

[Part 4] 3팀 최종 프로젝트 보고서



소상공인을 위한 광고 제작 서비스 (Fit - AD)

슬로건 : 딱 맞는, 알맞는 이미지를 제공합니다! Fit - AD.

1. 프로젝트 개요

- 목표 :** 전 분야 소상공인을 위한 생성형 AI 활용 광고 제작 서비스를 구축한다.
 - 최근 소비 트렌드가 '경험'과 '비주얼' 중심으로 변화함에 따라 요식업, 뷰티, 헬스 케어, 소매업 등 모든 업종에서 SNS 마케팅의 중요성이 그 어느 때보다 커졌다. 소상공인 창업은 활발히 이루어지고 있지만, 그만큼 동종 업계 간의 경쟁은 더욱 치열해졌다.

고객 유치를 위해 인스타그램 등 SNS 관리는 필수적인 생존 전략이 되었으나, 본업에 쫓기는 소상공인이 전문적인 콘텐츠를 직접 기획하고 제작하는 것은 엄청난 시간과 노력을 요구한다. 단순히 스마트폰으로 찍은 투박한 제품 사진이나, "문의 환영"과 같은 익숙한 문구만으로는 더 이상 소비자의 눈길을 사로잡는 경쟁력을 갖기 어렵다.

본 프로젝트의 목표는 소상공인들이 겪는 이 '마케팅 격차'를 생성형 AI 기술로 메우는 것이다. 전문 마케터나 디자이너가 아닌, 각 분야의 전문가인 사장님들이 온라인 광고 콘텐츠 생성 과정에서 겪는 비용적, 심리적 부담과 어려움을 해소할 수 있음을 기대해본다.

1.1 프로젝트 기대 효과

- 시간 비용 절감:** 콘텐츠 기획부터 디자인, 문구 작성, 촬영까지 걸리던 수 시간과 비용을 획기적으로 압축한다.
- 전문성 극대화:** 소상공인이 가진 본업의 가치(맛, 기술, 서비스 등)가 잠재 고객에게 가장 매력적인 비주얼(마케팅 콘텐츠)로 전달된다.
- 경쟁력 확보:** 1인 사업자나 골목 상권의 소규모 매장도 대형 프랜차이즈와 동등하게 온라인 시장에서 경쟁할 수 있는 무기를 갖게 된다.

1.2 프로젝트 핵심 가치

생성형 AI를 활용해 각 업종별 특성과 SNS(인스타그램 등) 규격에 맞춰 지금 바로 업로드 할 수 있는 완벽한 광고 이미지를 생성한다.

1.3 Target & Persona

- **사용자:** 홍보가 필요한 모든 분야의 소상공인 (요식업, 카페, 미용실, 네일샵, 공방, 헬스 케어 등)
- **Pain Point :**
 - 회원 유치와 매출 증대를 위해 인스타그램을 해야 하는데, 사진 퀄리티가 낮고(단순한 음식 사진, 정리되지 않은 매장 내부 등) 전문 모델이나 스튜디오를 쓰기에는 비용이 부담됨.
 - 사진 배경에 주방 집기나 박스 등 원하지 않는 지저분한 요소가 찍힘.
 - 홍보 글을 쓰려니 "오픈 이벤트 진행 중", "예약 문의 환영", "첫 방문시 모든 시술 20% 할인" 같은 뻔하고 임팩트 없는 문구만 반복하게 됨.
 - 우리 매장의 주 타겟층(예: 2030 직장인, 주부 등)이 검색하는 트렌디한 해시태그나 키워드를 모름.

1.4 프로젝트 최종 목표

- 생성형 AI를 활용해 소상공인이 별도의 가공 없이 바로 SNS에 업로드할 수 있는 고품질의 맞춤형 광고 이미지 생성.
- 전문 마케터나 디자이너가 아닌 각 분야의 소상공인들이 **온라인 광고 콘텐츠 생성으로 인해 겪는 비용적, 심리적 부담과 기술적 장벽을 해소한다.**
- 광고 제작을 위한 시간과 비용을 절감하여 소상공인이 본업의 서비스 품질 향상에 집중 할 수 있도록 돋는다.

2. 프로젝트 의의

- **Baseline → 점진적 고도화**
 - 기본 생성형 AI 서비스 웹 구축 후 기능 확장
 - 문구 생성 → T2I → I2I → Text Overlay 순서로 지속적인 기능 확대
 - 제품·기능 축을 기준으로 “기능 우선 구현 → 성능·안정성 강화” 사이클을 반복하며 서비스 완성도 고도화
- **프롬프트 엔지니어링**
 - FLUX, SDXL 모델을 대상으로 다양한 프롬프트 실험 및 구조화된 템플릿 개발
 - 정밀하고 엄격한 프롬프트 설계로 이미지 출력의 환각/왜곡/오류 최소화

- 이미지 품질 개선 및 세밀한 제어 강화를 통해 안정적인 시각 결과물 제공
- CLIP 77 토큰 상한을 고려한 프롬프트 압축·정규화 전략 도입으로 출력 안정성 향상

- **팀워크와 협업**

- 모든 팀원이 역할을 분담하여 효율적으로 기능 개발 및 테스트 진행
- 코드 리뷰 및 재확인 절차를 통해 오류를 사전에 방지하고 일관성 확보
- 지속적인 팔로우업을 통해 안정적 개발 리듬과 협업 환경 구축

- **ComfyUI 도입으로 효율적인 모델 관리**

- Diffusers 기반 단일 파이프라인 구조에서 ComfyUI 기반 모델 워크플로우로 전환
- 모델 로딩/언로드, 메모리 관리, 고비용 모델 전환 시 복잡도가 크게 감소
- gguf 기반 양자화 모델을 활용하여 GPU 메모리 효율성 극대화 및 속도 향상
- 커스텀 노드(얼굴/손 고정밀 처리 등)를 활용해 이미지 품질과 정확도 대폭 강화

- **모델 라이프사이클 운영 전략**

- FLUX/SDXL처럼 고메모리 모델 사용 상황에서 unload/캐시 관리 전략을 계획적으로 도입
- 메모리 피크를 낮추고 OOM(Out-Of-Memory) 리스크를 근본적으로 감소
- 서비스 운영 중 모델별 VRAM 사용량과 초기화 비용을 고려한 자동화 기반 확장성 확보

- **관측(Observability)·운영성 개선**

- 백엔드/프론트엔드 로그 스타일 표준화 → 디버깅 및 추적성 강화
- 최소 핵심 지표(이미지 처리 시간, VRAM/RAM 사용량, 오류 비율) 추적 체계 도입
- 향후 A/B 테스트 및 모델 교체 전략 운영을 위한 데이터 기반 로깅 기초 마련

- **확장성과 환경 이식성 강화**

- configs/*.yaml 기반 설정 중앙화로 경로, 모델 선택, 프롬프트, 폰트 등을 통합 관리
- 개발/로컬/클라우드(GCP) 간 환경 차이를 최소화하여 이식성과 유지보수성 향상
- 모델 추가, 기능 확장 시 YAML만 업데이트하면 되는 구조로 확장 비용 최소화

- **리스크 대응과 문서화**

- 완성되지 않은 기능은 명시적으로 스텝 처리하여 오작동을 차단

- 사용자/문서에 영향 범위를 명확하게 표시해 예측 가능성을 높임
 - 향후 기능 확장 시 리스크와 영향 분석을 기반으로 안정적인 배포 전략 수립
-

3. 최종 시스템 아키텍처

3-1. 페이지1 (문구 생성)

- 입력: 가게/서비스/특징/지역/톤
- 처리: FastAPI `/api/caption` → GPT 기반 문구·해시태그 생성, 비활성 시 규칙 기반 폴백
- 출력: 문구 3개·해시태그 15개, 세션 저장 후 페이지2로 전달

3-2. 페이지2 (T2I 이미지 생성)

- 입력: 사용자 프롬프트 + 페이지1 문구(보조), 사이즈/스텝/가이던스
- 처리: `/api/t2i` → ComfyUI FLUX.1-dev 워크플로우 실행, 프롬프트 v2 정규화(토큰 클램프)
- 출력: 3개 이미지, 다운로드·갤러리

3-3. 페이지3 (I2I 스타일 변경)

- 입력: 입력 이미지, 편집 프롬프트, Strength/Steps/Guidance
- 처리: `/api/i2i` → ComfyUI FLUX.1-dev I2I 워크플로우, 모델명 전달(페이로드 일치)
- 출력: 편집 결과 + 원본 비교

3-4. 페이지4 (배경 제거 + 고급 편집)

- 모드: Portrait/Product/Hybrid + FLUX Fill 실험
- 입력: 이미지, 프롬프트, `experiment_id`
- 처리: `/api/edit_with_comfyui` → 실험별 워크플로우 템플릿 로드·입력 바인딩 → BEN2/Rembg/ControlNet 조합
- 출력: 배경 제거/합성/채우기 결과

3-5. 페이지5 (캘리그라피 생성)

- 입력: 텍스트/폰트/색상/스타일

- 처리:
 - 기본: Pillow로 투명 PNG 렌더
 - 스타일: SDXL+ControlNet Depth+Rembg 파이프라인, 프롬프트 토큰 클램프 적용
 - 폰트: 기본 RiaSans-Bold.ttf + 폴백 검사
- 출력: 투명 배경 캘리그라피 PNG, 스타일별(emboss/carved/floating) 렌더

3-6. 페이지6 (슬라이더 기반 이미지 배치 시스템)

- 입력: 텍스트 이미지, 텍스트 크기(W/H), 위치 좌표(L/T), 배경 이미지
- 처리:
 - 모든 UI 조작은 슬라이더 기반 정밀 조절 시스템 제공
 - 조절된 값들은 Streamlit session_state로 자동 저장
 - 설정된 좌표, 크기 값을 기반으로 미리보기 800px로 축소한 캔버스에서 PIL로 실시간 렌더
- 출력: 최종 합성 원본 해상도(업로드 이미지 기준)로 정확히 재랜더링

3-7. 전체

- 프론트: Streamlit(8501) → 폼·세션·갤러리
- 백엔드: FastAPI(8000) → 비즈니스 로직, ComfyUI 클라이언트
- 엔진: ComfyUI(8188) → 워크플로우 오케스트레이션(GGUF, ControlNet, BEN2, Rembg)
- 공통: 설정 중앙화(configs.yaml), 로그(logs)

4. 기술 스택

4.1 프론트엔드

- **Streamlit** (1.28+): 웹 UI
- **PyYAML**: 설정 관리
- **Pillow**: 이미지 처리

4.2 백엔드

- **FastAPI** (0.104+): REST API
- **Pydantic**: 데이터 검증
- **OpenAI API**: 텍스트 생성

4.3 AI/ML

- **ComfyUI**: 이미지 생성 오클리스터레이션
- **FLUX.1-dev** (양자화): 고속 T2I/I2I
- **FLUX.1-Fill** (양자화): 편집 시 이미지 합성을 매끄럽게 함.
- **SDXL ControlNet (Depth)**: 깊이 제어
- **BEN2**: 배경 제거(마스킹 하여 마스킹 이외의 부분 제거)
- **Rembg (U2Net)**: 텍스트 이미지 생성 후 배경 제거 후처리

4.4 GPU 최적화

- CUDA 12.8 기반
 - Torch 2.1+ (BF16/FP16 혼합 정밀도)
 - VAE Tiling, slicing : 메모리 사용량 감소
-

5. 비지니스 분석

우리 프로젝트의 비지니스적 관점을 분석한 결과입니다.

5.1 사전 목표 설정

- 핵심 강조 내용 : PEST 분석 중 가장 중요한 부분? (예: 시장 기회, 기술 우위, 규제 위험 등)
- Value Chain : 주요 단계는? (예: 소상공인 온보딩 → 프롬프트 입력 → 콘텐츠 생성 → 배포)
- 현재 상태 (AS-IS)와 목표(TO-BE):
 - AS-IS: 소상공인들이 어떤 어려움을 겪고 있는가?
 - TO-BE: 우리 서비스 후 어떻게 변하나?
- 발표 대상 선정: 누구에게 발표하나? (예: 투자자, 임원)

5.2 PEST 분석 (Why Now? - 시장의 기회)

투자자에게 “지금이 이 사업을 해야 할 적기(Timing)”임을 강조합니다.

- Technological (핵심): Stable Diffusion, FLUX 등 고성능 open-source model의 경량화 및 API 접근성 증대. 프롬프트 엔지니어링 없이도 자연어 처리(LLM)를 통해 전문가급 결과물 도출 가능.
- Economic: 경기 침체로 인해 소상공인들의 마케팅 외주 비용 절감 needs 폭발.
- Social: Instagrammable한 고품질 이미지 없이는 오프라인 매장도 생존 불가능한 시대.

5.3 Value Chain 혁신 (How? - 비즈니스 구조의 변화)

기존의 분절되고 값비싼 외주 프로세스를 우리 솔루션이 어떻게 단축(Short-cut)하고 내재화(Internalize)하는지 보여준다

- Traditional Chain (기존):
[기획/콘셉트] → [스튜디오 촬영/섭외] → [전문 디자이너 편집] → [즉시 배포/출력]
(단계별 비용 발생, 커뮤니케이션 비용 과다)
- Our AI Chain (혁신):
[사용자: 텍스트 입력] → [AI Core: 기획+생성+편집 통합] → [즉시 배포/출력]
 - 사진 촬영 대체: AI 이미지 생성을 통한 제품 가상 피팅/배치
 - 편집/디자인 대체: LLM이 광고 문구 생성 및 배너 레이아웃 자동화

5.4 AS-IS vs TO-BE (ROI Analysis - 비용과 시간의 가치)

가장 직관적인 수치 비교로 효용성을 증명합니다

구분	AS-IS (외주/전문가 의존)	TO-BE (AI 솔루션 도입)	효과 (Value)
비용	건당 5~20만 원 (디자이너/촬영)	건당 100원 미만 (토큰/GPU 비용)	99% 비용 절감
시간	3일 ~ 1주일 (수정 포함)	3분 이내 (실시간 생성)	즉각적 마케팅 대응
진입장벽	포토샵 기술, 디자인 감각 필수	모국어(프롬프트)만 입력	누구나 전문가급 제작

5.5 Marketing Positioning (Where? - 우리의 위치)

경쟁사 대비 우리의 명확한 타겟팅을 보여준다

- 전문가용(Adobe, Midjourney) vs 초보자용(Our Service)
- 범용 이미지 생성 vs 상업/광고 특화(Our Service)
- Position: "Tech-Blind SMB Owner"를 위한 "All-in-One 마케팅 비서"
 - Canva 보다 쉽고, Midjourney 보다 상업적임

"우리 서비스는 소상공인이 겪는 '촬영-디자인-편집'이라는 복잡한 밸류체인(Value Chain)을 AI 기술로 '입력-출력'이라는 단일 단계로 압축하여, 시간과 비용을 99% 절감 시켜주는 비즈니스 효율화 도구입니다."

5.6 수익 모델 구조 (3-Tier Strategy)

1. Free Tier(미끼 상품 & 진입)

- 목적 : "일단 써보세요" → 회원가입 유도 및 데이터 확보
- 제공 : 매일 3장 무료 생성 (워터마크 포함, 저해상도)
- 심리 : "오, 이거 쓸만한데? 근데 워터마크 지우고 싶네" → 결제 유도

2. Credit Pass (실속형 - Main Cash Cow)

- 목적 : 가끔 쓰는 대다수 사장님들의 자갑 열기
- 방식 : 10,000원 충전 시 12,000 크레딧 지급 (유효기간 설정)
- 논리 : "구독은 부담스럽지만, 만 원 정도는 내 가게 홍보를 위해 쓸 수 있지"

3. Pro Subscription (해비 유저 & 기업)

- 목적 : 고정 매출 확보 (Lock-in)
- 제공 : 무제한 생성, 워터마크 제거, 고해상도, 전용 템플릿 보관함

5.7 로드맵 & KPI (1년)

- Q1 : Launch (MVP) - 베타 오픈, 초기 유저 1,000명. Free Tier로 트래픽 확보 집중
- Q2 : PMF 검증 - Credit 상품 출시, 첫 매출 발생. 소상공인 커뮤니티 침투
- Q3 : Growth - User 10,000명, 재구매율(Retention) 30%. 프랜차이즈 제휴(B2B) 테스트

- Q4 : Scale-up - 월 매출 5,000만 원 돌파 (BEP 근점). 전용 mobile app 출시
-

6. 추후 개선 사항

6.1 GPU 메모리 최적화

- 페이지 이동 시 모델 누적 로드로 OOM 발생
→ **페이지별 모델 로딩/언로드 전략 도입**, 불필요 모델 자동 해제, GPU 메모리 임계값 모니터링으로 안정성 확보

6.2 ComfyUI 워크플로우 안정성

- 노드 ID 불일치·복잡한 연결 구조로 오류 발생
→ **워크플로우 자동 검증 시스템**, 모드별 단위 테스트, Visual Inspector 활용, 버전 관리 강화

6.3 extra_model_paths.yaml 기반 경로 중앙화

- 모델 경로 하드코딩으로 유지보수 불편
→ **YAML 기반 경로 관리**, 실행 시 동적 로딩, 경로 검증, 환경별 설정 분리로 이식성 향상

6.4 에러 처리 및 자동 복구

- 메모리 부족·모델 실패 시 사용자 영향 큼
→ **자동 재시도(OutOfMemory 대응)**, 상세 오류 메시지, 타임아웃 처리, 구조화된 로그로 복구력 강화

6.5 성능 측정 및 최적화

- 성능 지표 미확보로 병목 확인 어려움
→ **실행 시간·메모리 프로파일링**, 병목 구간 분석, 필요 시 성능 대시보드 도입

6.6 코드 구조화 및 중복 제거

- 프롬프트 처리·노드 입력 바인딩 로직 중복

→ **공용 유틸 통합**, 표준 API 스키마(`model_name`, `workflow_mode`), 죽은 코드 제거, 폰트·경로 폴백 지원

→ 전체 코드 일관성·가독성·유지보수성 향상

6.7 텍스트 스타일링 및 배경 제거(Segmentation) 고도화

- SDXL 기반 텍스트 디자인 품질은 우수하나, 객체 분리(누끼) 정밀도 및 경계면 처리 미흡
 - 생성(Generation)과 분리(Segmentation) 파이프라인 이원화, 전용 배경 제거 모델(Rembg 등) 파라미터 정밀 튜닝
 - 텍스트 심미성 유지 및 경계선 오염(Artifact) 없는 고품질 투명 레이어 확보

6.8 기술적 기대 효과

- 메모리 안정성 확보** → 기능 연속 실행 시 발생하는 메모리 누수 및 OOM 원천 차단
- 워크플로우 품질 향상** → 구조적 오류 감소 및 워크플로우 최적화를 통한 개발 속도 가속화
- 인프라 유연성 강화** → 설정 중앙화를 통해 로컬/클라우드 등 다양한 환경에서의 이식성 및 유지보수성 증대
- 사용자 경험(UX) 개선** → 자동 복구 매커니즘 및 표준화된 에러 메시지를 제공으로 서비스 신뢰도 제고
- 데이터 기반 최적화** → 구조화된 성능 로그 체계를 구축하여 지속적인 서비스 고도화 기틀 마련

7. 프로젝트 실험내용

이번 프로젝트에서 우리팀이 서비스 도입을 위해 시도했던 다양한 실험 내용에 대한 결과 보고서입니다.

- ▼ 1 양자화 모델 LoRA 학습 시도

실험 목표

24GB VRAM 환경에서 FLUX.1-dev(120억 파라미터) 모델의 LoRA 파인튜닝 가능 여부 검증 및 한계 분석

- 실험 환경:

- GPU: 24GB L4
 - 모델: FLUX.1-dev (T5 + CLIP Dual Encoder, Flow Matching 구조, Transformer 단독 크기 약 12B)
 - 데이터셋: 피트니스 스튜디오 이미지 520장
 - **기술적 난관:**
 1. **Flow Matching:** 기존 노이즈 기반 학습 코드와 호환 불가 → 수식/학습 스케줄 재설계 필요
 2. **Dual Encoder:** T5(4096d) + CLIP(768d) 동시 사용 → Tensor Dimension 불일치 및 Forward 오류 발생
 3. **Transformer 단독 크기:** FP16 기준 12B 파라미터 로드만으로 약 24GB 필요
→ 24GB 환경에서는 학습까지 포함하면 VRAM 부족으로 학습 불가
 4. **메모리 한계:** 단일 GPU(24GB)로 모델 로드 불가, CPU Offload 및 8-bit Quantization 필요
 - **트러블슈팅 시도:**
 - 모델 8-bit 양자화 및 CPU/GPU 분산 배치
 - Batch Size 최소화 (1)
 - 라이브러리 인자 불일치(Monkey Patch) 및 Forward 함수 점검
 - **최종 결과:**
 - 메모리 최적화 및 코드 불일치 문제는 해결
 - 그러나 **Device Deadlock** 발생:
 - 모델 분산 시 연산에 필요한 텐서 이동 실패
 - 단일 장치 집중 시 즉시 OOM 발생
 - 결론: 24GB 환경에서 LoRA 학습 불가, 구조적/하드웨어 제약 확인
 - **성과:**
 - FLUX 내부 구조(Flow Matching, Dual Encoder, Transformer) 심층 분석
 - 극한 환경에서 학습 최적화 및 라이브러리 디버깅 경험 확보
 - 향후 고사양 GPU 환경에서 LoRA 학습 재개 가능성 확보
- ▼ 2 이미지 생성 및 영상 생성 모델 학습

1. 2025년 발표된 CatV2Ton 모델 중에서 16GB Vram에서 사용 가능한 오픈 소스 모델 실험

- 입력, 전처리, 가상피팅(Virtual Try-On)까지 시도 해 봄 (1024x768)



- 상의 하의 적용은 되나 얼굴이나 디테일이 깨지는 문제가 있었음.

2. 여러 시도해 본 모델 중에 IDM-VTON 모델이 가장 결과가 좋게 나왔으나 기존 모델이 상의 생성만 지원하여 코드를 분석 해 하의와 원피스(dress)도 생성할 수 있게 개선함.

- 16G vram에 돌아갈 수 있게 최적화 하고 의류 길이를 자동 감지하게 했으며 메모리 효율화로 전처리 모델 자동 해제 시스템 도입. 적응형 마스크 생성으로 의류 길이에 따라 마스크 영역 자동 조절.



- 옆모습 + 머리카락 등으로 가리는 경우 + 뒷 배경 있는 경우도 해결 하는 실험 시도함.

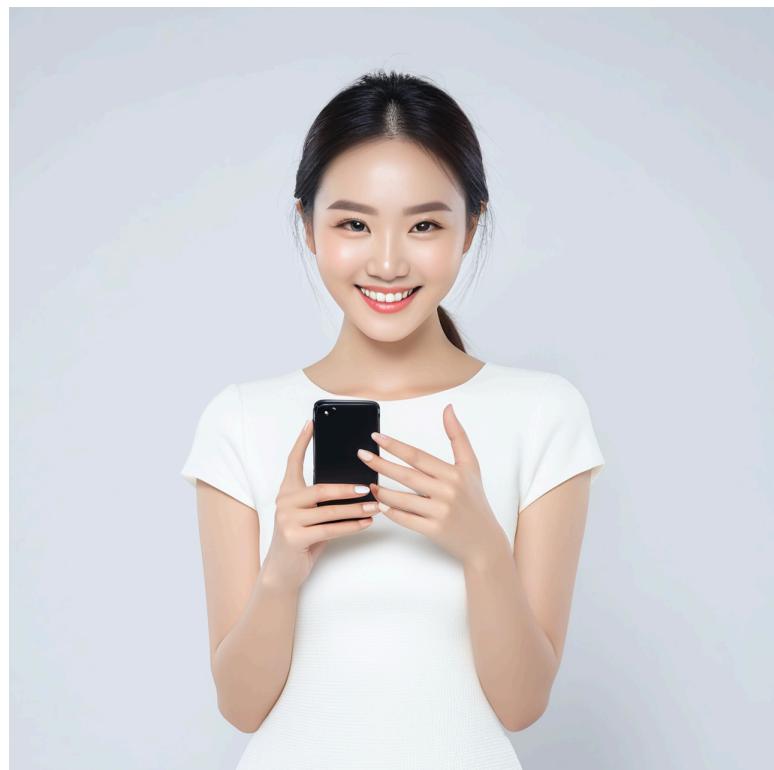


3. 16GB vram에서 가능한 여러 이미지 생성 모델들을 실험

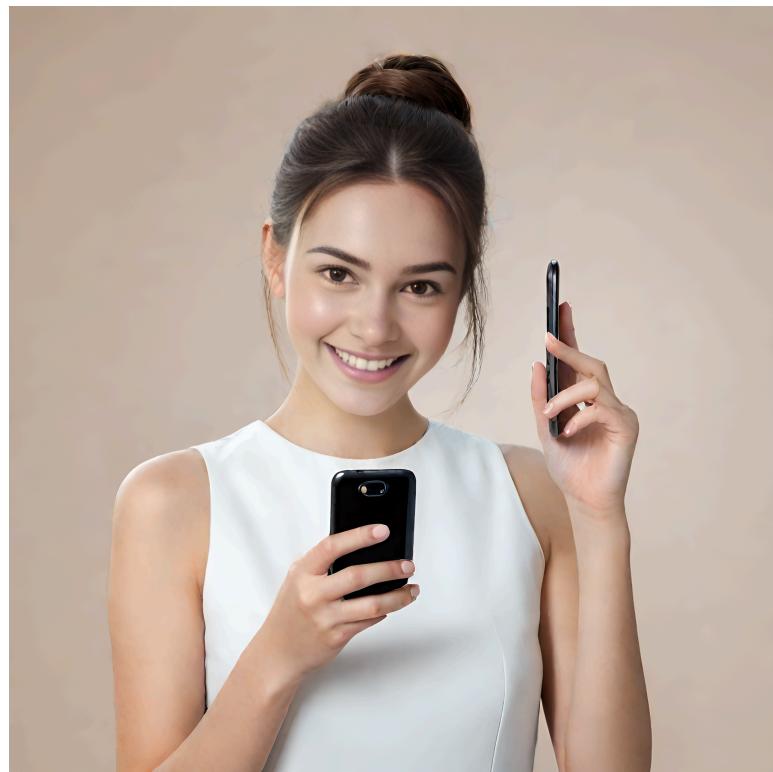
- sd35-medium : 초기에 사용 해 보았으나 이미지가 인위적으로 보이고 손가락 부분이 많이 어색했음.



- juggernaut-xl 모델 : 기존 모델들 대비 조금 더 자연스러워졌으나 고질적인 손가락 갯수 및 길이 오류 문제가 자주 등장함.



- **realistic-vision-xl** 모델 : 인물에 대한 명암이나 표현이 한층 자연스러워 졌음. 하지만 여전히 손가락 문제가 해결 안 됨.



- **flux-schnell** 모델 : **flux-dev** 모델의 경량화 모델. 기존 모델들 대비 훨씬 높은 품질의 이미지를 생성해 줬으나 Steps 수를 높일 수 없는 단점.



- **flux-dev** 모델 : 모델에 대한 질감, 광원, 아웃포커싱까지 높음 품질의 이미지를 생성함. 하지만 용량이 16GB vram에서 감당하기 어려워 cpuoffload등 여러 메모리 관리 작업을 해야 했고 50steps 이미지를 생성하려면 50분 가량 시간이 필요했음. 그리고 손가락 후처리를 하지 않아도 손가락 오류 문제가 가장 적게 발생했음.



- 이미지 품질과는 별개로 손가락 생성 때 갯수(6개, 4개) 문제가 꽤 자주 발생하여 특별히 손 보정 프로그래밍 후처리를 통해 디버깅을 시도하여 성공 비율을 높였으나 완전히 없애지는 못함.

4. 양자화 모델도 여러 가지 시도

- 양자화 모델도 여러 포맷이 존재하여 기존 프로그램과 호환성이 없으면 사용을 못함.
- flux-dev 모델의 BitsAndBytes 경량화 모델을 최종 실험하여 좋은 결과를 얻음. 8bit 모델은 24 vram 환경에서 4bit 모델은 16vram 환경에서 상당한 품질을 보여줌.

4bit



8bit



5. 고질적 메모리 문제를 해결하기 위해 ComfyUI라는 도구를 사용.

- 기존 작업환경에서 메모리 관리를 위해 모델 로딩 언로딩을 직접 일일이 수작업으로 진행하였으나 코드가 복잡해지고 후처리 부분도 무거웠는데 ComfyUI 도구 사용으로 메모리 로딩 부분이 자동화 되었고 GGUF 양자화 모델을 사용할 수 있게 되어 cpuoffload 사용을 하지 않아 이미지 생성 시간에 많은 시간 절약을하게 됨.
- 기존 모델 로더 코딩에서는 직접 ADtailer라는 방법을 코딩하여 손가락 오류 문제를 해결하였으나 ComfyUI 툴에서는 ADtailer를 툴에서 face-yolo, hand-yolo 등을 제공 받아 코드 작성 및 디버깅 문제가 많이 해소됨.

6. 하이브리드 워크 플로우 작업 실험 (Dev + Fill)

- 제품 이미지를 Flux-dev t2i 모델로 생성하여 i2i로 편집하는 방법은 i2i의 편집 범위가 너무 작아서 배경이나 인물의 의상, 디자인, 색상 등을 창의적으로 수정하기 어려웠음. 그래서 배경 생성도 따로 t2i 생성하여 Ben2 모델로 누끼(마스킹 이외 부분을 삭제함) 작업을 하고 만들어 둔 배경 이미지를 편집 전용 모델 flux-fill로 합성하는 실험을 함.
- flux-fill 모델은 단독으로 이미지 생성은 부족하지만 Ben2가 누끼 딴 이미지 마스크의 경계선(Edge)부분만 살짝 확장(Grow)하거나 Denoise를 낮춰 줌.
- 정리하면 Dev + Dev 합성 → Fill 합성 마무리 파이프라인 구성

7. 16GB vram에서 영상 작업 실험

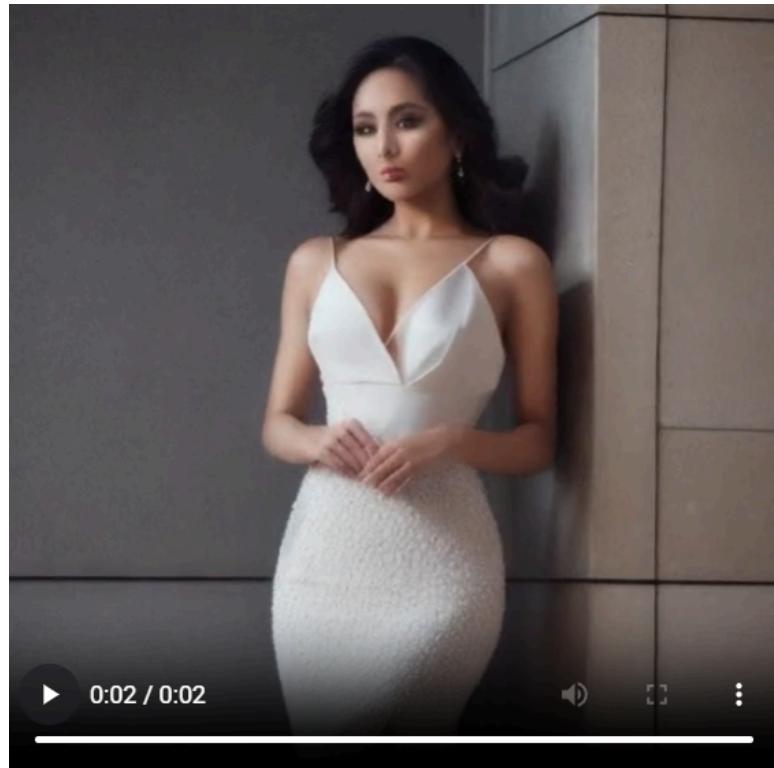
- 애초에 16GB vram에서 영상 작업은 불가능 해 보였으나 최근 업그레이드 된 오픈 소스 모델이 많이 출시되어 가능한지 실험 해 봄.
- svd video : 영상 생성은 가능했지만 액티브한 동작이 들어가면 이미지가 뭉개지는 현상 발생



- itx video : 조금 더 나은 품질을 보여줬으나 역시 액티브한 동작에서 무너짐.



- animatediff 모델 : 동작을 실행해도 이미지가 뭉개지지는 않았지만 약간 씩 프레임이 끊겨 보였고 얼굴 같은 부분의 디테일이 부족했음.



- hunyuan 1.5 모델 : 2025년 새로 나온 모델로 적은 메모리에서도 상당한 품질을 보여주었지만 이미지 생성에 몇 시간이 걸리는 문제가 있음.



- 전반적으로 실제 서비스하기에는 부족한 품질과 생성 시간을 보여 주었으나 해마다 최적화된 생성 모델이 쏟아져 나오고 있는 상황을 보면 조만간에 적은 메모리에서도 가능한 영상 모델이 출시 될 것으로 예상됨.

8. 시연 시나리오(동영상 촬영)

1. 사용자가 웹(Fit -AD)에 접속

2. **page1** 에서 광고 문구 추천을 실행, 이때 사용자가 원하면 **연결 모드** 를 **on** 하여 다음으로 진행
 3. **page1** 추천받은 문구를 활용해 **page2** 에서 바로 광고 이미지를 생성(**T2I**) → 최대 3개 까지 셈플 출력 후 선택
 4. **page2** 에서 만들었던 이미지를 **page3** 에서 원하는 분위기로 편집(**I2I**) → 편집 결과 + 원본 비교
 5. **page4** 에서 배경 제거 및 고급 편집 기능을 이용해 **배경 합성·채우기** → 편집 결과 + 원본 비교
 6. **page5** 에서 원하는 문구를 배경없는 png 파일의 캘리그라피(텍스트 이미지)를 생성
 7. **page6** 에서 광고 이미지와 텍스트 이미지를 합쳐 간단하게 sns 업로드용 광고 이미지를 생성
-

9. 핵심 설계 포인트 (요약)

9.1 AI 홍보 문구 & 해시태그 생성

- **GPT-5 Mini** 기반 전문적인 마케팅 문구 자동 생성
- 서비스 정보 입력 → AI가 3가지 버전의 매력적인 문구 생성
- Instagram 최적화 해시태그 15개 자동 추천
- 4가지 톤 선택 (친근함, 전문적, 트렌디, 감성적)
- **통합 프롬프트 엔진**: 비용 66% 절감, 속도 50% 향상

9.2 Text-to-Image (T2I) 이미지 생성

- **FLUX** 또는 **SDXL** 모델 기반 고품질 이미지 생성
- ComfyUI 워크플로우를 통한 전문가급 결과물
- 3가지 후처리 옵션:
 - **없음**: 빠른 생성 (4-8초)
 - **Impact Pack**: YOLO + SAM 기반 얼굴/손 자동 보정
 - **기존 ADetailer**: 레거시 후처리
- Instagram 최적화 해상도 지원 (4:5, 1:1, 16:9)
- GPU 메모리 효율적 관리

9.3 Image-to-Image (I2I) 이미지 편집

- 기존 이미지를 AI로 스타일 변환 및 재편집
- 변화 강도 조절 (0.0 ~ 1.0)
- 페이지1 문구 기반 보조 프롬프트 자동 생성
- 3가지 보조 프롬프트 방식:
 - 단순 키워드 변환
 - GPT 기반 자연스러운 확장
 - 사용자 조절형 혼합
- 원본 vs 편집본 비교 뷰(View)

9.4 고급 이미지 편집 실험 (3가지 모드)

- **1. 인물 모드:** 얼굴 보존, 배경/의상 변경
 - FLUX.1-dev (Q8/Q4) + ControlNet (canny/depth)
 - UNET: flux1-dev-Q8_0.ggf 또는 flux1-dev-Q4_0.ggf
 - ControlNet(Union safetensors): "InstantX-FLUX.1-dev-Controlnet-Union.safetensors"
- **2. 제품 모드:** 제품 보존, 배경 창의적 변경
 - FLUX.1-dev(Q4) + FLUX.1-Fill(Q8) + BEN2
 - Fill UNET: FLUX.1-Fill-dev-Q8_0.ggf
 - BEN2: BEN2_Base.pth 또는 BEN2_ComfyUI가 자체 다운로드 관리
- **3. 하이브리드 모드:** 얼굴+제품 동시 보존
 - FLUX.1-dev (Q8/Q4) + ControlNet (canny/depth)
 - UNET: flux1-dev-Q8_0.ggf 또는 flux1-dev-Q4_0.ggf
 - ControlNet(Union safetensors): "InstantX-FLUX.1-dev-Controlnet-Union.safetensors"
 - ControlNet 통합 (Depth, Canny)
 - 실시간 프리뷰 및 비교

9.5 3D 캘리그라피 생성

- **AI 배경 제거 (rembg)** 기반 입체적인 텍스트 이미지 생성
- 배경 투명 PNG 출력 → 다른 이미지 위에 합성 가능
- 주요 기능:
 - 한글/영문 지원
 - 커스텀 폰트 경로 지정
 - 자동 후처리 (Threshold → Erosion → Gaussian Blur)
- 사용 예시: 광고 문구, 이벤트 제목, 강조 텍스트