

2025년 전기 졸업과제 착수 보고서

컴퓨터 비전 기반의 개인 맞춤형 사진 관리 서비스 개발



Broom	
202255615	조수영
202255582	이서연
202255611	정지민

지도교수 : 박진선

목차

1. 연구 배경 및 목표-----	2
1.1 연구 배경-----	2
1.2 연구 목표-----	3
1.3 현실적 제약 및 대책-----	4
2. 시스템 설계 및 구성-----	5
2.1 전체 시스템 구조-----	5
2.2 기능 정의-----	5
3. 사용 기술 및 개발 환경-----	7
3.1 활용 기술 스택-----	7
3.2 개발 플랫폼 및 도구-----	8
4. 개발 계획 및 역할 분담-----	10
4.1 개발 일정-----	10
4.2 역할 분담-----	11
5. 참고문헌-----	11

1. 연구 배경 및 목표

1.1 연구 배경

스마트폰의 보급과 카메라 기술의 발달로 인해 사용자의 사진 촬영 빈도는 꾸준히 증가하고 있다. 그러나 사용자는 촬영된 수많은 사진을 정리하지 않고 단순히 저장하는 경우가 많아, 사진 관리에 대한 피로가 지속적으로 누적되고 있다. 특히 흐릿한 사진, 중복된 이미지 등의 불필요한 이미지가 갤러리에 혼재되어 있음에도 불구하고, 이를 효과적으로 필터링하고 정리할 수 있는 시스템은 부족한 실정이다.

Google Photos, Apple Photos 등 기존의 사진 관리 시스템은 인물, 장소, 시간 등을 기준으로 사진을 자동 분류하고 일부 흐릿한 사진이나 스크린샷 제거를 제안하는 기능을 제공한다. 그러나 이러한 기능들은 여전히 수동 정리에 대한 사용자 의존도가 높고, 분류 정확도나 개인화 수준이 충분하지 않아 사용자 만족도가 높지 않다. 특히 best shot 추천의 경우, 사용자에게 따라 잘 나온 사진의 기준이 다르기 때문에 개별 취향을 고려하지 못한다는 한계가 있다.

또한 대부분의 기존 시스템은 해외 사용자 환경을 중심으로 설계되어 있어 국내 사용자 특성을 반영하지 못하는 경우가 많다. 예를 들어 한국 사용자들은 카카오톡 대화 캡처, 필기 사진 등 다양한 맥락에서 사진을 활용하지만, 현재 시스템은 이러한 스크린샷을 단순히 삭제 대상으로 분류하거나 목적과 의미를 고려하지 않고 처리한다.

이와 같은 이유로 컴퓨터 비전 기술 기반의 사진의 품질과 내용을 정밀하게 분석하는 개인 맞춤형 사진 정리 시스템을 개발하여, 기존 시스템의 한계를 보완하고 실용적인 사진 정리 방식을 제안하고자 한다.

1.2 연구 목표

본 과제는 스마트폰 사용자들이 촬영한 대량의 사진들 중에서 불필요하거나 품질이 낮은 이미지를 자동으로 선별하고, 사용자가 원하는 기준에 따라 사진을 분류 및 정리할 수 있는 컴퓨터 비전 기반의 사진 관리 서비스를 개발하는 것을 목표로 한다.

첫 번째로, 흐릿한 사진, 중복 이미지, 문서 이미지 등 사진의 품질과 형태를 분석하여 자동으로 분류 및 제거하는 이미지 필터링 기능을 구현한다. 이를 위해 OpenCV, YOLO, OCR 등 컴퓨터 비전 기술을 활용하여 다양한 유형의 사진을 정량적으로 평가하고 분류한다.

두 번째로, 사용자가 직접 제공한 예시 이미지를 기반으로 유사한 사진들을 자동으로 선별하고 그룹화하는 개인 맞춤형 정리 기능을 구현한다. 예를 들어 사용자가 ‘고양이 사진’을 예시로 제공하면, 시스템은 이를 기반으로 전체 사진 중 고양이가 탐지된 이미지를 사용자 맞춤 폴더로 분류한다.

마지막으로 기능들을 통합하여 모바일 환경에서 작동하는 사진 정리 애플리케이션으로 구현하고, 사용자에게 직관적인 UI를 통해 정리 결과를 제공하는 서비스를 개발하는 것을 최종 목표로 한다.

1.3 현실적 제약사항 및 대책

개발 및 배포 과정에는 다음과 같은 현실적 제약이 존재한다.

1) 모바일 디바이스의 처리 성능 제한

고성능 AI 모델을 모바일 기기에서 직접 실행하기에는 리소스가 부족할 수 있다. 이에 따라 일부 기능은 경량화된 모델을 통해 온디바이스에서 제공하고, 고비용 연산이 요구되는 작업은 서버 기반으로 처리함으로써 연산을 효과적으로 분산할 수 있도록 한다.

2) 다양한 스크린샷 유형에 대한 분류 정확도 확보

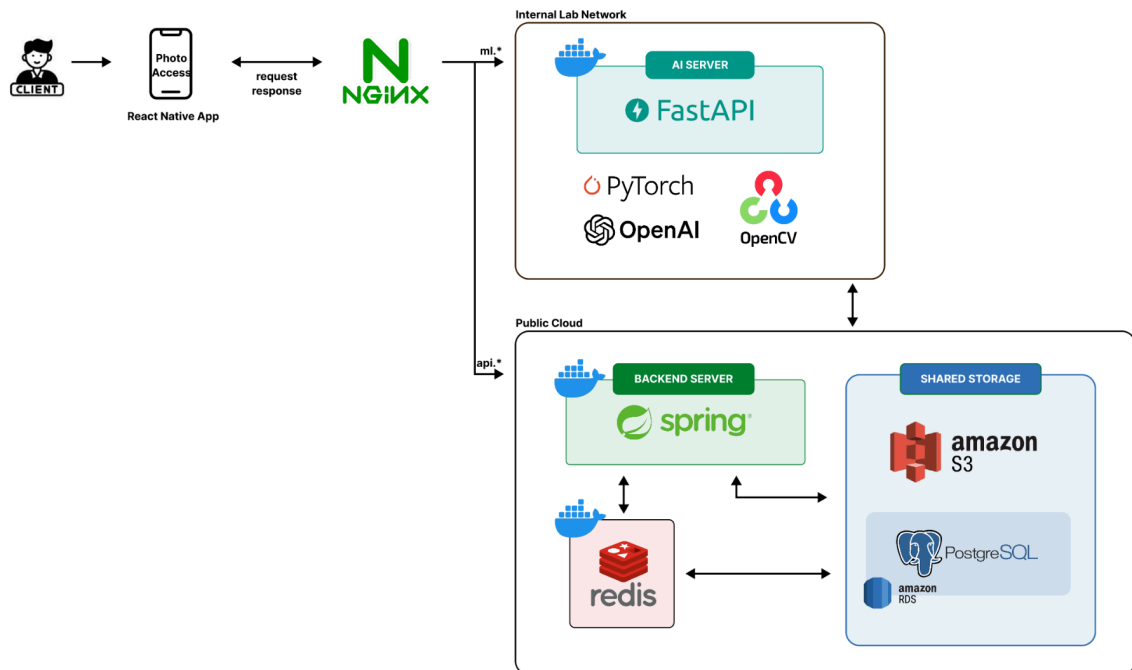
사용자의 다양한 스크린샷 유형을 자동으로 분류하는 데에는 여러 어려움이 존재한다. 이를 해결하기 위해 해상도 비율, 텍스트 밀도, UI 구성 요소 등의 다양한 시각적 특징을 종합적으로 분석한다. Rule-based 방식과 머신러닝 기반 모델을 함께 적용하여 분류 정확도를 높이고 이후 사용자 피드백을 바탕으로 지속적으로 모델을 개선할 예정이다.

3) 대용량 이미지 업로드 시 서버 부하

사진 정리 서비스의 특성상 사용자는 한 번에 수천 장의 이미지를 업로드할 수 있으며, 이는 서버의 네트워크 트래픽과 처리 비용을 급격히 증가시킬 수 있다. 이러한 문제를 해결하기 위해 Amazon S3의 Presigned URL 기능을 활용한다. 이를 통해 사용자는 백엔드 서버를 거치지 않고 S3에 이미지를 직접 업로드할 수 있으므로, 서버 부하를 효과적으로 분산시키고 전체 시스템의 안정성을 확보할 수 있다. 아울러 Presigned URL은 유효 시간이 제한되어 있어 인증된 사용자만 일정 시간 동안 접근할 수 있게 함으로써 보안적인 측면에서도 유리하다.

2. 시스템 설계 및 구성

2.1 전체 시스템 구조



2.2 기능 정의

사용자의 사진 데이터를 자동으로 분류하고 정리할 수 있는 시스템을 구축하는 것을 목표로 하며, 주요 기능은 다음과 같이 정의된다.

1) 중복 사진

이미지 유사도 분석을 통해 중복된 사진을 자동으로 식별한다. 연속 촬영이나 실수로 여러 번 촬영된 유사 이미지들을 그룹화하고, 대표 이미지를 제외한 나머지는 별도 폴더로 분류한다.

2) 흐릿한 사진

초점이 맞지 않거나 모션 블러가 심해 식별이 어려운 사진을 탐지하여 자동으로 분류한다. 이는 이미지 품질 분석을 통해 흐림 정도를 판단한다.

3) 스크린샷

해상도 비율, UI 요소, 텍스트 비율 등을 기준으로 스크린샷을 자동 식별하고, 목적별로 세분화하여 분류한다.

- a) 채팅 아카이빙: 대화 UI 구조 인식 기반
- b) 정보글: 이미지 내 텍스트 비율이 높은 스크린샷
- c) SNS 콘텐츠: 인스타그램 등 UI 아이콘을 기반으로 식별
- d) 상품 정보: 객체 탐지를 통해 상품 정보 포함 여부를 판단
- e) 미디어 스크린샷: 게임, 영화, 드라마 등 콘텐츠 이미지
- f) 기타 스크린샷: 상기 항목에 해당되지 않는 일반 스크린샷

4) 고대비 및 단색 사진

지나치게 어둡거나 밝은 이미지, 또는 단일 색상으로 채워진 사진을 자동 식별한다. 조명이 꺼졌을 때 찍은 사진처럼 픽셀 분포가 지나치게 단색에 가까운 이미지를 분류한다.

5) 문서 및 영수증

OCR 기술을 활용해 이미지 내 과도한 텍스트가 포함되어 있거나 문서 형식이 인식되는 경우 이를 자동으로 식별한다. 영수증, 문서 스캔본 등 종이 기반 이미지를 별도로 분류한다.

6) 객체 탐지 결과가 없는 사진

YOLO, Detectron2 등의 객체 탐지 알고리즘을 활용해 유의미한 객체가 전혀 탐지되지 않은 사진을 분류한다. 예를 들어, 벽이나 바닥만 찍힌 사진과 같이 배경 이외의 객체가 포함되지 않은 사진이 해당된다.

7) 사용자 개인 맞춤형 폴더

사용자가 일정 수의 예시 이미지를 제공하면 해당 패턴을 학습하여 유사한 이미지를 자동 분류하고 별도의 폴더로 그룹화한다. 예를 들어, 고양이 사진 또는 게임 승리 장면 등 개인화된 분류 기준 기반 자동 정리가 이루어진다.

3. 사용 기술 및 개발 환경

3.1 활용 기술 스택

1) 사용자 인터페이스

iOS 기반 사용자에게 직관적인 이미지 정리 기능을 제공하기 위해, React Native를 기반으로 프론트엔드 애플리케이션을 개발한다. 사용자는 앱을 통해 갤러리 접근 권한을 부여하고, 흐릿한 사진이나 유사 사진에 대한 정리 제안을 받아볼 수 있다. 상태 관리는 React Context와 Custom Hook을 활용하여 전체 앱의 상태 일관성을 유지하며, 사용자 경험을 향상시킨다.

2) 서버

서버는 Spring Boot와 JVM 기반 언어를 사용하여 RESTful API 서버로 구축한다. 해당 백엔드는 사진 정리 요청 처리, 사용자 피드백 저장, 요청 로그 관리 등의 기능을 담당하며, 멀티스레드 기반의 비동기 처리 방식으로 대량 이미지 요청에도 안정적으로 대응할 수 있도록 구성한다. 또한 Amazon EC2 인스턴스 위에서 Nginx 및 Docker 환경을 활용하여 지속적인 서비스의 통합과 배포를 관리한다.

3) 인공지능 분석 모듈

모델 훈련 및 검증은 연구실에서 제공한 GPU 서버에서 수행되며, 실제 서비스 배포 시에는 일부 모델은 스마트폰 내부에서 직접 실행될 수 있도록 배포한다. OpenCV를 활용하여 흐릿하거나 노출 및 색상이 부적절한 사진을 감지하고, YOLO를 통해 스크린샷이나 의미 없는 객체만 포함된 사진을 자동으로 분류한다. 또한 PyTorch 기반의 이미지 임베딩 기법을 적용하여, 사용자가 제공한 예시 이미지와 유사한 사진을 탐색하고 분류하는 개인 맞춤형 정리 기능을 구현한다.

4) 스토리지

이미지와 관련된 메타데이터 및 사용자 설정 정보는 Amazon RDS에서 운영중인 PostgreSQL 데이터베이스에 저장된다. 한편, 실제 이미지 파일은 Amazon S3에

저장된다. 사용자는 로컬 기기에서 직접 이미지를 관리하며, 서버 분석이 필요한 경우에만 해당 이미지를 선택적으로 업로드하는 방식으로 설계하였다. 사용자가 서버에 분석을 요청할 때 선택적으로 업로드 되는 구조로 설계한다. 또한 FastAPI 기반 AI 서버는 Amazon S3에서 이미지를 직접 읽어와 분석할 수 있도록 구성되어 있어 분석 처리 효율성을 높인다.

5) 보안 및 네트워크

전체 백엔드 인프라는 AWS VPC 내부에 구성되어있으며 보안 강화를 위해 HTTPS 및 JWT 기반의 사용자 인증 및 인가 체계를 적용한다. 인공지능 서버는 연구실 내부망의 별도 장비에서 운영되며 외부 요청은 Nginx를 통해 프록시 방식으로 전달된다.

3.2 개발 플랫폼 및 도구

분류	사용 도구
모바일 앱 플랫폼	React Native (iOS 대응), macOS, VS Code
서버 플랫폼	Spring Boot, IntelliJ, DataGrip, Postman
AI 분석 서버	Python, FastAPI, PyTorch, Jupyter Notebook
클라우드 인프라	AWS(EC2, S3, RDS, VPC), Docker, GitHub Actions
협업/디자인 도구	GitHub, Notion, Figma, Postman

본 과제는 모바일 앱, 백엔드 서버, 인공지능 분석 서버, 클라우드 인프라로 구성되고 각 구성 요소의 개발을 위해 다양한 플랫폼과 도구를 활용하고 있다.

모바일 앱 개발은 iOS 플랫폼을 대상으로 하며, React Native를 기반으로 하여 단일 코드베이스로 앱을 구현한다. 프론트엔드 개발 환경은 macOS 기반에서 구성되며, 에디터로는 Visual Studio Code를 주로 사용하고 있다.

백엔드 서버 개발은 Spring Boot 기반으로 진행되며, 개발 환경은 IntelliJ IDEA를 사용한다. 데이터베이스 설계 및 질의 작업은 DataGrip을 통해 수행하며, API 테스트에는 Postman을 활용하고 있다.

AI Vision 분석 서버는 Python 및 FastAPI 기반으로 구성되며, 모델 학습 및 실험에는 PyTorch를 활용한다. 인공지능 관련 개발은 주로 Jupyter Notebook 환경에서 수행하며, 코드 편집 도구로는 Visual Studio Code를 사용한다.

클라우드 인프라는 AWS를 기반으로 하며, EC2, S3, RDS, VPC 등을 활용하여 백엔드 서버와 스토리지를 구성한다. 컨테이너 기반 배포는 Docker를 통해 이루어지며, CI/CD는 GitHub Actions를 통해 자동화하고 있다.

프로젝트 협업 및 관리 도구로는 GitHub를 이용한 버전 관리, Notion을 통한 문서 관리 및 회의록 기록, Figma를 활용한 UI/UX 시안 설계가 사용된다.

4. 개발 계획 및 역할 분담

4.1 개발 일정

Task	May	June	July	Aug	Sept
ERD 및 DB 스키마 설계					
서버 구성 및 기술 스택 결정					
사진 데이터 수집					
UI/UX 디자인 작업					
핵심 API 개발					
도메인 구조 설계 및 엔티티 구현					
JWT 인증 및 사용자별 설정 저장 기능 구축					
핵심 컴포넌트 구현					
블러 필터 기반 흐릿한 사진 감지					
YOLO 기반 객체 탐지 모델 학습					
중간 보고서 작성					
OCR 기반 텍스트 영역 추출 및 분석					
해시 기반 유사 이미지 탐지 및 중복 분류					
통합 분류 결과 API 연동					
Spring Boot + Docker 이미지화					
AWS EC2 + Nginx 배포, Github Actions 통한 CI/CD 구성					
정리 제안 페이지 + 사용자 폴더 커스터마이징 기능 개발					
앱 빌드 자동화 구성					
배포용 APK 및 TestFlight 배포 준비					
사용자 맞춤 이미지 분류 모델 설계					
리팩토링 및 보완					
최종 보고서 작성					

4.2 역할 분담

공통	관련 레퍼런스 조사, 테스트 및 보완, 보고서 작성, 발표 및 시연
이서연	사용자 인터페이스 개발 전담 React Native를 활용해 iOS에서 작동 가능한 애플리케이션 제작 사용자 경험을 고려한 인터랙션 설계와 상태 관리 로직 개발 앱 컴포넌트 구현, 빌드 및 배포 준비
정지민	전체 시스템의 서버 아키텍처 설계 및 API 구현 Spring Boot를 기반으로 한 RESTful API 서버 개발 데이터베이스 설계 및 AWS 배포 JWT 기반 인증 시스템 구축, CI/CD 환경 구성 등 총괄
조수영	OpenCV, YOLO, OCR 등을 활용한 사진 분류 및 필터링 기능 구현 사용자 맞춤형 필터링 기능 구현 AI 모델 최적화 및 서버 연동 API 설계 등 전반적인 AI 파이프라인 담당

5. 참고문헌

- [1] Fraga Pérez, A., & Forti Buratti, M. A. (2017). *New habits in smartphones photo management*. adComunica. Revista Científica de Estrategias, Tendencias e Innovación en Comunicación, (13), 135–156.
- [2] Goyal, M., Munjal, R. S., Moharana, S., Garg, D., Mohanty, D. P., & Thota, S. P. (2021). *ScreenSeg: On-Device Screenshot Layout Analysis*. arXiv preprint arXiv:2104.08052.
- [3] Lonn, S., Radeva, P., & Dimiccoli, M. (2019). *Smartphone picture organization: A hierarchical approach*. arXiv preprint arXiv:1803.05940v2.