

OpenCV 과제 2

이미지 처리 필터의 소개

1. Box Filter

- 함수: `cv::blur()` 또는 `cv::boxFilter()`
- 예제 코드

`cv::blur()` 함수는 평균 블러(Mean Blur)를 적용하여 커널 영역의 모든 픽셀의 평균값을 중심 픽셀에 할당하는 방식으로 이미지를 흐릿하게 만듭니다.

Ex) `cv::blur(InputArray src, OutputArray dst, Size ksize, Point anchor = Point(-1,-1), int borderType = BORDER_DEFAULT);`

- 각 파라미터에 대해 설명하겠습니다.

src: 입력 이미지(`cv::Mat` 형식)입니다.

dst: 출력 이미지로, 블러링 결과가 저장될 `cv::Mat` 객체입니다.

ksize: 커널의 크기를 `cv::Size(width, height)` 형식으로 지정합니다. 커널 크기가 클수록 블러 효과가 커집니다.

anchor (기본값 `Point(-1, -1)`): 커널 내에서 고정할 위치입니다. `(-1, -1)`로 지정하면 커널의 중심이 기준으로 설정됩니다.

borderType (기본값 `BORDER_DEFAULT`): 이미지 외곽의 픽셀을 처리하는 방식을 결정합

니다. 외곽 처리 방법에는 여러 옵션이 있으며, 기본값으로 BORDER_DEFAULT를 사용합니다.

- **특징:**

이미지의 각 픽셀을 주변 픽셀의 평균 값으로 대체하여 블러링을 수행합니다. 커널 내 모든 픽셀에 같은 가중치를 적용합니다. 매우 단순한 연산으로 빠르게 블러링을 적용할 수 있습니다.

- **기대 효과:**

이미지의 노이즈가 감소하고, 전체적으로 부드러워지는 효과가 있습니다. 그러나, 경계선이 흐릿해질 수 있고, 특정 패턴이 있는 노이즈가 제거되지 않을 수 있습니다. 가우시안 블러보다 세부 사항이 더 많이 제거되며, 비교적 인공적인 블러 효과를 얻습니다.

- 예시 이미지

$\frac{1}{9} \times$	1	1	1
	1	1	1
	1	1	1

$\frac{1}{25} \times$	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1

실제 영상에 평균 값 필터를 적용한 결과



원본 영상



3x3 크기의 마스크



5x5 크기의 마스크

- 영상의 특정 좌표값을 주변 픽셀 값의 산술 평균으로 계산합니다.
- 이는 픽셀들 간의 그레이스 케일 값 변화가 줄어 엣지를 무뎠지게 하는 효과가 있습니다
- 위 예시 사진을 통해 마스크 커널의 크기를 키울 수 록 이미지가 더욱 부드러워 지는것을 확인할 수 있습니다.

2. Median blur

- 함수 : `cv::medianBlur()`
- 예제 코드 : `cv::medianBlur(InputArray src, OutputArray dst, int ksize);`
- 각 파라미터에 대해 설명하겠습니다.

src: 입력 이미지(`cv::Mat` 형식)입니다. 미디언 블러를 적용할 원본 이미지입니다.

dst: 출력 이미지(`cv::Mat` 형식)입니다. 필터링 결과가 저장됩니다.

ksize: 필터 커널의 크기를 지정하는 홀수 정수입니다. 커널 크기는 블러의 강도를 결정하며, 값이 클수록 더 많은 영역의 노이즈가 제거되지만, 처리 속도는 느려질 수 있습니다. 예를 들어, 3, 5, 7과 같은 홀수 값을 사용합니다.

- 특징

커널 내의 픽셀 값을 정렬한 후 중간 값을 사용하여 중심 픽셀 값을 대체합니다. 엣지 보존 능력이 뛰어나고, 소금-후추 노이즈(salt-and-pepper noise) 제거에 특히 효과적입니다.

- 기대 효과

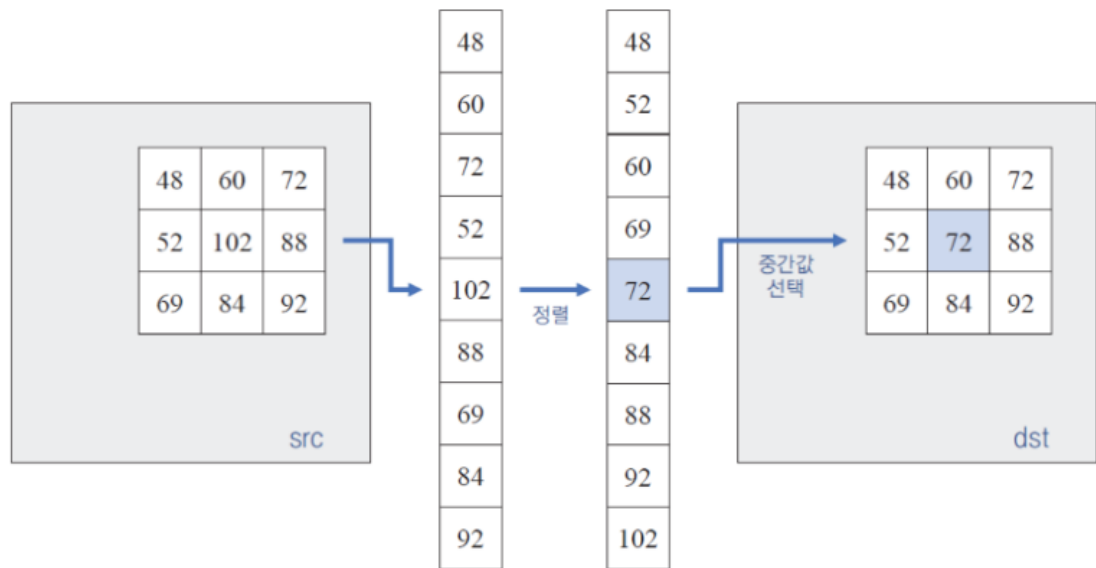
노이즈가 많은 이미지에서 잡음을 효과적으로 제거할 수 있습니다. 엣지 부분이 유지되기에 선명한 경계를 얻을 수 있습니다. 특히 소금-후추형태의 잡음을 제거할때 탁월한 성능을 보입니다.

- 예시 이미지



미디언 필터(Median filter)

- 주변 픽셀들의 값을 정렬하여 중앙값으로 픽셀 대체
- 소금&후추 잡음 제거에 효과적



3. 라플라시안 필터

- 함수 : `cv::Laplacian()`
- 예제 코드 : `cv::Laplacian(InputArray src, OutputArray dst, int ddepth, int ksize = 1, double scale = 1, double delta = 0, int borderType = BORDER_DEFAULT);`
- 특징

2차 미분 연산을 통해 이미지의 엣지나 경계선을 강조합니다. 한번의 연산으로 모든 방향의 경계선을 검출 할 수 있습니다. 밝기 차이가 큰 부분을 강조하여, 강한 엣지 검출효과를 제공합니다.

- 기대 효과

이미지의 경계선을 더욱 강조하여 세밀한 엣지 검출을 수행합니다. 특히 전체 이미지의 구조를 파악하고 경계선을 더욱 날카롭게 만들고자 할때 유리합니다. 밝기 차이가 큰 부분이 강조되며, 텍스처나 패턴이 두드러진 이미지에서 구조적인 특징을 파악하기 쉽습니다.

- 예시 이미지

