

# OpenCV 실시간 시설물 이상 징후 감지 프로젝트

## 프로젝트 정보

- 프로젝트명: OpenEye\_CCTV
- 개발 기간: 2025년 8월 12일(수) ~ 8월 22일(금)
- 개발 인원: 개인 프로젝트
- GitHub 링크: [https://github.com/guysaint/openeye\\_cctv](https://github.com/guysaint/openeye_cctv)
- Live Demo: <http://bit.ly/4lBkV9q>

## 문제 정의 및 목표

해결하고자 한 구체적 문제: 기존 아파트나 공장, 창고 등에 설치된 CCTV는 단순 녹화만 하는 기능으로 사고 발생 후 녹화된 영상을 찾아보는 정도에 그쳐 사전에 예방할 수 있는 고장난 시설물의 장기간 방치나, 그로 인한 인명 피해 등을 예방해 기존 CCTV의 한계를 보완해보자 했습니다.

## 설정한 측정 가능한 목표

- 시설물 고유의 패턴을 인식해 템플릿 매칭으로 패턴에서 벗어난 것을 감지.
- 패턴에서 일정 시간 이상 벗어났다고 판단되면 경고문구를 띄움.
- 한 앵글에 여러 시설물이 있을 경우 여러 객체를 인식하도록 개선

**문제 선택 이유:** 이상을 보이는 시설물이 장기간 방치되어 사고를 당한 개인적인 경험에서 충분히 예방할 수 있는 문제를 어떻게 해결 할 수 있을까 고민하다가 각 시설물들이 일정한 패턴이 있다는 것을 발견하여 충분히 구현이 가능해 문제를 해결 할 수 있을 것이라 판단했습니다.

## 성과 측정 결과

측정 항목	개선 전	개선 후	개선율
의자 패턴 이상 감지	0%	100%	+ 100%
Usb stick 패턴 이상 감지	80%	100%	+ 25%
그립톡 패턴 이상 감지	80%	80%	0%
벨브(모형)과 LED 이상 감지	100%	100%	0%
인식 실패율	50%	0%	-50%

## 기술 스택

- 주요 언어: Python 3.10.7 / Python 3.11.2(라즈베리파이)
- 핵심 프레임워크/라이브러리: OpenCV-contrib 4.12, Numpy
- 개발도구: VSCode

## 핵심 기능

### 1. 각 시설물 템플릿 이미지 저장.

- 기능 설명: 현재 CCTV의 앵글에 있는 시설물을 크기에 맞게 드래그하여 템플릿 이미지로 저장한다.
- 사용된 핵심 기술: OpenCV 관심 영역 설정, 파일 저장(자동 번호 붙이기)

### 2. 시설물 인식 및 패턴 이상 감지

- 기능 설명: 템플릿 매칭으로 시설물의 위치와 최초 모양을 인식, 시설물이 회전하거나, 불이 꺼지거나 하는 등을 인식하여 화면에 패턴 이상 경고문을 띄움.
- 사용된 핵심 기술: 템플릿 매칭, ROI(관심영역)
- 달성한 성과: 실시간으로 CCTV 한 앵글에 여러 개의 시설물 감지 가능

## 문제 해결 과정 및 역량

### 기술적 문제 해결 사례

사례 1: 미세한 움직임에도 패턴 이상이라고 감지하는 문제

- **직면한 구체적 문제:** 영상 촬영 시 손떨림으로 인해 약간의 변화도 패턴 이상이라고 인식하는 문제
- **문제 분석 과정:** 손떨림으로 인한 움직임, 야외 CCTV의 경우 바람에 의해 흔들릴 수 있음을 깨닫고 해결 방법을 모색함.
- **시도한 해결 방법:**
  1. 중심점을 인식하고 그 중심점을 중심으로 ROI(관심영역)을 설정해서 영역 안을 벗어나지 않으면 약간의 움직임이 있더라도 패턴 이상이 아니라고 판단하도록 함.

**해결 결과:** 허용 범위 안에서 움직임은 패턴 이상이라고 판단하지 않음.

**학습한 내용:** ROI(관심 영역)의 활용법 이해.

사례 1: 다른 객체가 가까워지면 처음 인식한 객체를 벗어나는 문제

- **직면한 구체적 문제:** 시설물이라고 인식하여 그 시설물을 인식했으나 사람이 가까이 가면 인식 범위가 사람으로 옮겨 가서 패턴 이상이라고 감지하는 문제.
- **문제 분석 과정:**
  - 처음 시설물을 인식하지만 옆에 사람이 있으면 인식 영역이 사람으로 이동하는 문제
- **시도한 해결 방법들:**
  1. 캐니 엣지를 사용하여 템플릿 매칭을 시도해 봄.
  2. 템플릿 이미지를 화면 전체가 아닌 시설물만 드래그로 선택하여 저장함.
  3. 로컬 검색 적용 -> 템플릿 이미지를 토대로 전역으로 의자를 찾고 그 다음부터 그 주변에서만 계속 매칭해 나가는 방법.
  4. 시설물의 벨브와 유사한 usb 메모리 스틱으로 테스트로 바꿈.

**해결 결과:** 캐니 엣지를 사용한 시도는 큰 차이점이 없었음. 템플릿 이미지를 시설물만 드래

그하여 배경을 최소화한 이미지로 매칭을 하니 인식률이 확연하게 올라감.

**학습한 내용:** 템플릿 매칭의 인식률 개선 방법 이해, 템플릿 매칭 시 실시간 영상 CCTV와 같은 해상도 상에서 템플릿 이미지를 생성해야 함. 로컬 검색에 대한 이해.

## 기술적 학습 성과

### 1. 템플릿 매칭

- **프로젝트 적용:** 고정된 시설물의 패턴 이상 감지에 효과적임.
- **학습 과정의 어려움:** 정확한 사용법을 몰라 실시간 영상과 매칭 시키는 것에 어려움이 있었음.
- **극복 방법:** 템플릿 이미지와 실시간 영상의 해상도 일치시킴, 시설물 영역만 잘라내어 매칭 시키는 등의 반복적인 시도.

### 2. 로컬 검색에 대한 이해

- **프로젝트 적용:** 실시간 영상의 경우 계속 매칭을 시키는 상황에서 오인식을 할 확률을 줄이는 방법에 로컬 검색을 사용함.
- **학습 과정의 어려움:** 로컬 검색에 대한 이해도가 없어 기술 적용에 대한 어려움.
- **극복 방법:** 로컬 검색에 대한 이해를 위해 학습. ChatGPT를 활용하여 이해를 더함.

## 코드 품질 및 협업

### Git 커밋 히스토리

- 총 33회(일일 평균 4회)

### 코드 리뷰 경험

- 동료 개발자 2명으로 부터 약 8회 피드백 수렴.

### README 문서화 수준:

- 간략한 개발 과정과 피드백, 실행 데모 영상 업로드.
- 설치부터 실행까지 단계별 가이드 작성 예정.
- 문제 해결 FAQ 섹션 포함 예정