

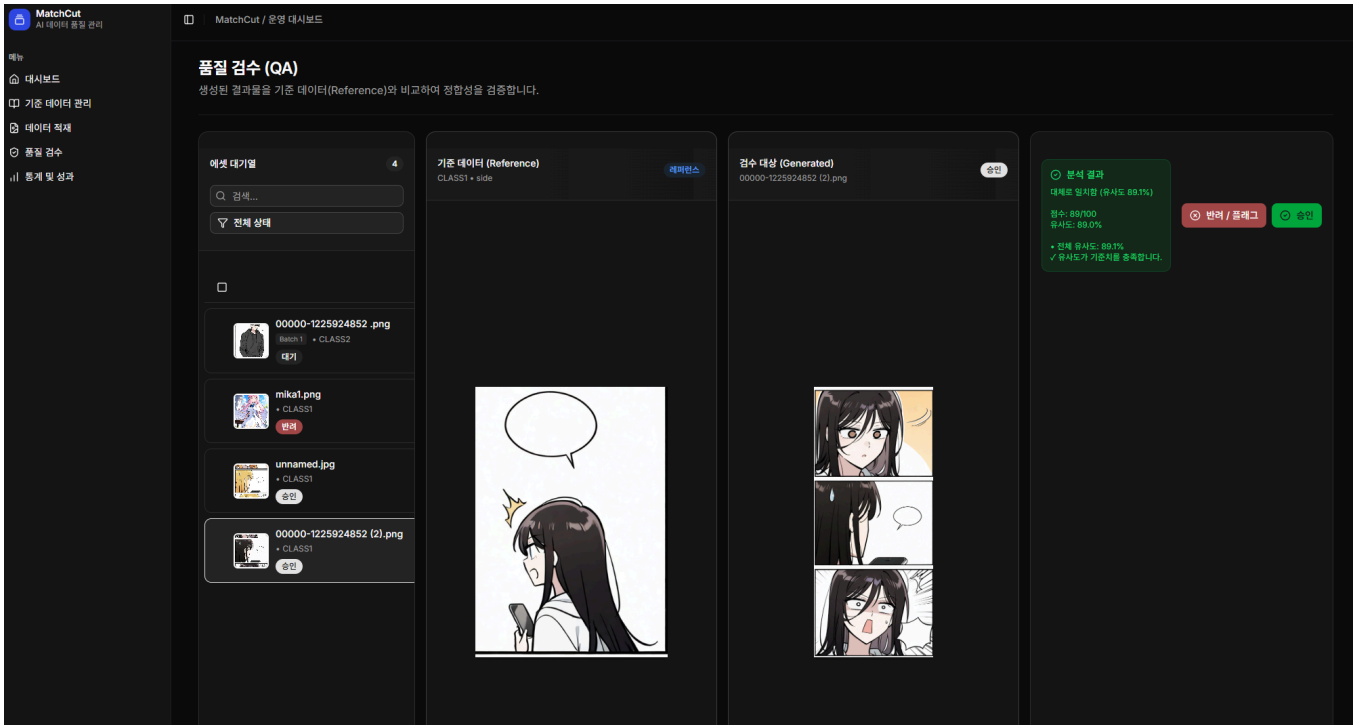
# MatchCut

| 💡 AI 기반 이미지 정합성 검증 솔루션

## 1. Overview

구분	내용
유형	개인 프로젝트
Stack	<code>Next.js</code> , <code>Transformers.js</code> , <code>Supabase</code> , <code>Zustand</code>
Github	<a href="https://github.com/jeonsavvy/MatchCut">https://github.com/jeonsavvy/MatchCut</a>
Live Demo	<a href="https://match-cut-jeonsavvys-projects.vercel.app/">https://match-cut-jeonsavvys-projects.vercel.app/</a>

## 2. 📺 Service Demo



레퍼런스로 지정한 이미지와 검수 대상 이미지를 대조한 결과, 자동 검수가 승인된 화면

## 3. Problem & Solution

- **Problem:** 생성형 AI 이미지를 검수하는 데 API 호출 비용이 과다하게 발생하고, 네트워크 지연으로 대량 처리가 비효율적임.
- **Solution:** 브라우저 탑재에 적합한 경량 AI 모델(`openai/clip-vit-large-patch14`) 탑재

## 4. System Architecture



프로젝트 작동 구조도

## 5. Key Features

1. **Hybrid Ingest System:** Drag & Drop으로 수천 장의 이미지를 올려도 멈추지 않는 처리 큐 설계
2. **On-Device Verification:** 서버 통신 없이 브라우저 메모리 위에서 정합성 판별
3. **Smart Curation:** AI 신뢰도가 낮은 데이터(Gray Zone)만 사람이 보게 하여 검수 효율 증대

## 6. Troubleshooting

### Issue 1: 브라우저 프리징

- **Challenge:** 수백 장의 이미지를 동시에 분석할 때 메인 스레드가 차단되어 UI가 멈추는 현상 발생
- **Action:** Web Worker를 도입하여 무거운 연산을 백그라운드 스레드로 격리 및 병렬 처리
- **Result:** 1,000장 업로드 중에도 UI가 반응하도록 최적화

### Issue 2: 모델 로딩 속도

- **Challenge:** 모델을 매번 다운로드하여 로딩이 느림
- **Action:** 브라우저 Cache API로 모델을 로컬 캐싱
- **Result:** 재방문 시 로딩 시간 대폭 감소

## 7. Retrospective

- **Impact:** 서버 비용 제거 및 데이터 처리 속도 향상
  - **Growth:** 브라우저 기반 AI 파이프라인 구축 경험
-