



2021 로봇스터디 교육 2주차

영상처리 기본

# 목차

---

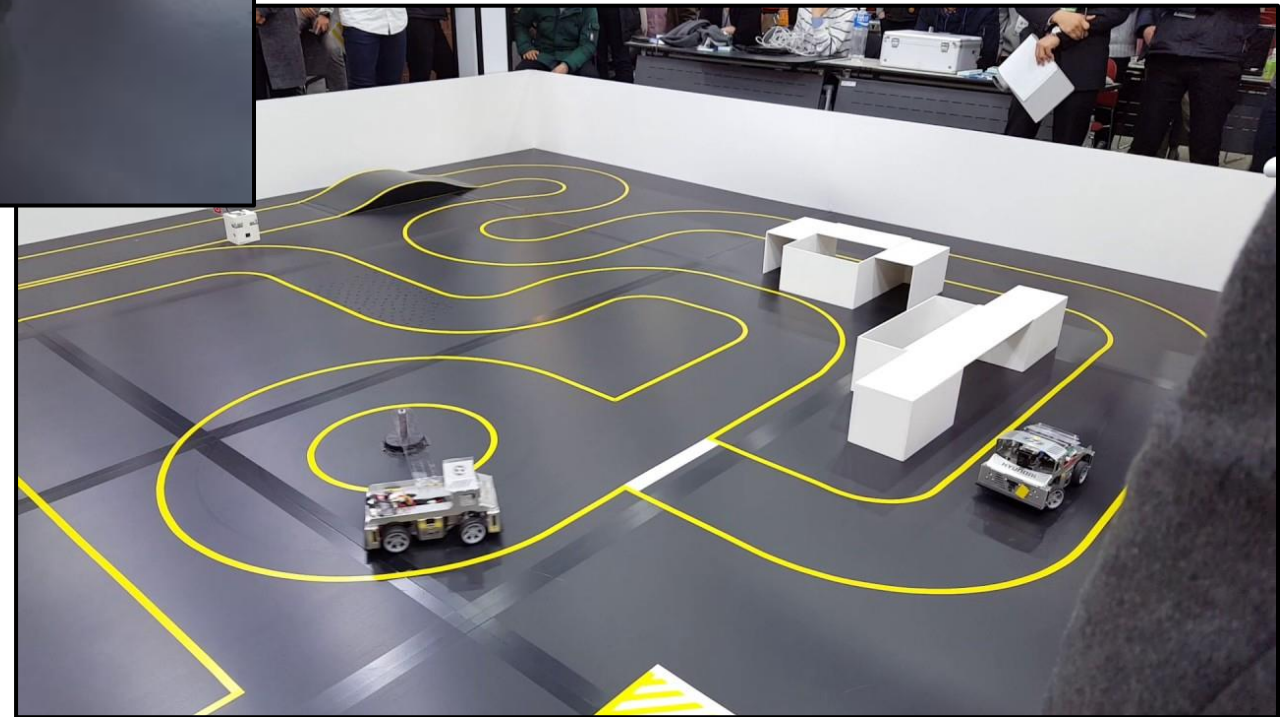
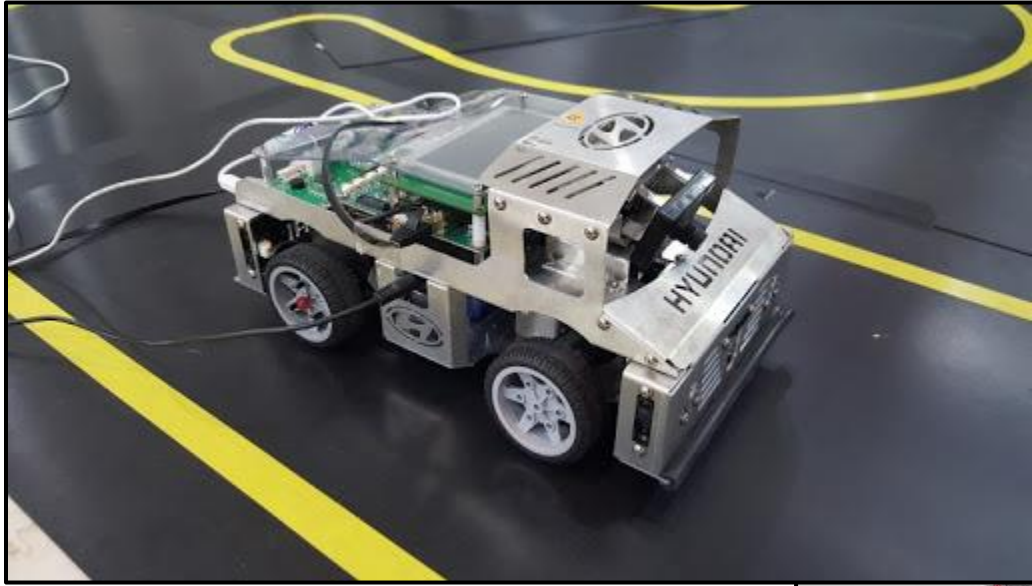
1. 픽셀 처리

2. 컨볼루션 연산

3. 이진화

4. 블러링 & 샤프닝

?





1. grayscale



2. blur



3. canny edge



4. ROI



5. hough transform

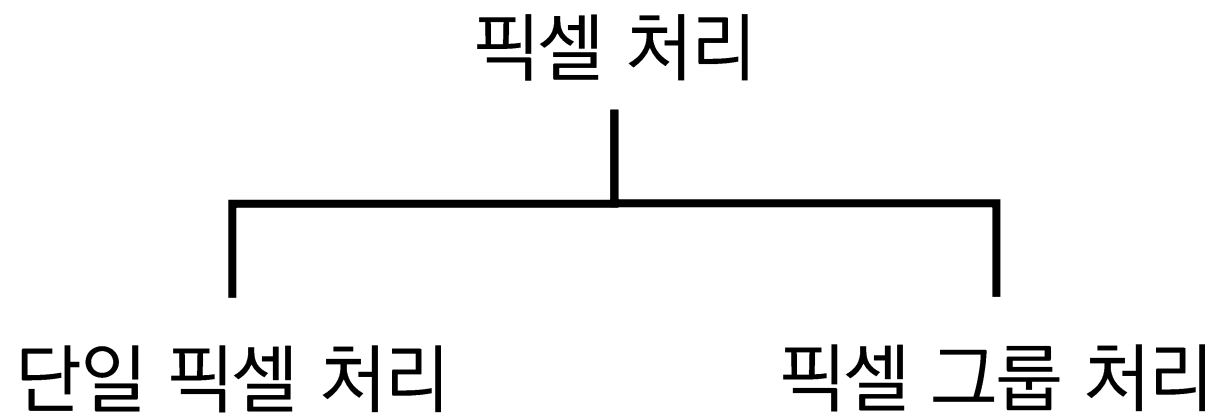


6. Result

## 1. 픽셀 처리

## 1. 픽셀 처리

---

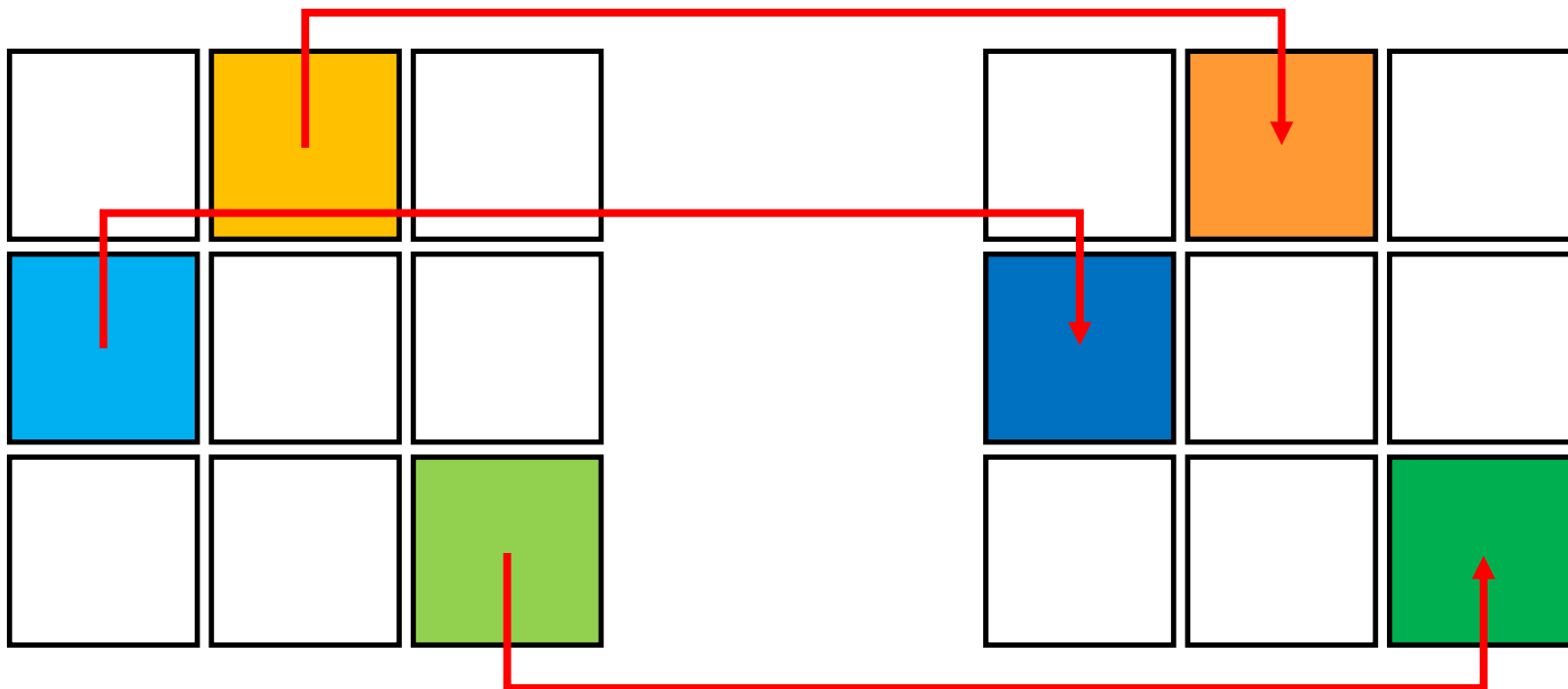


# 1. 픽셀 처리

---

## 단일 픽셀 처리 (Pixel Point Processing)

이웃 픽셀과는 **독립적**으로 입력 영상의 각 픽셀 값을 변환한 후 결과 영상의 동일한 위치에 출력하는 연산





# 1. 픽셀 처리

## 단일 픽셀 처리 기법의 종류

### 1. 산술 연산 : 덧셈, 뺄셈, 곱셈, 나눗셈



덧셈 연산 (+50)



뺄셈 연산 (-50)



곱셈 연산 (\*1.2)



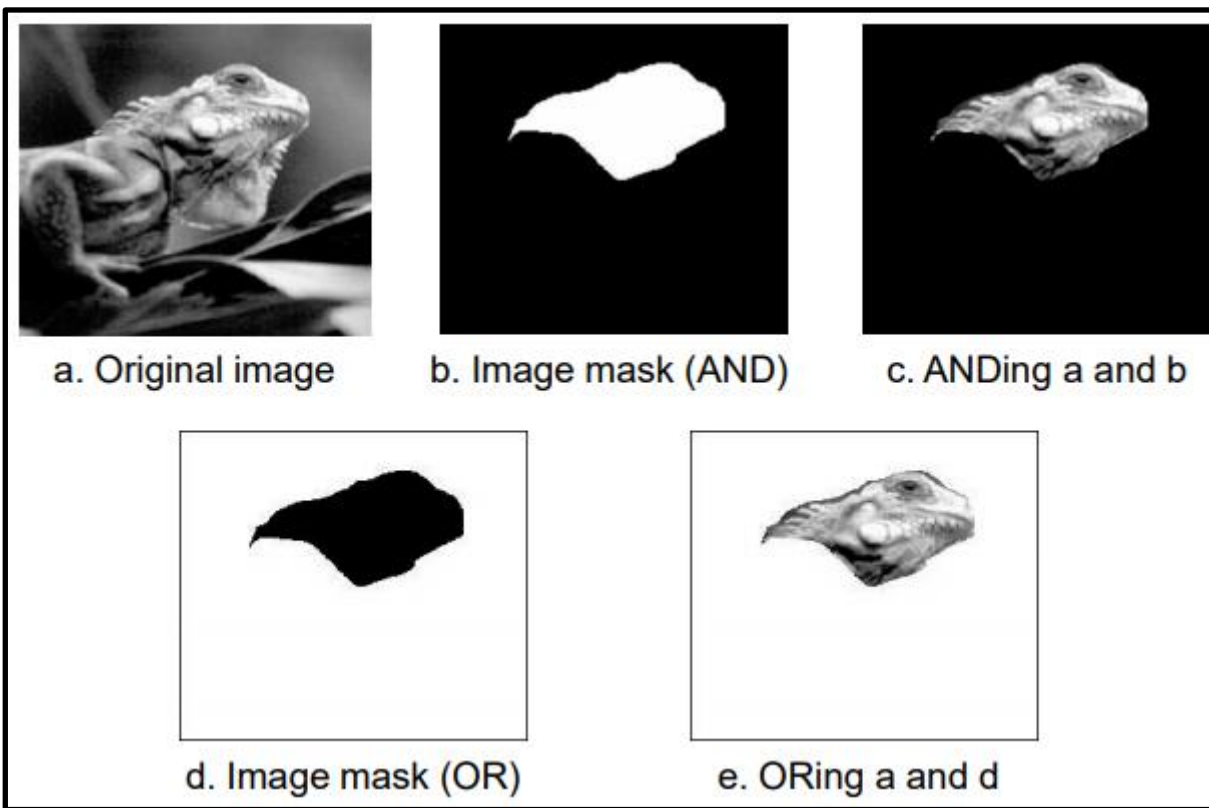
나눗셈 연산 (/1.2)

# 1. 픽셀 처리

---

## 단일 픽셀 처리 기법의 종류

2. 논리 연산 : AND, OR, NOT, NAND, NOR, XOR 등



# 1. 픽셀 처리

---

## 단일 픽셀 처리 기법의 종류

3. Grayscale : 컬러 이미지를 구성하는 각 픽셀의 R, G, B 값의 연산을 통해 흑백(회색조) 이미지로 변환 (3채널 → 1채널)



[224, 224, 3]  
HEIGHT WIDTH RGB



[224, 224]  
HEIGHT WIDTH

# 1. 픽셀 처리

---

## 단일 픽셀 처리 기법의 종류

4. 이진화 (고정 임계값) : 각 픽셀 값이 임계값보다 작으면 0, 크면 255로 변경

\* 아래는 uint8 기준, float64일 경우 0, 1로 변경

\* 임계값(threshold) : 이진화를 하기 위한 기준값



### 3. 이진화

---

#### 실습 1 (복습)

for 문을 이용하여 덧셈 산술 연산을 구현하기 (+100)

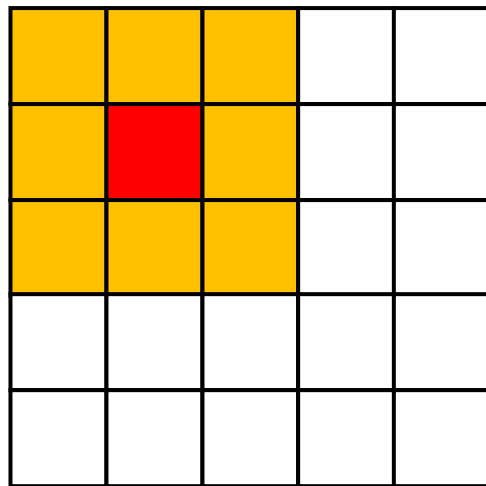


# 1. 픽셀 처리

---

## 픽셀 그룹 처리 (Pixel Group Processing)

- 출력 영상의 새로운 픽셀 값을 결정하기 위해 해당 픽셀 뿐만 아니라  
그 주위의 이웃 픽셀들도 함께 고려하는 공간 영역 연산



- 컨볼루션(Convolution) 연산으로 처리

# 1. 픽셀 처리

---

## 픽셀 그룹 처리 기법의 종류

1. 블러링 (Blurring) & 샤프닝 (Sharpening)

2. 지역 이진화 (가변 임계값)

: 픽셀 그룹 단위에 따른 임계치를 설정하고 그 값에 따라 0 또는 255로 변경

\* 아래는 uint8 기준, float64일 경우 0, 1로 변경

## 2. 컨볼루션 연산



## 2. 컨볼루션 연산

### 컨볼루션 연산 (Convolution, 합성곱)

처리하고자 하는 픽셀 값을 이웃 픽셀의 각각에 대응하는 2차원 배열 내의 가중치(계수)를 곱하고  
그 값들을 모두 더한 값으로 변경하는 연산

1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved  
Feature

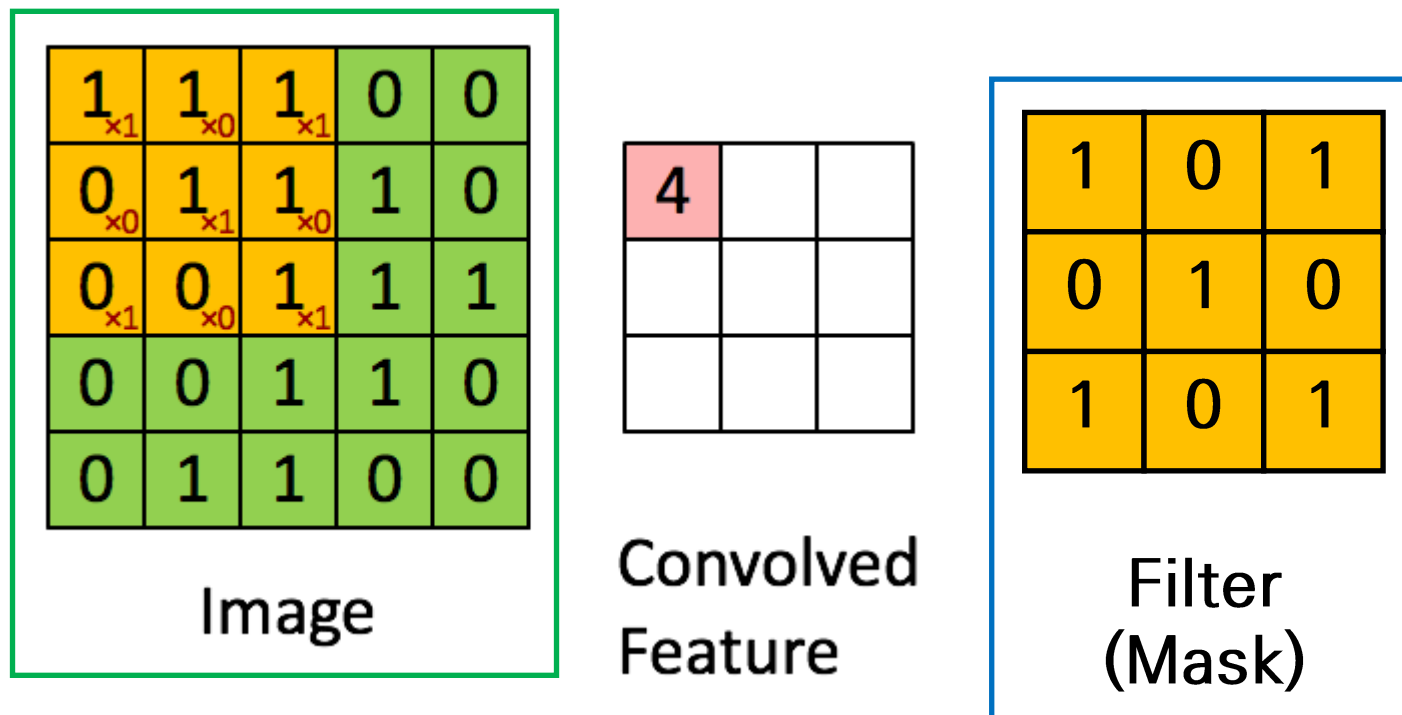
1	0	1
0	1	0
1	0	1

Filter  
(Mask)

## 2. 컨볼루션 연산

### 컨볼루션 연산 (Convolution, 합성곱)

처리하고자 하는 픽셀 값을 이웃 픽셀의 각각에 대응하는 2차원 배열 내의 가중치(계수)를 곱하고  
그 값들을 모두 더한 값으로 변경하는 연산



## 2. 컨볼루션 연산

---

### 필터 (Filter, Mask)

입력되는 신호의 일부 성분을 제거하거나 일부 특성을 변경하기 위해 설계된 하나의 시스템

1	0	1
0	1	0
1	0	1

Filter  
(Mask)

## 2. 컨볼루션 연산

---

### cv2.filter2D

컨볼루션 연산을 수행하는 함수

```
cv2.filter2D(src, ddepth, kernel)
```

src : 입력 영상

ddepth : 출력 영상의 테이터 타입

kernel : 필터 행렬

### 3. 이진화

### 3. 이진화

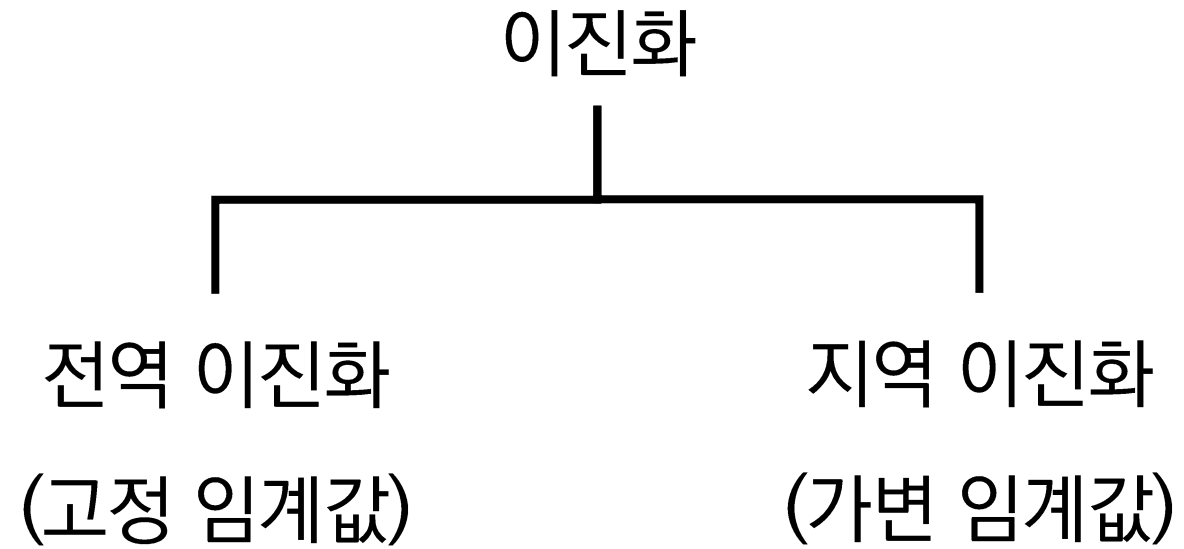
---

\* 임계값(threshold) : 이진화를 하기 위한 기준값

어떤 주어진 임계값(threshold)보다 밝은 픽셀들은 모두 흰색으로,  
그렇지 않은 픽셀들은 모두 검은색으로 바꾸는 작업.

### 3. 이진화

---



### 3. 이진화

---

#### 전역 이진화 (고정 임계값)

\* 아래는 uint8 기준, float64일 경우 0, 1로 변경

**각 픽셀** 값이 임계값보다 작으면 0, 크면 255로 픽셀 값 변경

\* 임계값(threshold) : 이진화를 하기 위한 기준값



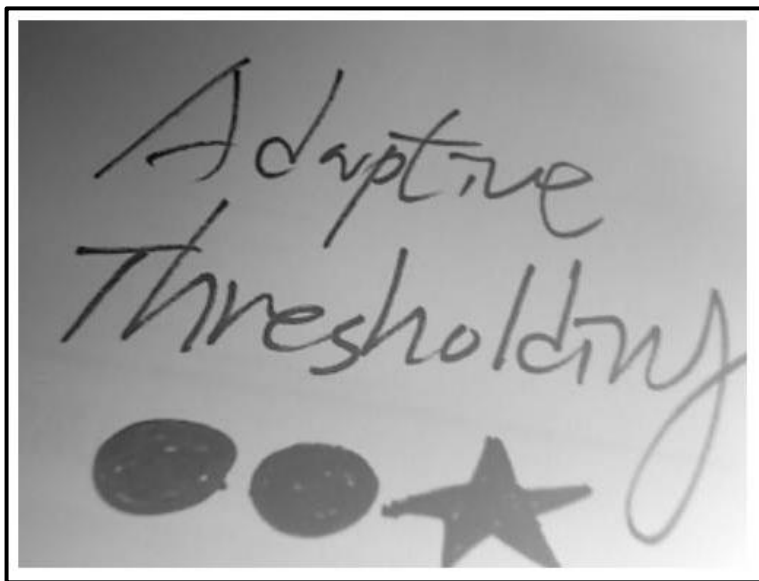


### 3. 이진화

---

#### 전역 이진화의 단점

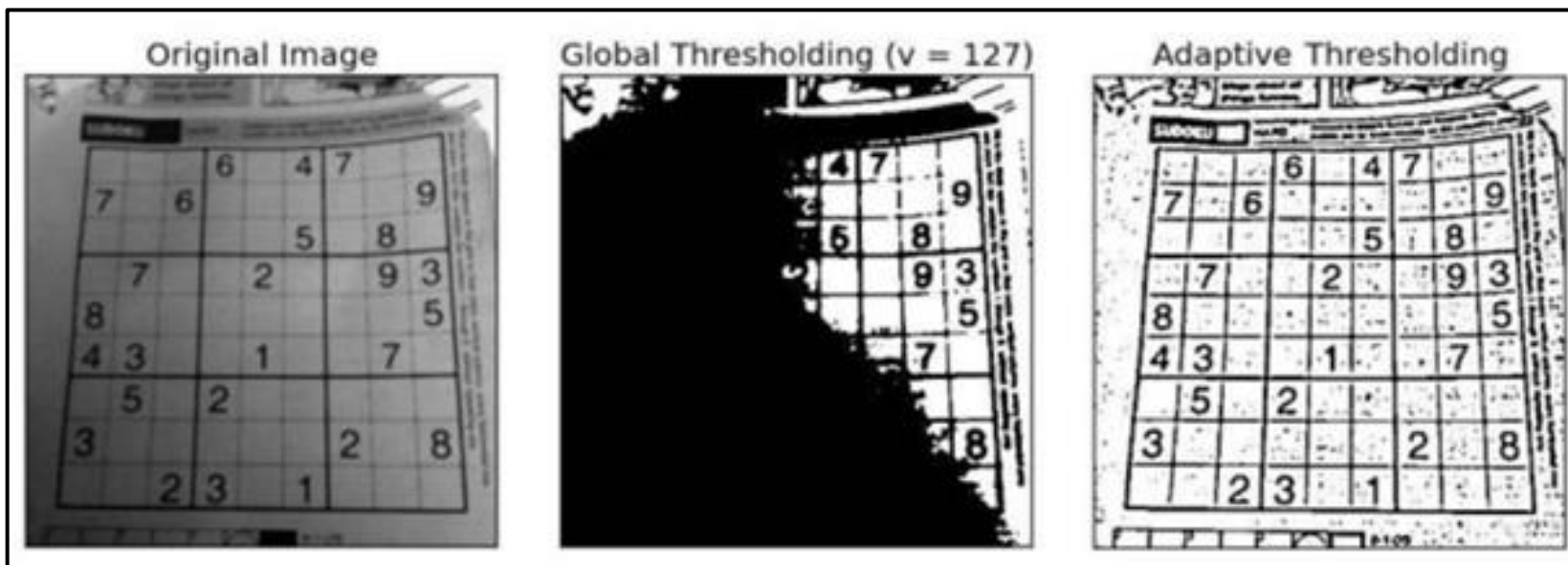
밝기 변화가 심한 영상에는 임계값 설정이 어려움



### 3. 이진화

#### 지역 이진화 (가변 임계값)

픽셀 그룹 단위마다 임계값을 설정하고 그 값보다 작으면 0, 크면 255로 픽셀 값 변경



### 3. 이진화

---

#### 실습 2

for 문을 이용하여 영상의 픽셀 값을 threshold 값보다 작은 값은 0, 큰 값은 255로 바꾸기



### 3. 이진화

---

## cv2.threshold

전역 이진화 함수

```
cv2.threshold(src, thresh, maxval, type)
```

src : 입력 영상

thresh : 임계값

maxval : 임계값보다 클 때 적용되는 최대값

type : 임계값 적용 방법

### 3. 이진화

---

## cv2.adaptiveThreshold

지역 이진화 함수

```
cv2.adaptiveThreshold(src, maxValue, adaptiveMethod, thresholdType, blockSize, C)
```

src : 입력 영상

maxValue : 임계값보다 클 때 적용되는 최대값

adaptiveMethod : 적용할 지역 이진화 알고리즘

thresholdType : 임계값 적용 방법

blockSize : 지역 이진화를 할 때 고려할 주변 픽셀 크기 (블록 크기)

C : 블록 내 평균값에서 뺀 값 (임계값을 결정하는 파라미터)

## 4. 블러링 & 샤프닝

## 4. 블러링 & 샤프닝

---

### 블러링 (Blurring)

이웃 픽셀들의 차이를 감소시켜 영상을 부드럽게 만드는 기법

영상의 잡음(Noise)을 제거하기 위해 쓰임



원본 영상



블러링 영상

## 4. 블러링 & 샤프닝

---

### 블러링 (Blurring)

블러링에 사용되는 필터 : 평균 값 필터, 미디언 필터, 가우시안 필터

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

평균 값 필터

73	142	156
52	85	120
64	102	85

미디언 필터

$1/16$	$2/16$	$1/16$
$2/16$	$4/16$	$2/16$
$1/16$	$2/16$	$1/16$

가우시안 필터



## 4. 블러링 & 샤프닝

---

### 평균 값 필터

영상의 특정 픽셀과 주변 픽셀들의 산술 평균을 결과 영상 픽셀 값으로 설정하는 필터

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

가장 간편하게 사용할 수 있는 블러링 방법

필터의 크기가 커질수록 더 흐려짐

## 4. 블러링 & 샤프닝

---

### 평균 값 필터

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

x

1	1	1
1	1	1
1	1	1

3 x 3 평균 값 필터

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

x

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

5 x 5 평균 값 필터

## 4. 블러링 & 샤프닝

---

### 실습 3

직접 평균 값 필터를 만들어 제공된 영상에 컨볼루션 연산 적용하기

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

3 x 3 평균 값 필터

$$\frac{1}{25} \times \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

5 x 5 평균 값 필터

## 4. 블러링 & 샤프닝

---

### 미디어 필터

입력 영상에서 주변 픽셀들의 값들을 오름 또는 내림 차순으로 정렬하여 그 중앙에 있는 값으로 픽셀 값을 대체하는 방식의 필터

73	142	156
52	85	120
64	102	85

소금 & 후추 잡음(Salt & Pepper Noise)을 효과적으로 제거



## 4. 블러링 & 샤프닝

### 미디어 필터

73	142	156
52	82	120
64	102	85

1차원 변환

73	142	156	52	82	120	64	102	85
----	-----	-----	----	----	-----	----	-----	----

정렬

52	64	73	82	85	102	120	142	156
----	----	----	----	----	-----	-----	-----	-----

중간값 선택

73	142	156
52	85	120
64	102	85

### 3. 이진화

---

## cv2.medianBlur

미디언 필터를 적용해주는 함수

```
cv2.medianBlur(src, ksize)
```

src : 입력 영상

ksize : 필터 크기

## 4. 블러링 & 샤프닝

---

### 가우시안 필터

가우시안 분포를 이용한 정규화 필터

$$\frac{1}{16} \times$$

1	2	1
2	4	2
1	2	1

모든 가우시안 필터의 합은 1

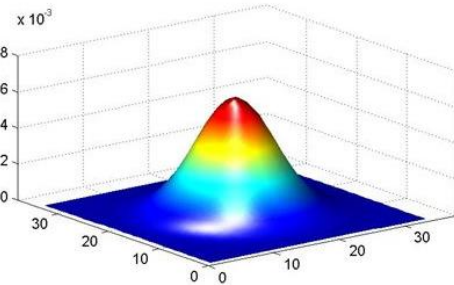
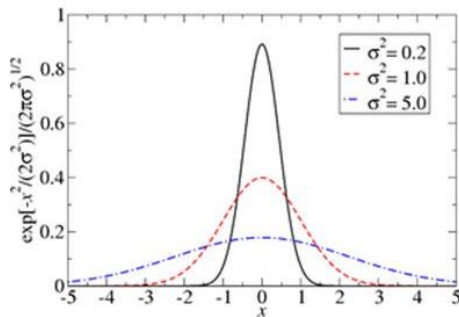
빠른 처리 가능

## 4. 블러링 & 샤프닝

### 가우시안 분포

#### Gaussian Filter

- 1D Gaussian filter:  $f(x) = e^{-\frac{x^2}{2\sigma^2}}$
- 2D Gaussian filter:  $f(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$



모든 과학분야에서 가장 보편적인 정규 분포  
실험 오차 측정 등에 많이 사용



## 4. 블러링 & 샤프닝

---

### 가우시안 필터

$$\frac{1}{16} \times$$

1	2	1
2	4	2
1	2	1

3 x 3 가우시안 필터

$$\frac{1}{256} \times$$

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

5 x 5 가우시안 필터

## 4. 블러링 & 샤프닝

---

### 실습 4

직접 가우시안 필터를 만들어 제공된 영상에 컨볼루션 연산 적용하기

$$\frac{1}{16} \times$$

1	2	1
2	4	2
1	2	1

3 x 3 가우시안 필터

$$\frac{1}{256} \times$$

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

5 x 5 가우시안 필터

### 3. 이진화

---

## cv2.getGaussianKernel

가우시안 필터를 만들어주는 함수

```
cv2.getGaussianKernel(ksize, sigma, ktype)
```

ksize : 필터 크기

sigma : 가우시안 함수 식 중 시그마 값

ktype : 필터 타입 (default : float64)

### 3. 이진화

---

## cv2.GaussianBlur

가우시안 필터를 적용해주는 함수

```
cv2.GaussianBlur(src, ksize, sigmaX)
```

src : 원본 이미지

ksize : 필터 크기

sigmaX : 가우시안 함수 식 중 시그마값

## 4. 블러링 & 샤프닝

---

### 샤프닝 (Sharpening)

이웃 픽셀들의 차이를 크게 만들어 영상을 날카롭게 만드는 기법



원본 영상



샤프닝 영상

## 4. 블러링 & 샤프닝

---

### 샤프닝 필터

0	-1	0
-1	5	-1
0	-1	0

0	-2	0
-2	5	-2
0	-2	0

-1	-1	-1
-1	9	-1
-1	-1	-1

## 4. 블러링 & 샤프닝

---

### 실습 5

직접 샤프닝 필터를 만들어 제공된 영상에 컨볼루션 연산 적용하기

0	-1	0
-1	5	-1
0	-1	0

0	-2	0
-2	5	-2
0	-2	0

-1	-1	-1
-1	9	-1
-1	-1	-1

# 과제

---

## 로버츠(Roberts), 프리윗(Prewitt), 소벨(Sobel) 필터 사용해보기

| **형식**    보고서 형식(한글, 워드) 또는 PPT 형식

| **내용**    각 필터의 특징    \* 간단하게 작성해도 괜찮습니다

파이썬 코드    \* 각 필터를 만들어 filter2D 함수 사용

결과 화면 캡처    \* 원본 영상 + 필터 적용 영상

| **기한**    2월 16일 화요일 18:00



# 참고 자료

---

다크프로그래머 <https://darkpgmr.tistory.com/>

영상처리 카테고리

슬라이싱 <https://dojang.io/mod/page/view.php?id=2208>

조건문을 이용한 인덱싱 <https://rfriend.tistory.com/291>



감사합니다