OpenCV를 활용한 Contour 검출 보고서

여기에 텍스트를 입력하세요 여기에 텍스트를 입력하세요

**뮤텍코리아 지원자 이지혁**

2025년 6월 20일

# 

# 개요

본 보고서는 OpenCV 라이브러리를 활용하여 이미지 처리 및 Contour 검출을 수행하는 과정을 기술합니다. RAW\_02-66\_0309.bmp 이미지를 대상으로 Grayscale 변환, Binary 변환 및 윤곽선 검출 및 시각화 작업을 단계별로 진행하였습니다. 또한 추가기능1) 임계점 변화에 따른 Contour 검출의 차이, 추가기능2) 검출된 Contour의 기하학 모양에 따른 분류, 추가기능3) Binary 전후 가우시안 필터를 통한 이미지처리에 따른 Contour 검출 변화를 서술하고 있습니다.

# 개발 환경

1. VScode
2. Python 3.13.5
3. opencv-python 4.11.0.86 사용

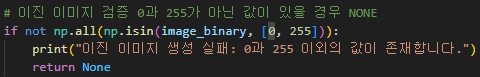
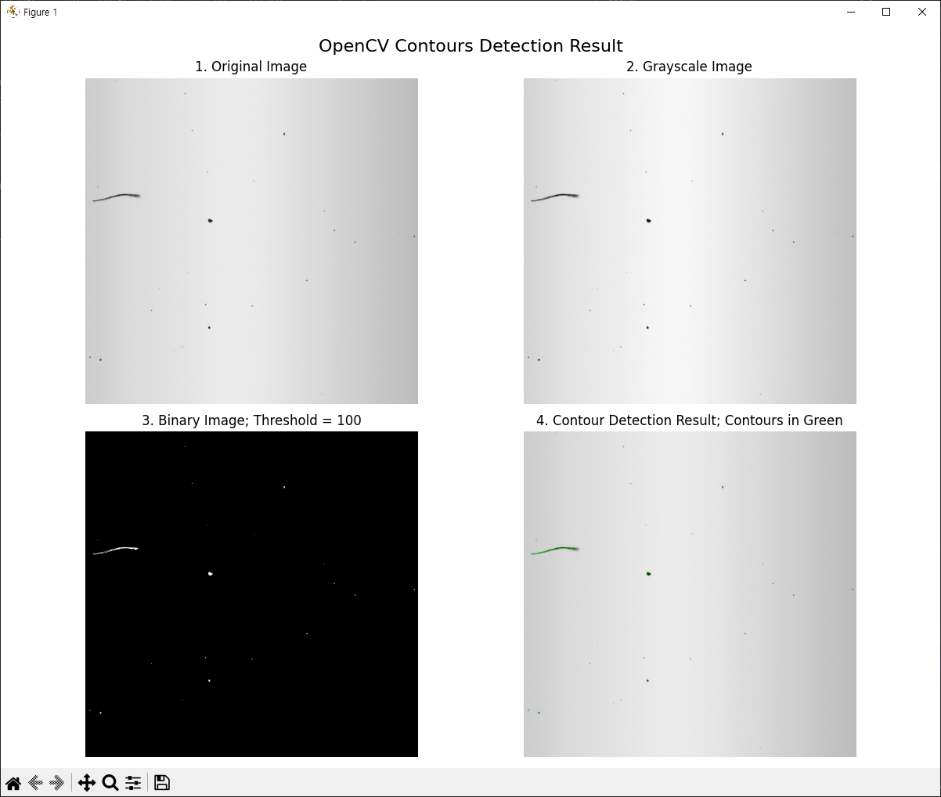
# 실행 절차

1. 프로젝트 다운로드
2. 가상환경 설치
   1. 가상환경 설치 명령어 입력
      1. python -m venv venv
   2. 필요 모듈 설치
      1. pip install -r “requirements.txt”
3. 파이썬 인터프리터 설정
4. 프로젝트 실행

# 전체 소스 코드

1. 프로젝트 폴더의 [Readme.md](http://readme.md) 참조
2. 제공된 전체 코드 및 주석 참조

# 단계별 구현 설명

1. cv2.imread() 를 통한 **이미지 읽기**   
   
   1. [ERROR] 이미지경로 오류시 프로그램 종료  
      
2. cv2.imread()를 통한 **Grayscale 이미지 생성**  
   
3. Binary 이미지 생성 함수 set\_image\_binary() 호출  
   
4. np.where()를 통해 임계값 기준으로 **Binary 이미지 생성**  
   
   1. Binary 이미지 검증 실패시 None 반환  
      
5. Contour 검출 함수 detect\_contous() 호출  
   
6. cv2.findContours() 함수를 통해 Contour 검출  
   
   1. mode : cv2.RETR\_EXTERNAL
   2. method : cv2.CHAIN\_APPROX\_SIMPLE
7. 검출된 Contour를 원본 이미지 위에 그리는 display\_result()함수 호출  
   
8. pyplot 을 이용한 결과 이미지 display  
   
9. 사진 결과 저장하는 함수 save\_results() 호출  
   

# 추가기능

1. 임계점 변화에 따른 Contour 검출 변화
2. 검출된 Contour의 기하학 모양에 따른 분류
3. Binary 전후 가우시안 필터를 통한 이미지처리에 따른 Contour 검출 변화

# 기술적 고려사항

여기에 텍스트를 입력하세요

1. 여기에 텍스트를 입력하세요
2. 여기에 텍스트를 입력하세요
3. 여기에 텍스트를 입력하세요 여기에 텍스트를 입력하세요

# 결론

여기에 텍스트를 입력하세.

# 추후 개선사항

1. 여기에 텍스트를 입력하세요
2. 여기에 텍스트를 입력하세요
3. 여기에 텍스트를 입력하세요 여기에 텍스트를 입력하세요