/Grad-Cache/

QA를 통한 소설 삽화 생성 - CLIP Guidance Diffusion model(2022.01 ~ 2022.03)

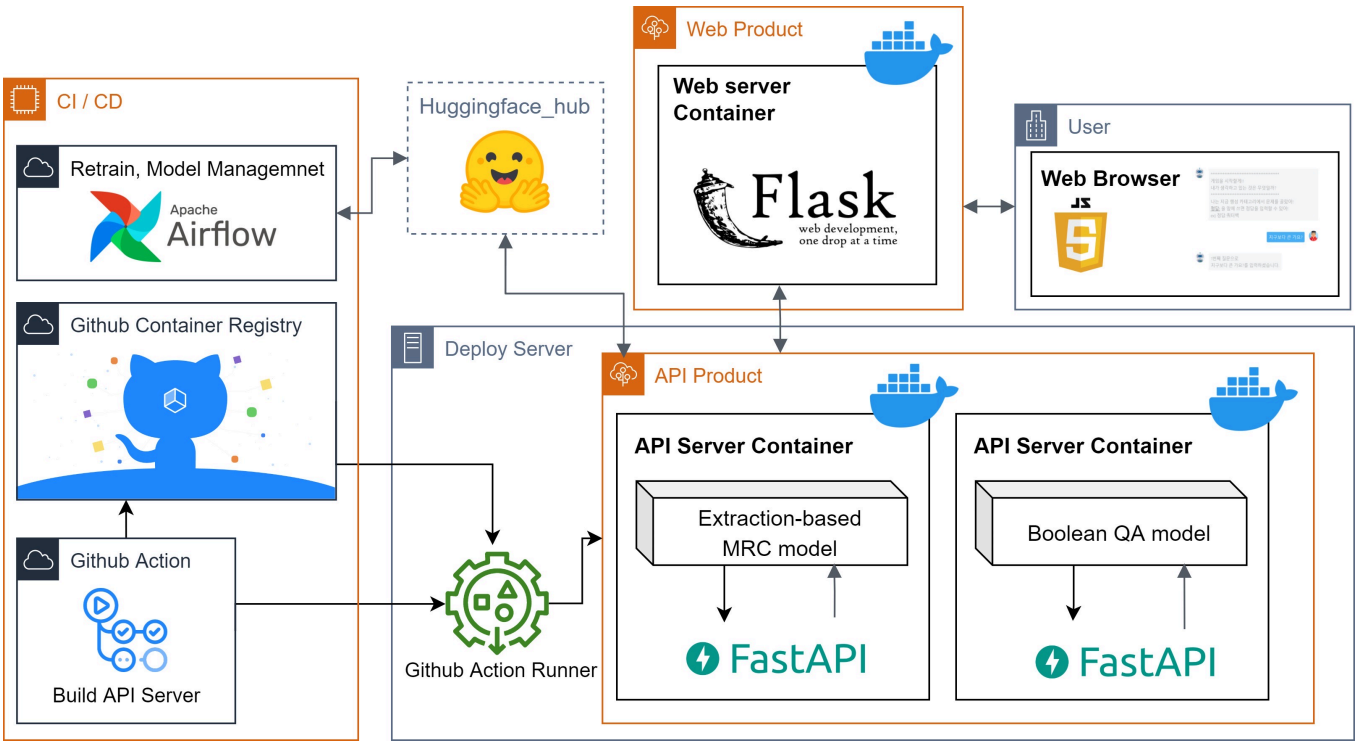
- 리뷰 논문
    - CLIP, https://naem1023.notion.site/CLIP-d0caa5302fcb47f3897a5892ac41ad81
    - Diffusion Models Beat GANs on Image Synthesis, https://naem1023.notion.site/Diffusion-Models-Beat-GANs-on-Image-Synthesis-f21eb3a1530840fdaf4fdbbc2b58dfbc
    - GLIDE, https://naem1023.notion.site/GLIDE-Towards-Photorealistic-Image-Generation-and-Editing-with-Text-Guided-Diffusion-Models-27a5fb0e375745c8b30acdb701da5098

Recsys2022 - Transformers4Rec(2022.04 ~)

NVIDIA의 Recommendation system model인 Transformers4Rec을 분석하면서 대회 참여 중입니다.
Github: https://github.com/naem1023/rec-sys-2022-challenge

네이버 부스트캠프 딥러닝 MRC를 활용한 스무고개 게임

프로젝트 기간	담당역할	개발 언어	기타 사용 IT Tool
2021.11 ~ 2021.12	Project Manager, 모델 서비스 개발 담당	Python, Shell script	Docker, FastAPI, Flask, HuggingFace, Elastic Search



코드: https://github.com/boostcampaitech2/final-project-level3-nlp-09

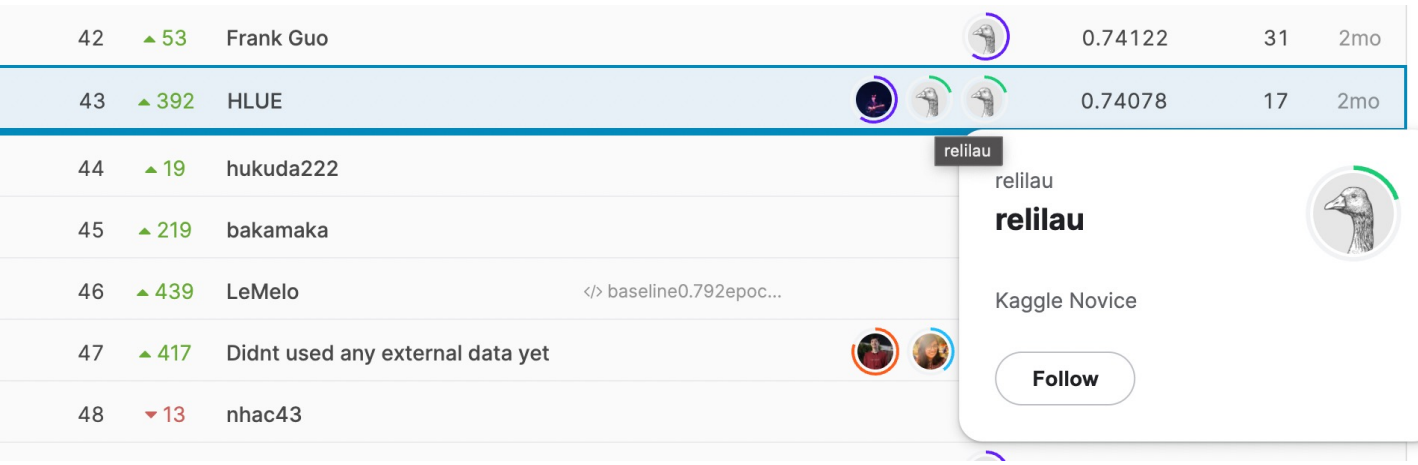
스무고개 게임을 위한 질의응답을 딥러닝을 활용해서 해결.
- 일반적인 예/아니오에 대한 질문은 Boolean QA Model로 해결.
- 힌트에 사용되는 주관식 질문은 Extraction-based MRC Model로 해결.

[도커라이징을 통한 모델 서빙 담당]
- 팀원들이 개발한 연구 단계의 inference 코드를 서비스 단계로 변형하기 위해 모듈화 및 리팩토링
- 서로 다른 환경이 요구되는 두 개의 모델 서빙을 위해 모델별로 별도의 쿠다 컨테이너 구축
- FastAPI로 모델 조작, 추론을 위한 비동기 API 개발

[프로젝트 설계, 역할 분배, 병합]
- 모델 재학습, 모델 서빙, CI/CD 파이프라인 설계.
- 팀원들의 개발 수준을 고려하여 업무 할당.
- e.g., 모델 재학습은 제가 Airflow, Docker를 활용해 프로토타이핑으로 개발한 코드를 할당된 팀원에게 전달 및 교육.
- 매일 이뤄지는 회의에서 팀원들의 개발 사항 확인, 해결하지 못한 이슈를 같이 해결

Kaggle, Hindi and Tamil Question Answering

프로젝트 기간	담당역할	개발 언어	기타 사용 IT Tool
2021.10 ~ 2021.11	모델 학습/평가, 앙상블 시도	Python	Pytorch,HuggingFace



Google 주관의 힌디, 타밀어 기반 Closed domain Question Answering Kaggle 대회
Github: https://github.com/quarter-100/Hlue
해당 대회 Silver medal, 43등.

- 힌디, 타밀어 관련 전처리는 Kaggle Discussion의 여러 코드들을 통합해 처리
- 사용 가능한 Backbone 모델을 여러 개 실험. (Hindi bert, xlm-roberta-large, bert, roberta)
- 가장 성능이 좋았던 backbone 모델에 대해 앙상블 시도

AiFactory 텍스트 요약 온라인 해커톤 대회

프로젝트 기간	담당역할	개발 언어	기타 사용 IT Tool
2021.11 ~ 2021.12	모델 학습/평가, 앙상블 시도	Python	Pytorch,HuggingFace

Github: https://github.com/quarter-100/text-summarization
해당 대회 2등.

- Abstractive summarization: KoBART, KoGPT2를 활용.원본보다 길거나, 모델에 저장된 불필요한 정보들이 요약 결과에 포함되어 폐기.
- Extractive summarization: Pororo API의 brain-bert 활용. 최적의 top_k, top_p, beam_search 값을 탐색.

네이버 부스트캠프 KLUE Open-Domain Question Answering 대회

프로젝트 기간	담당역할	개발 언어	기타 사용 IT Tool
2021.11 ~ 2021.12	Reader model 개선을 위해 Question Generation를 이용한 Data Augmentation 개발/수행	Python, Shell script	Pytorch, HuggingFace, Elastic Search

코드: https://github.com/boostcampaitech2/final-project-level3-nlp-09
대회 개요: Open Domain Question Answering 모델 개발 및 개선
최종 성적: 2등

train data의 수가 적어서 대회에서 제공해준 Wikipedia 데이터를 활용하기 위해 Question generation을 활용했습니다. 생성된 question과 context의 title을 답으로 사용해서 Reader model을 학습시키고자 했습니다.

[Generation model]
KorQuAD-Question-Generation라는 GPT base 모델을 사용했습니다. Context와 Answer만을 사용해서 Question을 생성해주는 모델입니다.

[Wikipeida 데이터 전처리]
title이 context에 포함되지 않은 Wikipedia 데이터는 사용하지 않았습니다. title을 answer로 간주해서 question을 생성하고 해당 qusetion을 사용해 Extract-based MRC를 수행해야 하기 때문입니다.
context가 너무 길어진다면 tokenizer 단계에서 truncate를 하도록 설정했습니다. context의 길이가 너무 길어진다면 Reader model이 제대로 학습을 진행하지 못할 가능성이 높기 때문입니다.

[Generation 모델 성능 개선]
Generation 모델 자체의 성능을 개선해보기로 했습니다. 동일한 학습 데이터와 설정을 유지하고 epcoh를 변경하면서 generation 모델의 성능을 평가했습니다.
perplexity를 확인해보니 epoch가 증가할수록 성능이 급격하게 하락했습니다. epoch=5로 학습된 기본 배포 weight를 사용하기로 했다.

[Post-Processing]
질문에 정답이 들어간 Question은 제외했습니다. 정답이 포함된 질문은 의미가 없다고 판단했습니다.

[BM25 scoring]
Question generation의 context와 question 간의 BM25 score를 산출하여 의미 없는 generation 결과를 걸러내기로 했습니다.
왜냐하면 첫번째로 의미 있는 generation 결과만을 사용해야 했습니다. 의미없는 question을 학습시킬 경우 오히려 Reader model의 성능이 하락할 수 있기 때문입니다.
두번째로 Question generation의 결과가 너무 많았기 때문입니다. train data는 3700개였는데 generation data는 11007개였습니다. 이는 train data의 경향성을 generation data가 해칠 수 있을 정도의 비율입니다.

BM25 top k만을 변인으로 둬서 실험을 하고 싶었지만 대회의 빠듯한 시간 관계 상 그럴 수 없었습니다. 또 다른 변인으로 Negative sampling의 top k를 설정해서 실험을 진행하여 최적의 Trained Reader model을 찾았습니다.

네이버 부스트캠프 KLUE Relation Extraction대회

프로젝트 기간	담당역할	개발 언어	기타 사용 IT Tool
2021.109~2021.11	Relation Extraction Model 개선을 위해 Back Translation으로 Task Adaptive Pre-training 개발, 수행	Python, Shell script	Pytorch, HuggingFace

Github: https://github.com/boostcampaitech2/klue-level2-nlp-09
대회 개요: Relation Extraction 모델 개발 및 개선
최종 성적: 2등

[Overview]
1. Papago crawler를 사용해 train data에 대해 한글 → 영어, 영어 → 한글 번역을 수행
2. 변형된 train data를 사용해 klue/roberta-large를 다시 Pre-training
3. 2번에서 생성된 Pre-trained model을 팀 내에서 정해진 Best method에 적용

[Crawler]
Selenium과 chormedriver를 활용했다. User-agent는 일반 사용자의 브라우저에서 발췌해서 수정.

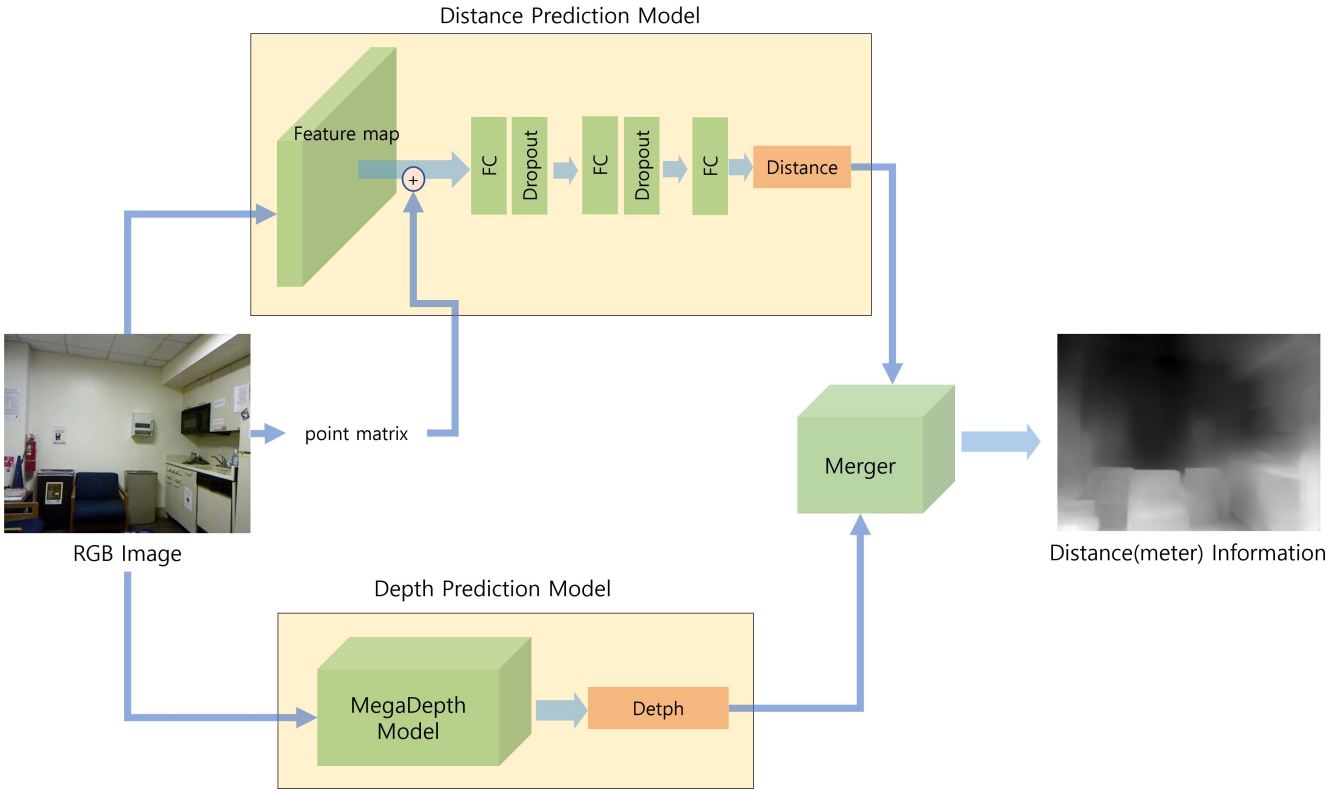
[Pre-training process]
팀 내 Best method가 RoBERTa 계열이었다. 따라서 RoBERTa의 Pre-training 방법을 사용했다.
- Dynamic masking BERT의 statical masking과 다르게 RoBERTa는 Epoch별로 Masking을 다르게 했다.
- Full sentence BERT의 NSP는 비효율적이라는 것이 RoBERTa 논문의 주장이었다. 학습 데이터에는 한 번에 여러 문장이 동시에 존재하도록 구성했다.
- Modify pytorch model dictionary Language model의 뒤에 Classifier가 붙은 HuggingFace의 Masked Language model을 불러와서 Pre-train을 진행했다. 하지만 우리 팀의 Best method는 순수한 Language model을 불러와서 학습을 진행한다. 따라서 모델이 pth 파일의 형태로 저장된 후 Best method에서 load 할 때 다음과 같이 weight, bias dictionary를 수정했다.
klue/roberta-large의 형식에 맞게 weight, bias의 key를 모두 변경한다.
Classifier에서 klue/roberta-large에 맞도록 FC layer의 weight, bias를 삭제, 수정한다.

[결과]
기존 Best method 단일 모델 F1 score: 75
TAPT with Back Translation을 Best method에 적용했을 때의 F1 score: 72

[실패 원인 분석]
- Overfitting 이미 KLUE Benchmark를 위한 데이터로 학습이 된 모델을 KLUE Benchmark 데이터로 다시 학습했기 때문에 train data에 overfitting 됐을 가능성이 있다.
- Best method 재현 실패 Best method를 재현하지 못했던 것 같기도 하다.

2D 이미지의 픽셀 단위 미터 거리 측정 졸업프로젝트

프로젝트 기간	담당역할	개발 언어	기타 사용 IT Tool
2021.03 ~ 2021.06	모델 개발, 학습/평가, 보고서 작성	Python, Shell Script	Pytorch, FastAPI, Rabbit-MQ, Docker



https://github.com/naem1023/Measuring-Image-Distance

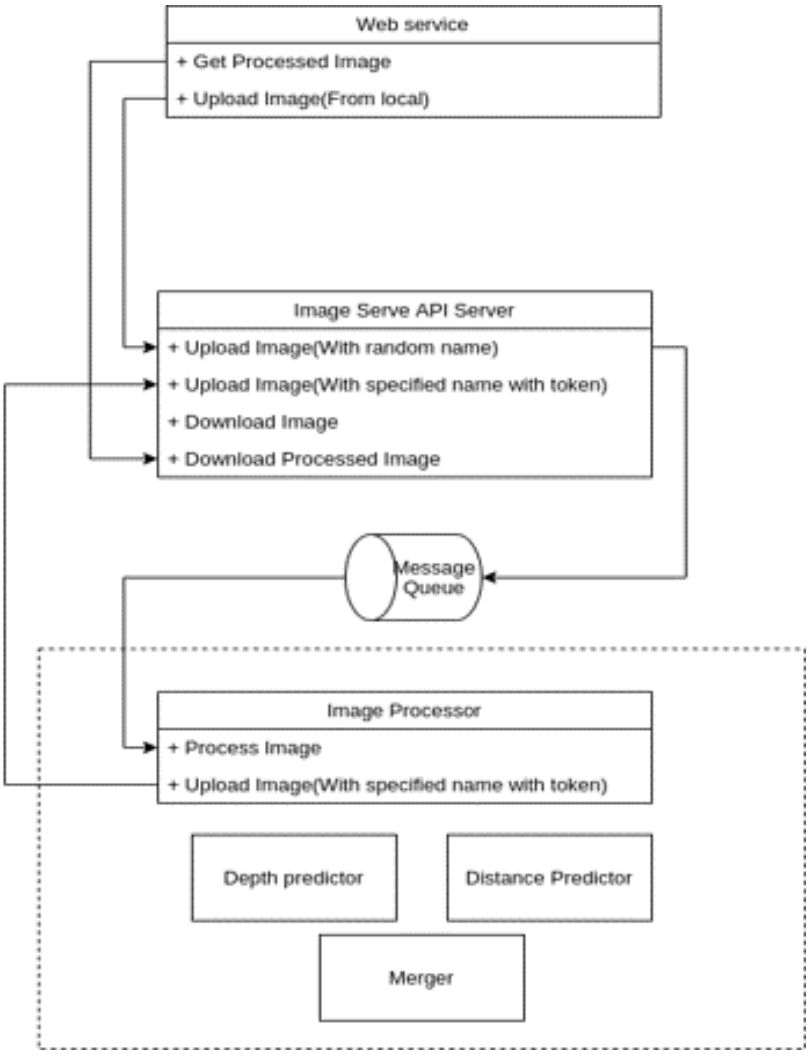
[모델 개발]
아래의 두 논문과 해당 모델들을 결합하여 개발.
Distance Predictor: Learning Object-Specific Distance From a Monocular Image
Depth Predictor: MegaDepth: Learning Single-View Depth Prediction from Internet Photos

[Predictor overview]
1. n개의 픽셀을 개별적으로 Distance Predictor에 통과시켜 n개의 픽셀에 대한 미터 단위의 거리 정보를 획득.
2. Depth Predictor로 전체 이미지에 대한 깊이 정보 획득.
3. Merger에서 1번, 2번의 정보를 조합.
4. 전체 픽셀에 대한 미터 단위의 거리 정보 획득.

- Test dataset 평가지표: RMSE 2.39

2D 이미지의 픽셀 단위 미터 거리 측정 졸업프로젝트

프로젝트 기간	담당역할	개발 언어	기타 사용 IT Tool
2021.03 ~ 2021.06	모델 개발, 학습/평가, 보고서 작성	Python, Shell Script	Pytorch, FastAPI, Rabbit-MQ, Docker



Front-end Web service: React
Image Serve API Server: FastAPI
- 원본 이미지를 업로드, depth 정보가 담긴 이미지르 다운로드
- Message Queue로 사진 전송

Message Queue: RabbitMQ
Image Processor
- Message Queue로부터 이미지를 획득.
- Predictor 호출.
- API Server로 depth 정보가 담긴 이미지 반환

네이버 부스트캠프 Mask Image Classification 대회

프로젝트 기간	담당역할	개발 언어	기타 사용 IT Tool
2021.08 ~ 2021.09	전처리, Augmentation/TTA 개발, Hyperparameter Tuning	Python	Pytorch, Albumentation

코드: https://github.com/naem1023/boostcamp-pstage-image
대회 회고: https://velog.io/@naem1023/%EC%B2%AB%EB%B2%88%EC%A7%B8-Ai-Competition-%EB%A7%88%EB%AC%B4%EB%A6%AC

대회 개요: Mask, Gender, Age 정보에 따라서 18개의 label로 이미지를 Classification하는 대회
최종 성적: 5등

[Backbone model]
실험을 위한 모델(Resnet18, efficientnet-b2)과 검증을 위한 모델(Efficientnet-b7)을 분리. 최종적으로 Efficientnet-b7을 사용.

[Transformation]
강한 augmentation을 통해서 robust한 모델을 구성.

[Label 기준 변경]
Label의 기준을 변경해서 Age feature의 class 불균형 문제를 해결.
- 30세~60세 그룹의 데이터 수를 늘리기 위해 30세가 아닌 29세를 기준점으로 삼음
- 60세 이상에 해당하는 그룹의 데이터 수를 늘리기 위해 60세가 아닌 59세를 기준점으로 삼음

[Feature 분할]
Feature 별로 Classification을 수행하는 모델을 개발.
- Mask, Age, Gender 간의 상관관계, 인과관계가 없다고 판단. 따라서 feature 별로 별도의 모델을 학습.

[구성 방법]
- train, validation set에 동일한 사람이 존재하지 않도록 구성. 학습 과정에서 이미 봤던 사람의 경우, validation에서 쉽게 예측할 가능성 방지.
- train, test set이 동일한 class 분포를 형성하도록 구성.

[CutMix]
- CutMix를 직접 구현해 CutMix를 위한 loss 계산을 수행.

[TTA]
- Albumentation을 활용해 Test Time Augmentation. Soft voting으로 구현.