딥러닝을 활용한 실시간 가상 착장 시스템

Real-Time Deep Learning-Driven Virtual Try-On System

신 은 호, 이 유 진, 전 윤 호*

Eun-Ho Shin, Yu-Jin Lee, Yun-Ho Jeon*

요 약 : 본 논문에서는 온라인 소핑몰의 높은 반품률 문제를 해결하기 위한 딥러닝을 활용한 실시간 가상 착장 시스템을 제안한다. 기존 온라인 반품률이 약 30%에 달하는 문제를 해결하기 위해, CatVTON 모델을 기반으로 하여 NVIDIA jetson Orin Nano 환경에서 실시간 가상 착장 시스템을 구현했다. 모델 경량화 기술을 적용하여 전체 추론 시간을 30초에서 11초로 단축하였으며, 세그멘테이션 모델을 통해 정확한 마스크 생성을 구현하였다. 최종 시스템은 14.2초의 처리시간으로 엣지 환경에서 실용적인 가상 의류 착용 서비스를 제공할 수 있음을 입증하였다.

Abstract : This paper proposes a real-time virtual try-on system using deep learning to address the high return rate problem in online shopping malls. To tackle the issue of return rates reaching approximately 30% in online commerce, we implemented a real-time virtual try-on system based on the CatVTON model on the NVIDIA Jetson Orin Nano platform. By applying model compression techniques, we reduced the overall inference time from 30 seconds to 11 seconds and achieved accurate mask generation through a segmentation model. The final system demonstrated its practicality for virtual clothing try-on services in edge environments with a processing time of 14.2 seconds.

Key Words : VITON, Diffusion model, Jetson Nano

Ⅰ. 서 론

현재 온라인 쇼핑몰의 반품률은 약 30%로 오프라인 매장의 10%에 비해 3배 높은 수준[1][2]이다. 이러한 차이는 주로 실제 착용감을 미리 확인할 수 없어 발생하는 사이즈 불일치 문제로 인한 것으로, 물류 비용 증가와 소비자 만족도 저하의 주요 원인이 되고 있다.

이러한 문제를 해결하기 위해 VITON(Virtual Try-On Network) 기술에 대한 연구가 활발히 진행되고 있다. VTON은 사용자의 이미지와 선택한 의류를 자연스럽게 합성하여 실제 착용과 유사한 가상 이미지를 제공함으로써 구매 전 착용감을 미리 확인할 수 있도록 하는 기술이다. 특히 최근 Diffusion 모델 기반의 VITON 시스템들이 높은 품질의 합성 이미지를 생성하며 주목받고 있다.

그러나 기존의 Diffusion 기반 VITON 시스템들은 주로 서버 환경에서의 추론을 전제로 하고 있어, 사용자의 개인 이미지가 외부 서버로 전송되는 과정에서 보안과 프라이버시 문제가 발생할 수 있다. 특히 개인의 신체 이미지라는 민감한 정보를 다루는 VITON 서비스에서 이러한 문제는 더욱 심각하게 고려되어야 한다.

본 논문에서는 이러한 보안 및 프라이버시 문제를 해결하기 위해 CatVTON 모델을 엣지 디바이스에서 실행할 수 있도록 최적화한 실시간 가상 착

장 시스템을 제안한다. 모든 데이터 처리를 스마트폰 내에서 수행하는 온디바이스 컴퓨팅 방식을 채택하여 개인정보 보안을 강화하였으며, Jetson Orin Nano 플랫폼에서 최적화된 성능을 구현하였다. 특히 세그멘테이션 모델 개선, 추론 속도 향상, 메모리 사용량 최적화를 통해 제한된 자원을 가진 엣지 디바이스에서도 실시간 가상 피팅 서비스가 가능하도록 하였다.

Ⅱ. 관련 연구

2.1 Jetson Orin Nano

본 논문에서는 고성능의 실시간 가상 착장 서비스를 엣지 환경에서 제공하기 위해, NVIDIA Jetson Orin Nano[3]를 On-Device AI 컴퓨팅 플랫폼으로 활용하였다. Jetson Orin Nano는 Ampere 아키텍처 기반 GPU (CUDA 코어 1024개, Tensor 코어 32개)를 탑재하고, 최대 472GFLOPs의 AI 연산 성능을 제공한다. 또한, 7W~25W의 유연한 전력 모드를 지원하여 배터리 기반 시스템에서도 효율적으로 동작할 수 있다.

JetPack SDK(버전 6 이상)를 기반으로 CUDA, cuDNN, TensorRT 등의 소프트웨어 스택을 구성하였으며, CatVTON 모델 구동을 위한 환경으로 Python 3.10, PyTorch 소스 빌드, ARM 아키텍처용 OpenCV 등을 포함한 Conda 및 venv 기반 구성을 적용하였다. 특히, 모델 경량화 및 최적화를 통해 기존 30초 이상의 추론 시간을 Jetson Orin Nano 환경에서 11초까지 단축함으로써, 실제 서비스에 적합한 경량 고속 추론 시스템을 실현하였다.

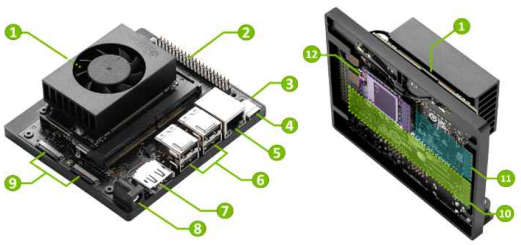

그림 1. Jetson Orin Nano 개발자 키트.
Fig. 1. Jetson Orin Nano Developer Kit.

2.2 Segmentation

가상 착장 시스템의 정확도는 사용자의 신체 부위를 정밀하게 구분하는 Segmentation 성능에 크게 의존한다. 본 논문에서는 ResNet-50을 백본으로 사용하는 U-Net 형태의 세그멘테이션 네트워크를 설계하였다. 이 구조는 입력 이미지를 인코딩-디코딩 형태로 처리하며, 사용자 인체의 주요 부위를 픽셀 단위로 구분하는 마스크를 출력한다. 출력은 채널 수가 1인 단일 마스크 형태로 제공되며, 시그모이드(Sigmoid) 함수를 출력 활성화 함수로 사용하여 각 픽셀의 클래스 확률을 0과 1 사이로 정규화한다. 이러한 구조는 정확한 인체 마스크 추출을 가능하게 하며, 후속 가상 의류 합성 과정의 품질을 크게 향상시킨다.

2.3 Diffusion

최근에는 Diffusion 모델[4]이 이미지 생성 및 조작 분야에서 높은 성능을 보이며, 가상 착장 시스템에도 응용되고 있다. Diffusion 모델은 이미지에 점진적으로 노이즈를 주입한 후 이를 복원하는 과정을 통해 고품질 이미지를 생성하는 방식으로, 기존 GAN 기반 기법보다 안정성과 표현력이 뛰어나다. 예를 들어, Stable Diffusion과 같은 모델은 사용자 이미지와 의류 조건을 바탕으로 자연스러운 가상 착장 이미지를 생성하는 데 사용될 수 있다.

2.4 Virtual Try-on Network

VITON(Virtual Try-On Network)[5]은 사용자의 이미지와 원하는 의류 이미지를 입력으로 받아 착장된 이미지를 출력하는 시스템을 의미한다. 초기 VITON 연구에서는 2단계 구조를 사용하여 먼저 사용자와 옷의 위치를 정렬하고, 이후 합성을 수행했다. 대표적인 모델로는 VITON[5]과 그리고 HR-VITON[6]이 있으며, 해당 모델들은 의류의 질감 유지와 신체 왜곡 최소화를 위해 다양한 정합 알고리즘과 합성 네트워크를 도입했다.

다양한 VITON 모델의 성능을 평가 비교하여 Jetson Orin Nano 키트에 올릴 모델을 선택하였다. 표 1은 VITON-HD[6] 데이터셋에서 주요 가상 피팅 모델을 NVIDIA RTX 4090 환경에서 벤치마크한 결과이다. 입력 해상도는 1024×768, batch size는 1로 고정하였다. 평가 지표는 추론 시간으로 삼았다.

표 1. VITON 모델 벤치마크 결과
Table. 1. Benchmark of the VITON Model

사용한 모델	모델 이미지	의상	결과물	추론 시간
HR-VITON [6]				0.125s
StableVITON [7]				35s
IDM-VTON [8]				5s
CatVTON [9]				11s
Leffa [10]				6s

이와 같은 벤치마크 결과, CatVTON은 실시간 처리에 요구되는 속도와 한정된 메모리 환경에서도 안정적인 가상 피팅 품질을 제공하여 서비스 적용에 가장 적합한 모델로 판단되었다.

2.5 CatVTON

CatVTON(Cascaded Try-On Network)[9]은 기존 VITON 시스템의 성능 한계를 개선하기 위해 제안된 모델로, 세분화된 단계별 네트워크 구조와 세그멘테이션 마스크 기반 제어를 통해 더욱 정밀한 합성을 구현한다. 특히 CatVTON은 상의와 하의, 인체 부위별 구간에 따라 단계적으로 의류를 입히며, 합성된 이미지의 자연스러움을 극대화한다. 또한 최신 Diffusion[4]모듈이나 StyleGAN[11] 기반 기술과도 결합될 수 있어, 고품질의 실사형 착장 이미지 생성을 가능케 한다. CatVTON의 전체 구조는 그림 3와 같다.

CatVTON은 인물 이미지와 의류 이미지를 하나의 VAE Encoder-UNet 구조를 통해 동시에 처리함으로써, 기존에 사용되던 텍스트 인코더, Reference U-Net, CLIP, DINOv2와 같은 외부 모듈을 제거하였다. 이로 인해 전체 파라미터 수와 연산량이 크게 줄어들었으며, 별도의 외부 네트워크 없이도 U-Net 기반의 denoising 단계만으로 고품질의 합성이 가능해졌다. 이러한 구조적 단순화는 추론 속도를 획기적으로 향상시키고, 메모리 사용량을 대폭 줄여 모바일이나 임베디드 환경에서의 활용 가능성을 높였다.

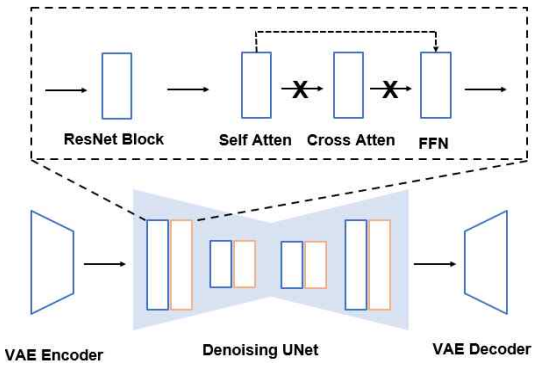

그림 2. CatVTON 전체 구조.
Fig. 2. Overall architecture of CatVTON.

VITON 모델은 연산량이 많고 구조가 복잡하여, Jetson Orin Nano와 같은 엣지 디바이스에서 추론 시 시간이 과도하게 소요된다. 정밀도 FP16에 해상도 512×384, 이미지 1장 생성에서 실제 측정된 단계별 추론 시간은 아래 표 2와 같다.

표 2. Jetson Orin Nano CATVTON 모델 단계별 추론 시간
Table. 2. Step-by-Step Inference Time of the CATVTON Model on Jetson Orin Nano.

처리 단계	시간 (초)
이미지 전처리	0.006초
VAE Encoder	0.06초
Unet denoising (50step)	29.5초
VAE Decoder	0.05초
Safety Checker	0.34~0.36초

실험 결과, 단일 추론에 30초 이상 소요되어 사용자 응답성이 떨어지며 시스템 활용도와 사용자 만족도를 저하시킬 수 있다. 특히 Unet denoising이 전체 시간의 대부분을 차지하여 병목을 형성하고 있다.

Ⅲ. 본론

3.1 제안 방법

최근 엣지 디바이스를 기반으로 하는 다양한 인공지능 서비스가 활발히 개발되면서, 제한된 연산 자원 내에서 효율적으로 동작할 수 있는 경량화 모델의 필요성이 대두되고 있다. 특히, Jetson Orin Nano와 같은 임베디드 AI 플랫폼을 활용한 응용에서는 연산 효율성과 리소스 소비의 최적화가 핵심 요구사항으로 떠오르고 있다. 본 논문에서는 반복적인 추론 요청에도 안정적으로 대응할 수 있는 모델 구조를 설계하고, 연산 속도를 개선함으로써 실시간 응답성을 확보하는 것을 주요 목표로 삼는다. 이를 위해 다음과 같은 구체적인 경량화 목표를 설정하였다.

- 전체 추론 시간을 약 30초에서 10초 이내로 단축
- 모델의 파라미터 수 및 메모리 크기를 기존 대비 50% 이상 축소
- 이미지 출력 품질은 기준 모델 대비 유사한 수준으로 유지

이러한 목표는 모델 구조의 경량화 및 연산 최적화를 통해 달성하였으며, 이후 항목에서 제안하는 모델 크기 최적화 방안, 추론 속도 향상 기법, 그리고 경량화 모델의 성능 분석을 통해 구체적인 방법론과 실험 결과를 상세히 제시한다.

3.2 Segmentation 모델 성능 개선

CatVTON 모델은 의상 이미지를 적용하기 위해 사람 이미지와 해당 인물의 의상 위치를 나타내는 마스크를 입력으로 필요로 한다. 본 논문에서는 사람 이미지를 입력으로 받아 해당 마스크를 예측하는 모델을 사용하여 이러한 역할을 수행하였다.

3.2.1 마스크 생성 모델 설계

Segmentation models pytorch 라이브러리를 이용해 U-Net 기반 Segmentation model을 정의하였다. backbone으로는 ResNet-50을 사용하였다. 출력 채널은 1, 활성화 함수는 시그모이드 함수로 적용한다. Dice Loss는 mask 품질을 측정하는 손실함수로 클래스 불균형에 강인한 특징이 있다. IoU 평가 지표는 예측 마스크와 정답 마스크의 겹치는 정도를 평가하는 지표로 정답 마스크는 VITON-HD의 agnostic mask에 해당한다.

VTION-HD 데이터셋에 맞춰 정의한 모델과 손실함수를 바탕으로 학습을 진행하였으며, 이를 통해 U-Net segmentation model을 효율적으로 학습 및 검증하면서 검증 성능이 향상될 때마다 자동으로 최적의 모델을 저장할 수 있다.

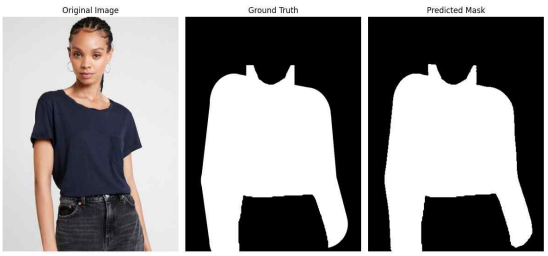

그림 3. VITON-HD 기반 segmentation 결과.
fig 3. Segmentation results based on VITON-HD.

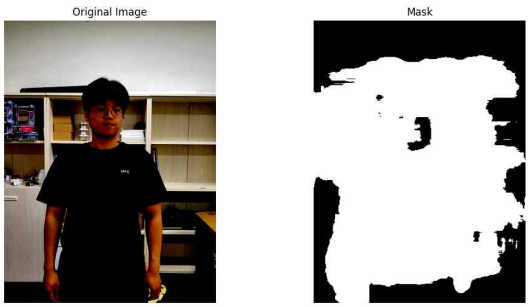

그림 4. 실제 촬영 이미지 기반 segmentation 결과.
fig 4. Segmentation results on images captured from real scenes.

3.2.2 배경 이미지 합성

그림 4과 같이 실제 환경에서 수집된 이미지로 segmentation한 경우, mask가 잘 예측되지 않는

것을 볼 수 있다. segmentation 학습 시 배경 이미지가 단조로운 VITON-HD 데이터셋으로 진행하여 발생한 문제라고 판단하였다. 이를 해결하기 위해 배경 데이터셋인 bg-20k 데이터셋[12]으로 배경을 합성하였다.

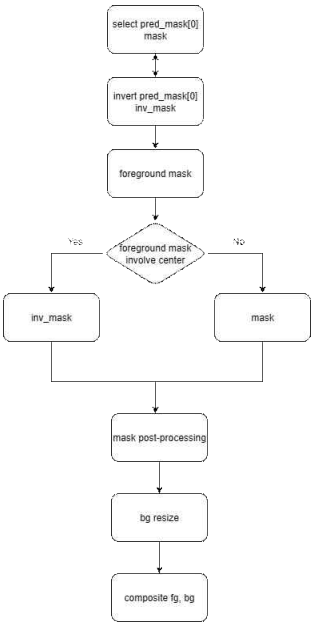

그림 5. SAM 기반 전경/배경 합성 알고리즘 순서도.

fig 5. Flowchar of SAM-based foreground/background composition algorithm.

VITON-HD 데이터셋의 인물을 전경으로 추출하기 위해 segmentation 모델인 SAM[13]을 활용하였으며, 합성 알고리즘을 반복적으로 개선하여 최종 프로세스를 확립하였다. 최종 알고리즘의 전체 절차는 그림 5의 순서도와 같이 요약할 수 있다.

그림 6. 배경 합성 결과.

fig 6. Background image synthesis.

알고리즘을 통해 생성된 합성 이미지는 그림 6과 같다.

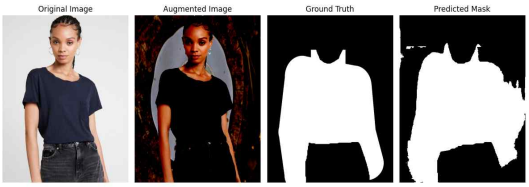

그림 7. VITON-HD 학습 segmentation 모델의 합성 이미지 추론

fig 7. Inference on synthetic images by the VITON-HD trained segmentation model.

합성한 이미지로 재학습하기 전 VITON-HD로 학습한 segmentation 모델로 추론한 결과는 그림 7과 같다. VITON-HD 데이터셋으로만 segmentation model을 학습할 경우, 복잡한 배경에서 segmentation 모델의 일반화 성능이 저하되어 결과적으로 정확한 마스크 분리에 실패한 모습을 확인할 수 있다.

3.2.3 합성된 이미지 기반 마스크 생성 모델

기존에 설계한 모델을 활용하여 합성 이미지를 학습하였다. 그러나 실제 환경에서 수집된 이미지를 그대로 사용할 경우, 입력 이미지의 종횡비 차이로 인해 비율 왜곡이 발생할 수 있다. 이를 방지하기 위해 전처리 단계에서 원본 이미지의 비율을 유지한 채 입력 크기(512×384)를 완전히 덮도록 확대 또는 축소하였다. 이후 목표 크기에 맞게 중앙을 크롭하여 최종 입력 이미지를 생성하였다. 추론 단계에서 전처리 과정을 적용하여 모델 추론을 수행하였다.

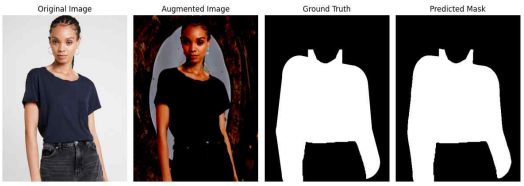

그림 8. 재학습된 모델을 활용한 합성 이미지 추론 결과.

fig 8. Inference results on synthetic image using the retrained model.

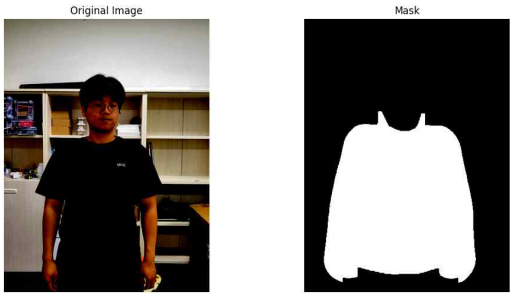

그림 9. 재학습된 모델을 활용한 실제 이미지 추론 결과.
fig 9. Inference results on real image using the retrained model.

그림 8, 9와 같이 합성된 VITON-HD 이미지와 실제 이미지 모두에서 segmentation 모델이 마스크를 정확하게 예측하는 것을 확인할 수 있다. 이를 통해 CatVTON 모델의 입력 단계에서 해당 마스크가 안정적으로 활용될 수 있음을 알 수 있다.

3.3 CatVTON 모델 최적화

CatVTON 모델의 추론 성능 분석 결과, 전체 처리 흐름에서 가장 큰 병목은 denoising 단계로 확인되었다. 해당 단계는 각 timestep마다 반복되는 U-Net 추론 + guidance 보정 + noise scheduler 연산으로 구성되며, 전체 추론 시간의 대부분을 소모한다. Jetson Orin Nano 환경에서는 이 병목을 줄이기 위한 여러 최적화 기법이 적용되었다. 추론을 위한 데이터 셋은 VITON-HD의 test을 사용하였다.

3.3.1 정밀도

Jetson Orin Nano는 FP32 연산보다 FP16 연산에 최적화된 구조를 가지고 있다. 이에 따라, 전체 모델을 BFP16(BFloat16) 모드로 변환하여 추론을 수행하였다. BFP16은 FP16과 달리 FP32와 동일한 지수 비트 수를 가져 동적 범위가 넓고, 수치적 안정성이 우수한 특징을 가진다.

3.3.2 이미지 크기

입력 이미지의 해상도는 추론 속도에 큰 영향을 미친다. 연산량은 해상도의 제곱에 비례하여 증가하므로, 적절한 해상도 설정이 성능 최적화의 핵심이다. 본 논문에서는 VITON-HD[6]테스트셋을 기반으로 다양한 해상도에서의 추론 속도와 메모리 사용량을 분석하였다.

3.3.3 Denoising 스케줄러

CatVTON 모델의 denoising 단계는 기본적으로 DDIM(Deterministic Denoising Implicit Models) 스케줄러[14]를 활용한다. 본 논문에서는 더 효율적인 DPM++(DPM-Solver++) 스케줄러[15]로 변경하고 Time Step 수를 축소하는 방법을 제안한다. DPM++ 스케줄러는 수치적 미분방정식 해법을 개선하여 적은 Time Step으로도 고품질 이미지 생성이 가능한 특징을 가진다.

3.3.4 메모리 최적화 기법

Diffusion 모델의 denoising 과정에서 다수의 iteration 수행 시 각 단계마다 intermediate activation 값들이 메모리에 누적 저장되어 OOM(Out of Memory) 문제가 발생한다. 이를 해결하기 위해 다음과 같은 메모리 최적화 기법을 적용하였다:

시스템 레벨 최적화:

- GPU 메모리 사용량을 전체의 85%로 제한
- cuDNN Conv 알고리즘 튜너 활성화

모델 레벨 최적화:

- VAE Slicing: VAE 디코더를 채널 축으로 슬라이싱 후 순차 계산
- Attention Slicing: Q, K, V를 채널 축으로 슬라이싱 후 순차 계산

3.4 실험 결과

Jenson Orin Nano 에서 각 기법에 대한 적용 결과를 실험했다. 실험 환경은 JetPack 버전 6.2 버전에 miniconda 가상환경 Python 3.10, 전원 모드는 MAXN SUPER mode[16]에서 진행했다. 모델 설정은 정밀도 BFP16, 입력 이미지 크기는 512×384, Denoising Time Step은 50 Step, Denoising 스케줄러는 DDIM으로 고정하여 비교 실험을 진행하였다.

3.4.1 정밀도 실험

Jetson Orin Nano는 FP32 연산보다 FP16 연산에 최적화되어 있다. 따라서 모델의 연산 정밀도에 따른 최적화을 수행하기 위해 모델 연산 정밀도별 추론과 메모리 사용량을 비교 실험하였다. 모델 전체를 BFP16 모드로 변환한 결과, 표 3과 같은 속도 향상이 확인되었다.

표 3. 연산 정밀도 별 추론 속도
Table. 3. Inference Speed
by Computation Precision

정밀도	총 추론 시간 (512×384 해상도, 50 step 기준)
FP32	약 1분 24초
FP16	약 30초
BFP16	약 29초

연산 정밀도가 FP32일 경우 약 1분 24초 소요됐다. FP16인 경우에는 약 30초, BFP16일 경우 약 29초로 정밀도 전환만으로도 약 62%의 속도 향상이 이루어졌으며, 그림 4와 같이 이미지 품질 저하는 거의 없는 수준이었다. 또한 모델 크기도 정밀도에 따라 변하였는데 UNet 모델 기준 FP32 연산이 3.2GB이고 FP16 연산이 1.67GB로 약 48% 감소하였다.

그림 10. 연산 정밀도 별 생성된 이미지
Fig 10. Images Generated
by Computation Precision

3.4.2 이미지 크기 실험

VITON-HD 테스트셋을 기반으로 다양한 해상도에서의 추론 속도를 비교한 결과는 표 4과 같다.

표 4. 입력 이미지 크기 별 추론 속도
Table. 4. Inference Speed
by Input Image Size

이미지 크기	총 추론 시간 (BFP16, 50 step)
1024×768	약 3분 3초
512×384	약 29초
256×192	약 10초

입력 해상도를 1024×768로 설정한 경우 약 3분 3초 추론 속도가 걸렸고 이미지 생성 중 메모리 부족으로 저장 과정에서 Jetson Orin Nano가 멈췄다. 입력 해상도를 512×384로 설정한 경우 추론 속도는 약 29초 걸렸고 입력 해상도를 256×192로 설정한 경우 추론 속도는 약 10초 걸렸다. 하지만 이미지 해상도가 256×192일 때 이미지 품질이 떨어지는 결과를 그림 5에서 확인할 수 있다. 해상도 축소가 연산량을 줄여 실질적인 속도 개선으로 이어짐을 확인하였다. 512×384 해상도에서 실용성과 품질의 적절한 균형점을 찾을 수 있었다.

그림 11. 입력 이미지 크기 별 생성된 이미지
Fig 11. Images generated by Input Image Size

3.4.3 Denoising 스케줄러 실험

DDIM 스케줄러는 512×384, BFP16, 50 Step 기준 약 30초가 소요되며, 전체 추론 성능에 큰 영향을 미친다. 이를 개선하기 위해 Time Step 수를 줄일 수 있는 스케줄러 교체를 적용하고자 DDIM 스케줄러와 DPM ++스케줄러의 비교실험을 진행했다.

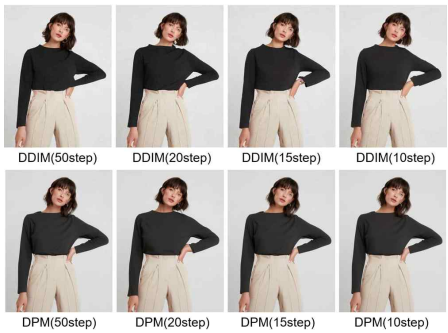

그림 12. 스케줄러와 Time Step 별 생성된 이미지
Fig 12. Generated Images by Scheduler and
Number of Time Steps

스케줄러에 따라 1 Time Step의 추론 속도의 차이점이 없다는 것을 확인하였고 시각화로 정성적 평가로 판단하면 DPM++ 스케줄러를 활용할 경우, 더 적은 Time Step으로도 DDIM과 유사한 화질과 디테일을 유지할 수 있었다. 또한 Time Step 갯수 별 추론 속도를 확인하였는데 결과는 표 5와 같다.

Table. 5. Inference Speed by Number of Time Steps

Time Step	추론 시간 (BFP16, 512×384 해상도)
50 Step	약 30초
20 Step	약 11초
15 Step	약 8초
10 Step	약 5초

이러한 변경을 통해 기존 50 Step에서 30초 걸리던 추론을, 20 Step으로 줄여 11초까지 단축 가능함을 확인하였다.

3.4.4 메모리 최적화 실험

최적화 기법을 적용하지 않은 상태에서는 모델 단독으로도 약 6~7 GB의 GPU 메모리를 사용하였다. 가상 피팅 서비스와 함께 실행할 경우 메모리 부족으로 인해 swap 메모리까지 사용되는 현상이 발생하였다.

하지만 제안한 메모리 최적화 기법을 적용한 결과, 최대 메모리 사용량은 7.4 GB로 감소하였고, swap 메모리 사용 없이 안정적인 실행이 가능해졌다. 단, 이 과정에서 10 Step 기준 추론 시간은 5초에서 9초로 소폭 증가하는 트레이드오프가 발생하였다.

3.5 모델 통합 및 추론 수행

본 논문에서는 다양한 모듈과 전처리 과정을 통합하여 전체 파이프라인을 구성하고, CatVTON 모델을 중심으로 통합된 시스템에서 추론을 수행하였다. Segmentation 모델과 CatVTON 모델을 유기적으로 연결함으로써 입력 이미지의 전처리부터 최정 가상 착용 이미지 생성까지 일괄적으로 처리되는 통합 환경을 구축하였다. 이러한 모델 통합 전략은 추론 과정에서의 효율성을 높이고, 엣지 환경에서 실시간 응용 가능성을 검증하였다. 결과 이미지는 그림 14에 제시되어 있다.

그림 13. 입력 이미지
Fig 13. Input image

그림 14. Segmentation 결과 및 최종 출력
Fig 14. Segmentation result and final ouput

Ⅳ. 결 론

본 논문에서 Jetson Orin Nano를 기반으로 CatVTON 기반 가상 의류 착용 시스템의 성능을 정량적으로 측정하고 분석하였다. 전체 파이프라인의 처리 시간은 약 9.2초로, Segmentation 모듈이 0.2초(1.4%)로 매우 빠른 속도를 보였으며, 메인 프로세스인 CatVTON(10 steps)이 9초(63.4%)로 전체 처리 시간의 가장 큰 비중을 차지하였다. 엣지 컴퓨팅 환경에서 본 시스템은 9.2초의 전체 처리 시간을 통해 상용 서비스에 적합한 응답성을 확보하였다. 특히 클라우드 기반 서비스 대비 네트워크 지연이 없으며, 개인 정보 보호와 오프라인 환경에서도 안정적으로 동작할 수 있다는 장점을 갖

는다.

참 고 문 헌

[1] 한국경제, perfitt 관련, url:https://www.hankyung.com/article/2024080531631.

[2] 물류신문, 반품 관련. url:https://www.klnews.co.kr/news/articleView.html?idxno=312851.

[3] NVIDIA Corporation, "Jetson Orin Nano Developer Kit - Hardware Specification," NVIDIA Developer, [Online]. Available: https://developer.nvidia.com/embedded/learn/jetson-orin-nano-devkit-user-guide/hardware_spec.html. [Accessed: Jul. 24, 2025].

[4] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," Advances in neural information processing systems, vol. 33, pp. 6840-6851, 2020.

[5] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, "Viton: An image-based virtual try-on network," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7543-7552, 2018.

[6] S. Lee, G. Gu, S. Park, S. Choi, and J. Choo, "High-resolution virtual try-on with misalignment and occlusion-handled conditions," arXiv preprint arXiv:2206.14180, 2022.

[7] J. Kim, G. Gu, M. Park, S. Park, and J. Choo, "Stableviton: Learning semantic correspondence with latent diffusion model for virtual try-on," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 8176-8185, 2024.

[8] Y. Choi, S. Kwak, K. Lee, H. Choi, and J. Shin, "Improving diffusion models for authentic virtual try-on in the wild," in European Conference on Computer Vision, pp. 206-235, Springer, 2024.

[9] Z. Chong, X. Dong, H. Li, S. Zhang, W. Zhang, X. Zhang, H. Zhao, D. Jiang, and X. Liang, "Catvton: Concatenation is all you need for virtual try-on with diffusion models," arXiv preprint arXiv:2407.15886, 2024.

[10] Z. Zhou, S. Liu, X. Han, H. Liu, K. W. Ng, T. Xie, Y. Cong, H. Li, M. Xu, J.-M. P´erez-R´ua, et al., "Learning flow fields in attention for controllable person image generation," in Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 2491-2501, 2025.

[11] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4401-4410, 2019.

[12] J. Li, J. Zhang, S. J. Maybank, and D. Tao, "Bridging composite and real: Towards end-to-end deep image matting," International Journal of Computer Vision, pp. 1-20, 2022.

[13] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment Anything," arXiv preprint arXiv:2304.02643, pp. 1-12, 2023.

[14] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," arXiv preprint arXiv:2010.02502, 2020.

[15] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models," Machine Intelligence Research, pp. 1-22, 2025.

[16] By Shashank Maheshwari and Chen Su, "NVIDIA JetPack 6.2 Brings Super Mode to NVIDIA Jetson Orin Nano and Jetson Orin NX Modules", url :https://developer.nvidia.com/blog/nvidia-jetpack-6-2-brings-super-mode-to-nvidia-jetson-orin-nano-and-jetson-orin-nx-modules/