

미니 프로젝트 ver 2.0

디지털 영상 처리 w. C++, MFC

[Intel] 엡지 AI SW 아카데미 - 객체지향 프로그래밍

Contents

- 프로젝트의 목표
- 개발 환경
- 화면 구성 및 기능
- 부가 기능
- 마치며

프로젝트의 목표



비전

C++를 사용하는 MFC로
윈도우 개발을 진행하여
추후 컴퓨터 비전에 초석으로 함



미션

C++과 MFC로
영상 처리 프로그램을
구현하는 것

개발 환경

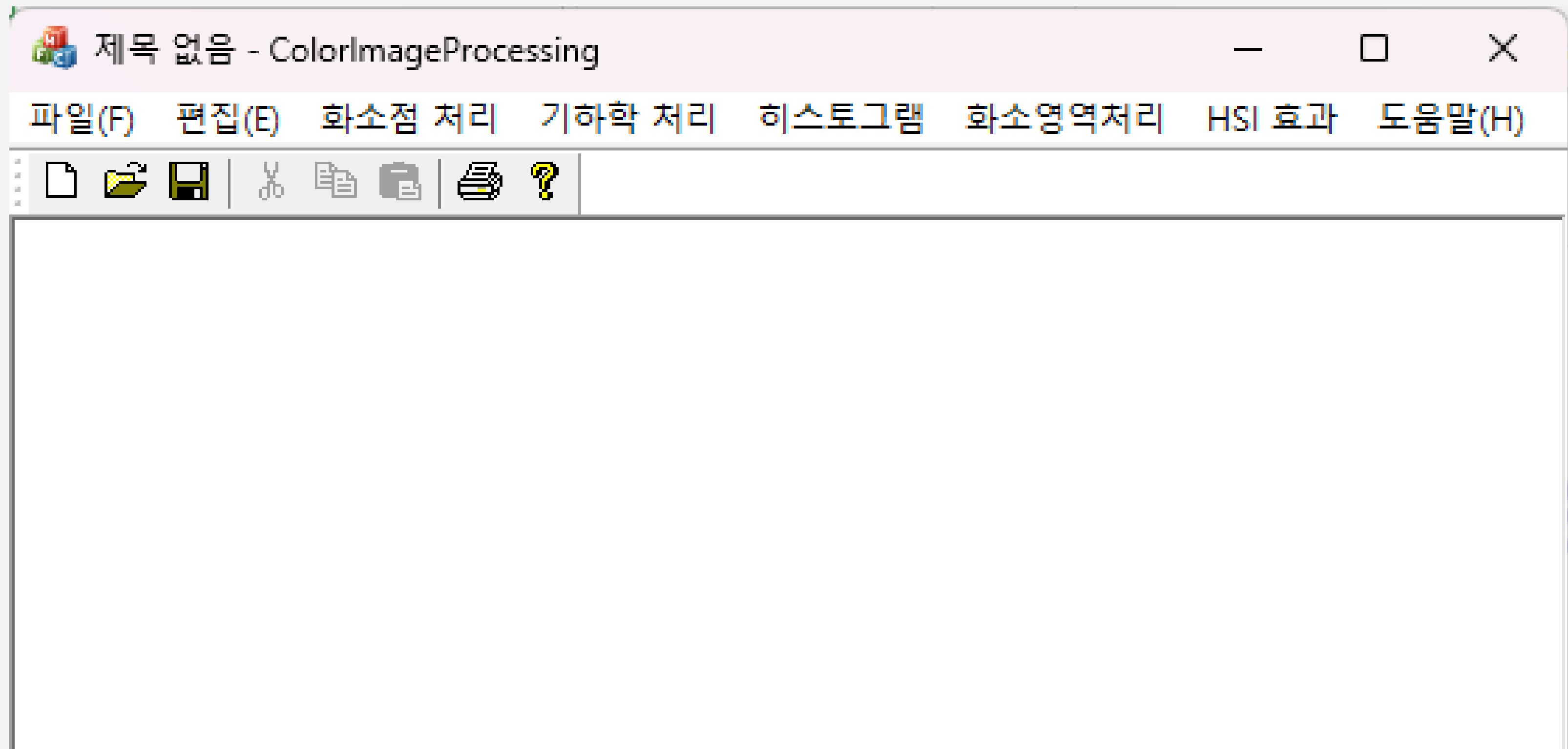


**Microsoft
Visual Studio
Community 2022**

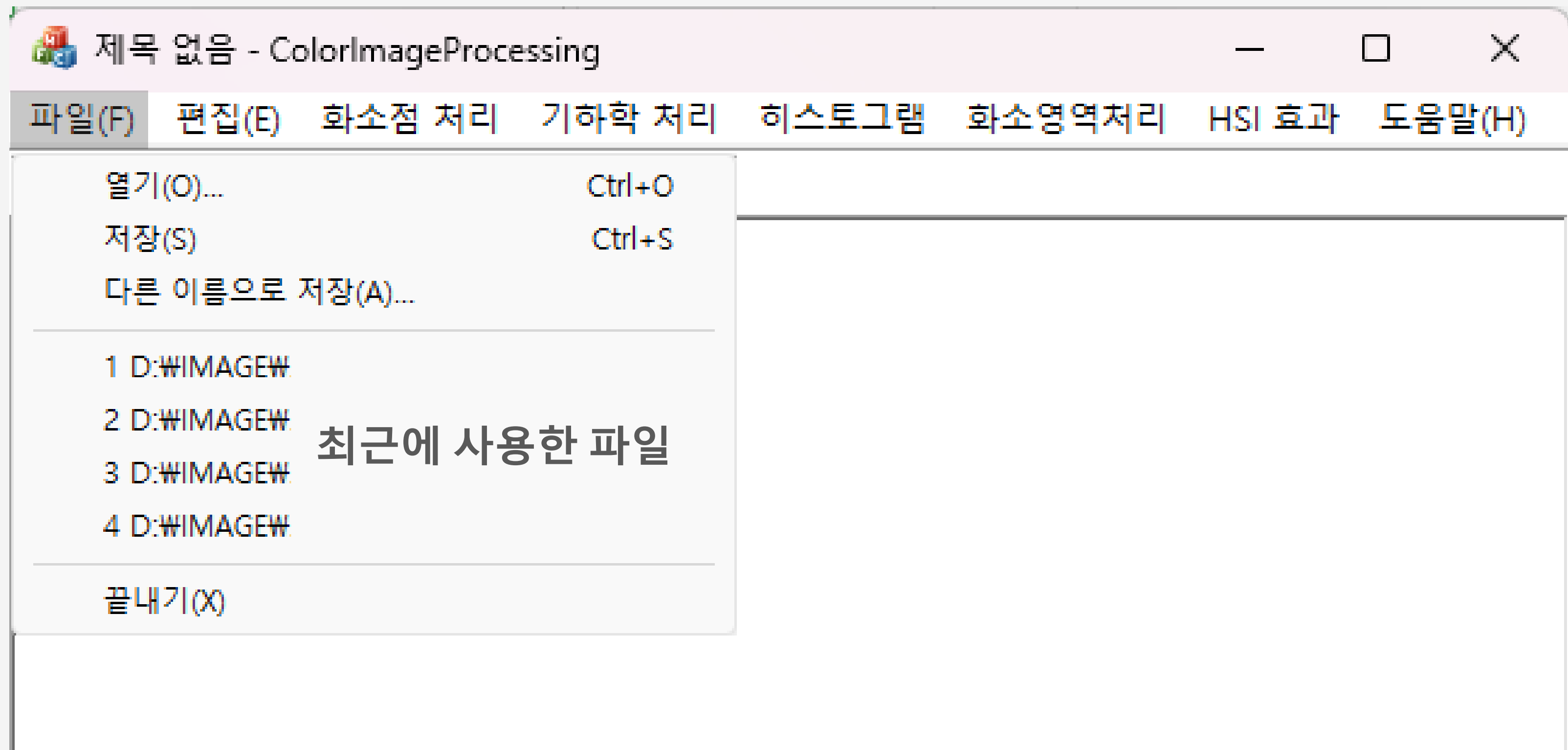


Visual C++ 2022

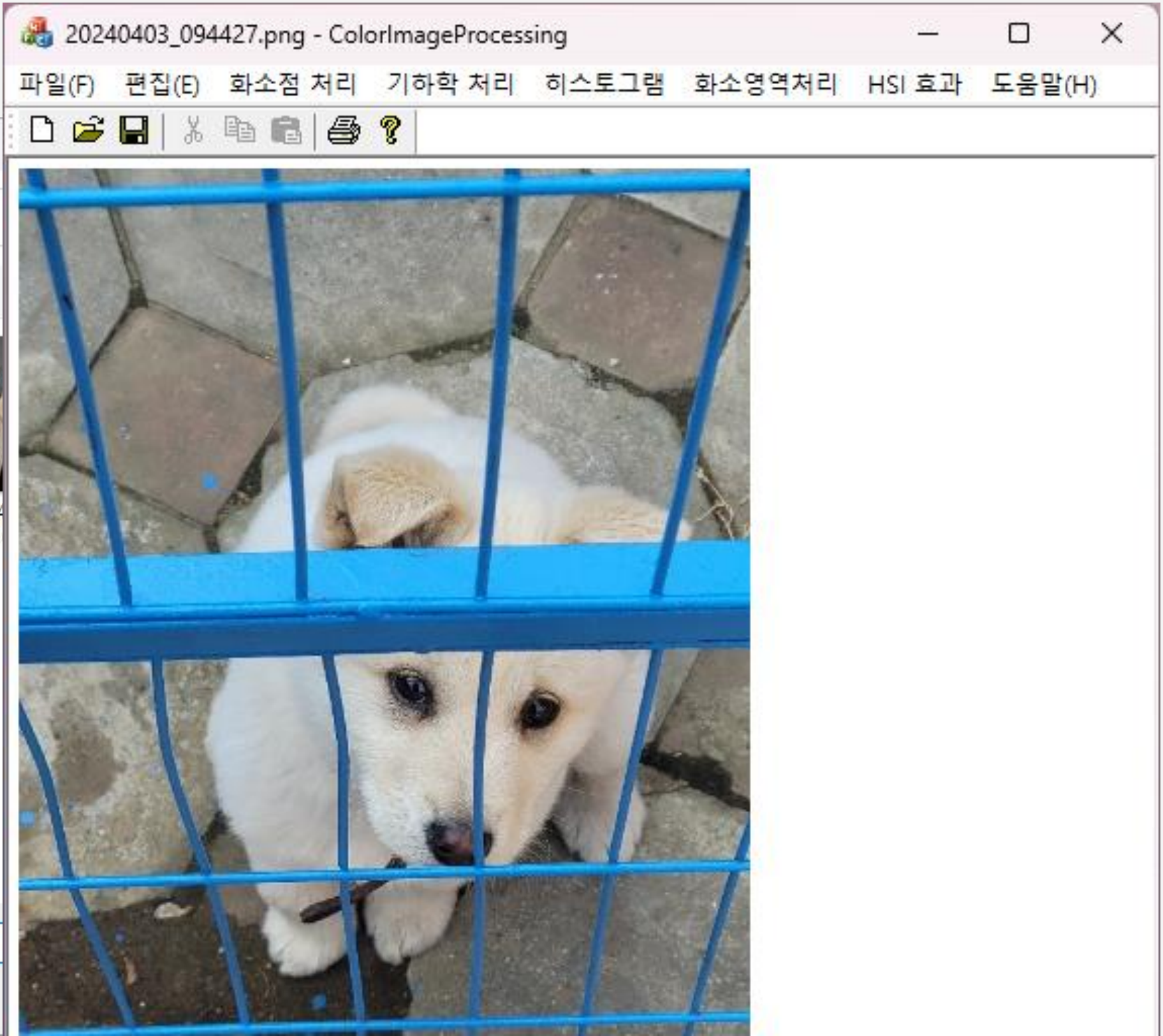
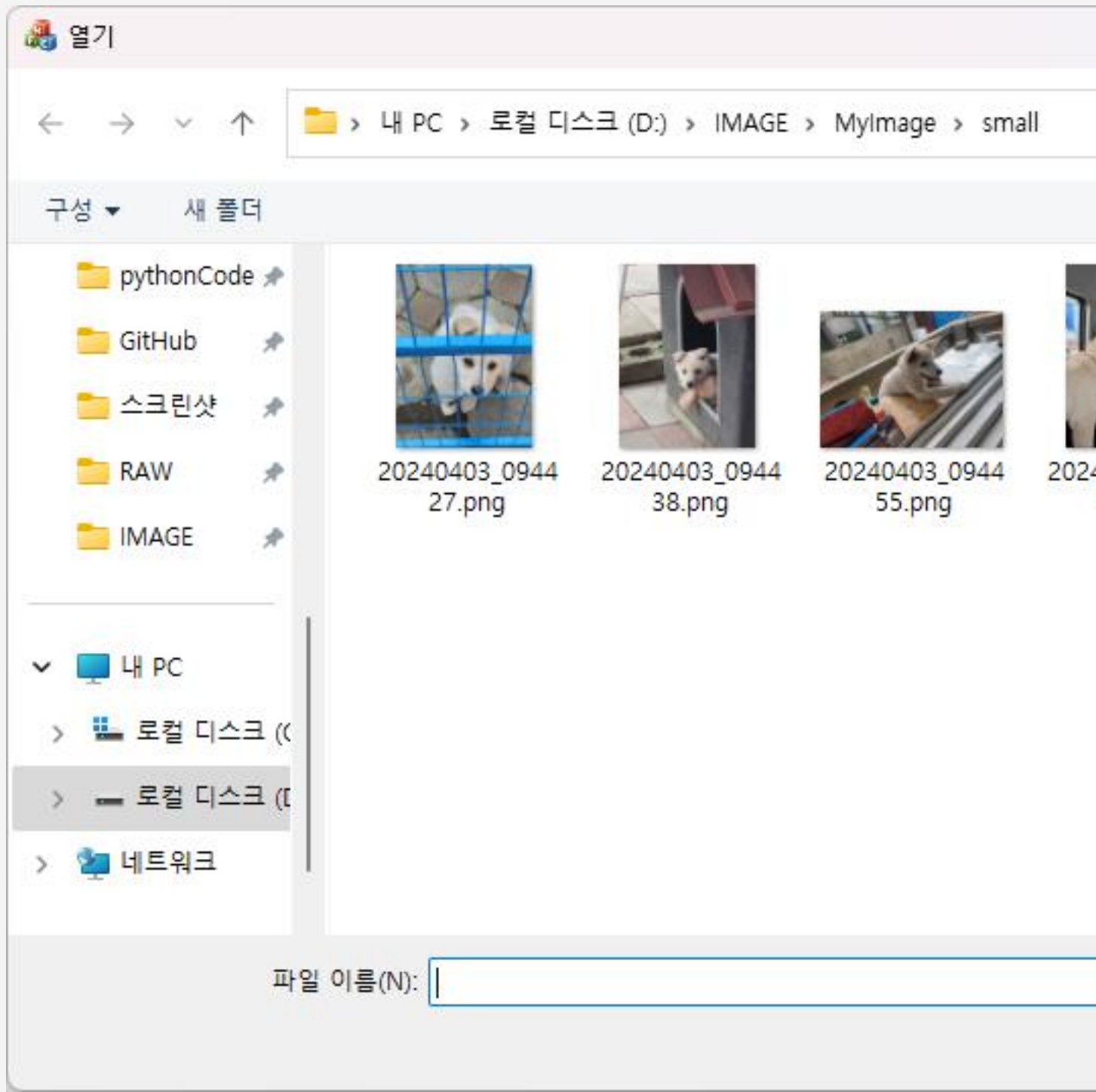
화면 구성 및 기능 초기 화면



화면 구성 및 기능 파일 메뉴



화면 구성 및 기능 파일 > 열기



화면 구성 및 기능

파일 > 저장

이미지를 편집하지 않고 저장

저장 실패



이미지를 편집하고 저장해주세요.

확인

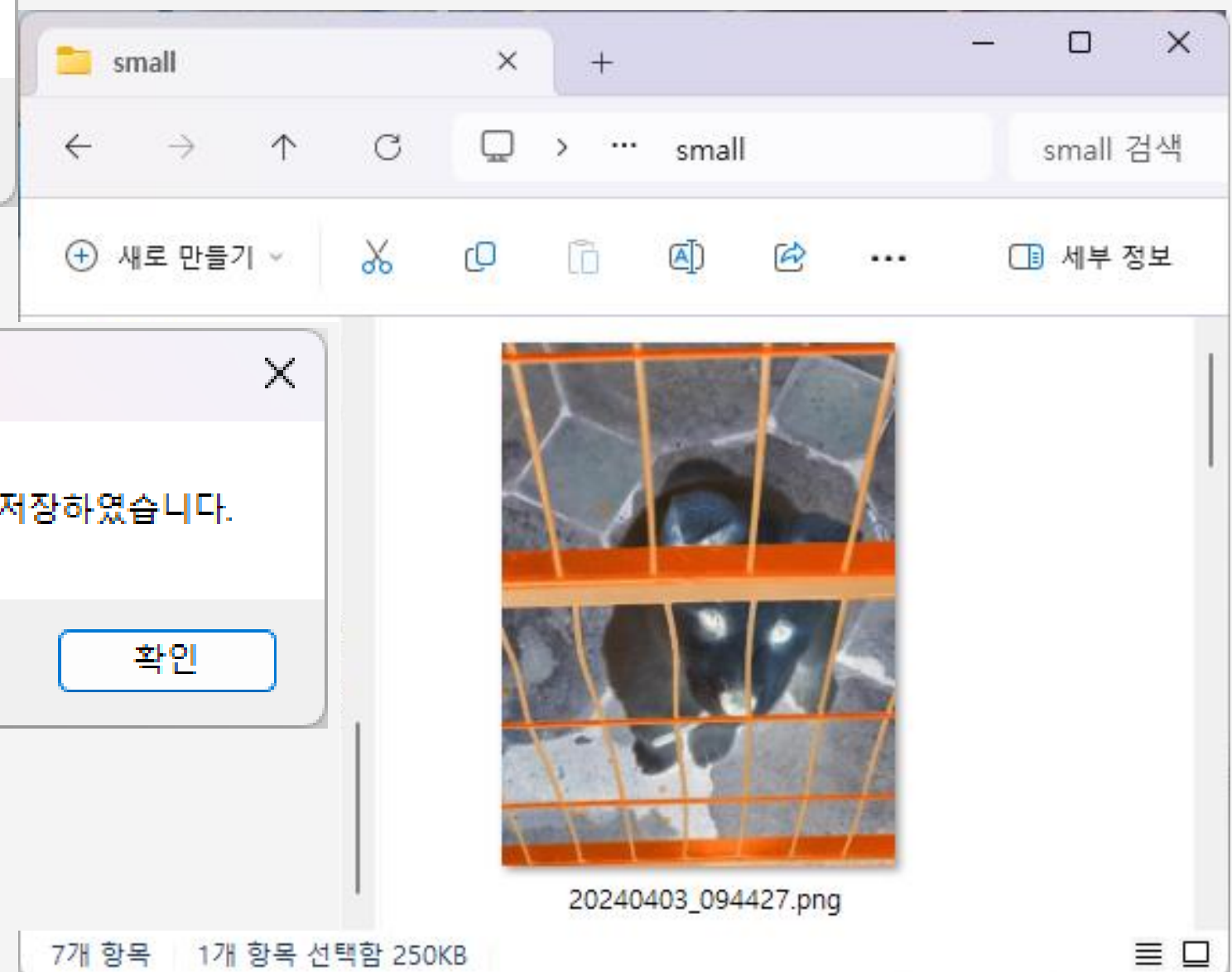
이미지를 편집하고 저장
(임시로 반전 효과)

저장 성공

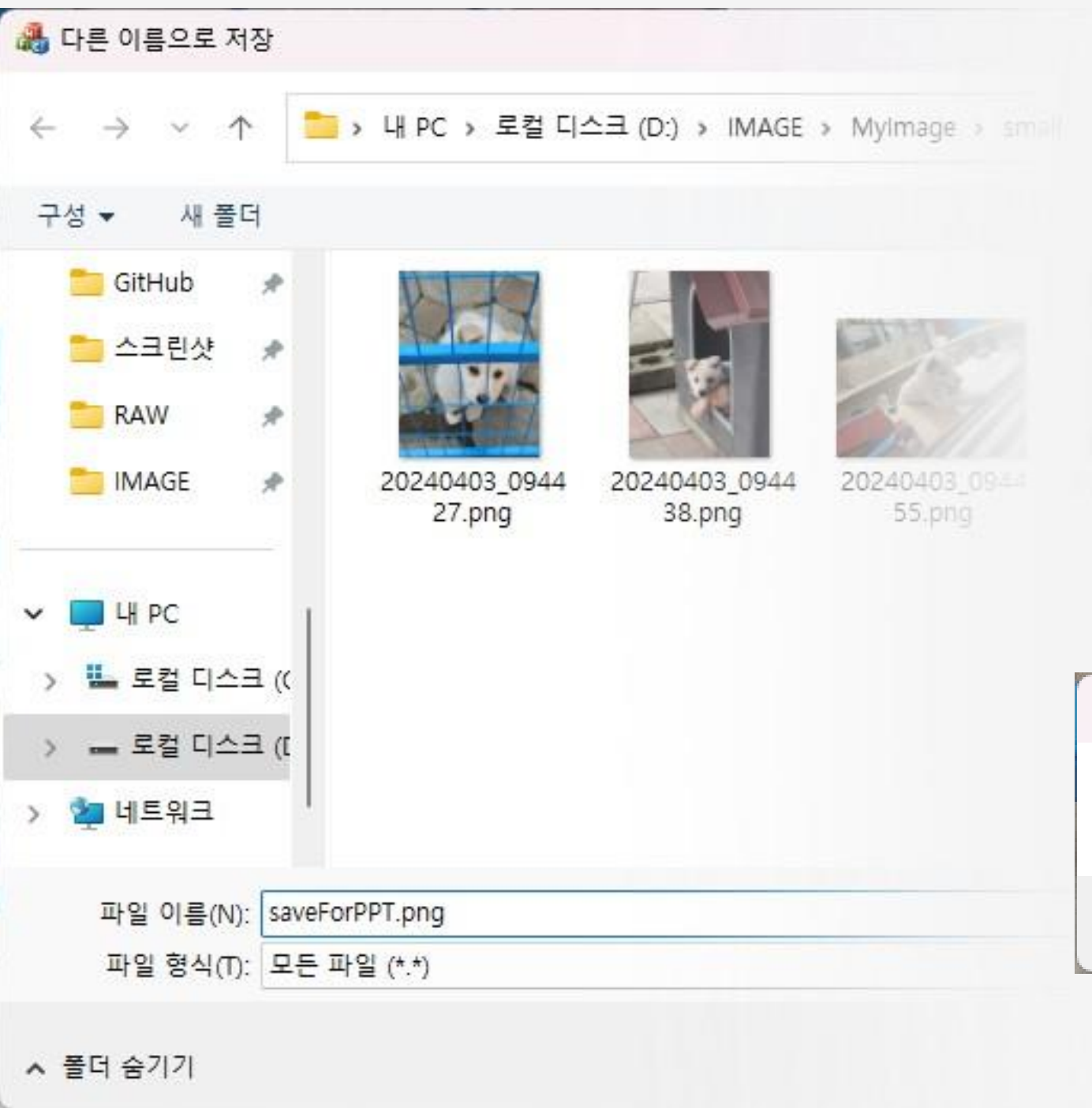


D:\IMAGE\MylImage\small\20240403_094427.png으로 저장하였습니다.

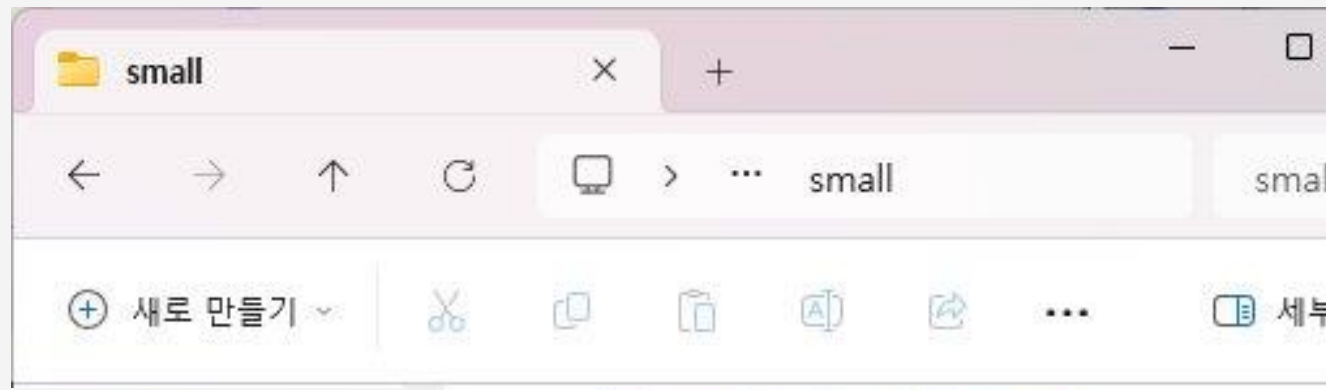
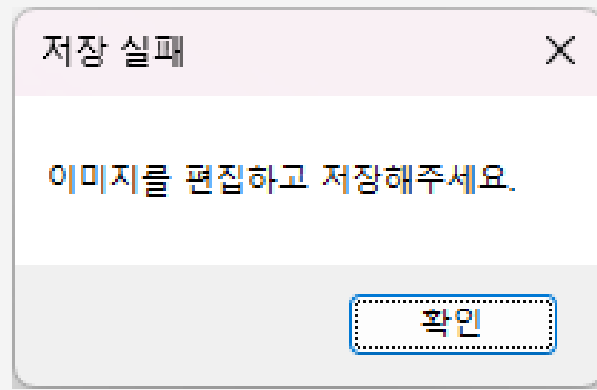
확인



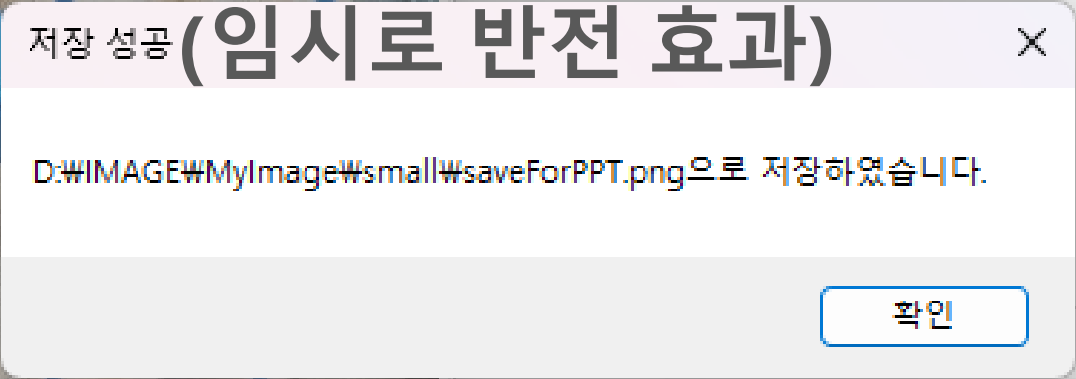
화면 구성 및 기능 파일 > 다른 이름으로 저장



이미지를 편집하지 않고 저장

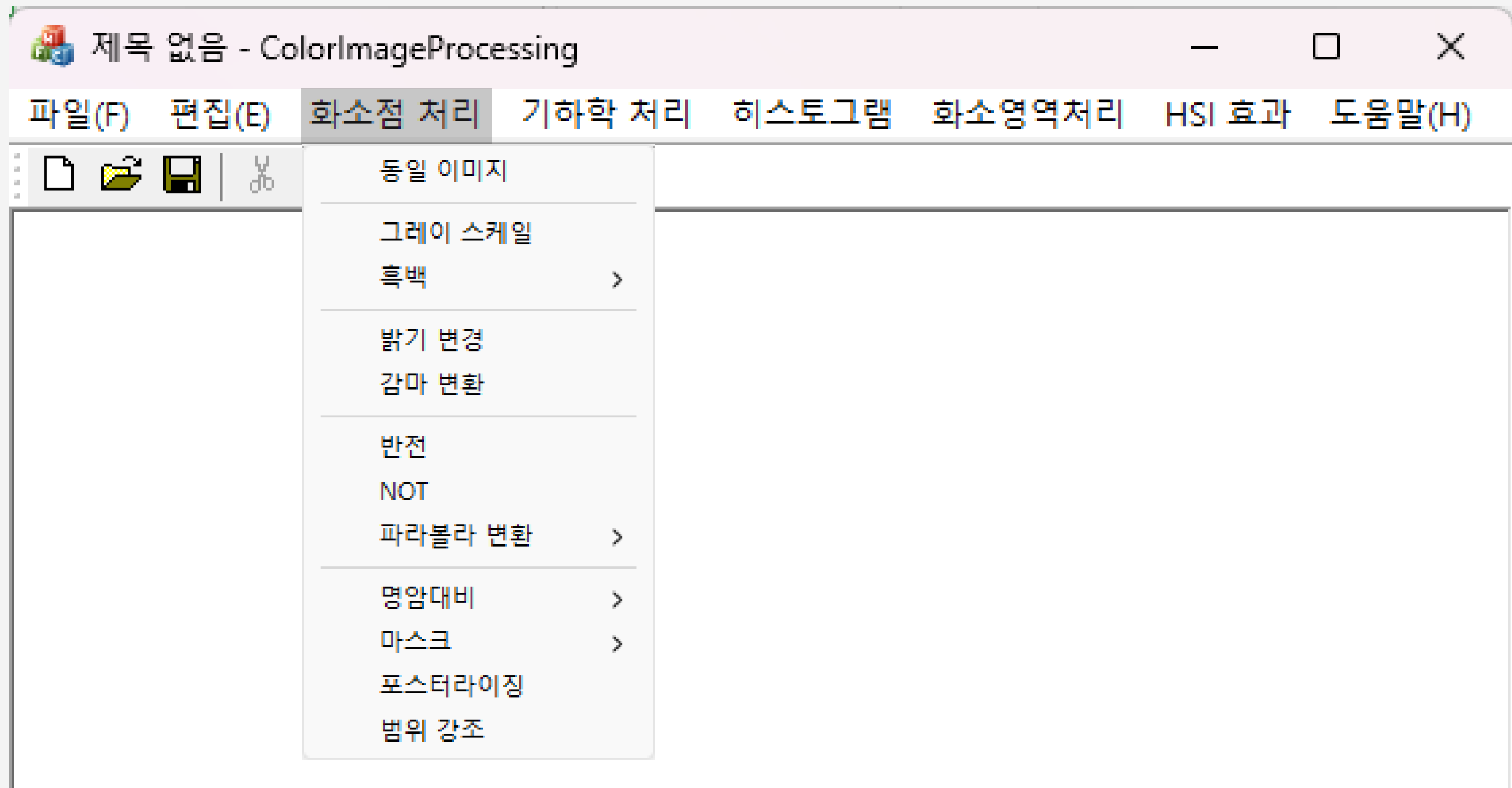


이미지를 편집하고 저장



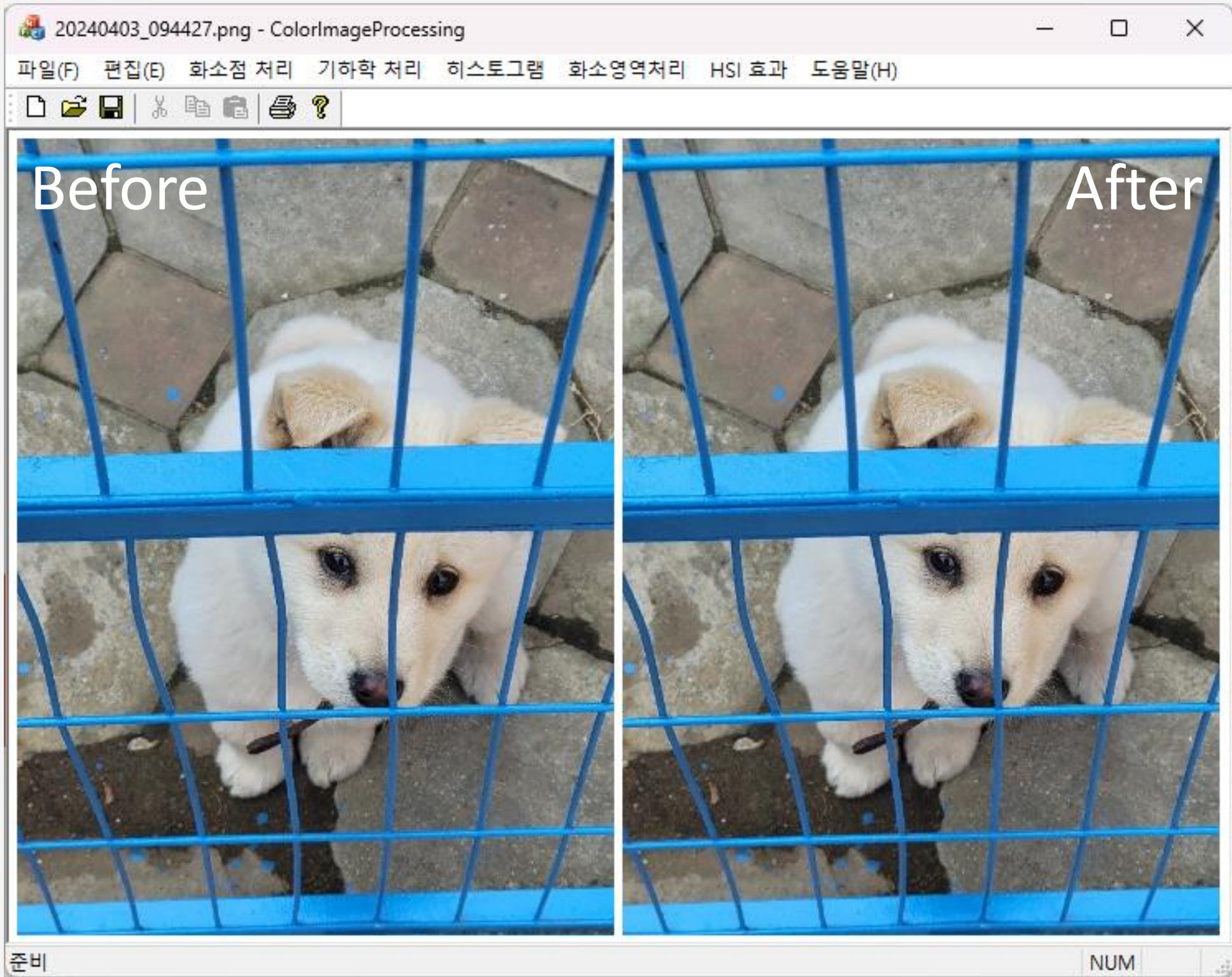
8개 항목

화면 구성 및 기능 화소점처리 메뉴



동일 이미지	반전
그레이 스케일	NOT
흑백	파라볼라 변환
밝기 변경	명암대비
감마 변환	마스크
	포스터라이징
	범위 강조

화면 구성 및 기능 화소점처리 > 동일 이미지



처음 불러온 이미지를 그대로 출력

출력 이미지 = 입력 이미지

화면 구성 및 기능 화소점처리 > 그레이 스케일



흔히 아는 흑백 이미지

사실 회색이 제일 많아
원래 이름은 그레이 스케일

출력 이미지 = 입력 이미지의 RGB 평균값

- 동일 이미지
- 그레이 스케일
- 흑백**
- 밝기 변경
- 감마 변환
- 반전
- NOT
- 파라볼라 변환
- 명암대비
- 마스크
- 포스터라이징
- 범위 강조

화면 구성 및 기능 화소점처리 > 흑백 흑과 백으로만 이루어진 이미지

Before



기준?
127



평균값

평균값

×

평균값: 132

확인



중간값

중간값

×

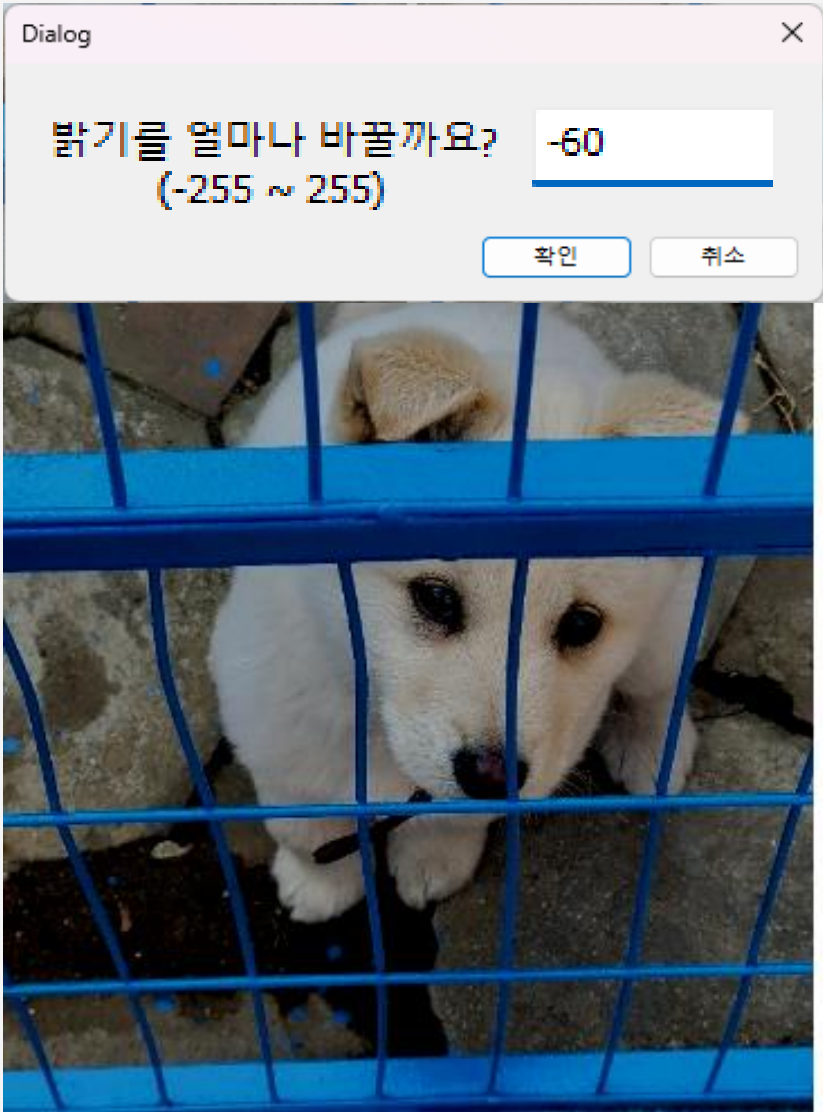
중간값: 154

확인



출력 이미지 = 입력 이미지 값이 기준보다 큰지 작은지

화면 구성 및 기능 화소점처리 > 밝기 변경



빛의 세기는 값이 클수록 밝음

출력 이미지 = 입력 이미지 값 + 입력 받은 값

화면 구성 및 기능 화소점처리 > 감마값 변경

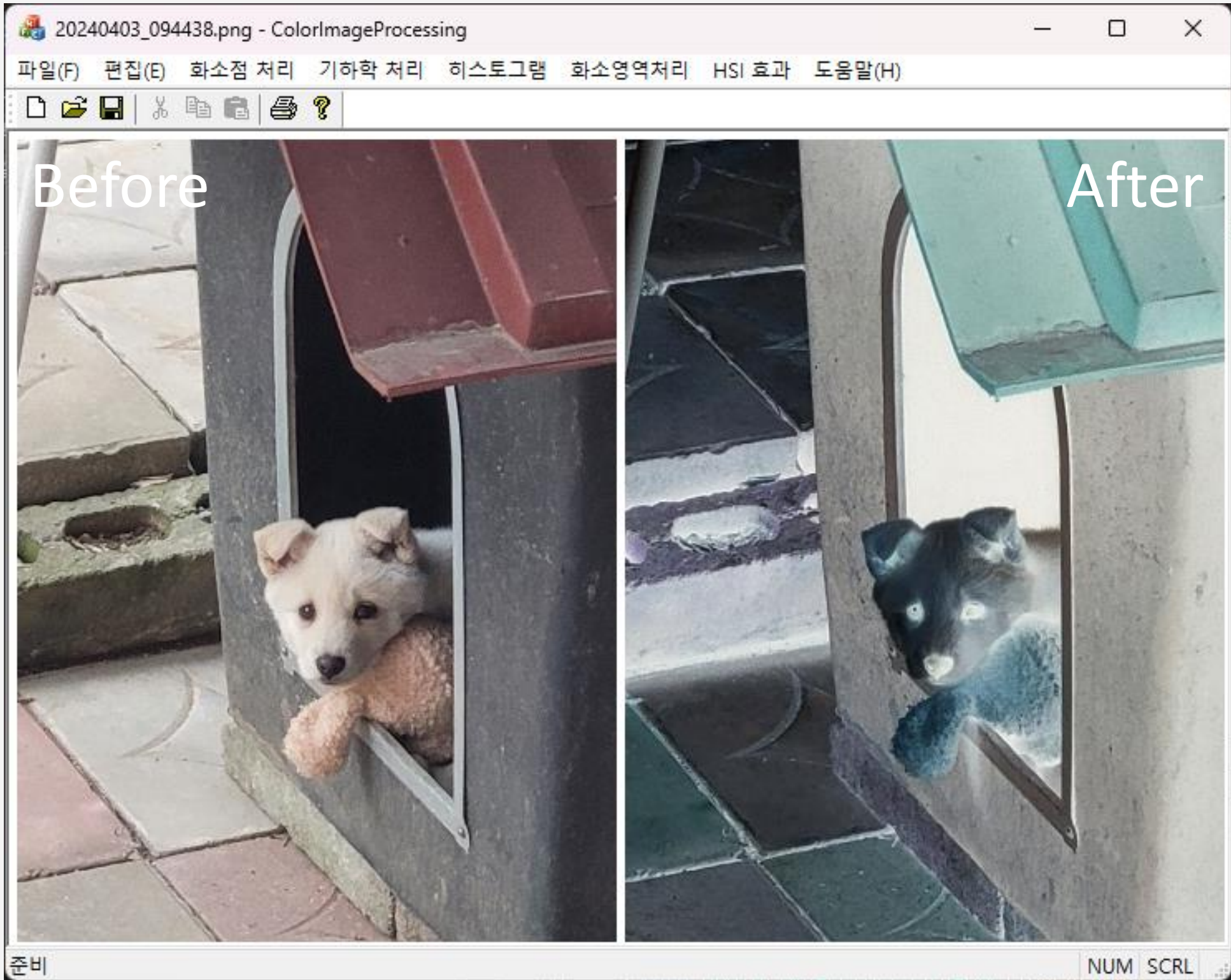


사람 눈이 왜곡하는 밝기를
 조정하기 위한 기능

$$\text{출력 이미지} = 255 \times \left(\frac{\text{입력 이미지 값}}{255} \right)^{\frac{30}{\text{입력값}}}$$

동일 이미지	반전
그레이 스케일	NOT
흑백	파라볼라 변환
밝기 변경	명암대비
감마 변환	마스크
	포스터라이징
	범위 강조

화면 구성 및 기능 화소점처리 > 반전

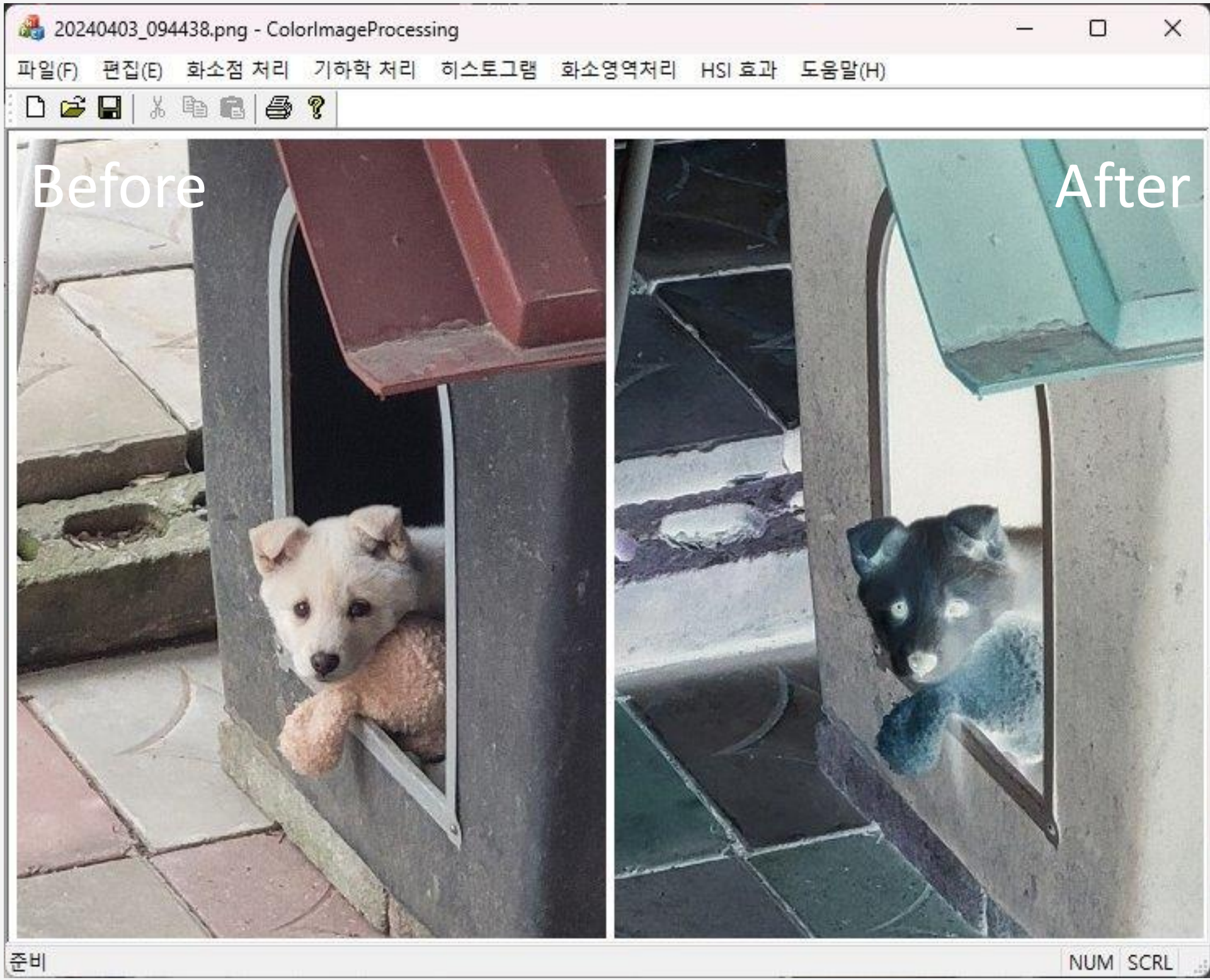


0을 255로
255를 2으로

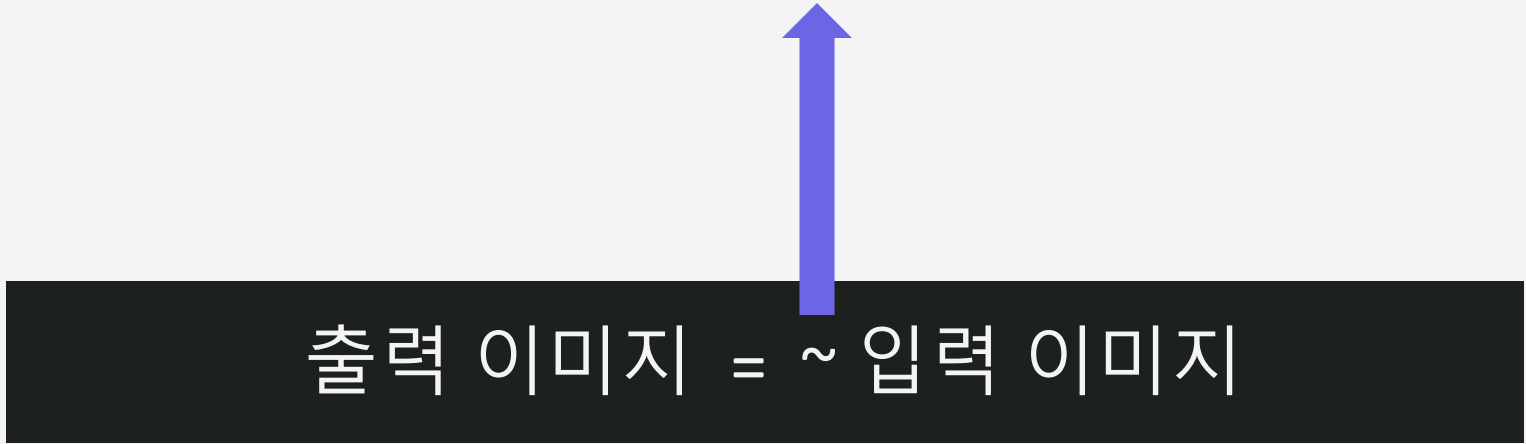
출력 이미지 = 255 - 입력 이미지

동일 이미지	반전
그레이 스케일	NOT
흑백	파라볼라 변환
밝기 변경	명암대비
감마 변환	마스크
	포스터라이징
	범위 강조

화면 구성 및 기능 화소점처리 > NOT 반전



이진수 기준으로 0을 1으로
01010101을 10101010으로



동일 이미지	반전
그레이 스케일	NOT
흑백	파라볼라 변환
밝기 변경	명암대비
감마 변환	마스크
	포스터라이징
	범위 강조

화면 구성 및 기능 화소점처리 > 파라볼라 변환



사진의 일부를 입체적이게

CAP-밝은데 CUP-어두운데

CAP 수식

$$\text{CUP 출력 이미지} = 255 - 255 \times \left(\frac{\text{입력 이미지 값}}{128} \right)^2$$

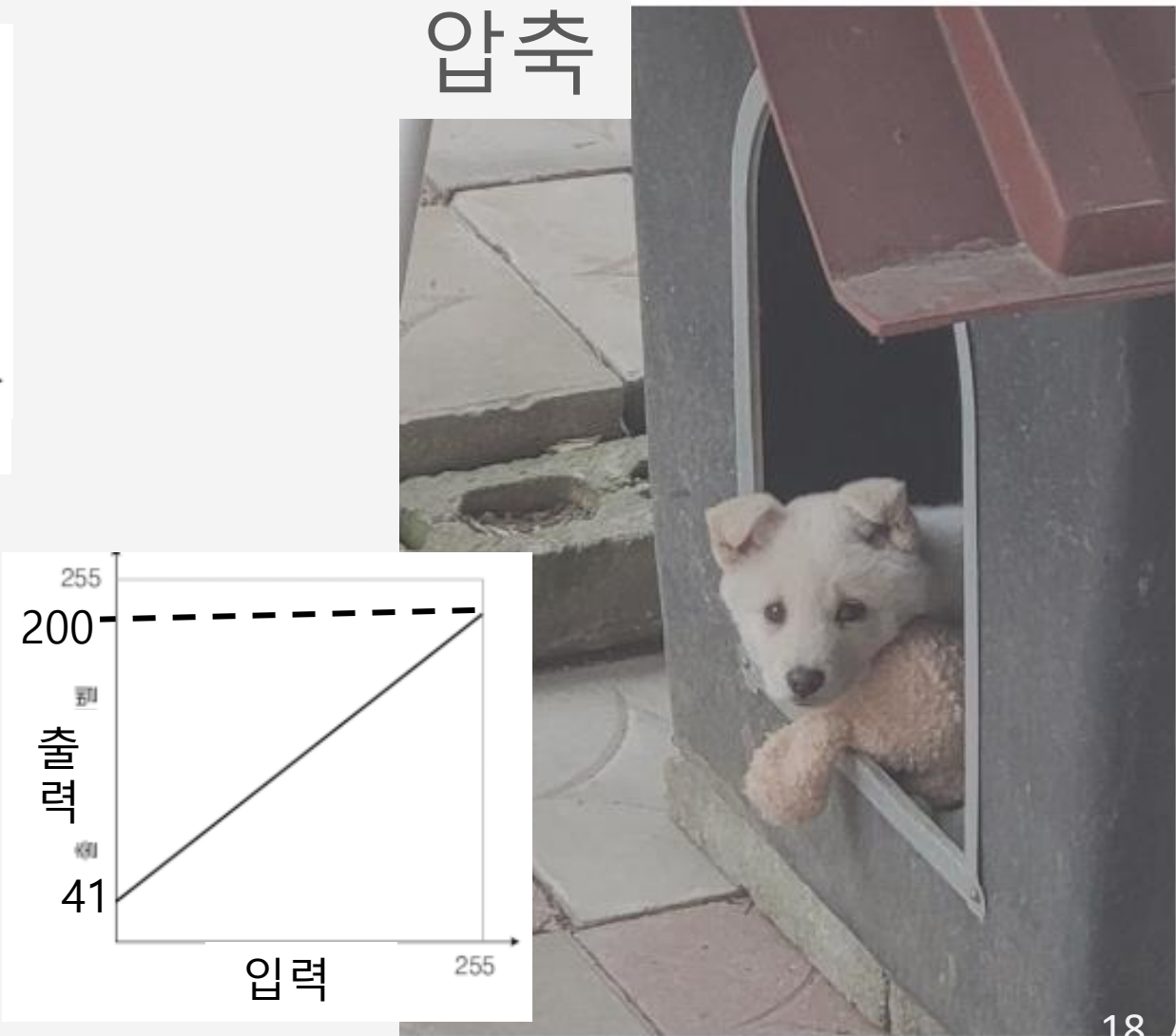
동일 이미지	반전
그레이 스케일	NOT
흑백	파라볼라 변환
밝기 변경	명암대비
감마 변환	마스크
	포스터라이징
	범위 강조

화면 구성 및 기능 화소점처리 > 명암대비 ~



값을 식으로 변경

스트레칭-선명하게 압축-흐리게



동일 이미지	반전
그레이 스케일	NOT
흑백	파라볼라 변환
밝기 변경	명암대비
감마 변환	마스크
	포스터라이징
	범위 강조

화면 구성 및 기능 화소점처리 > 마스크

두 사진을 bit 연산을 통해서 합침



Before



AND



OR



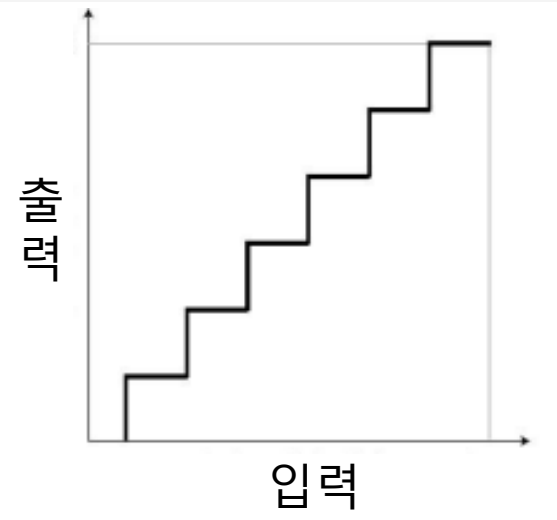
XOR



동일 이미지	반전
그레이 스케일	NOT
흑백	파라볼라 변환
밝기 변경	명암대비
감마 변환	마스크
	포스터라이징
	범위 강조

화면 구성 및 기능

화소점처리 > 포스터라이징



0~255를 단계별로

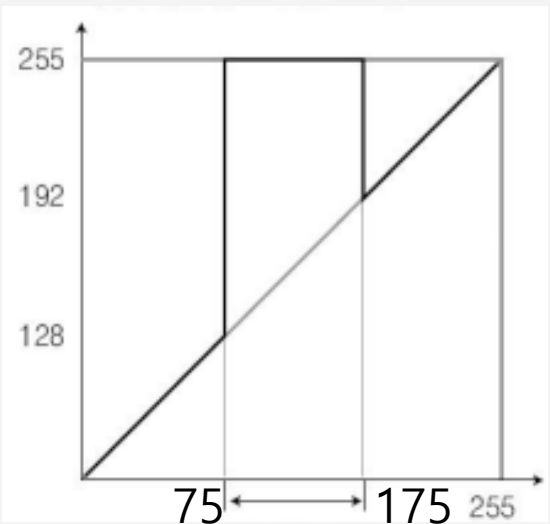
8단계로 비교적 촘촘하게 나눔

출력 이미지 = (입력 이미지 / 32) 단계 값

32 x 8 = 256

동일 이미지	반전
그레이 스케일	NOT
흑백	파라볼라 변환
밝기 변경	명암대비
감마 변환	마스크
	포스터라이징
	범위 강조

화면 구성 및 기능 화소점처리 > 범위 강조



75~175 사이 = 255

8단계로 비교적 촘촘하게 나눔

입력 이미지가 75 ~ 175 사이

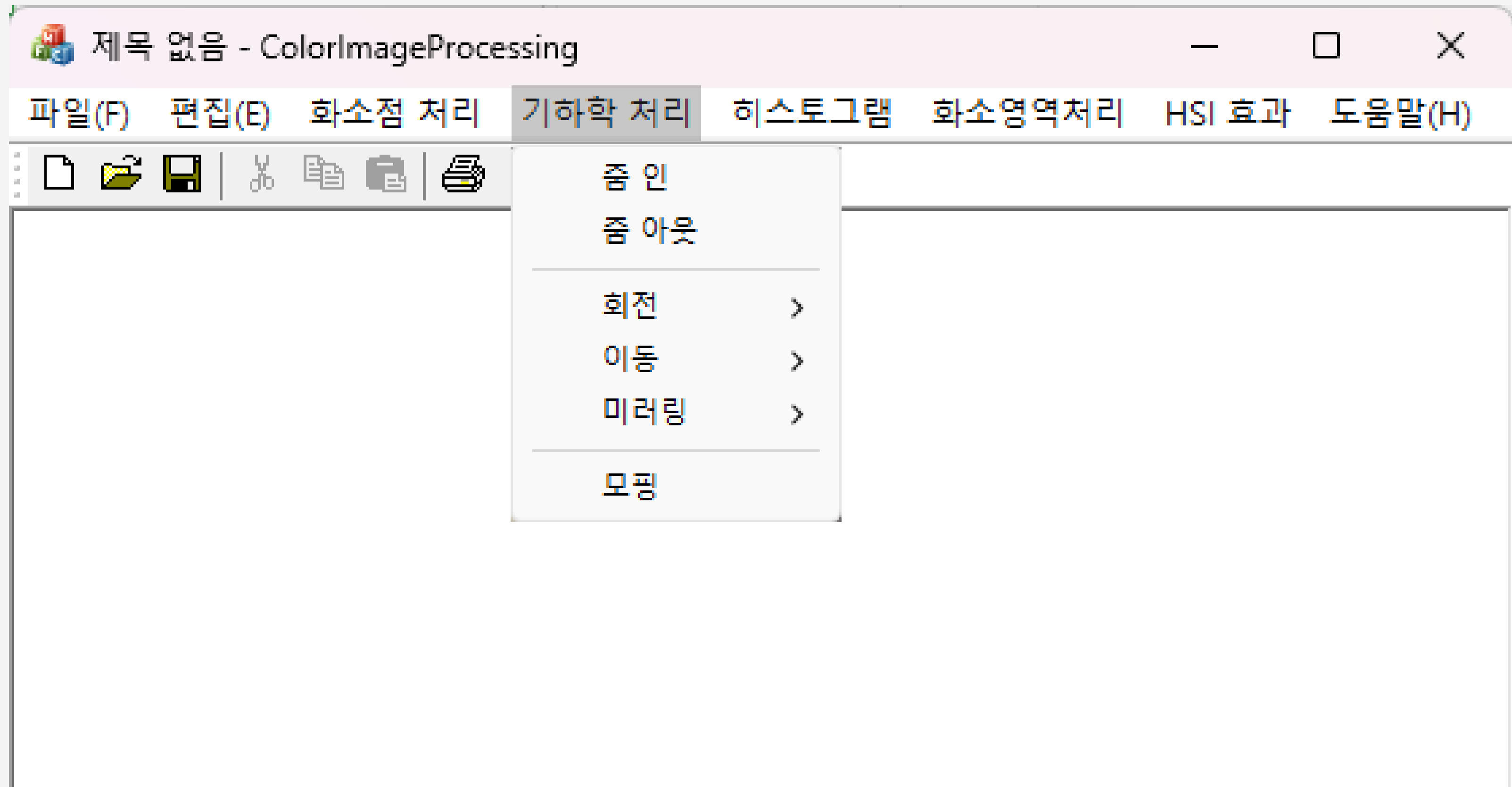
↗

↘

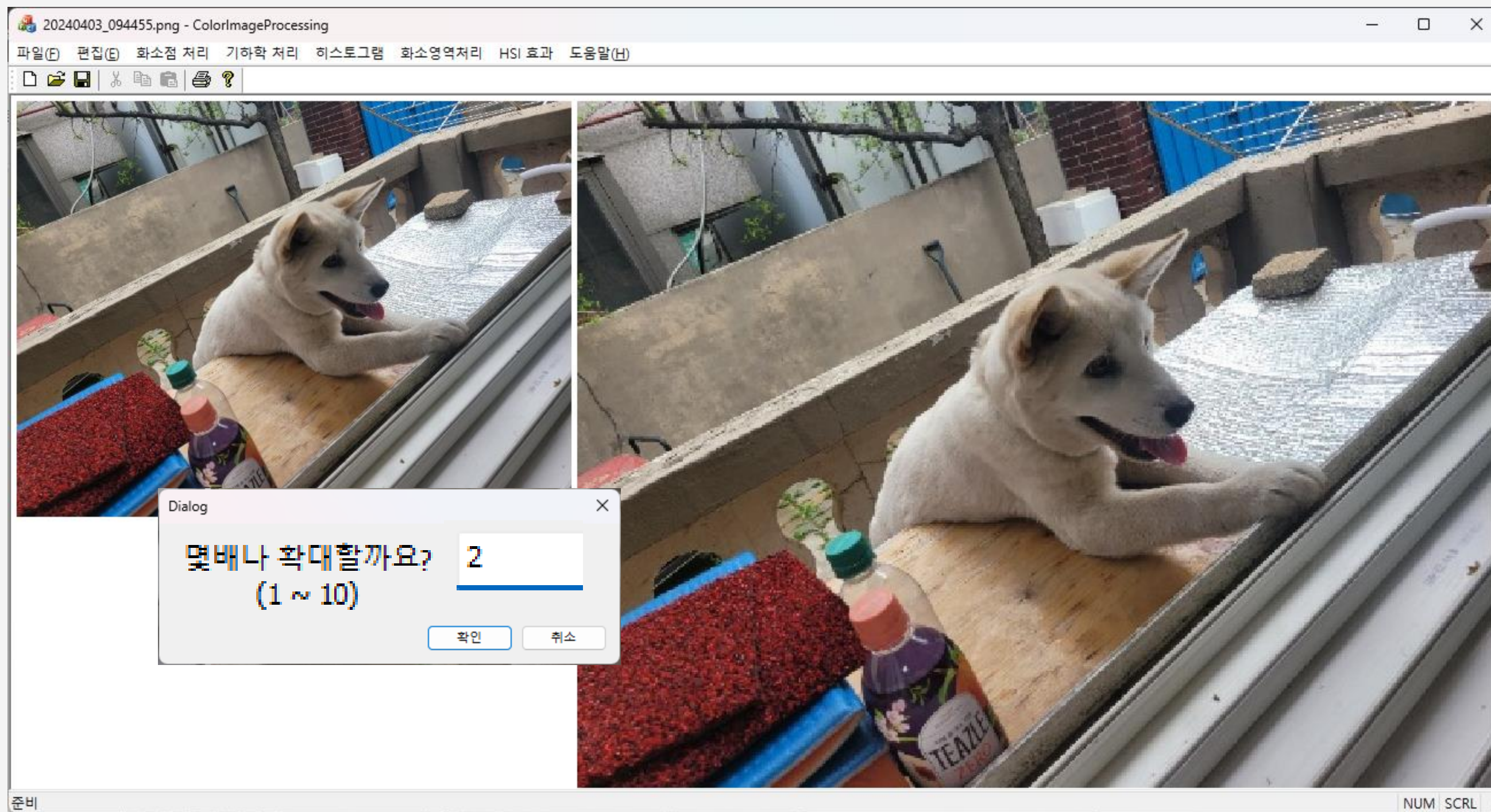
맞음) 출력 이미지 = 255

아님) 출력 이미지 = 0

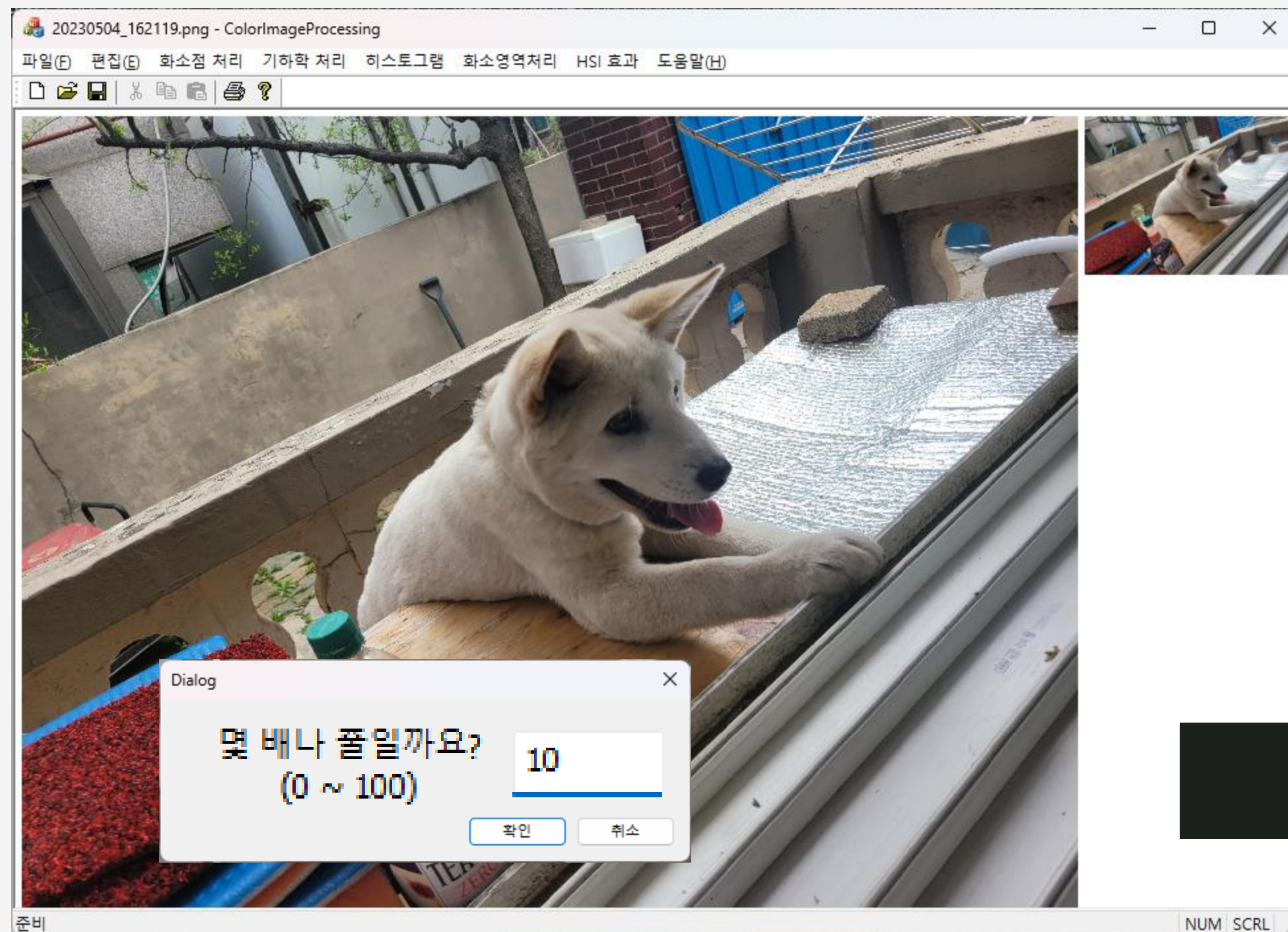
화면 구성 및 기능 기하학 처리 메뉴



화면 구성 및 기능 기하학처리 > 줌인



화면 구성 및 기능 기하학처리 > 줌아웃



전 후 크기가 10배?

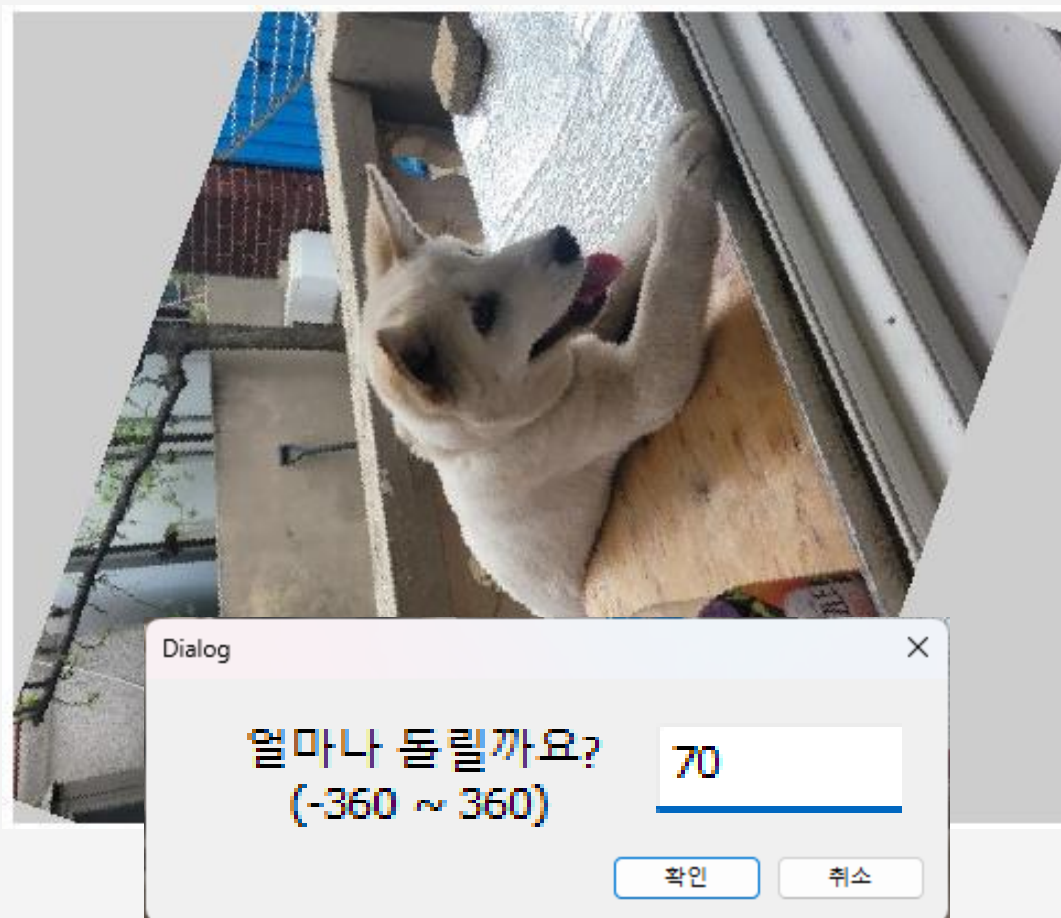
원본이미지 크기
1612 x 1209

화면에 잘리지 않게 나오도록
출력 시에만 크기 조정

식은 코드 참조

화면 구성 및 기능

기하학처리 > 회전



낮설고도 익숙한 삼각함수

cos sin
-sin cos

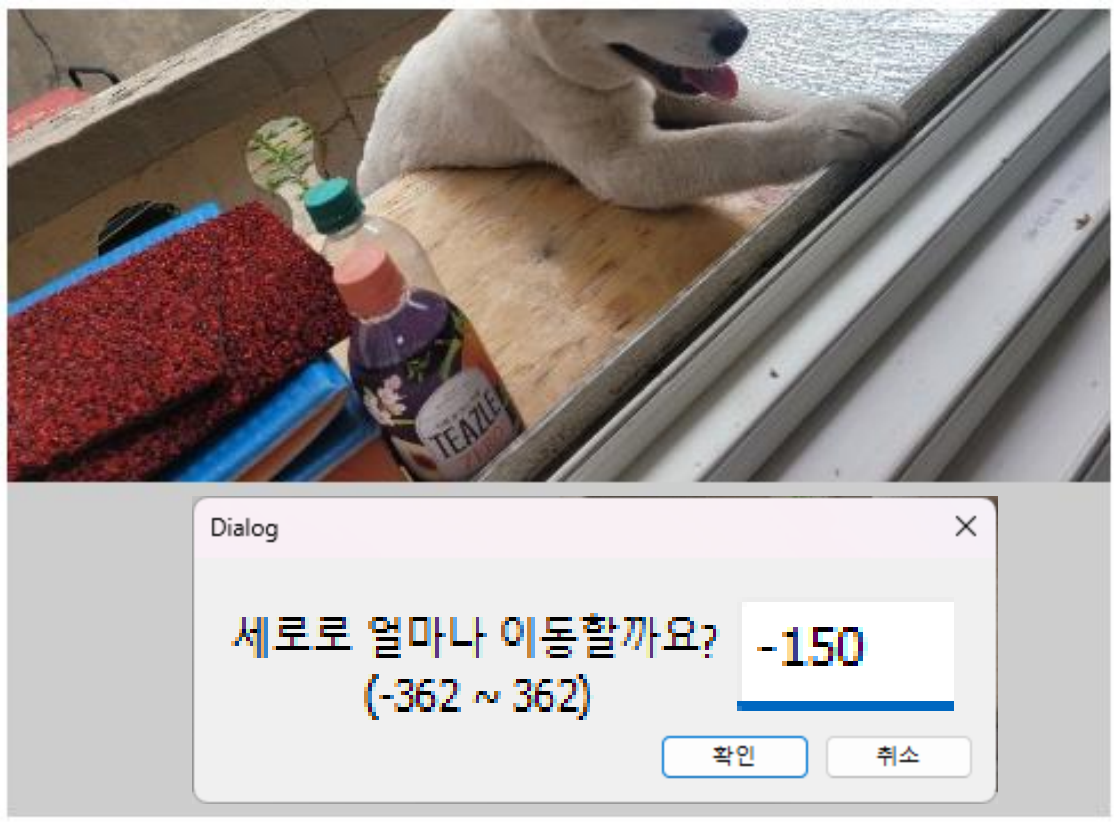
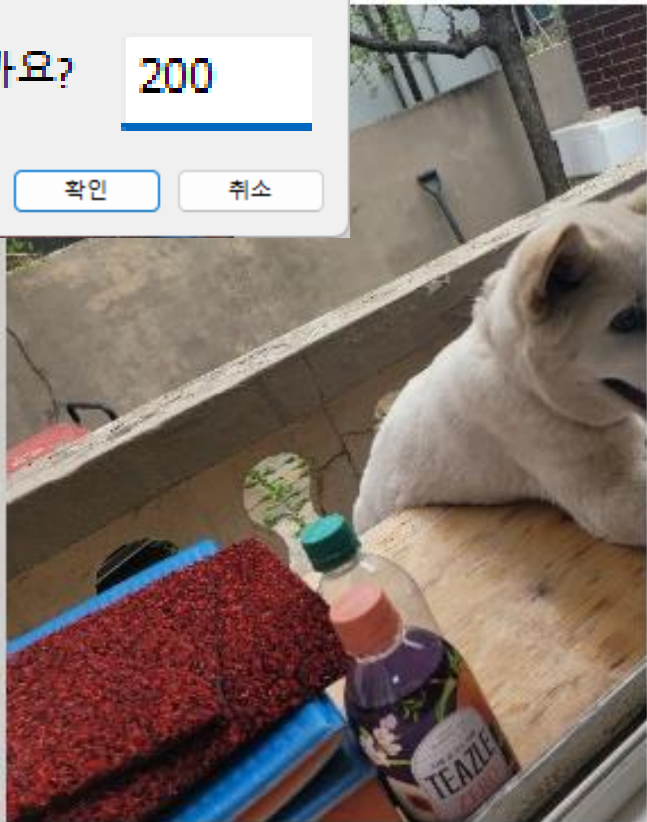
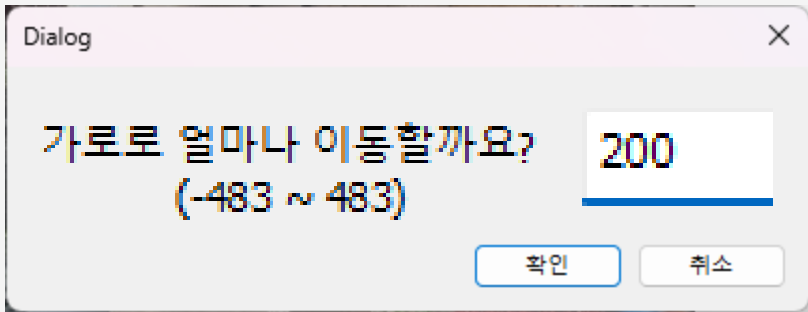
식은 코드 참조

화면 구성 및 기능 기하학처리 > 이동



이동의 범위 확인이 어렵다는 점
 원본이미지의 크기에 따라 출력 변경

이미지
 사진 크기 483 x 362



가로 출력의 가로 칸 + 입력 값
 = 입력의 가로 칸
 세로 출력의 세로 칸 + 입력 값
 = 입력의 세로 칸

화면 구성 및 기능 기하학처리 > 이동



가로 출력의 가로 칸 - 입력할 칸
= 입력의 가로 칸

세로 출력의 세로 칸 - 입력할 칸
= 입력의 세로 칸

화면 구성 및 기능 기하학처리 > 모핑

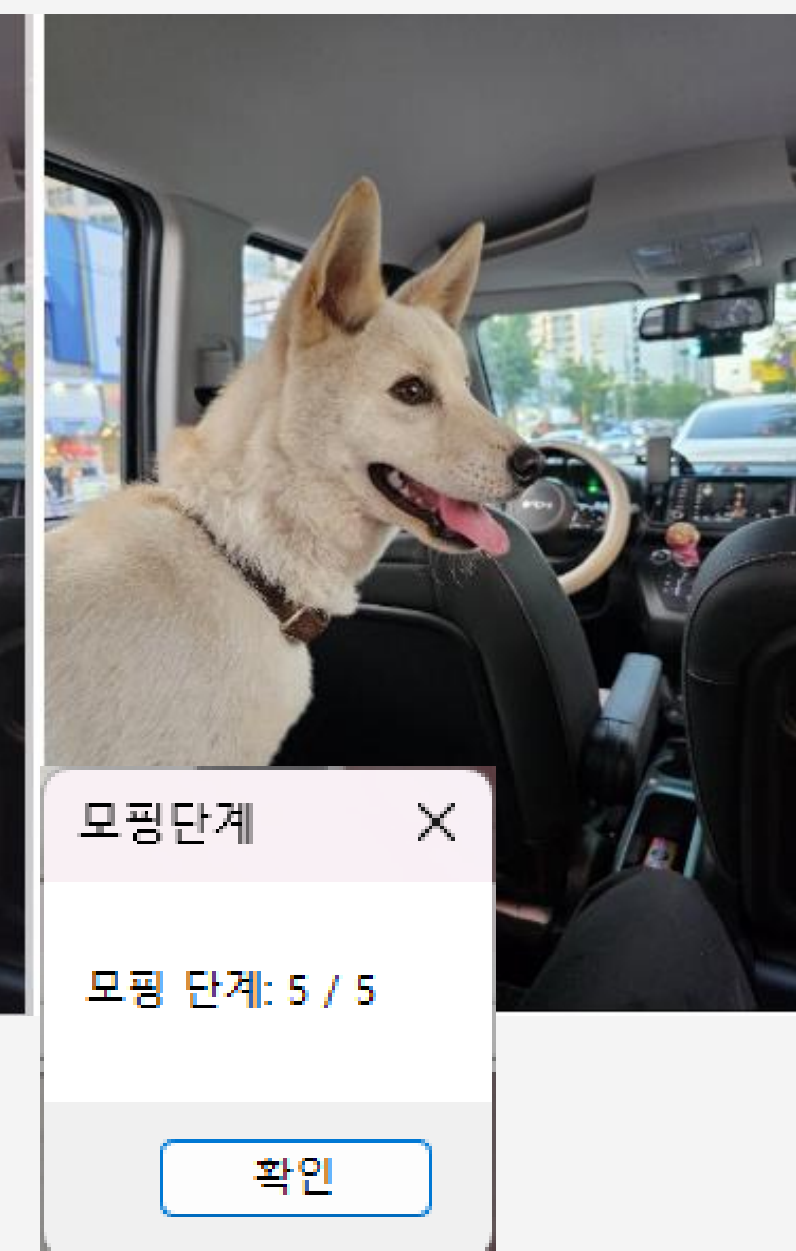
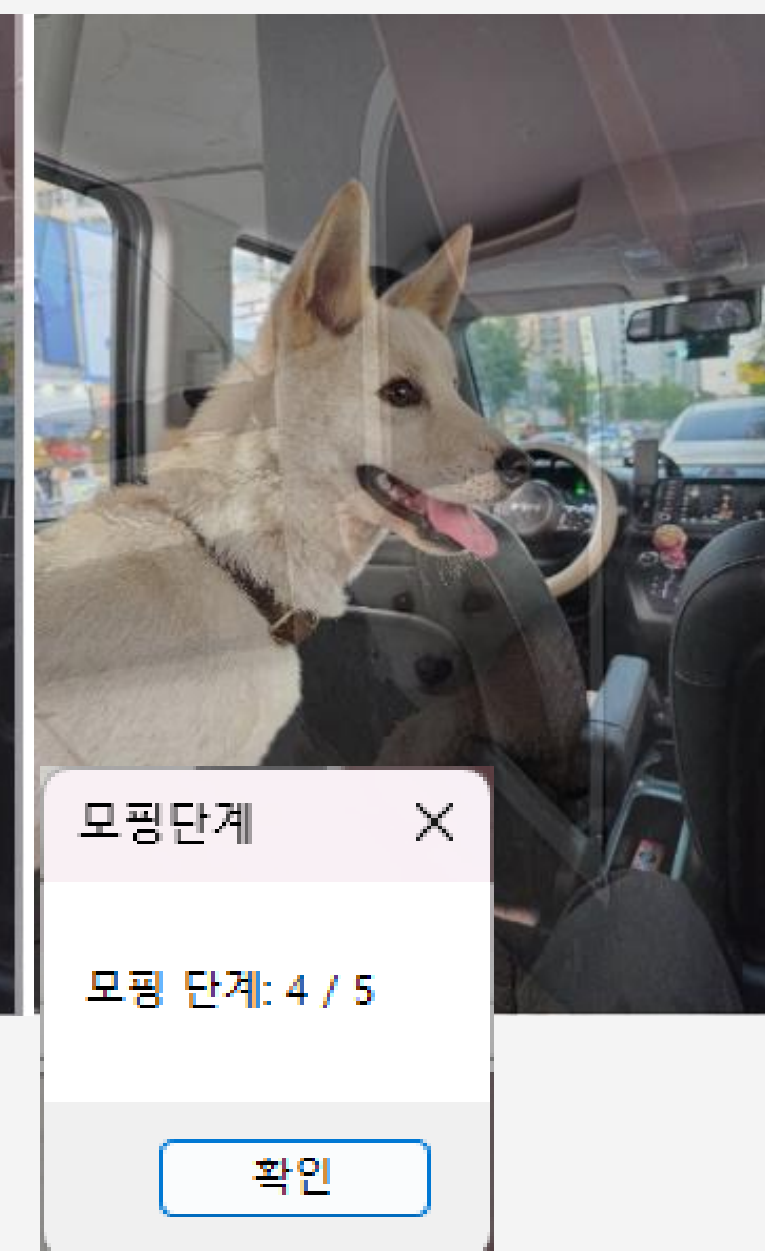
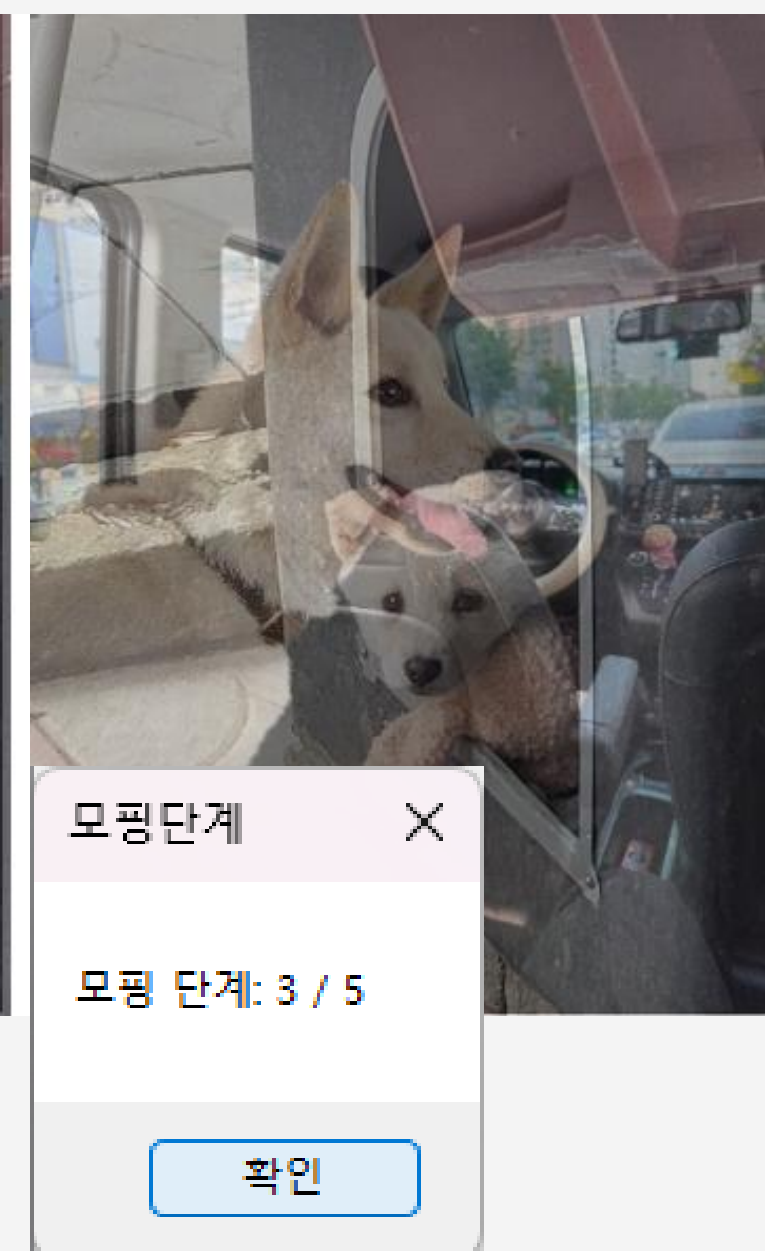
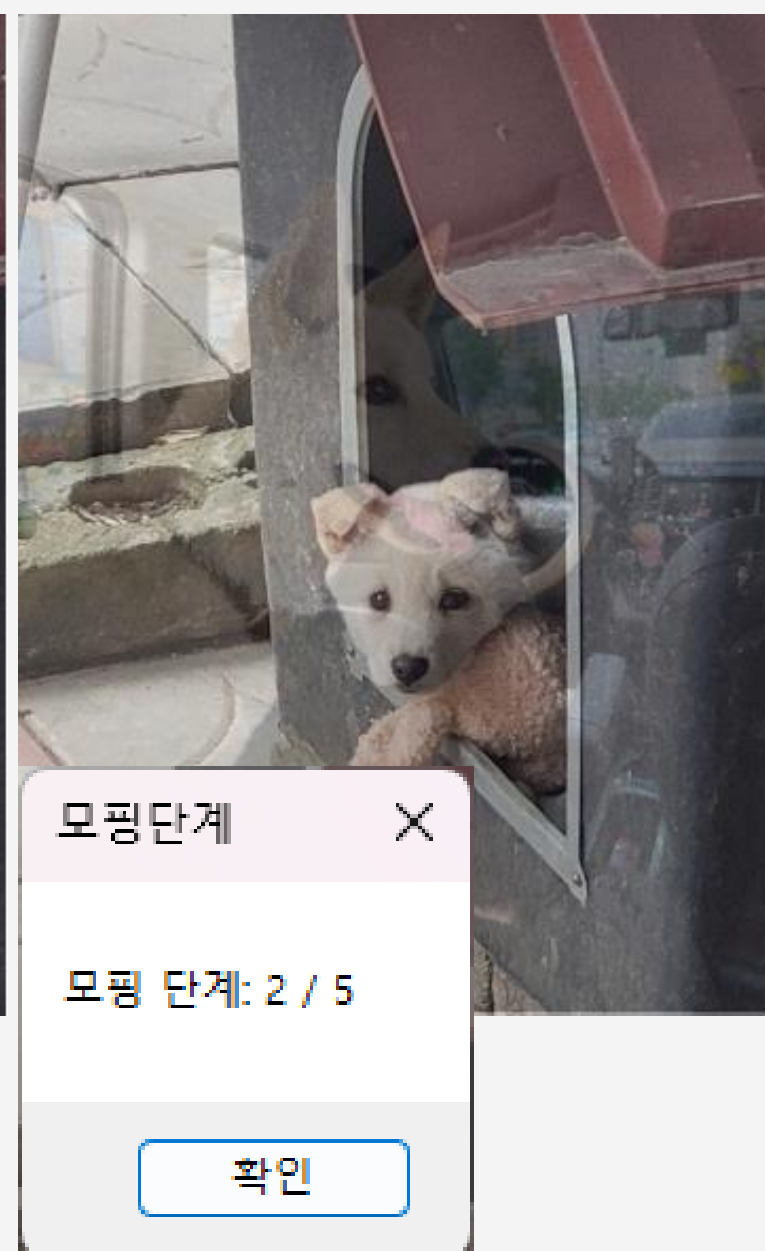
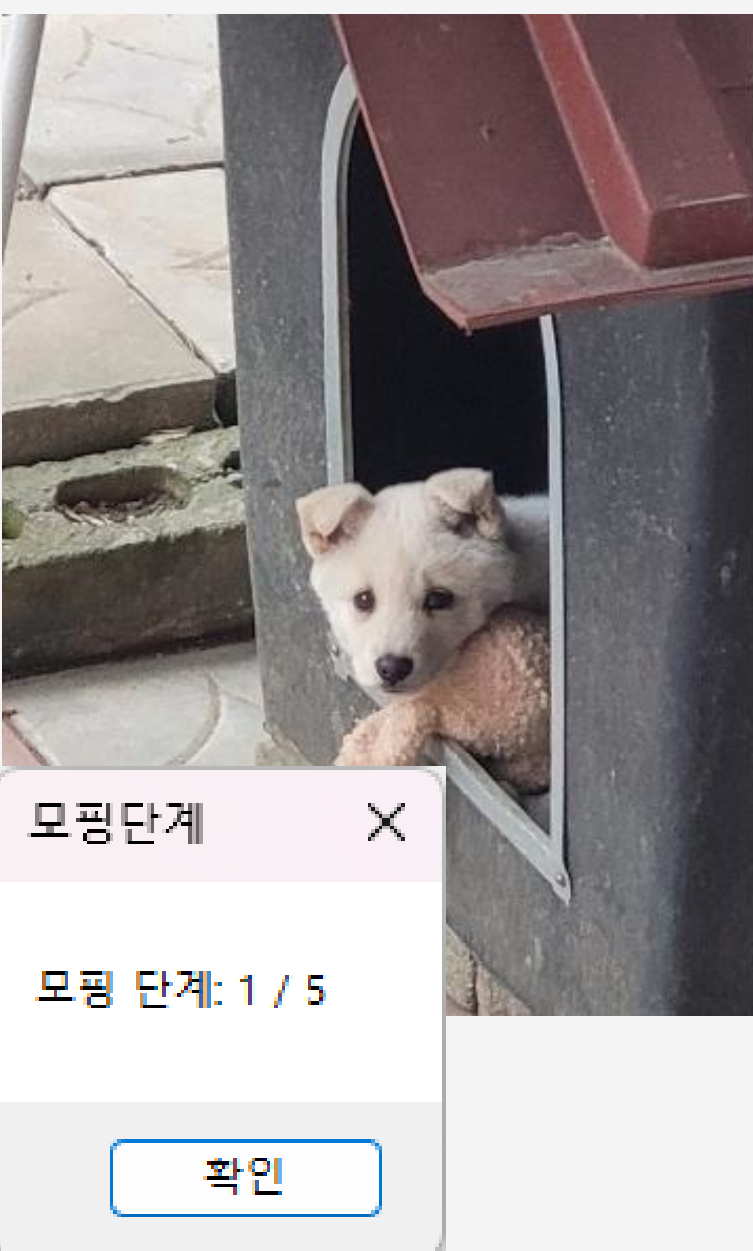
Dialog

모핑을 몇 단계로 할까요?
(2 ~ 100)

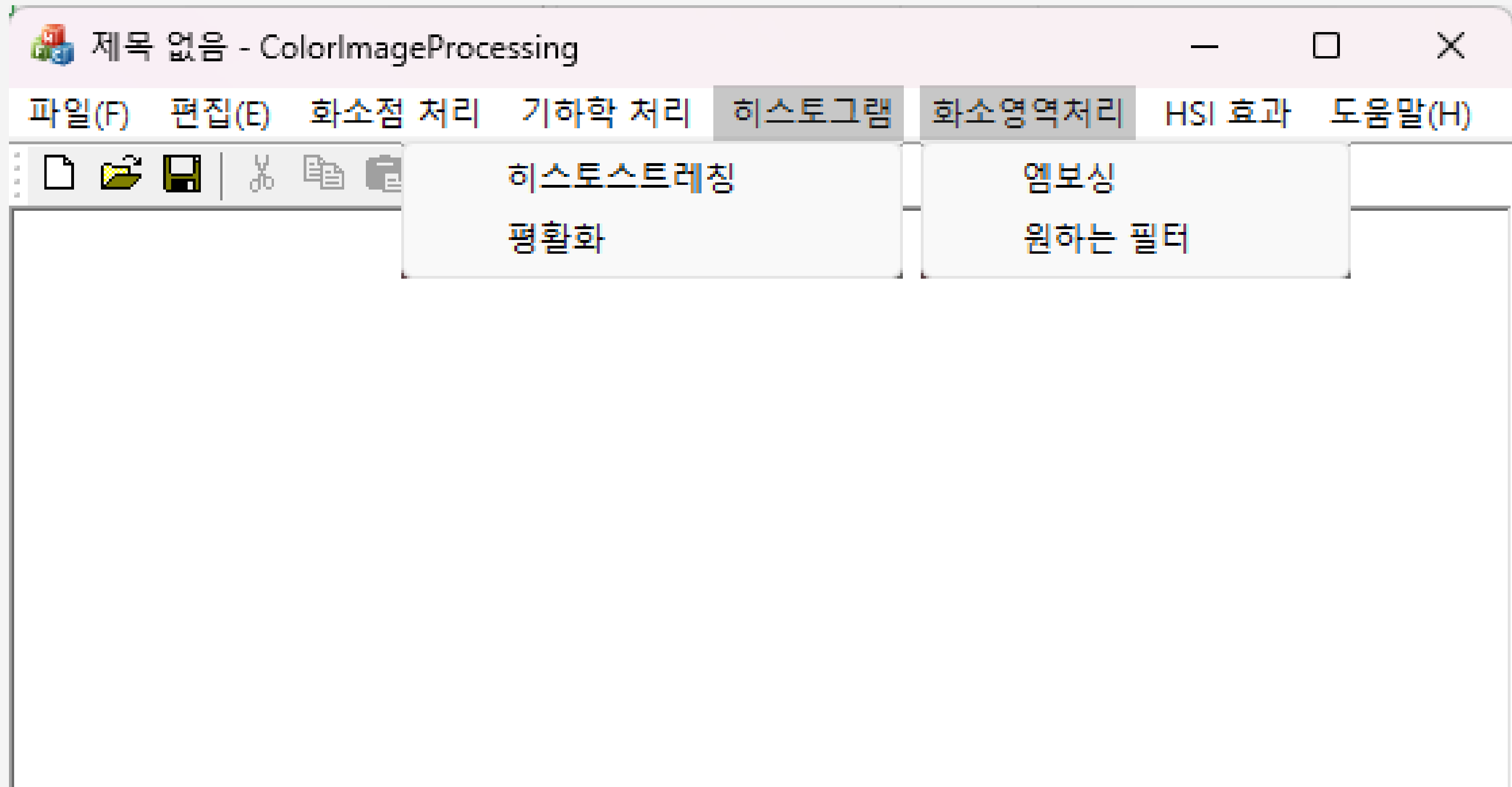
5

확인취소

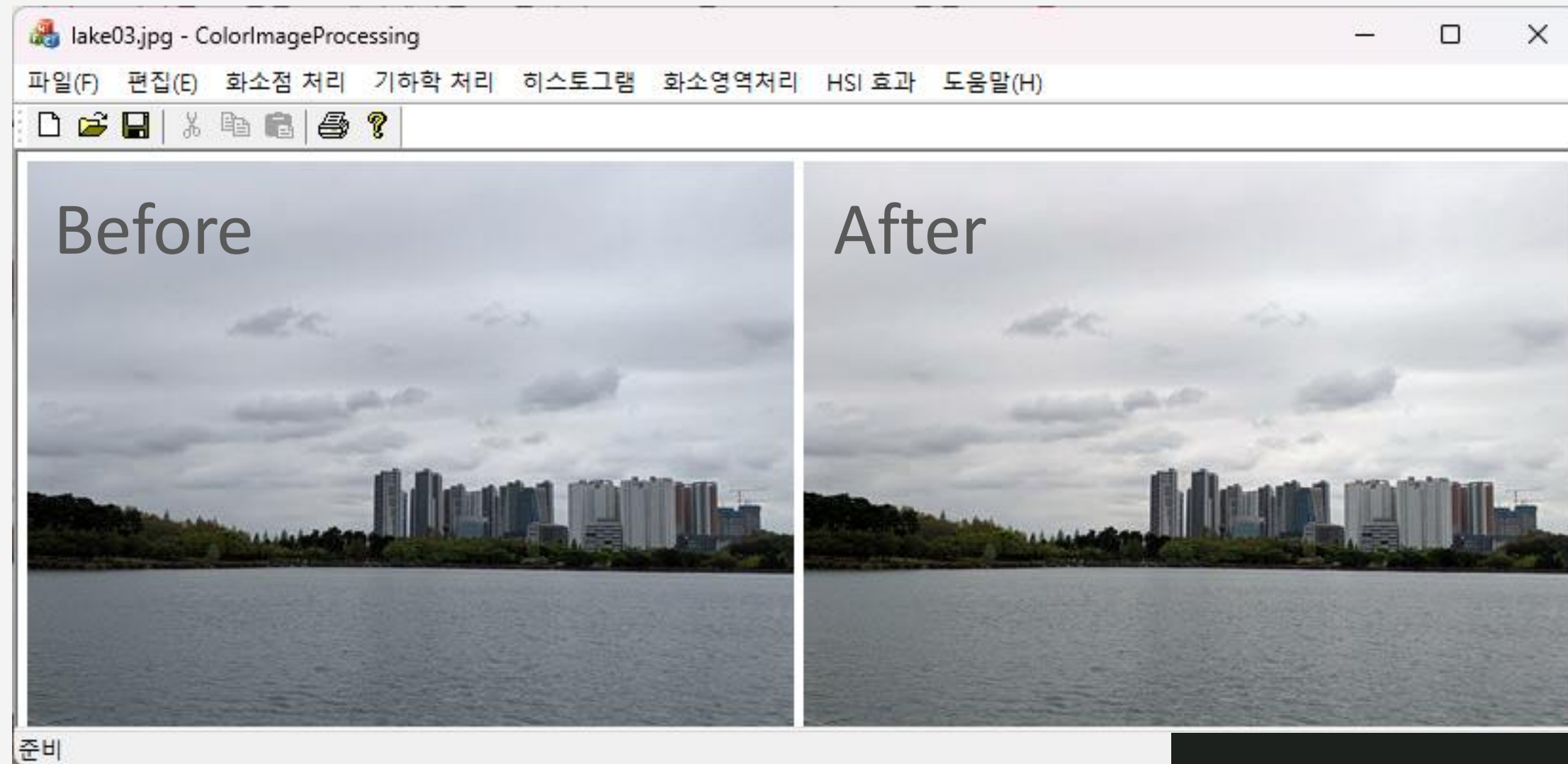
줌 인
줌 아웃
회전
이동
미러링
모핑



화면 구성 및 기능 히스토그램 & 화소 영역 처리 메뉴



화면 구성 및 기능 히스토그램 > 히스토스트레칭

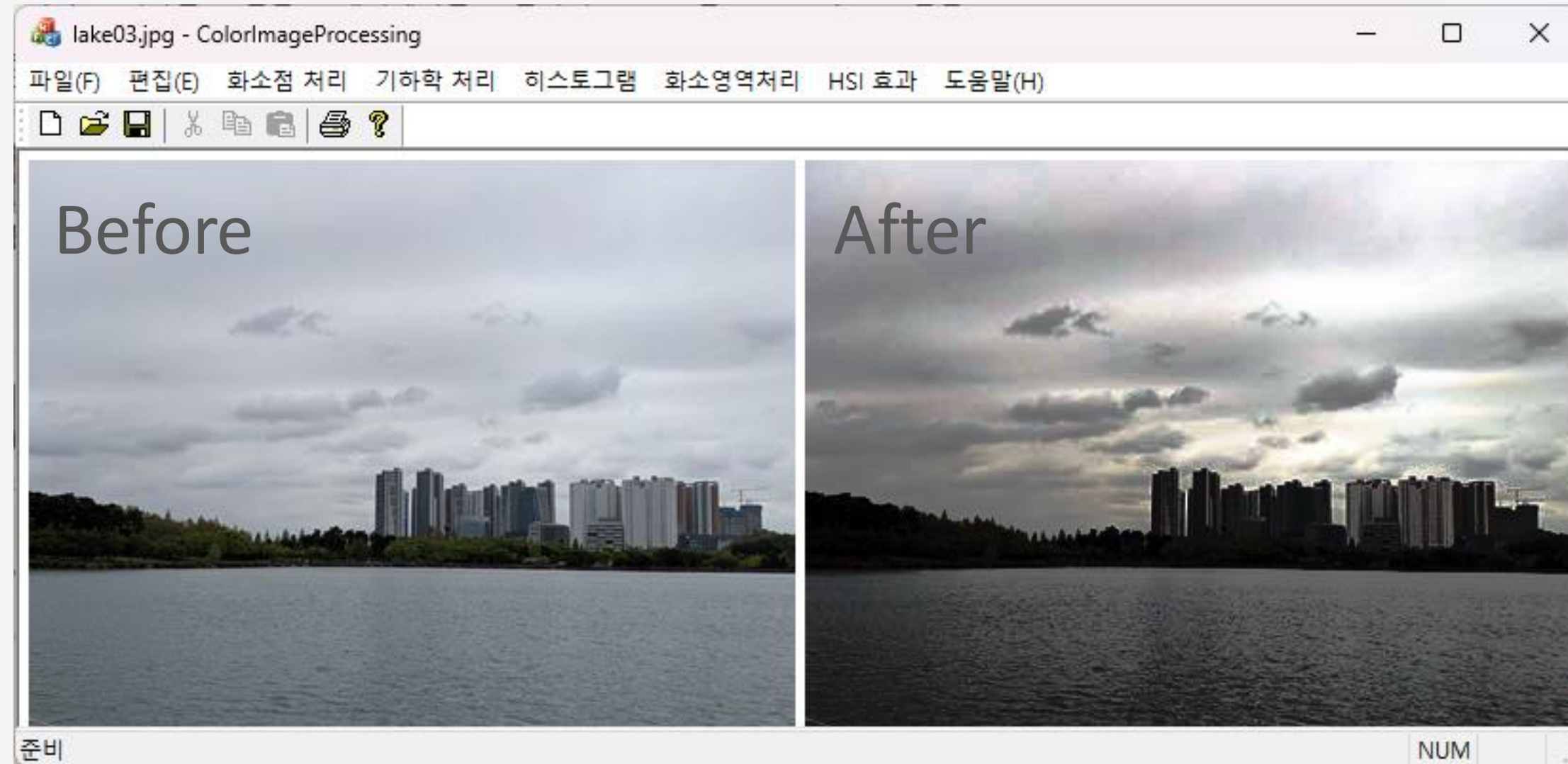


해당 사진의 픽셀 중
최대 밝기와 최저 밝기를 찾음

최대 밝기와 최저 밝기를 기반
기존 픽셀 값을 갱신

$$\text{출력 이미지} = \frac{\text{입력 이미지} - \text{최솟값}}{\text{최댓값} - \text{최솟값}} \times 255$$

화면 구성 및 기능 히스토그램 > 평활화



각 밝기의 빈도를 일정하게
만드는 처리

밝은 사진은
전체적으로 어두워짐

어두운 사진은
전체적으로 밝아짐

히스토그램 생성
각 픽셀값의 빈도를 셈



누적 히스토그램 생성
히스토그램 누적합 배열 생성



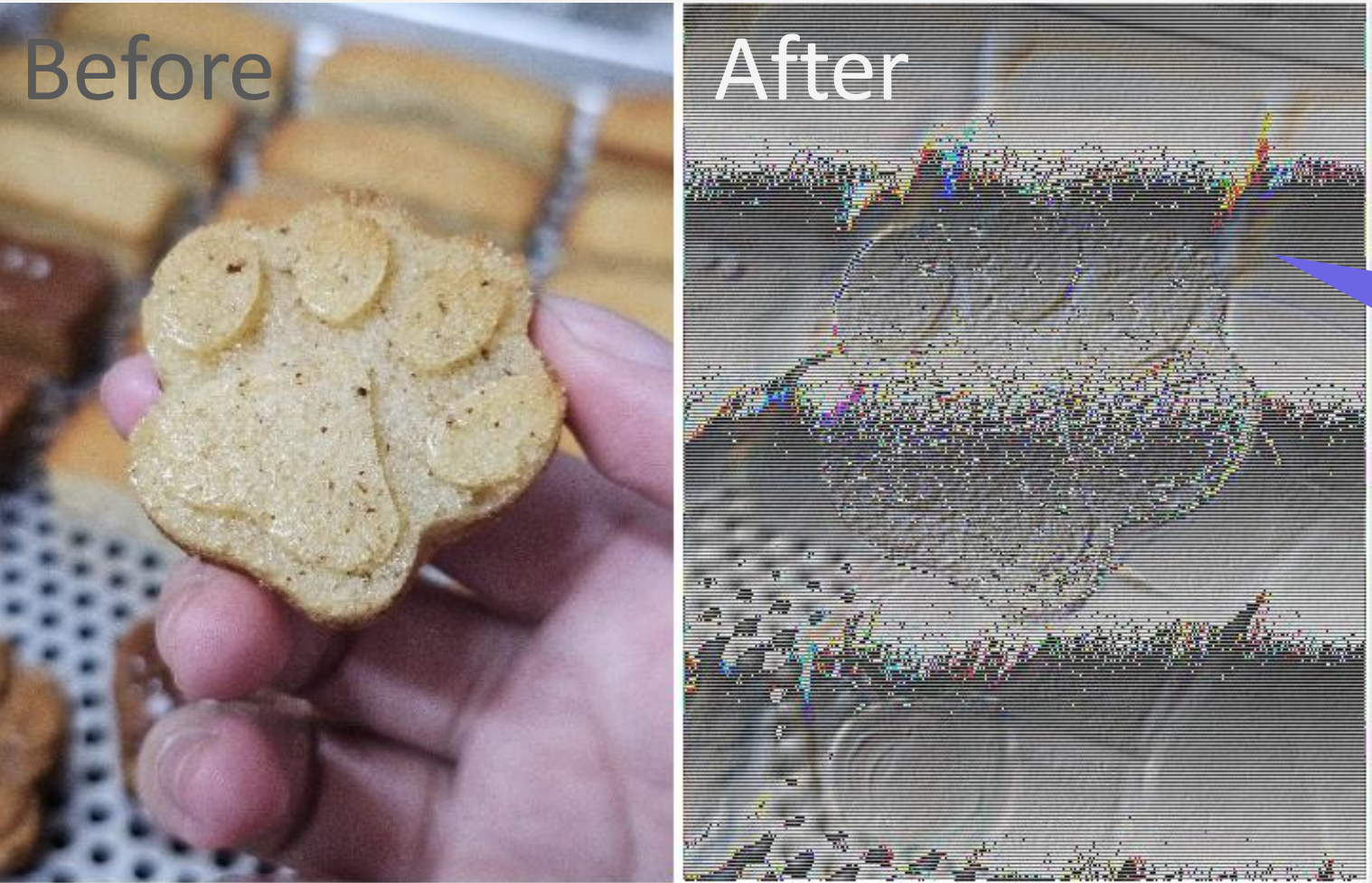
히스토그램 정규화
누적합 배열을 평균냄

화면 구성 및 기능

화소영역처리 > 엠보싱

필터 :

-1	0	0
0	0	0
0	0	1



사진에 일부만 튀어나오게 하는
마스크를 씌움

임시배열의 (1, 1)부터
출력배열에 복사



필터의 합이 0이면
임시배열에서 127을 더함



임시 저장 배열 생성
사진 크기보다 높리와 너비가
(필터크기-1)만큼 크게 생성



127으로 초기화
입력 사진 복사
(1, 1)부터 시작



임시배열의 (1, 1)부터 마스크를 씌움
마스크와 겹쳐진 곳끼리는 곱하고
곱해진 값의 평균으로 임시배열을 변경

화면 구성 및 기능

화소영역처리 > 원하는 필터



고주파 필터 -

-1	-1	-1
-1	8	-1
-1	-1	-1

Dialog

필터 크기는 몇으로 할까요?
(3, 5, 7, 9)

3

Dialog

필터의 1행
예) 1 2 3

-1 -1 -1

Dialog

필터의 2행
예) 1 2 3

-1 8 -1

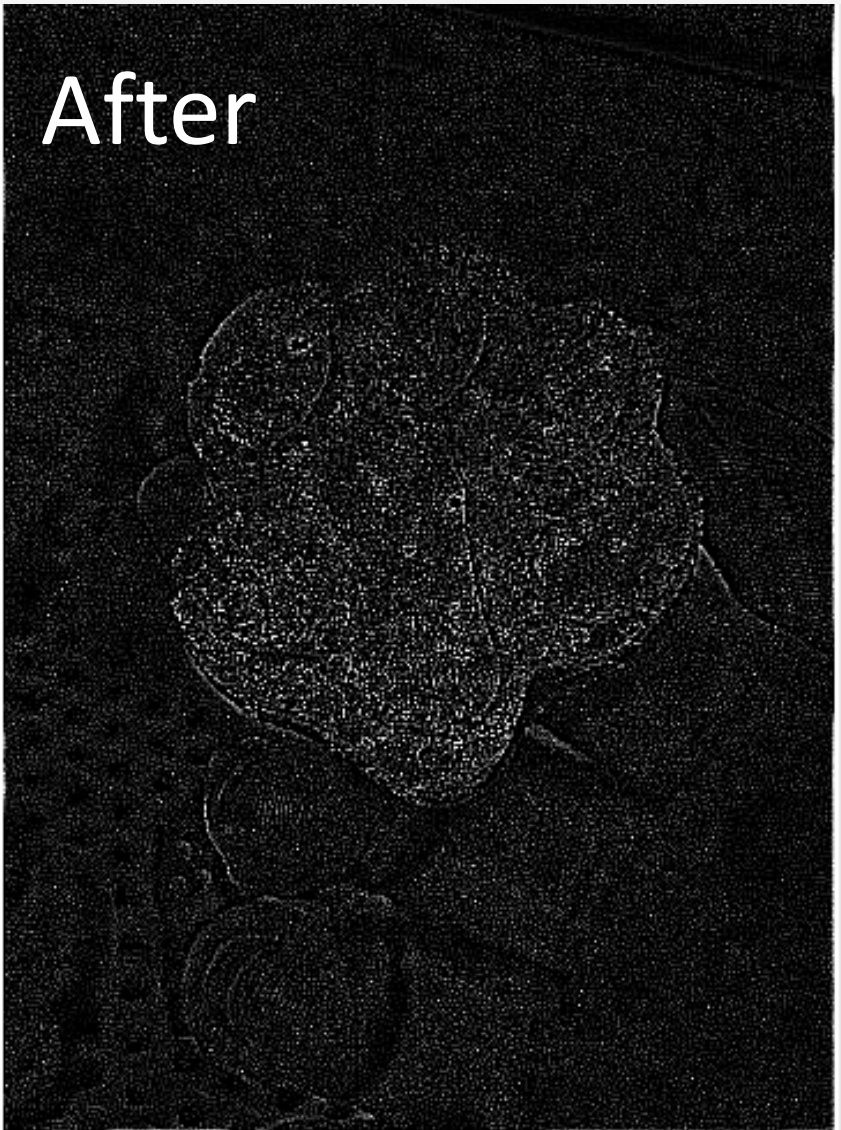
Dialog

필터의 3번째 줄을 입력해 주세요,
예) 1 2 3 4 5 6 7

-1 -1 -1

확인

취소



화면 구성 및 기능

화소영역처리 > 원하는 필터



LoG 필터 -

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Dialog

필터 크기는 몇으로 할까요?
(3, 5, 7, 9)

5

확인 취소

0 0 -1 0 0

0 -1 -2 -1 0

-1 -2 16 -2 -1

0 -1 -2 -1 0

0 0 -1 0 0



화면 구성 및 기능

화소영역처리 > 원하는 필터

필터 저장과 처리 방법

필터의 크기를
입력 받아 저장

필터의 한 줄을
입력 받음

입력 받은 한 줄의 값을
띄어쓰기를 기준으로 나누어
정수일 경우에만 배열에 저장

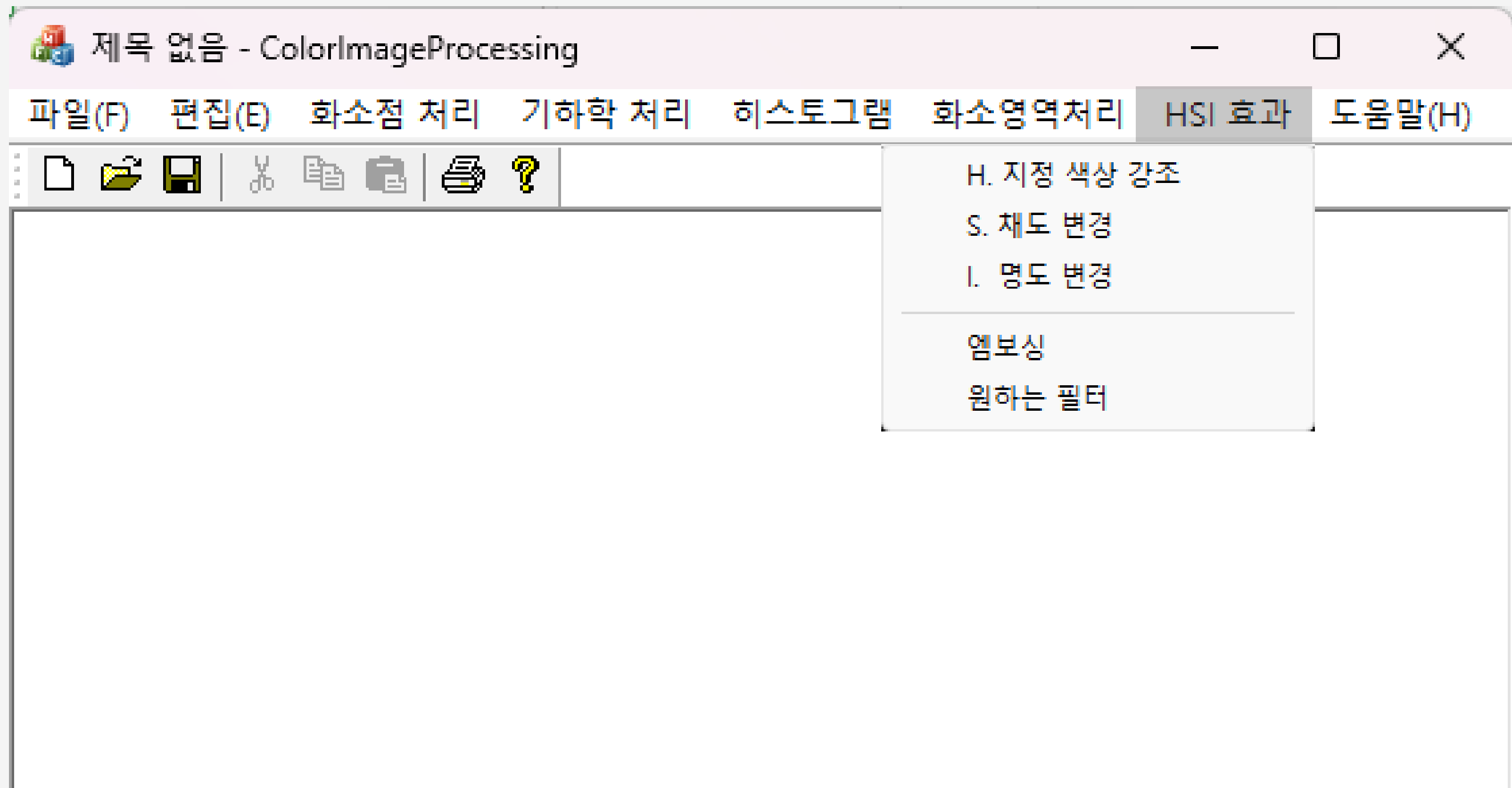
2번과 3번을
필터 크기만큼 반복

필터의 합을 확인하고
0이 아닐 경우,
1이 되도록 연산

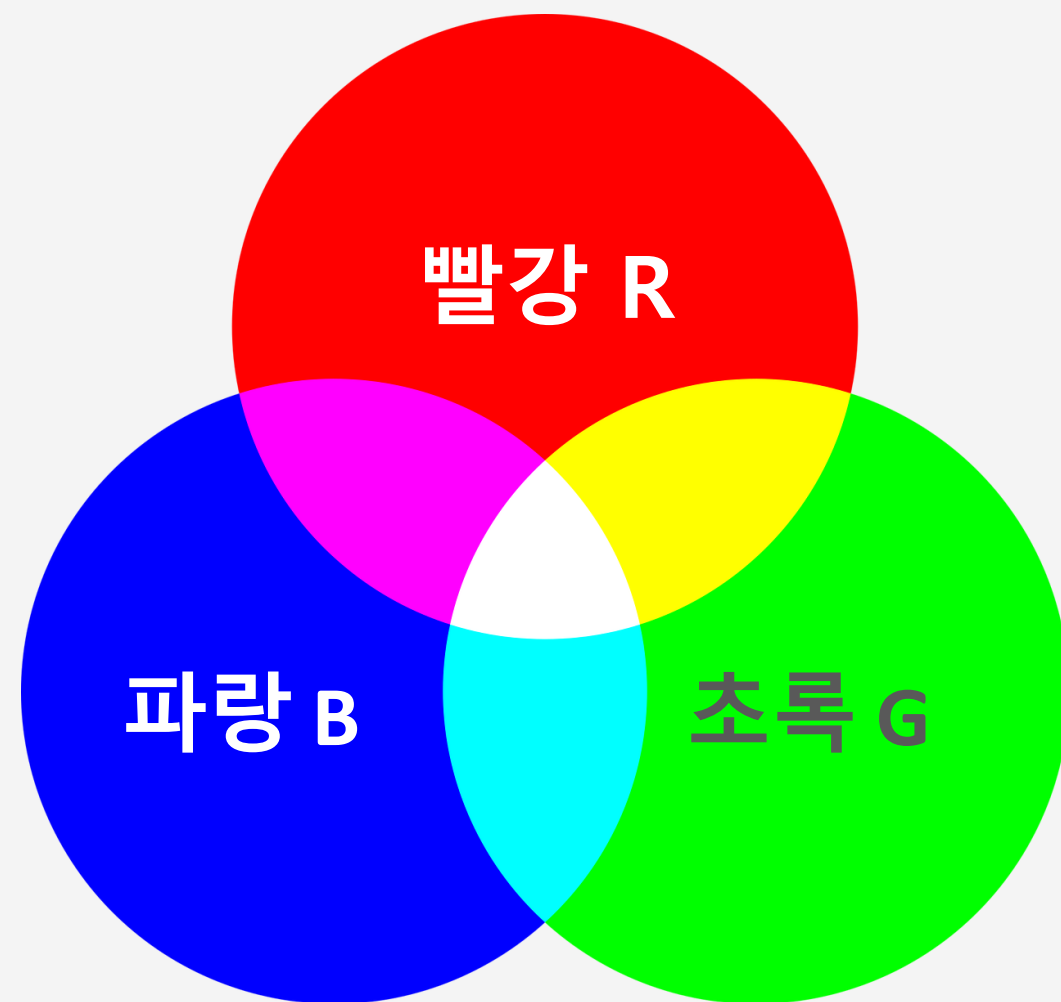
입력 이미지에
필터를 씌움

필터의 합이 0이었다면
각 픽셀에 127을 더함

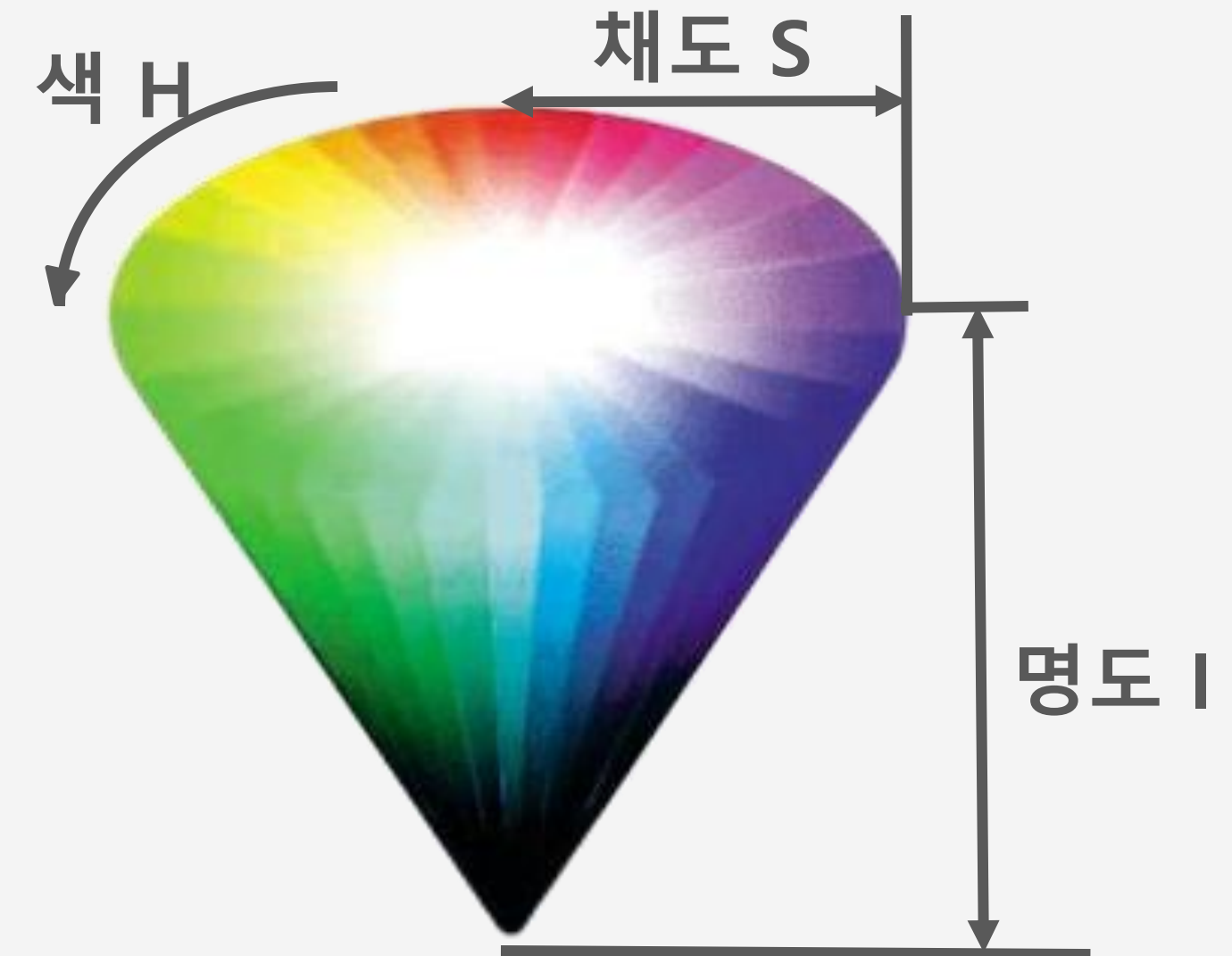
화면 구성 및 기능 HSI 효과 메뉴



화면 구성 및 기능 HSI?



모니터가 표현하는 색

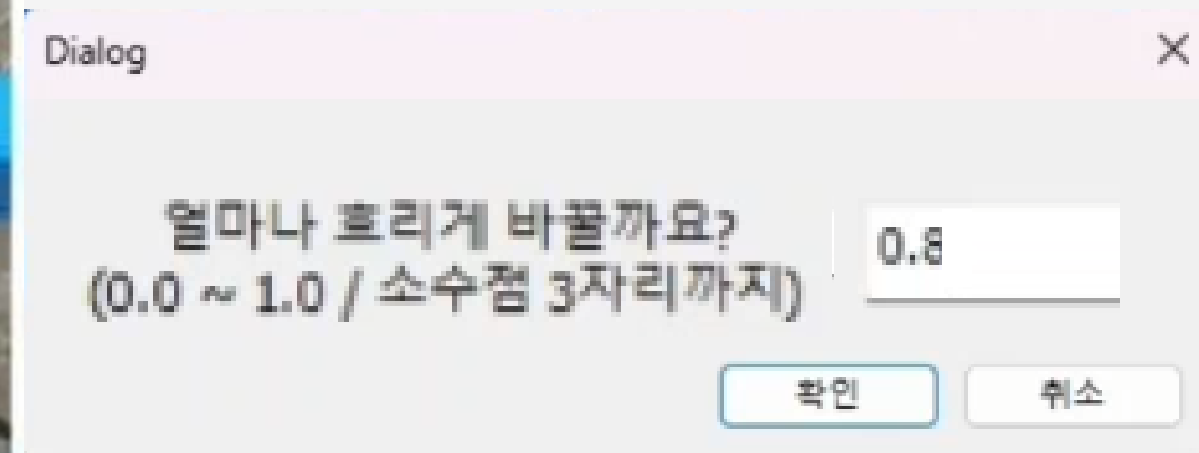


모니터는 표현 못하는 색

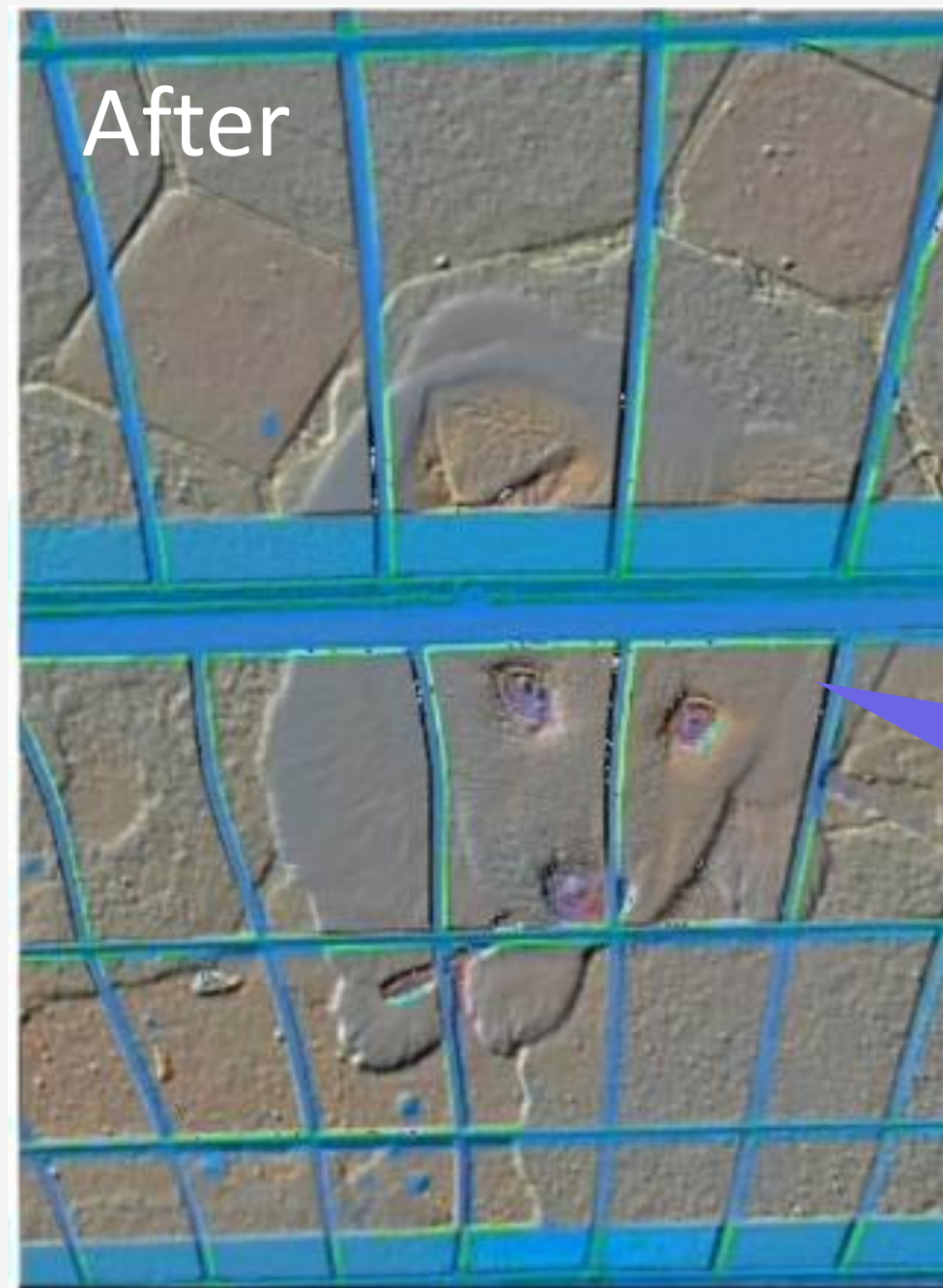
화면 구성 및 기능 HSI 효과 > H. 지정 색상 강조



화면 구성 및 기능 HSI 효과 > I. 명도 변경



화면 구성 및 기능 HSI 효과 > 엠보싱



이전과 다르게
RGB를 HSI로
바꾸어 I값에
엠보싱을 진행
색상도 비교적 적은 손실
상상과 유사한 결과

화면 구성 및 기능 HSI 효과 > 원하는 필터



LoG 필터 -

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Dialog

필터 크기는 몇으로 할까요?
 (3, 5, 7, 9)

5

확인
 취소

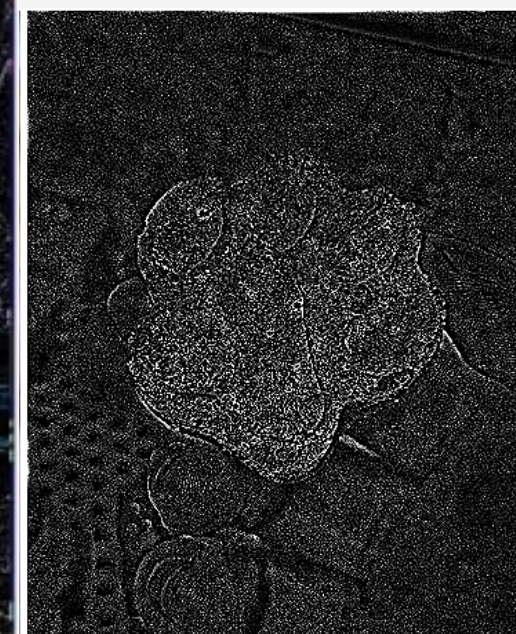
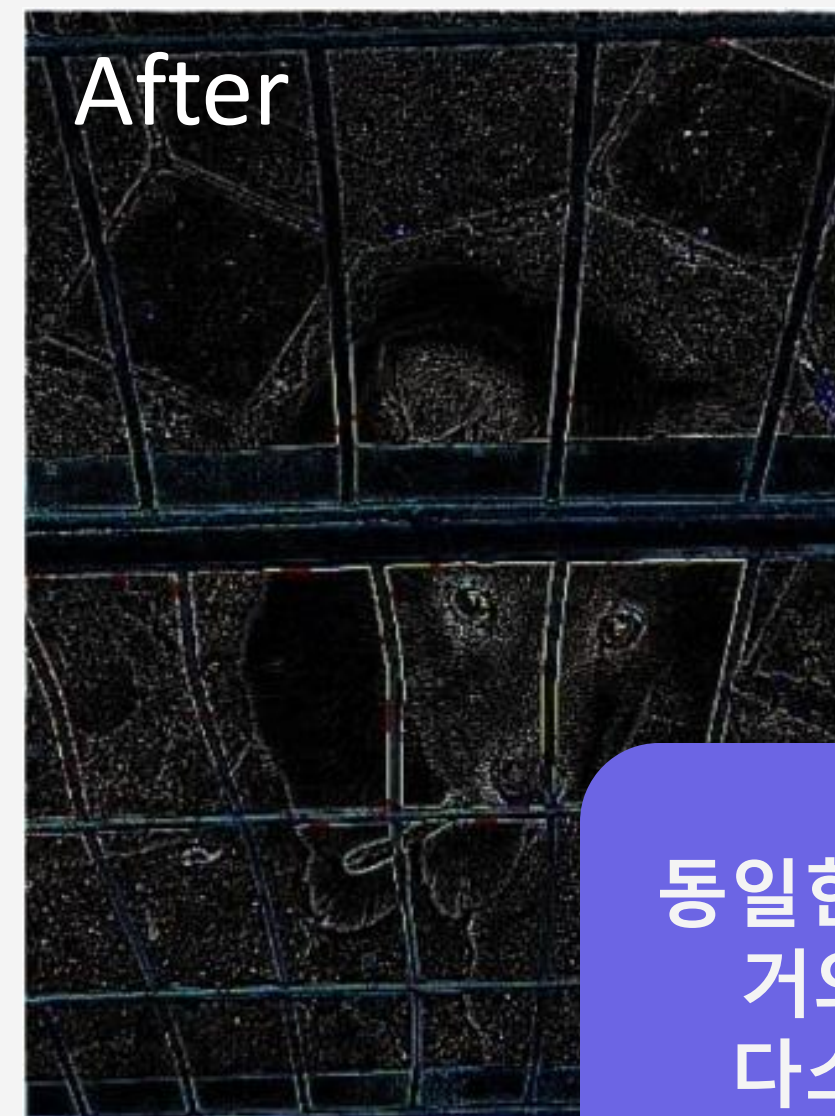
0 0 -1 0 0

0 -1 -2 -1 0

-1 -2 16 -2 -1

0 -1 -2 -1 0

0 0 -1 0 0



동일한 필터의 결과와 거의 유사하지만 다소의 색감 유지

마치며

좋았던 점

처음 사용해본 MFC으로 프로젝트를 만들어 봄

아쉬운 점

부가기능이나 예외처리 등을 진행하지 못함

추후 계획

아쉬운 점에 대한 보완과 아직 완성하지 못한 함수들의 완성

장혜원



시연영상



https://github.com/Jang-HW/Intel_Edge_AI_SW_Academy



hw11515@naver.com