
Digital Image Processing

Using JavaScript

FrontEnd Programming 1

박 영 호

(blog.naver.com/hkpyh)

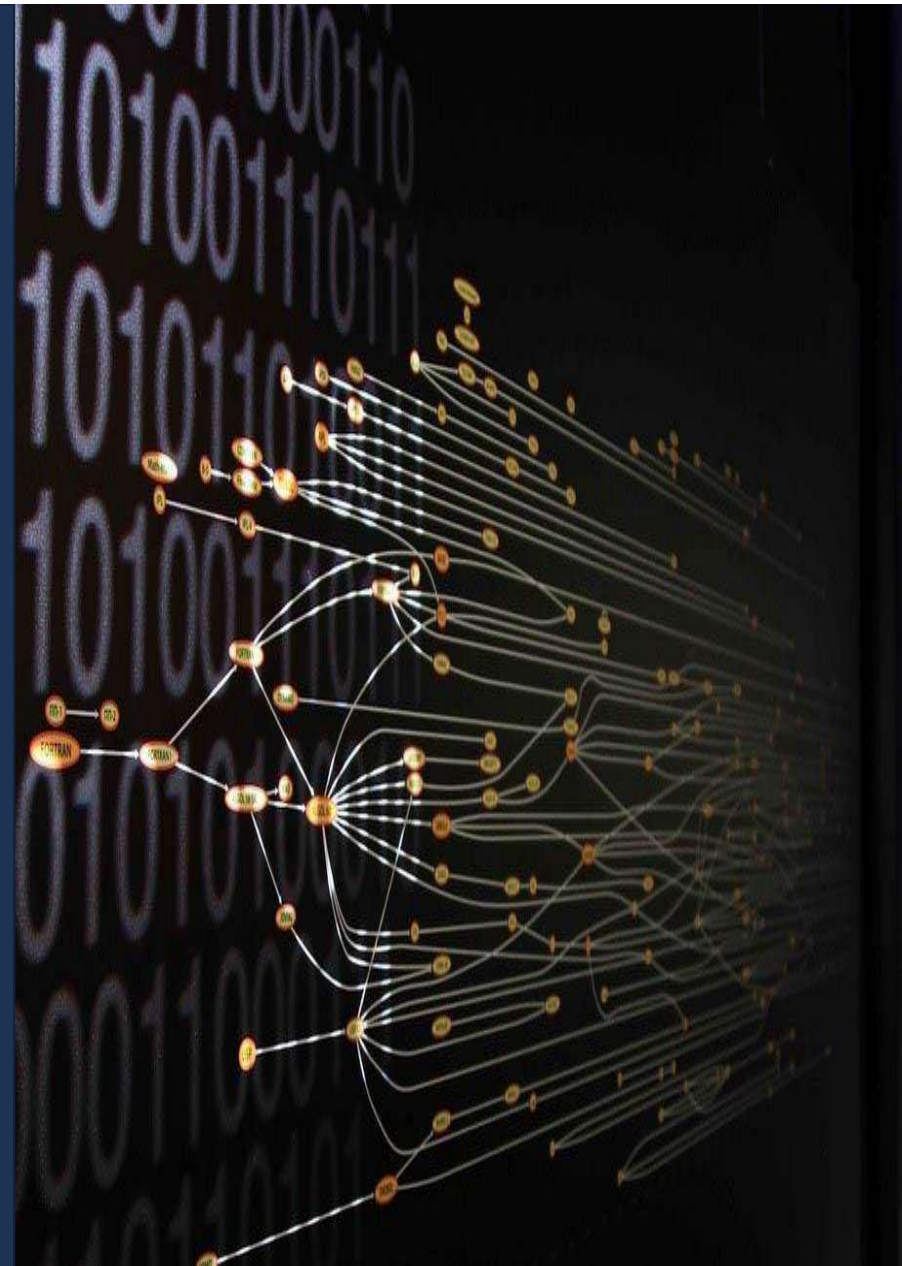
Contents

01 Introduce

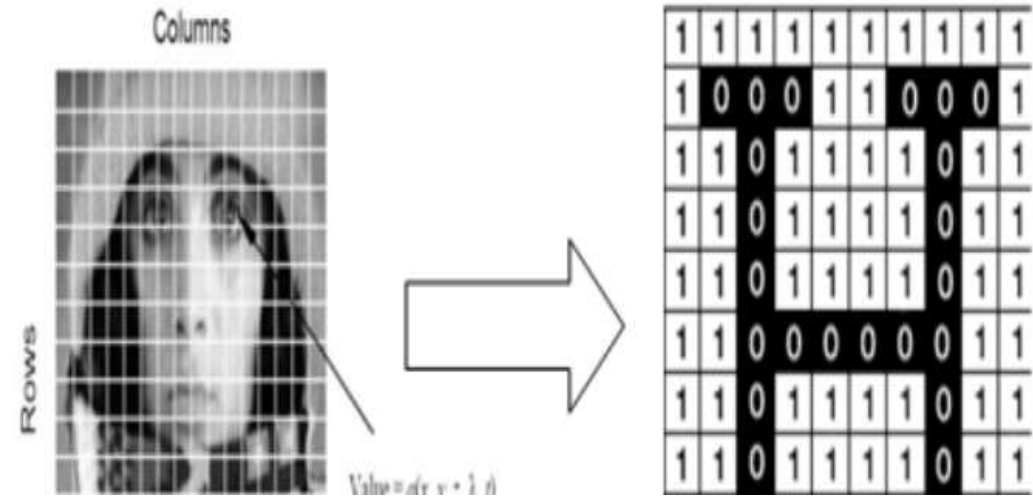
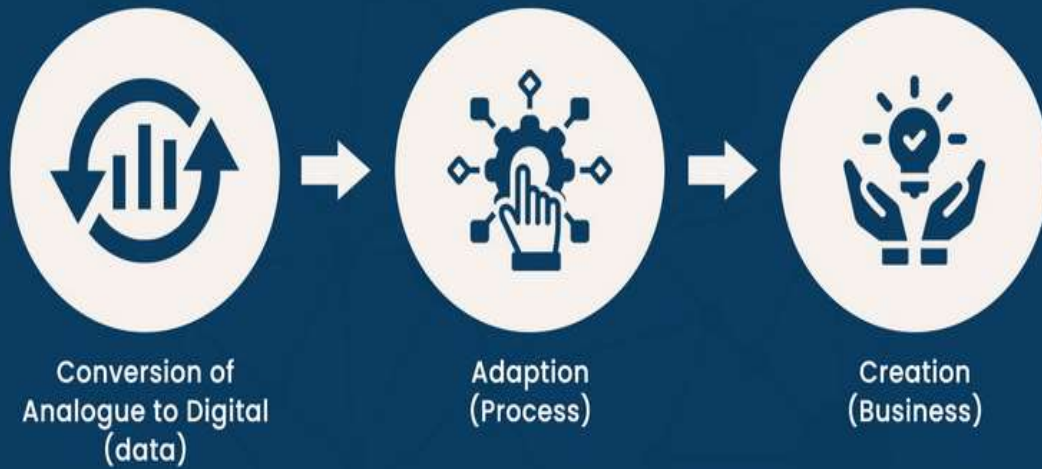
02 Algorithm

03 Future Plan

Introduce of Digital Image Processing



Introduce of Digital Image Processing



- 영상(Analog)을 Digital로 입력
- Program(APP)을 활용하여 영상처리
- 결과물을 화면에 시현 및 상업적 활용

- 영상을 각각의 좌표와 해당 숫자로 인식
- 각 좌표의 숫자에 필요한 Algorithm 적용
- Digital로 처리 및 활용

Why Using Web & JavaScript ?

As is

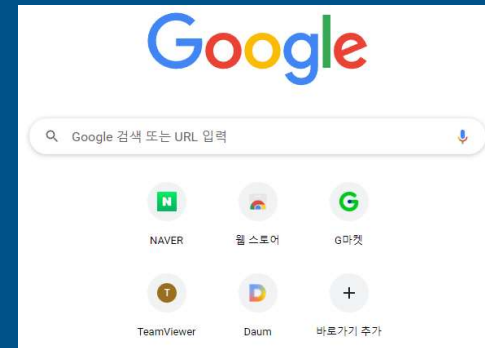


- PhotoShop 등 S/W 활용
- S/W 구입, 설치 필요
- 제한된 사용자

JavaScript

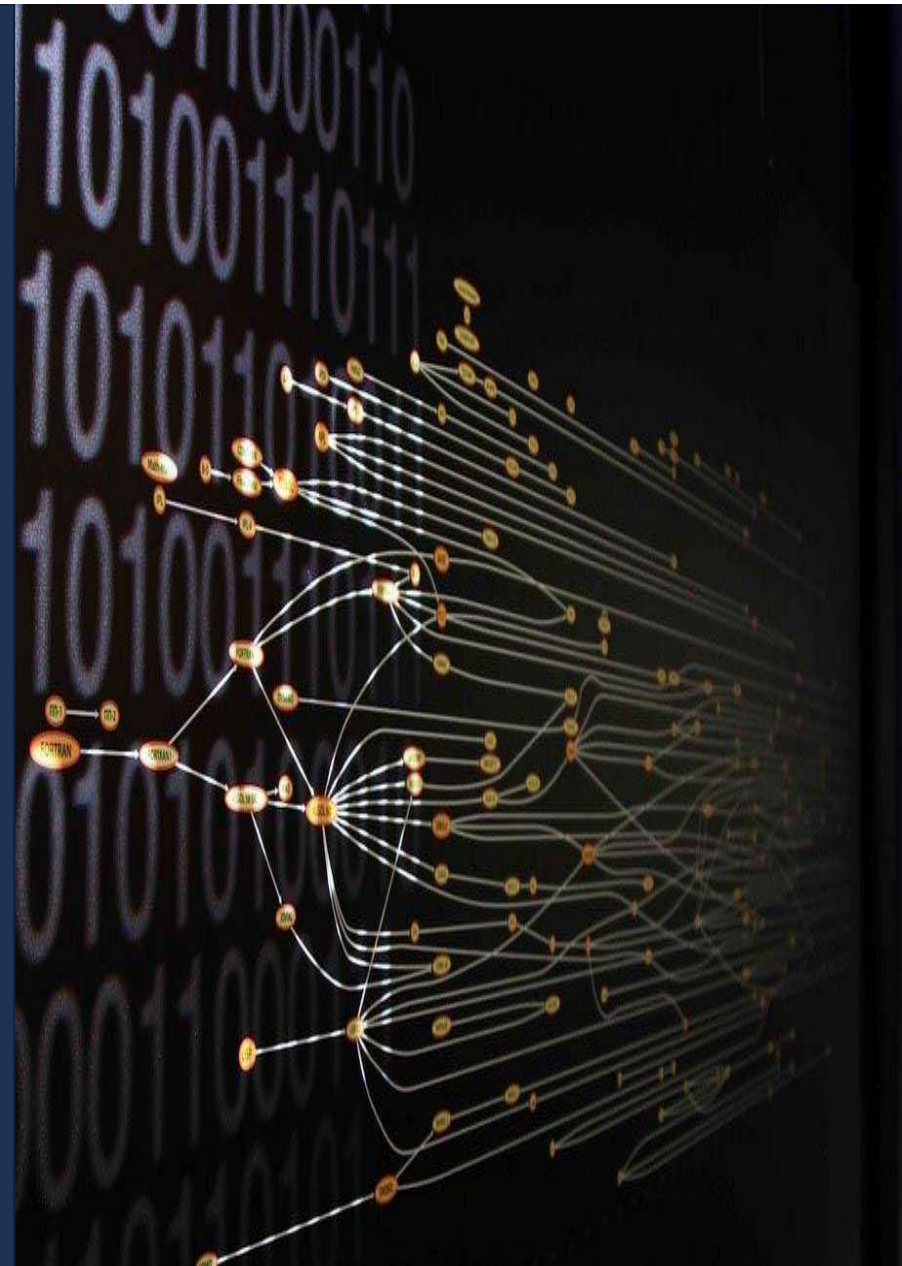
HTML5 기반으로
이미지, 영상 처리 가능

To Be



- Web Brower에서 활용
- S/W 구입, 설치 **불필요**
- **모든** 사용자

Algorithm of Digital Image Processing



Algorithm of Digital Image Processing

화소점 처리

- 화소점의 원래 값이나 위치를 기준으로 변경
- 사진의 밝기 조절, 사진의 반전 처리
- Stretching : 낮은 명암대비를 높게 향상
- 평활화 : 어두운 영상을 밝게 표현

영역 처리

- 이웃한 화소를 고려한 영상처리
- Blurring : 영상 흐리게
- Sharpening : 영상 대비 효과

기하학 처리

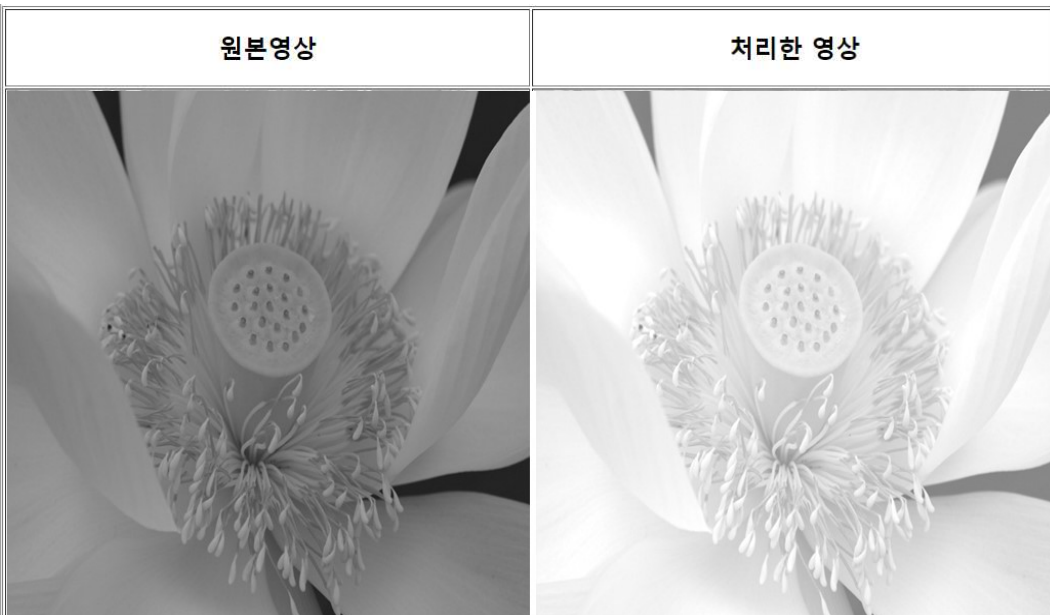
- 화소의 위치 또는 배열을 변화
- 영상 회전
- 영상 축소, 확대

프레임 처리

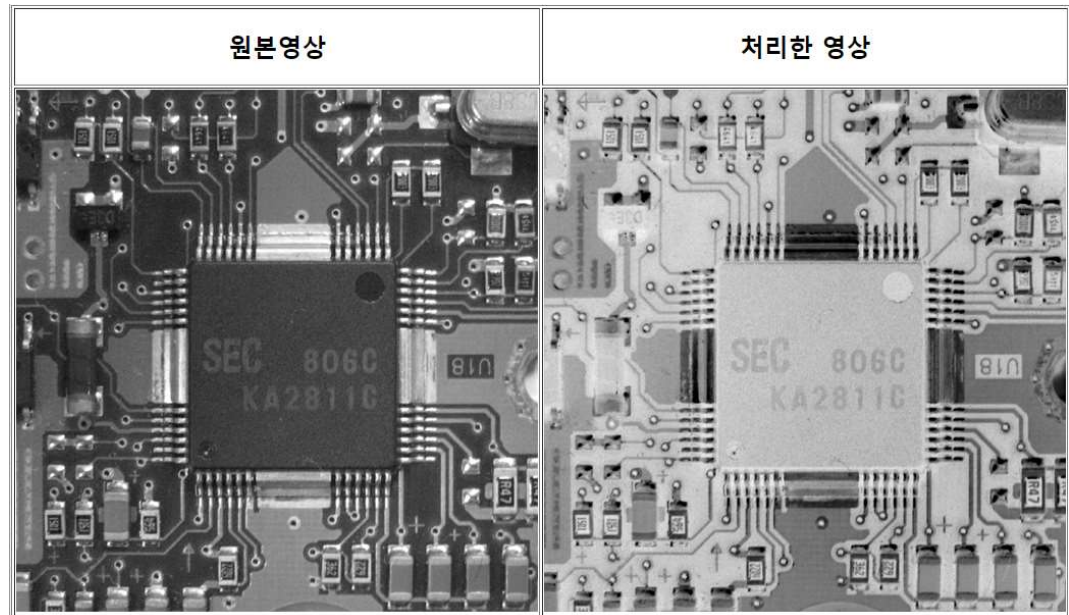
- 2개 이상의 서로 다른 영상 조합
- Mapping

Example of Digital Image Processing – 화소점 처리[1](밝게/반전)

유형/효과	밝게 / 영상을 밝게 출력
Algorithm	출력 영상 = 입력 영상 + 보정값(입력)



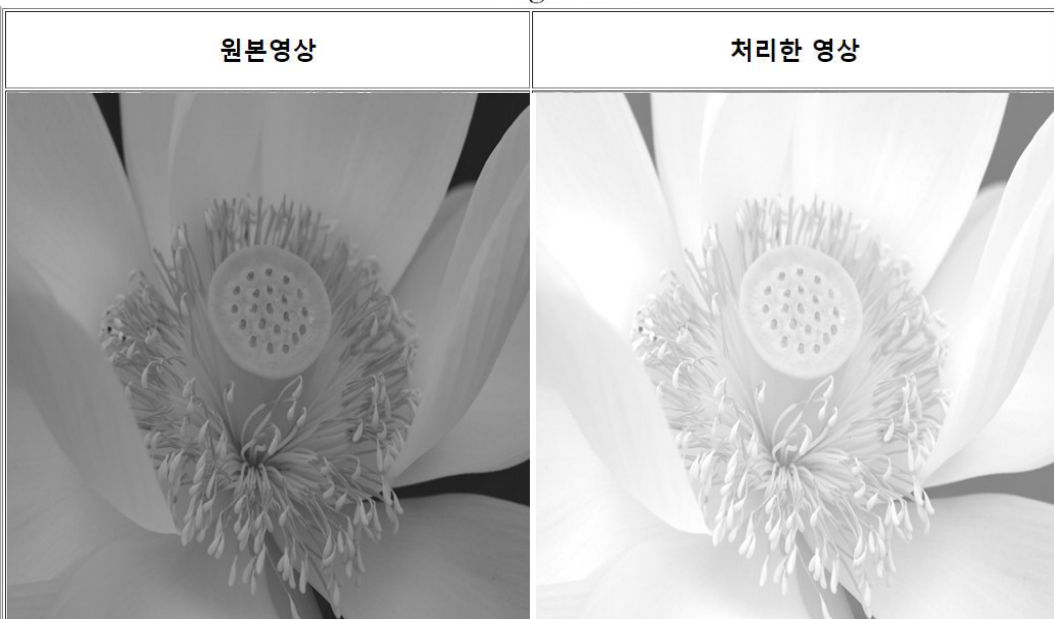
유형/효과	반전 / 영상의 밝기를 반대로 표현
Algorithm	출력 영상 = 최대 밝기(255) - 입력 영상



Example of Digital Image Processing – 화소점 처리[2](stretching, 평활화)

유형/효과	Stretching / 명암대비가 낮은 영상의 품질 향상
Algorithm	입력 영상의 최저, 최고 명도 값을 구한 후 아래 공식을 활용하여 향상된 출력 영상 구현

$$new\ pixel = \frac{old\ pixel - low}{high - low} \times 255$$

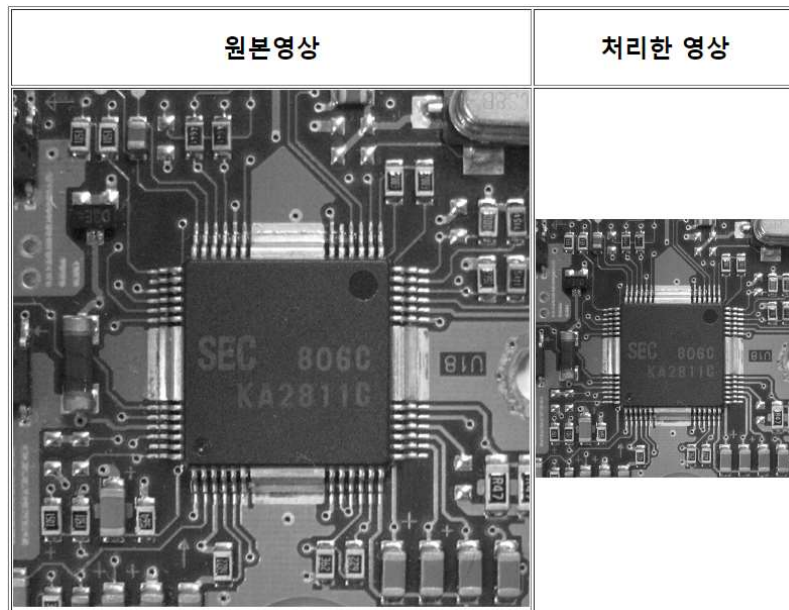


유형/효과	평활화 / 어두운 영상을 밝게 구현
Algorithm	입력 영상 명도의 누적합을 구한 후에 정규화 공식 활용하여 영상 보정

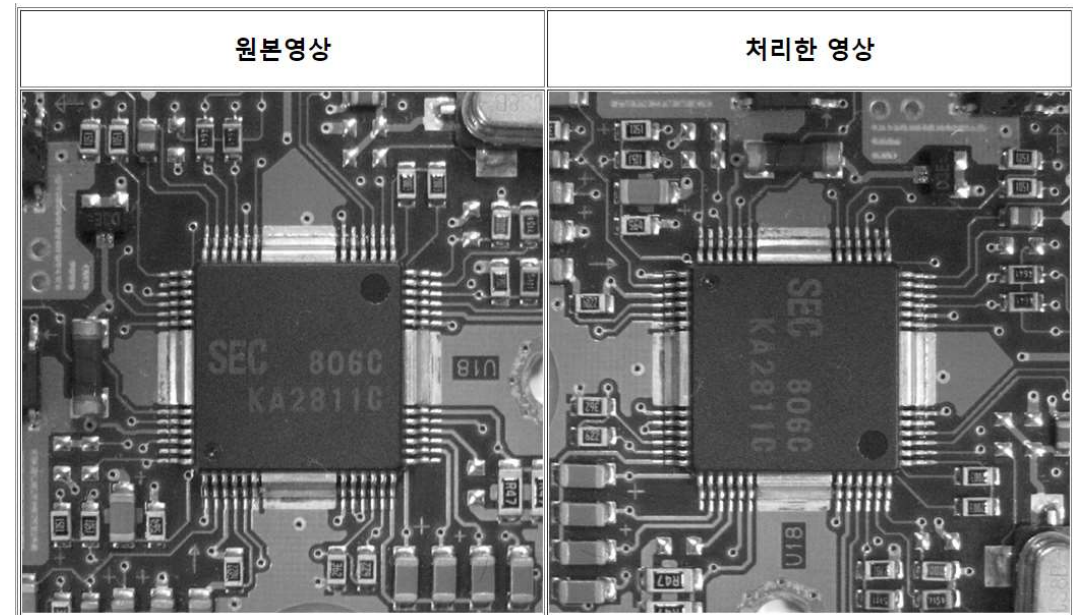


Example of Digital Image Processing – 기하학 처리(축소, 회전)

유형/효과	축소 / 영상을 사용자 입력 배율로 축소
Algorithm	출력 영상 = 입력 영상 / (사용자 배율)



유형/효과	회전 / 영상을 90도 회전하여 표현
Algorithm	영상의 X, Y 좌표를 서로 바꾸어서 출력



Example of Digital Image Processing – 영역 처리[1](blurring)

유형/효과	blurring / 영상을 흐리게 또는 부드럽게 표현
Algorithm	입력화소에서 이웃한 각 화소에 3x3 배열의 가중치(Mask)를 곱하여 출력

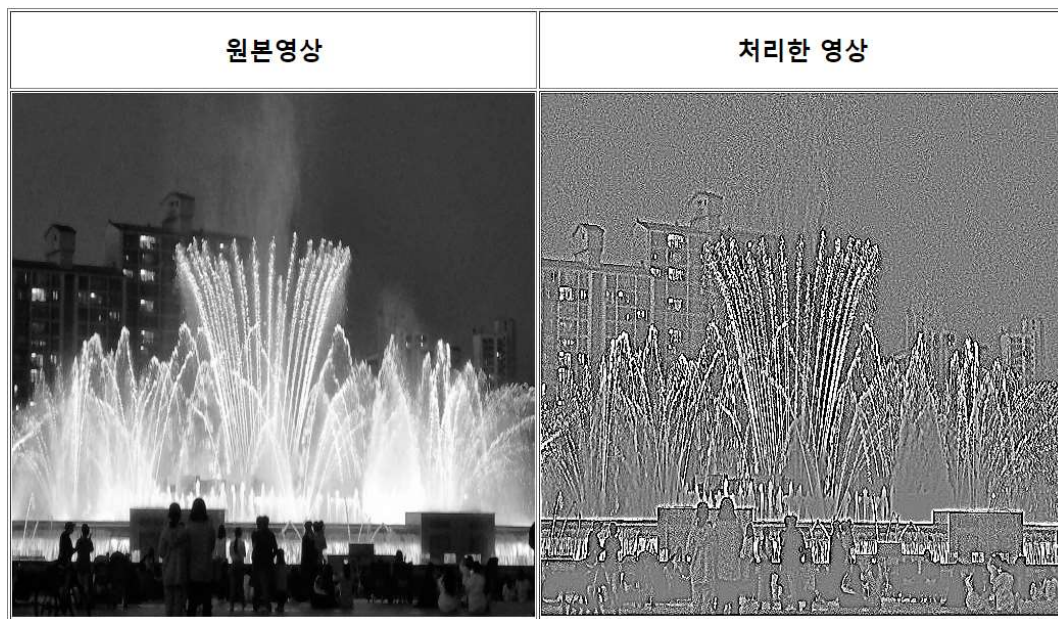


유형/효과	blurring / 영상을 흐리게 또는 부드럽게 표현
Algorithm	입력화소에서 이웃한 각 화소에 5x5 배열의 가중치(Mask)를 곱하여 출력



Example of Digital Image Processing – 영역 처리[2](edge 검출)

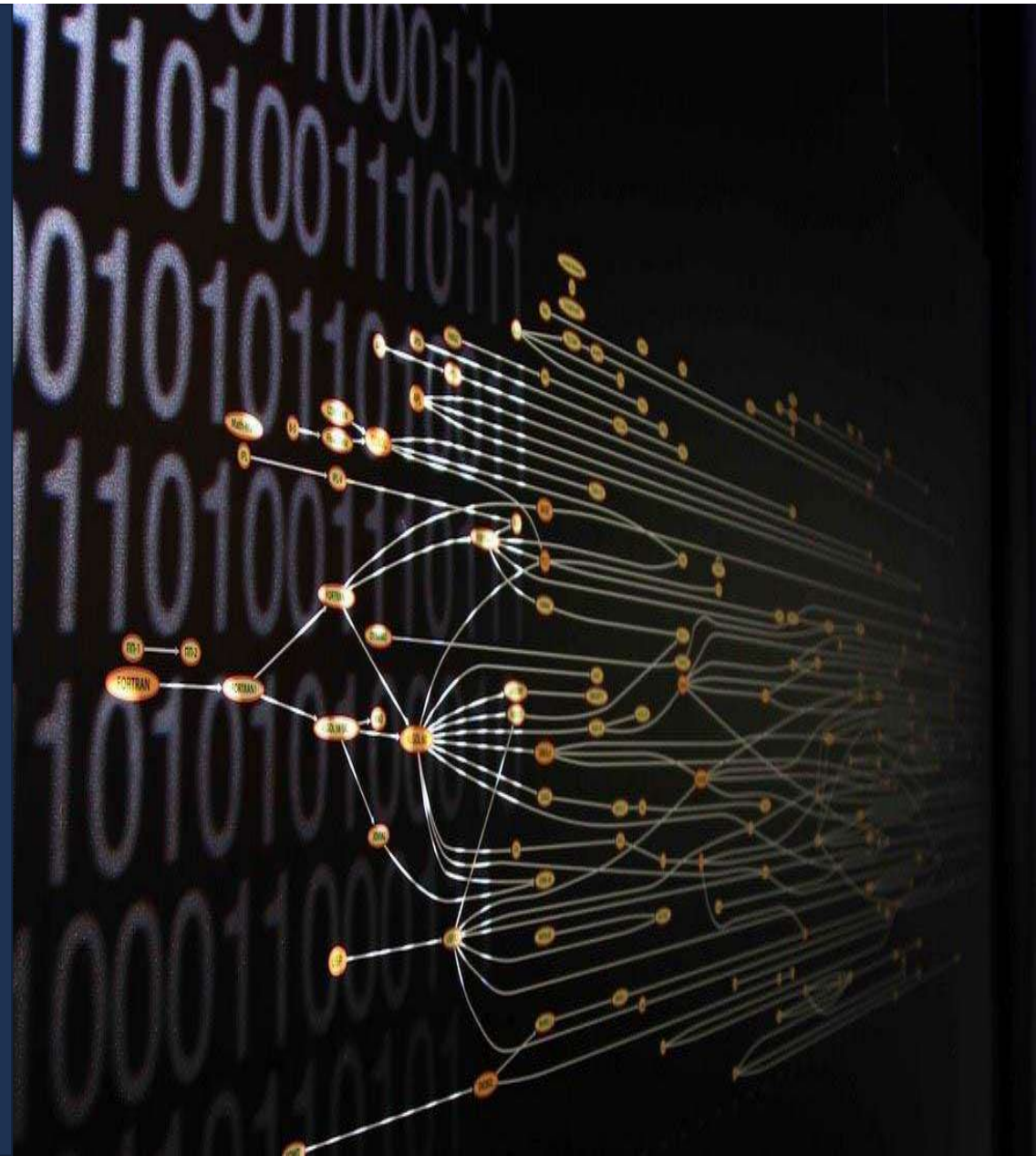
유형/효과	Edge 검출 / 물체식별, 위치/모양/크기등을인지하고방향성을탐지할수있는정보제공
Algorithm	입력화소에서 이웃한 각 화소에 5x5 배열의 가중치(Mask) 를 곱하여 출력



- Edge 검출은 여러 종류가 있으나, 기본 Algorithm은 동일 (3x3 배열의 가중치 변경으로 구현 가능)
- 좌측은 LoG(Laplacian of Gaussian) 기법의 예시이며, 영상의 잡음제거 이후 Edge 강조 기법 활용 (아래는 적용한 Mask 배열임)

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

Future Plan of Digital Image Processing



Future Plan of Digital Image Processing

'22. 9월

JavaScript
환경 구성
(완료)

현재 시점

'22. 9월

Color 영상
Code 작성

(추후 예정)

Web에 구현 및
상용화

'22. 9월

흑백(RAW) 영상
Code 작성(완료)

'22. 11월

동영상에
각종 효과 구현

<개선 필요 사항>

- Color 영상 및 동영상에 기 작성된 Code 적용 필요
- 홈페이지 개설 및 사용자 편의를 위한 화면 구성
- 상용화를 위한 추가 기능 개발(영상 다운로드 및 공유 / 스마트폰 환경에서 작동 가능 여부 검증 등)

“

Web에 Digital Image Processing을 개발 및 구현하여

사용자 편의성 증대 및 상용화를 통한 이익 추구

”