

Easy Diffuser 중간 보고서

박찬호, 허찬용



필수 환경

Clone this model repository

HTTPS

SSH

Make sure you have git-lfs installed (https://git-lfs.com)

```
git lfs install
```

```
git clone https://huggingface.co/leeyunjai/img2txt
```

if you want to clone without large files - just their pointers

prepend your git clone with the following env var:

```
GIT_LFS_SKIP_SMUDGE=1
```

Copy

brotli==0.7.0
charset-normalizer==3.1.0
colorama==0.4.6
contourpy==1.0.7
cycler==0.11.0
fonttools==4.39.3
fsspec==2023.5.0
huggingface-hub==0.14.1
kiwisolver==1.4.4
matplotlib==3.7.1
mkl-fft==1.3.6
mkl-service==2.4.0
mpmath==1.2.1
numpy==1.23.5
packaging==23.1
Pillow==9.4.0
pyparsing==3.0.9
python-dateutil==2.8.2
PyYAML==6.0
regex==2023.5.5
requests==2.30.0
six==1.16.0
tokenizers==0.13.3
torch==2.0.1
torchaudio==2.0.2
torchvision==0.15.2
tqdm==4.65.0
transformers==4.28.1
urllib3==2.0.2

Quick Start

```
import easy_diffuser

easy_diffuser=easy_diffuser.start("./image1.jpg")

easy_diffuser.run()

easy_diffuser.print_caption()

easy_diffuser.send2ui()
```

내부 구현을 추상화 한 API

`Easy_diffuser.start("이미지 주소")`: `easy_diffuser`를 위한 모델이 생성되고 이미지가 삽입된다.

`Easy_diffuser.run()`: 이미지에 따라 `pos_prompt`, `neg_prompt` 가 생성이 된다.

`Easy_diffuser.print_caption()`: 생성된 `prompt`들을 확인할 수 있다.

`Easy_diffuser.input_img("이미지 주소")`: 삽입하고자 하는 이미지를 재설정할 수 있다.

`Easy_diffuser.send2ui()`: 생성된 `prompt`를 `webui`로 전송하여 이미지를 생성한다.

Quick Start

API 구현부

```
class start():
    def __init__(self,img) :
        self.model = m.Model()
        self.img=img
        self.pos=None
        self.neg=None

    def run(self):
        self.pos,self.neg = self.model.predict(self.img)

    def print_caption(self):
        print(self.pos, self.neg)

    def input_img(self,img):
        self.img=img

    def send2ui(self):

        url = "http://127.0.0.1:7860"

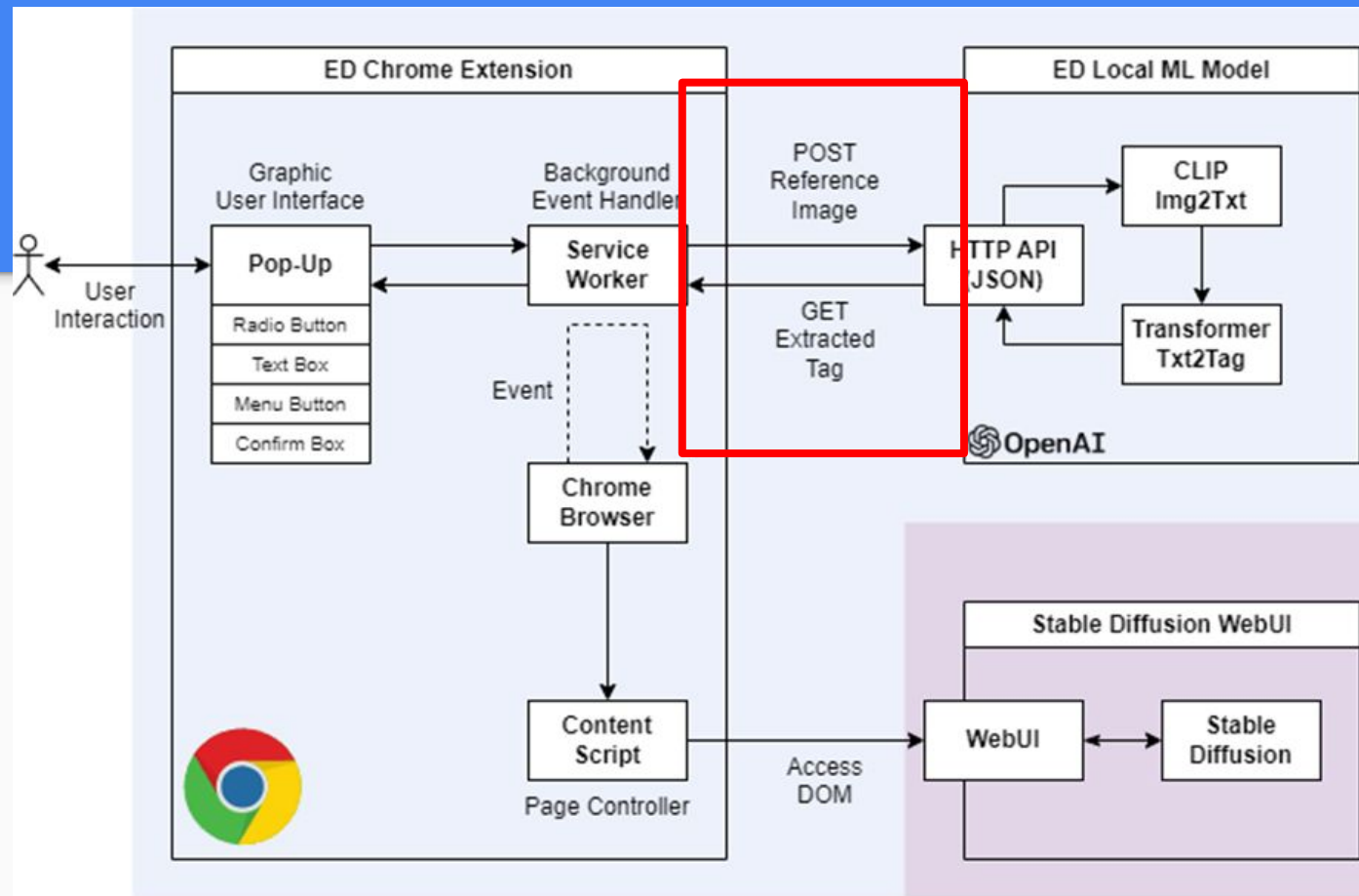
        payload = {
            "prompt": "{}".format(self.pos),
            "neg_prompt":"{}".format(self.neg),
            "steps": 10
        }

        response = requests.post(url=f'{url}/sdapi/v1/txt2img', json=payload)
        r = response.json()
        image = Image.open(io.BytesIO(base64.b64decode(str(r['images']).split(",")[0])))
        image.save('output.png')
```

아키텍처

빨간 박스 제외
구현 완료

모델 학습 중



CLIP

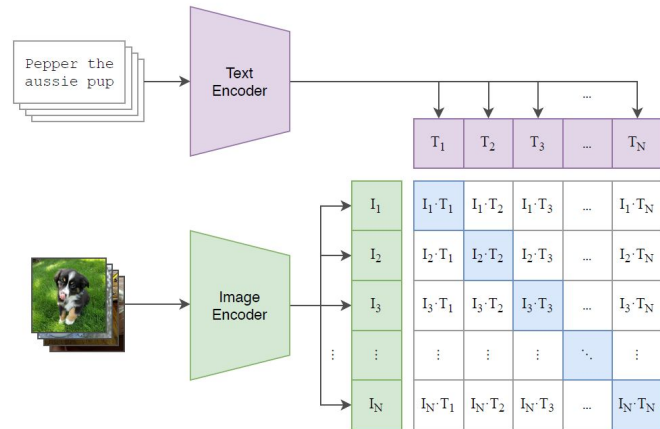
CLIP : OpenAI에서 개발

- 자연어를 supervision으로 하여 Zero-shot Transfer Learning이 가능하게 함

→ Upstream Task로 이미지를 이해 → Downstream Task에 zero shot으로 적용

- Image Encoder 와 Text Encoder를 pair로 하여 joint하게 학습
- Image Encoder – 이미지를 저차원으로 embedding (ResNet-50, ViT)
- Text – Transformer 사용

(1) Contrastive pre-training



학습 및 테스트

Positive prompt : 추출에 문제 없음

Negative prompt

- 태그의 집합에서 가장 낮은 연관성을 가지는 태그를 선택할 경우, 무의미한 태그가 추출
- 태그의 집합에서, 연관성을 가지는 부분집합 중 가장 낮은 연관성을 가지는 태그를 선택하여 **negative tag** 추출
- 연관성을 가지는 부분집합을 찾는 **threshold**를 학습 → **argmin**으로 가장 낮은 연관성을 가지는 태그 추출
- 통합 테스트에서의 **metric**으로 학습

통합 테스트 결과 평가

Metric : Binary Cross Entropy를 이용한 정량화

기존에는 **Inception Score**를 결과평가 척도로 삼았으나, 최종적으로는 **Binary Cross Entropy**를 사용하여 학습중이다. 이는, 다양성을 포함하는 **Inception Score**를 사용하지 않고, 사용자가 원하는 특성과 동일한 특성을 보유한 이미지를 생성하고, 이를 생성하는 태그를 추출하기 위함이다.

Optional : Input output 결과 비교

따라서, 사람이 느끼는 **semantic**한 특징을 읽어내기 위해 직접 **input output**을 비교한다. 테스트 코드를 사용하여 충분한 반복을 시행한 후, 유사도를 직접 매기고 이 결과를 평균 내어 모델의 결과를 평가한다.

예시 테스트 결과

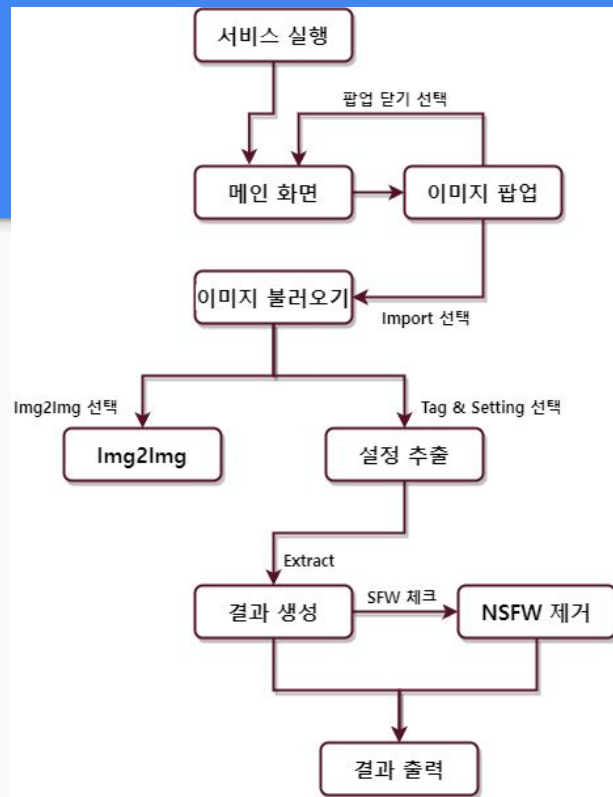
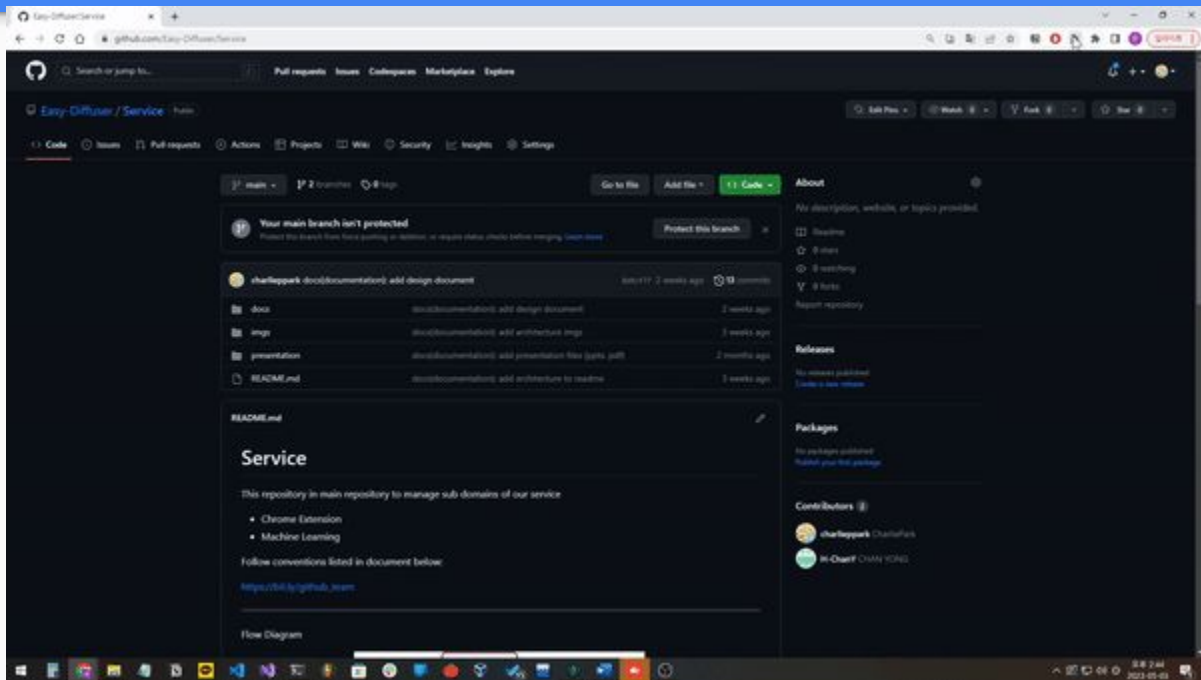


원본



생성

UI & 플로우 다이어그램



감사합니다.