

학사학위 청구논문

기억의 보관과 공유를 위한

AI 기반 추모 서비스 개발

(Development of an AI-Based Memorial Service for  
Preserving and Sharing Memories)

2023 년 12 월 22 일

승실대학교 IT 대학

전자정보공학부 IT 융합전공

박 성 호

학사학위 청구논문

기억의 보관과 공유를 위한

AI 기반 추모 서비스 개발

Development of an AI-Based Memorial Service for  
Preserving and Sharing Memories

지도교수 : 김 동 성

이 논문을 학사학위 논문으로 제출함

2023 년 12 월 22 일

승실대학교 IT 대학

전자정보공학부 IT 융합전공

박 성 호

# 박성호의 학사학위 논문을 인준함

심사위원장 신오순 (인)

심사위원 김동성 (인)

2023 년 12 월 22 일

승실대학교 IT 대학

# 목 차

표 및 그림 목차 .....	ii
국문초록 .....	iii
I. 서론 .....	1
II. Stable Diffusion 을 활용한 AI 이미지 생성 기술.....	3
II-1. Stable Diffusion .....	3
II-2. Textual Inversion .....	4
II-2. Dreambooth .....	4
II-2. Low Rank Adaptation .....	5
III. AI 이미지 생성 자동화 시스템 설계.....	6
III-1. FastAPI 를 활용한 API 설계 .....	6
III-2. 시나리오 .....	7
IV. AI 이미지 생성 자동화 시스템 제작 및 구현 .....	8
IV-1. Low Rank Adaptation 훈련 .....	8
IV-2. 이미지 생성 자동화 구축 .....	8
IV-3. AI 이미지 생성 확인 .....	9
V. 결론 .....	12
참고문헌 .....	13

## 표 및 그림 목차

그림 1.1 Diffusion Model 과 Stable Diffusion 비교.....	1
그림 2.1 Stable Diffusion Architecture .....	3
그림 2.2 Text embedding 과 Tokenization .....	4
그림 2.3 Dreambooth .....	5
그림 2.4 LoRA .....	5
그림 3.1 플랫폼 구조 .....	7
그림 4.1 프롬프트화된 이미지 파일 .....	8
그림 4.2 AI 이미지 자동 생성 시스템 code .....	9
그림 4.3 AI 이미지 생성 .....	10
그림 4.4 AI 종합 서비스 서버 반환 및 ControlNet 적용 code .....	11

국문초록

# 기억의 보관과 공유를 위한 AI 기반 추모 서비스 개발

전자정보공학부 IT 융합전공 박 성 호

지도교수 김 동 성

Stable Diffusion이란 잠재 확산 모델을 이용한 고해상도 이미지 합성 연구를 기반으로 개발된 딥러닝 인공지능 모델이다. 가우스 노이즈를 사용하여 이미지를 인코딩하며, 노이즈 예측기와 역방향 확산 프로세스를 함께 사용하여 이미지를 재현한다.

본 논문에서는 Stable Diffusion을 활용한 AI 이미지 자동 생성 시스템에 대해서 제시한다. 사전 준비된 이미지로 LoRA를 훈련시켜 원하는 화풍을 만들고, 이미지를 생성시 임베딩을 추가하여 실제 촬영한 사진을 변형시켜 실시간으로 텍스트, 음성과 함께 제공하는 방식을 제안한다. Nginx를 활용한 웹 서버 구축과 FastAPI를 활용한 API 설계를 통해 AI 이미지 자동 생성 시스템을 구현하고 시나리오를 설정한다. AI 이미지 자동 생성을 통한 자동화 시스템의 연구 결과 및 구현 결과를 제시한다.

## I. 서론

최근 AI 이미지 생성에 대해 관심이 커지며 GLIDE, Imagen, DALL.E2 등의 Diffusion Model 들이 많이 생겨나고 있다. 이 확산 모델들은 공통적으로 이미지 픽셀값을 직접 예측해야 해 저화질 이미지만 생성 가능하여 추가적으로 Upscale 용 모델을 적용해 고화질 이미지를 만들어야 한다는 단점이 존재한다.

Stable Diffusion 은 기존 확산 모델의 단점을 개선시켜 Auto Encoder 를 사용해 이미지 공간이 아닌 잠재 공간으로 이미지를 압축 후 Latent Embedding 을 예측함으로써 훨씬 더 빠른 연산속도를 가진다.

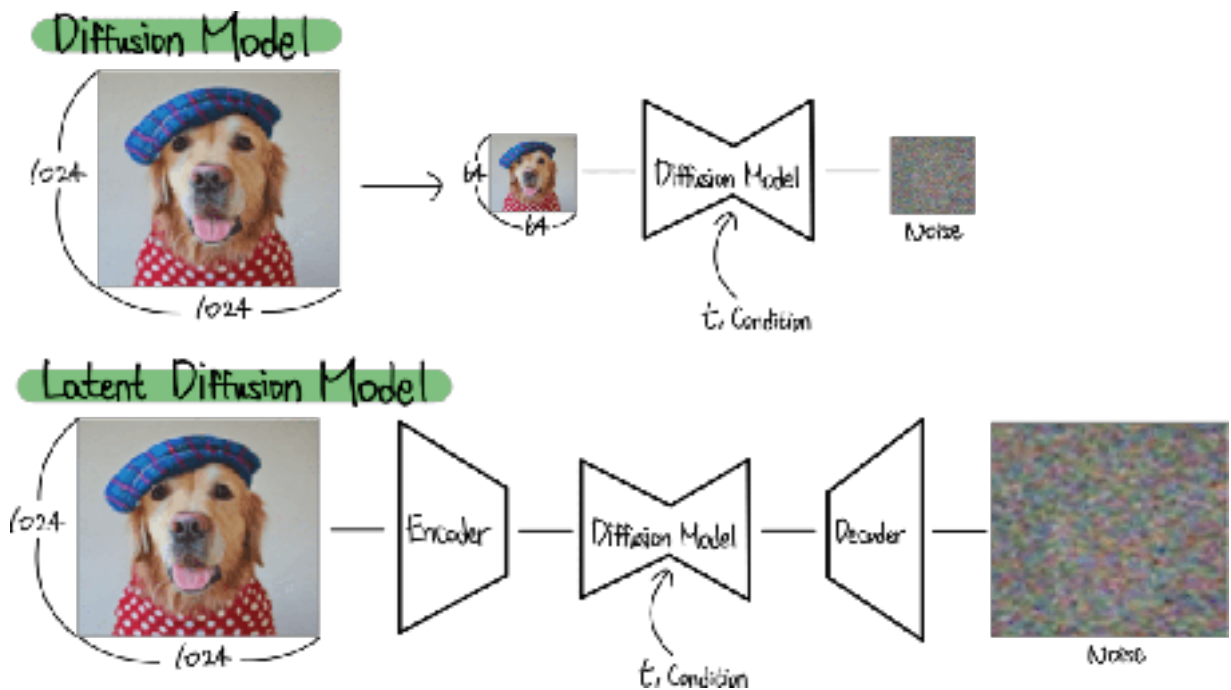


그림 1.1 : Diffusion Model 과 Stable Diffusion 비교

본 논문에서는 Stable Diffusion 을 활용해 실제 촬영된 사진을 원하는 화풍으로 자동 생성하는 시스템을 구축한다. 2 장에서는 Stable Diffusion 을 활용한 AI 이미지 생성 기술에 대해서 기술하고, 3 장에서는 AI 이미지 생성 자동화 시스템 설계 및 시나리오를 제안한다. 4 장에서는 AI 이미지 생성 자동화 시스템의 결과를 분석하고, 마지막으로 5 장에서는 본 시도를 통해 이 기술의 발전 방향을 제안한다.



## II. Stable Diffusion 을 활용한 AI 이미지 생성 기술

### II-1. Stable Diffusion

Stable Diffusion 은 VAE(Variational AutoEncoder), 가변 자동 인코더 기법을 사용해 실제 차원보다 낮은 차원 공간의 표현으로 압축하고 복원한다. Text Prompt 에 포함된 단어들을 토큰화하여 벡터값인 임베딩으로 변환시키고, 텍스트 변환기가 임베딩을 노이즈 예측기가 사용 가능하도록 처리한다. 잠재 공간에서 Latent noise 를 생성해 손상시킨 이미지에 임베딩을 추가하고 잠재 잡음을 제거해 원하는 이미지를 생성하도록 유도한다. 본 논문에서는 Img-to-Img 를 사용하므로 Input 이미지를 토큰화해 기본 임베딩으로 사용한다. 마지막으로 사전 설정된 횟수만큼 잡음 제거된 잠재 이미지를 VAE 의 디코더를 통해 픽셀로 만들어 이미지화한다.

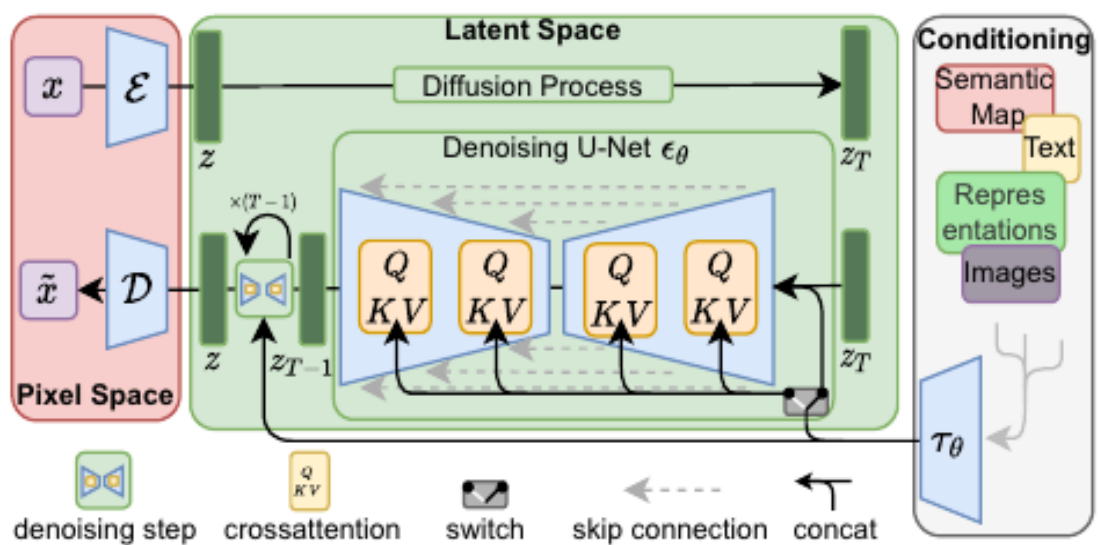


그림 2.1 : Stable Diffusion Architecture

## II-2. Textual Inversion

텍스트 반전은 Stable Diffusion 의 나머지 구성 요소들을 고정한 채 새롭게 추가된 3~5 장의 이미지에서 새로운 임베딩을 추출한다. 원하는 화풍을 만들기 위해서 지브리 스튜디오의 이미지들을 크롤링하여 준비하고, 텍스트 반전을 실행해 ckpt 파일을 생성해 임베딩을 추가했지만 원하는 화풍이 나오지 않았다. 모델을 직접적으로 변화시키는 것이 아닌 임베딩만을 추가하는 것이기에 이미지의 스타일을 변화시킬 수 없었다.

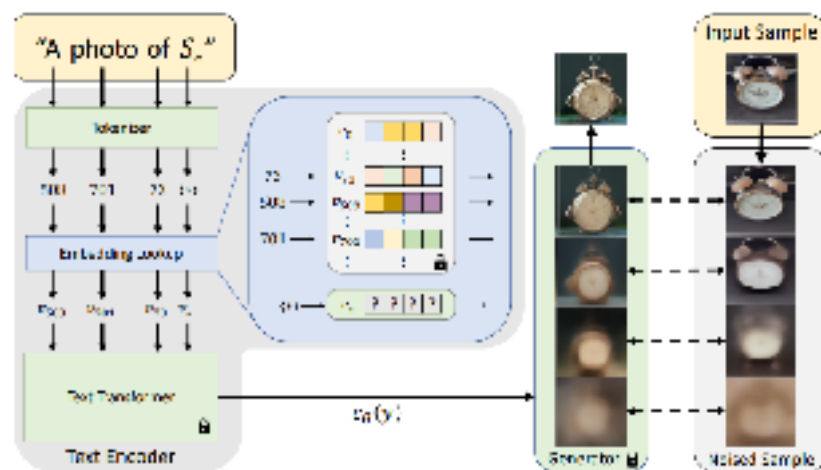


그림 2.2 : Text embedding 과 Tokenization

## II-3. Dreambooth

Dreambooth는 텍스트 반전과 다르게 전체 모델을 미세 조정하여 Input 이미지를 학습하고, 동시에 기존 모델로도 Class 이미지를 학습하여 기존 모델의 Overfitting을 막는다. 다만 단점으로는 모델을 조정하기 때문에 시간이 오래 걸리고, Gb 단위의 모델을 생성시킨다.

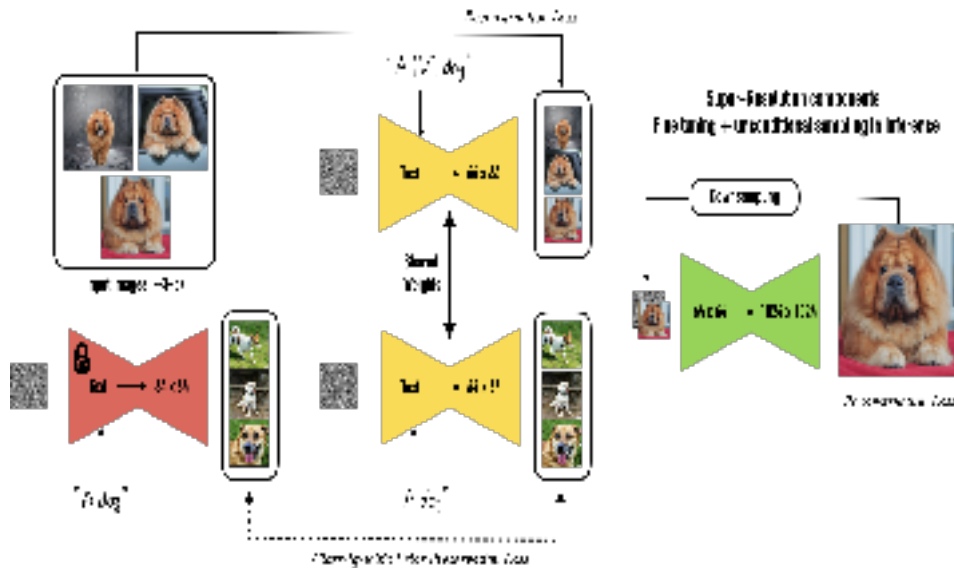


그림 2.3 : Dreambooth

## II-4. Low Rank Adaptation

LoRA 란 LLM(Large Language Models)을 Fine-tuning 할 때 시간과 용량을 줄이기 위해 만들어진 방법이다. 사전 훈련된 가중치는 동결시키고, Low Rank 로 분해된 가중치만 학습시켜 기존 가중치에 추가하기 때문에 매우 효율적이다. 또한 기존 모델의 끝에 추가하던 방식과 달리 새로운 레이어에 추가하기 때문에 Inference 대기시간이 생기지 않는다. 따라서 본 연구에서는 LoRA 를 사용하여 스타일을 변화시켰다.

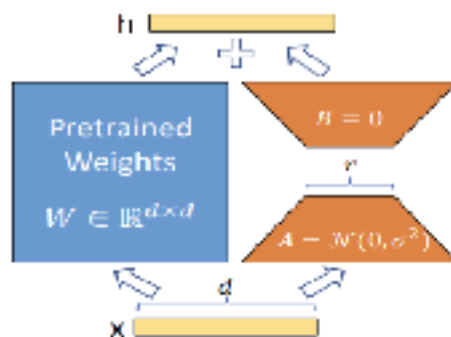


그림 2.4 : LoRA

### III. AI 이미지 생성 자동화 시스템 설계

#### III-1. FastAPI 를 활용한 API 설계

API(Application Programming Interface)는 정의 및 프로토콜 집합을 사용하여 두 소프트웨어 구성 요소가 서로 통신할 수 있게 하는 메커니즘이다. SOAP API, RPC API, Websocket API, REST API 중 본 논문에서는 REST API 방식을 선택하였다.

REST(Representational State Transfer)는 클라이언트가 서버 데이터에 액세스하는 데 사용할 수 있는 GET, PUT, DELETE 등의 함수 집합을 정의한다. 클라이언트와 서버는 HTTP 를 사용하여 데이터를 교환한다.

FastAPI 는 RESTful API 를 개발하기 위한 Python 기반 웹 프레임워크이다. Uvicorn ASGI Server(Asynchronous Server Gateway Interface)를 사용하여 빠르고 비동기 처리(async/await)가 가능한 것이 장점이다. 또한 Pydantic 을 사용해 데이터 검증 및 스키마 정의가 가능해 API 문서를 자동으로 생성해준다.(OpenAPI) 본 연구의 AI 이미지 생성에 사용되는 Stable Diffusion web UI 는 FastAPI 기반이므로 이미지 생성을 자동화하기 위해 API 형태로 설계하였다.

## III-2. 시나리오

본 연구의 목표는 실제 인물을 촬영해 로컬 웹 서버로 보내고, 받은 Input 이미지와 AI 이미지 생성을 위한 명령어, Prompt, 추가 기능들을 AI 이미지 생성 서버로 보내고, 생성된 이미지 파일을 AI 종합 서비스 서버로 반환하는 것이다. 순서는 다음과 같다.

1. Nginx 를 활용해 촬영한 Input 이미지를 받을 로컬 웹 서버를 구축한다.  
이후 촬영한 Input 이미지를 받는다.
2. Python 웹 프레임워크 FastAPI 를 활용해 API 를 구축하여 AI 이미지 생성 서버로 Input 이미지와 Prompt, 추가 기능 명령어들을 보내어 AI 이미지를 생성한다.
3. 생성된 AI 이미지를 로컬 웹 서버에 반환받고 이미지 파일로 변환해 저장한다.
4. 저장된 이미지 파일을 AI 종합 서비스 서버로 반환한다.

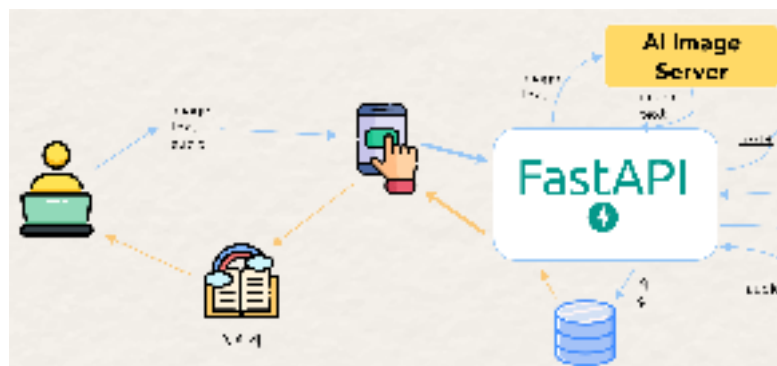


그림 3.1 : 플랫폼 구조

## IV. AI 이미지 생성 자동화 시스템 제작 및 구현

### IV-1. Low Rank Adaptation 훈련

원하는 화풍을 만들기 위해서 LoRA 를 훈련시켜 모델에 추가할 가중치를 얻어야 했다. 따라서 기존에 크롤링해놓은 데이터를 프롬프트화하여 LoRA 학습에 사용했다. 200 개의 이미지를 배치 사이즈 4, 학습률 0.0001, Optimizer 는 AdamW8bit, Rank 는 8 로 훈련시켜 LoRA ckpt 파일을 얻었다.

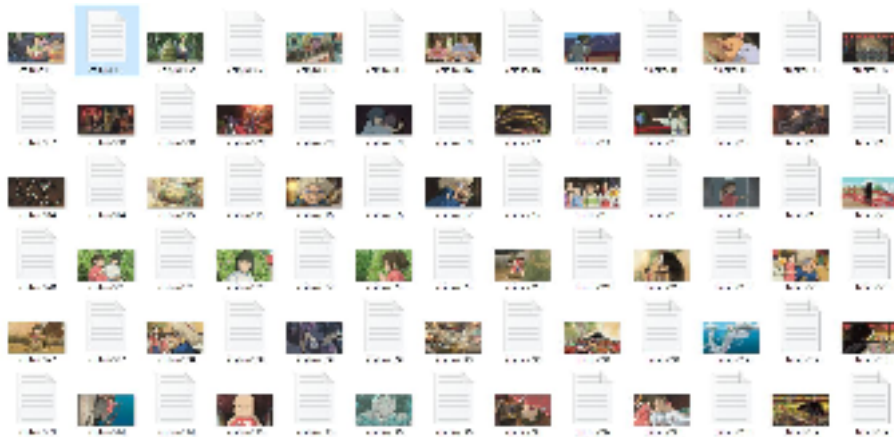


그림 4.1 : 프롬프트화된 이미지 파일

### IV-2. AI 이미지 생성 자동화 구축

AI 이미지 자동화 생성 시스템을 만들기 위해 API 형태로 구축해 클라이언트 호출이 가능해야 했다. 본 연구에서는 FastAPI 를 활용해 실제 촬영된 이미지를 로컬 웹 서버에 받아 Prompt 와 추가 기능 명령어들과 함께 POST Method 를 사용해 AI 이미지 생성 서버로 보낸다. 이미지 생성 서버에서는 Stable Diffusion 을 활용해 새로운 AI 이미지를 생성하고, 생성된 이미지를 다시 로컬 웹 서버로 반환받고 이미지 파일로 변환해 저장한다.

```

run: fastapi --reload --port 8080
run uvicorn main:app --port 8080
from fastapi import FastAPI, Request
from typing import Dict, List, Optional

app = FastAPI()

@app.get("/")
def root():
    return {"message": "Hello World"}

@app.post("/generate")
def generate_image(request: Request):
    data = request.json
    prompt = data.get("prompt", "")
    style = data.get("style", "realistic")
    model = data.get("model", "stable_diffusion")

    # Generate image using the specified model and prompt
    image = generate_image(prompt, style, model)

    # Save the generated image to a file
    image_path = f"generated_images/{style}/{model}/{prompt}.png"
    with open(image_path, "wb") as f:
        f.write(image)

    # Return the image path as a response
    return {"image_path": image_path}

if __name__ == "__main__":
    uvicorn.run(app, host="0.0.0.0", port=8080)

```

그림 4.2 : AI 이미지 자동 생성 시스템 code

### IV-3. AI 이미지 생성 확인

이미지 생성 결과 원하는 화풍이 제대로 구현된 것을 확인할 수 있다. 추가 기능으로 ControlNet 을 사용했다. ControlNet 이란 Stable Diffusion 모델을 제어하기 위한 신경망 모델로 이미지를 생성할 때 자세나 윤곽선 등을 참조하게 해 나온 품질의 이미지를 생성한다. Openpose 모델로 관절을 키 포인트로 뼈대를 추출해 자세를 잡고, depth-leres 모델로 사진의 depth 를 추정해 자세의 정확도를 높이고 형태를 잡고, mlsc 모델로 배경의 윤곽선을 따 생성했다.

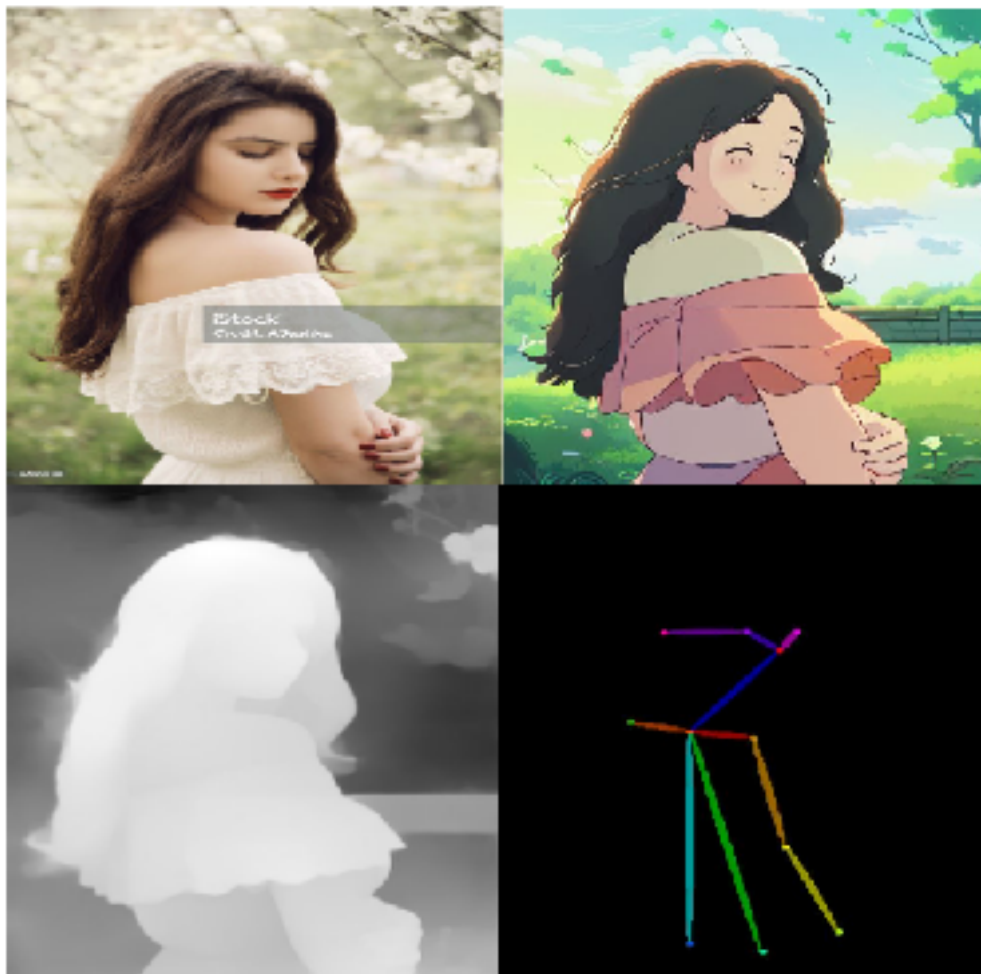


그림 4.3 : AI 이미지 생성



```

        'module': 'depth_image',
        'model': 'controlnet_cnet_depth20_480x480',
        'load_flag': True
    },

    {
        'module': 'depth_map',
        'model': 'controlnet_cnet_depth20_480x480 [9w2x2d]',
        'load_flag': True
    },

    {
        'module': 'final',
        'model': 'controlnet_cnet_depth20_480x480 [9w2x2d]',
        'load_flag': True
    }
]

response = {
    "req": [
        {
            'enable': True,
            'control_weight': 0.1,
            'temp_weight': 0.5,
            'transition_weight': 0.1,
        }
    ]
}

response = requests.post(url=url+'api/v1/req2ans', json=response)
r = response.json()

# 결과 출력스펙 변경
result = r['image'][0]
image_2 = Image.open(io.BytesIO(base64.b64decode(result.split(',')[1].strip())))
image_2.save('temp.png')
image_file = open('temp.png', 'rb')
response = requests.post(url=url+'api/v1/p_image', files={'image': image_file})
image_file.close()

```

그림 4.4 : AI 종합 서비스 서버 반환 및 ControlNet 적용 code

## V. 결 론

본 논문에서는 VAE(Variable Encoder/Decoder)를 활용한 Text embedding 방식의 Stable Diffusion 을 활용한 AI 이미지 자동화 생성 시스템을 구축하였다. Python 기반의 웹 프레임워크 FastAPI 를 활용해 API 를 구축하고 Nginx 로컬 웹 서버와 Stable Diffusion web UI, AI 종합 서비스 서버를 실시간으로 오가는 방식으로 이미지 데이터를 생성하였다. API 형태로 시스템을 설계하고 구축하였기 때문에 활용방안이 많을 것으로 기대된다.

## 참고문헌

[1] Ho, Jonathan, et al. "High-Resolution Image Synthesis with Latent Diffusion Models." arXiv preprint arXiv:2105.03831 , 2021.

[2] Lu, J., Xu, K., Feng, J., & Lapata, M. (2019). An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) , 4796-4802.

[3] Wang, Y., Chen, X., Wang, B., Zha, Z. J., Huang, Q., & Yuille, A. (2023). DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. arXiv preprint arXiv:2303.01234 , 2023.

[4] Jia, Y., Lu, S., & Wang, W. Y. (2022). LoRA: Low-Rank Adaptation of Large Language Models. arXiv preprint arXiv:2202.03476 , 2022.

[5] Ong, Shyue Ping, et al. "The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles." Computational Materials Science, 97 (2015), 209-21.

[6] AUTOMATIC1111. (2023). stable-diffusion-webui (Version v1.7.0).  
GitHub. <https://github.com/AUTOMATIC1111/stable-diffusion-webui>

[7] Illyasviel. (2023). ControlNet (Version v1.1). GitHub.  
<https://github.com/Illyasviel/ControlNet>