



# Introduction to Computer vision and template matching

Division of Electrical Engineering  
Hanyang University, ERICA Campus

# Contents

## ◆ 컴퓨터 비전

- 소개
- 대표적인 문제와 어플리케이션들

## ◆ Low-level 비전

- Color space
- Gamma correction
- Histogram equalization
- Edge detection

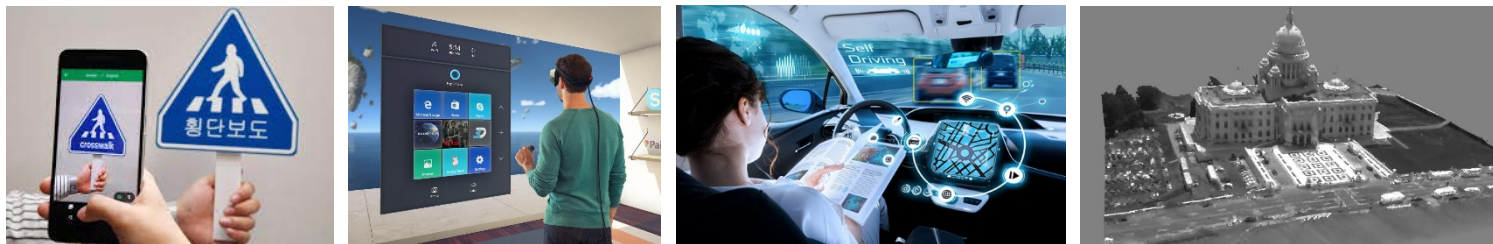
## ◆ Template matching

- Basic matching and patch distance
- MSE
- PSNR
- Cross Correlation
- Normalized Cross Correlation

## ◆ 실습 - 숨은 그림 찾기

# 컴퓨터 비전 : 소개

- ◆ **컴퓨터 비전** : 기계의 시각에 해당하는 부분을 연구하는 분야로써 사진이나 동영상과 같은 영상 정보를 통하여 필요한 정보를 획득하는 연구분야이다. 사진을 통한 번역, MR(mixed reality), 자율주행 자동차, 3차원 복원 등의 많은 응용분야를 가지고 있으며 최근 인공지능의 발전과 함께 더욱 활용 분야가 팽창하게 되었다.



- ◆ **계층적 처리** : 컴퓨터 비전의 처리 과정은 영상을 이해에 적합한 형식으로 변환하는 전처리와 다루기 편한 특징들로 요약하는 특징 추출 그리고 그 특징들로 부터 필요한 정보를 얻어내는 해석 부분으로 이루어져 있다. 전처리와 특징추출 부분은 low-level 비전, 해석 부분은 high-level 비전으로 불린다.



# 컴퓨터 비전 : 대표적인 문제와 어플리케이션들

## ■ Segmentation

<https://www.youtube.com/watch?v=KYNDzlcQMWA> (instance segmentation)

## ■ Detection

<https://www.youtube.com/watch?v=OOT3UIXZztE&t=12s> (4K mask RCNN)

## ■ 3d reconstruction

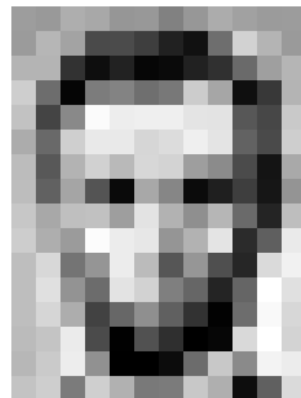
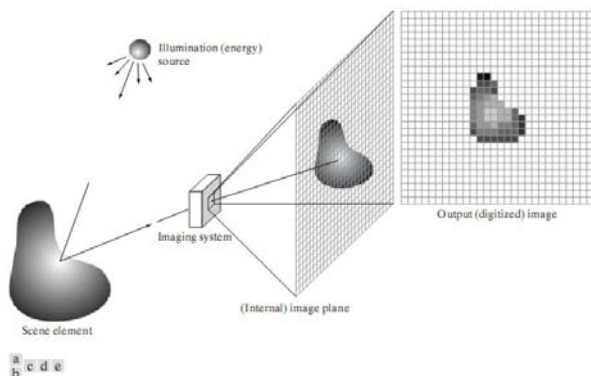
<https://www.youtube.com/watch?v=SkJG-uFU2yA> (high quality streamable)

## ■ Image Generation

<https://www.youtube.com/watch?v=XOxxPcy5Gr4> (GAN CelebA HD generation)

# Image formulation

- ◆ **이미지** : 카메라가 바라보고 있는 방향의 물체에 반사된 빛을 센서를 통해 이산화(discretize) 되어 데이터로써 저장된다. 이산적인 정보로써 2차원 행렬의 형태로 저장됨으로 대부분의 컴퓨터 비전 알고리즘을 이 2차원 행렬을 대상으로 작동하게 된다.

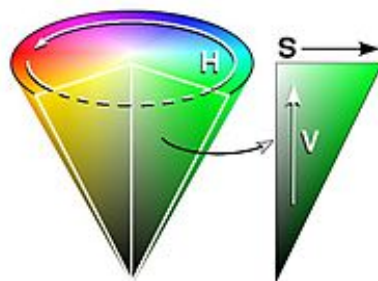
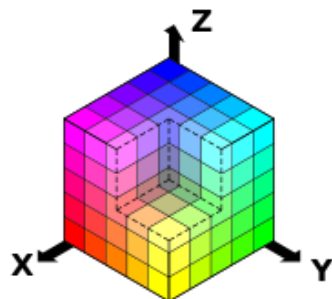


187	163	174	168	160	162	129	161	172	161	165	166
165	182	163	74	75	62	33	17	110	210	180	164
180	180	60	14	34	6	10	93	48	106	169	161
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	67	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	168	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	90	2	109	249	215
187	196	236	75	1	81	47	0	6	217	265	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

167	163	174	168	160	162	129	161	172	161	165	166
165	182	163	74	75	62	33	17	110	210	180	164
180	180	60	14	34	6	10	93	48	106	169	161
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	67	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	168	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	90	2	109	249	215
187	196	236	75	1	81	47	0	6	217	265	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

## Low-Level vision : color space

- ◆ **색 공간** : 색을 N 차원으로 표현한 공간 개념. 같은 색도 다른 표현을 가질 수 있다.
- **RGB** : 가장 기본적인 3차원의 색 공간 으로서 빨강, 초록, 파랑의 조합으로 대부분의 색을 표현한다.



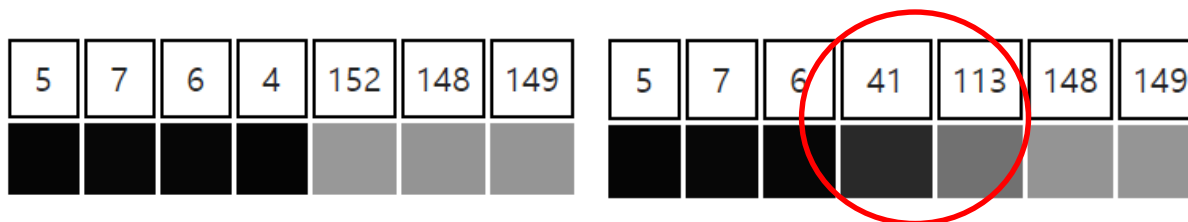
- **Gray scale** : 1차원의 색공간으로 흑백으로 색을 표현한다.
- **HSV** : 색, 채도, 명도를 기준으로 색을 표현하며 RGB 와 달리 채도와 명도를 구분하기 때문에 전처리로서 사용하는 경우가 있다.

## Low-level vision : edge detection

- Edge** : 밝기의 변화가 크거나 밝기 값의 불연속성을 보이는 점을 이야기한다. 물체의 경계선은 변화가 적은 평평한 영역에 비하여 많은 정보를 포함하고 있다. 또한 물체 간의 경계를 알 수 있게 됨에 따라 이후의 처리에 도움을 줄 수 있다. 따라서 이런 특성을 가지는 edge 나 corner를 찾아 그 정보를 활용하는 경우가 많다.



- Non-trivial task** : 밝기의 변화가 급격하게 일어나는 점을 찾는 것은 모호한 문제이다. 변화의 양을 정의하는 것은 가능하지만 어느 정도의 변화를 edge 로 볼지 애매하기 때문이다. 또한 노이즈로 인하여 edge 가 아닌 점이 강한 변화를 가지는 경우 또한 있을 수 있다.



어느 정도의 변화를 edge로 볼지 모호하다

# Low-level vision : edge detection (cont)

- Sobel edge detection** : 가장 기본적인 edge detection method 로써 노이즈로 인한 영향을 제거 하는 gaussian filtering 과 함께 1-차 미분 특성을 이용한다.



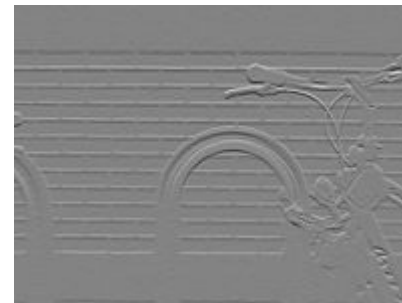
Image A



$|A_g|$  Normalized gradient magnitude



$A_{gX}$  X axis edge



$A_{gY}$  Y axis edge

-1	0	+1
-2	0	+2
-1	0	+1

$G_x$

+1	+2	+1
0	0	0
-1	-2	-1

$G_y$

0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	1	1	0	0	0	0
1	1	0	0	0	0	0

I

1	0	1
0	1	0
1	0	1

K

1	4	3	4	1
1	2	4	3	3
1	2	3	4	1
1	3	3	1	1
3	3	1	1	0

$I * K$

$$A_{gX} = A * G_x$$

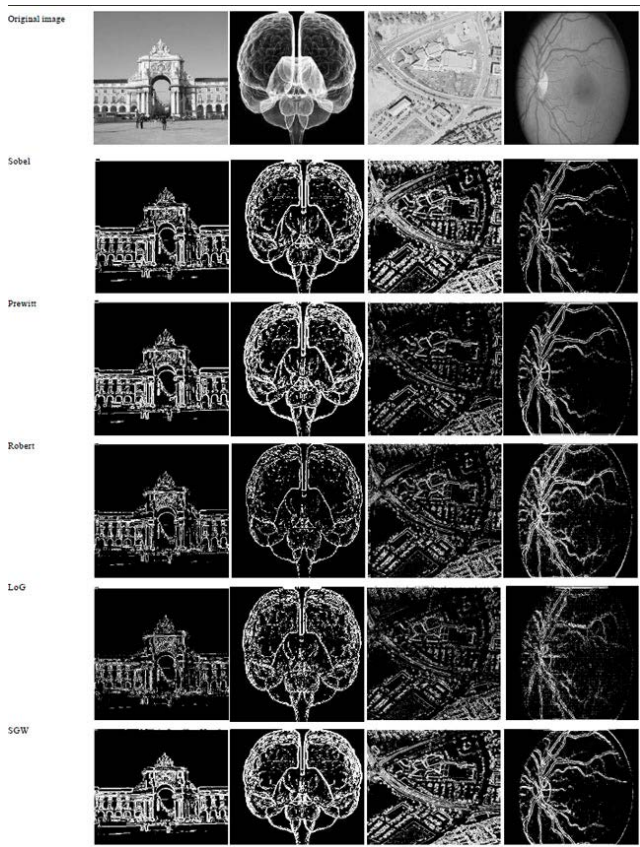
$$A_{gY} = A * G_y$$

$$|A_g| = \sqrt{A_{gX}^2 + A_{gY}^2}$$



# Low-level vision : edge detection (cont)

- **edge detection kernel:** 찾고자 하는 edge 의 특성에 따라 다양한 방식으로 kernel 을 구성하는 것이 가능하다.

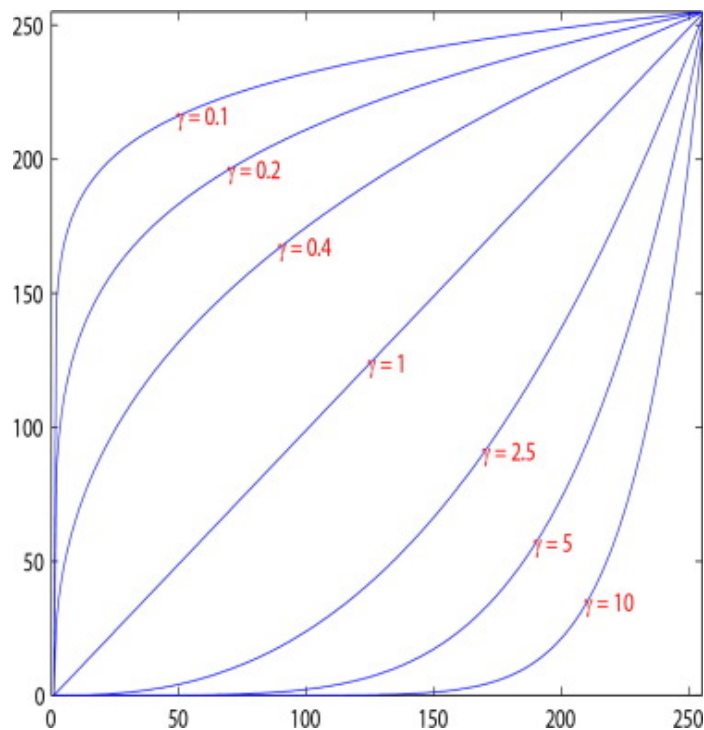


Operators	Gx	Gy
Sobel	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & -1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
Prewitt	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$
Robert	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	
LoG	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \\ & G & \end{bmatrix}$	

## Low-Level vision : gamma correction

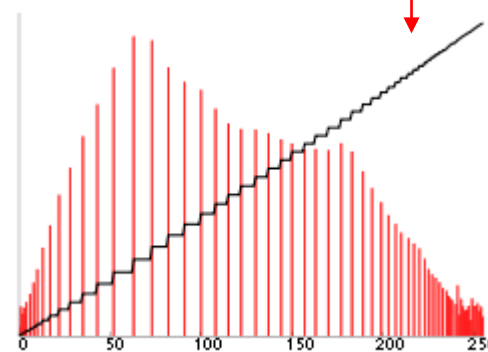
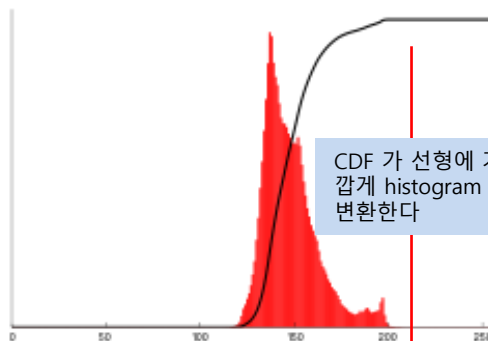
- **대비 증가** : 이미지의 밝음과 어두움 사이의 차이를 증가시키거나 줄임으로써 필요한 정보를 확장하거나 더 좋은 품질의 영상을 획득하기 위해 사용한다.

$$V_{\text{out}} = AV_{\text{in}}^{\gamma}$$



# Low-Level vision : histogram equalization

- 히스토그램** : 픽셀의 값과 그 값을 가지는 픽셀의 수를 표현한 것. 이미지의 전반적인 전역 특징을 나타낸다. 히스토그램을 이용한 영상 품질 개선의 예로써 히스토그램 평활화가 있다. 평활화는 통해 많은 수의 픽셀이 몰려 있는 부분은 넓은 칼라 공간을 할당한다.



이미지 I 의 histogram P를 계산

```
T[0] = P[0]
% cdf 계산
For k=1 to L
    T[k] = T[k-1] + P[k]
```

```
% 변환 계산
For r=0 to L
    S[r] = round(T[r]*L)
```

```
% 이미지에 적용
For y = 0 to M
    For x = 0 to N
        r = I[y, x]
        He[y, x] = S[r]
```

# Low-level vision : histogram equalization (cont)

0	3	1	1	1
1	1	1	1	3
1	1	1	3	1
2	3	1	1	2
1	1	2	1	1

Value	Cdf	Equalized v
0	1	0
1	18	2
2	21	2
3	25	3

Value	number
0	1
1	17
2	3
3	4

Normalization to max value  $4-1=3$

$$h(v) = \text{round} \left( \frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1) \right)$$

$\downarrow$                        $\downarrow$

$5 \times 5 = 25$                $1$

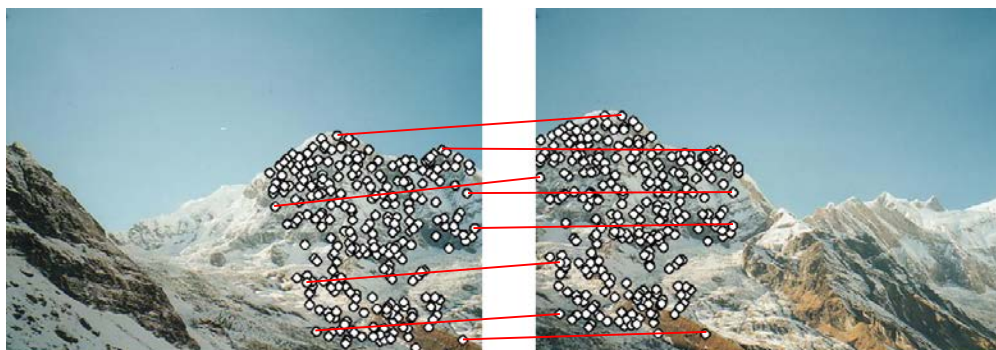
Round operation for discrete value

# Template matching : What is matching?

## ■ Matching

### ▶ Panorama image [1]

- Traditional Method 중 카메라로부터 출력된 여러 이미지 간의 "Align"과정
- Harris Corner, SIFT 등을 활용한 Feature map 추출과 동일한 지점을 찾는 과정



SIFT matching with 2 Image



Image aligned according to a Homography

# Template matching : What is matching?

## ■ Matching

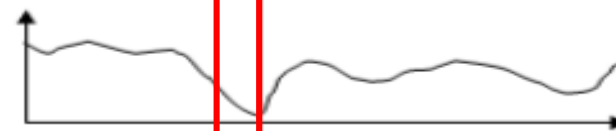
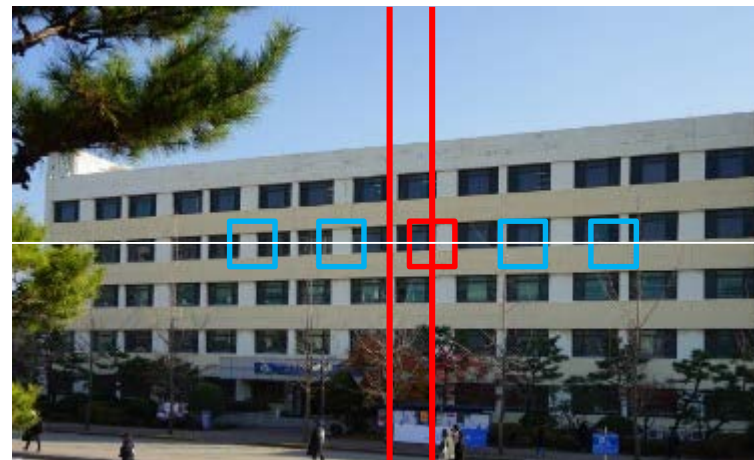
### ▶ Ex2. Stereo Matching

- Reference 이미지와 Target 이미지의 Depth map상의 matching을 통해 시차 (disparity)를 계산함을 목표로 함
- SAD(Sum of absolute difference), SSD(Sum of squared difference), NCC(Normalized cross correlation), Census transform, Rank transform 등

Left

Scan  
line

Right



disparity

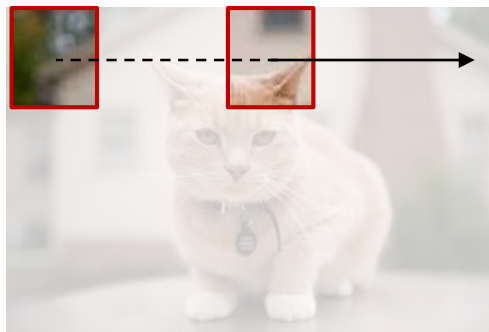
# Template matching : window sliding & patch distance

## ■ Template Matching

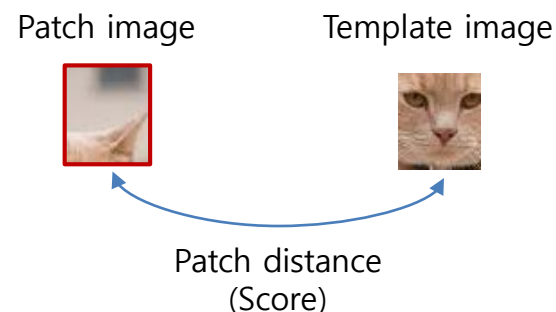
- ▶ Goal : Template 이미지가 주어졌을 때 Reference 이미지와 일치하는 부분을 탐색하는 방법
- ▶ Method
  1. Reference 이미지로 부터 **window sliding**을 통해 Template 이미지와 동일한 사이즈의 patch 추출
  2. Template 이미지와 patch 사이의 **patch distance**를 측정하여 score map을 출력
  3. Score map 최댓값의 patch 탐색



Reference image



Window sliding



# Template matching : window sliding & patch distance

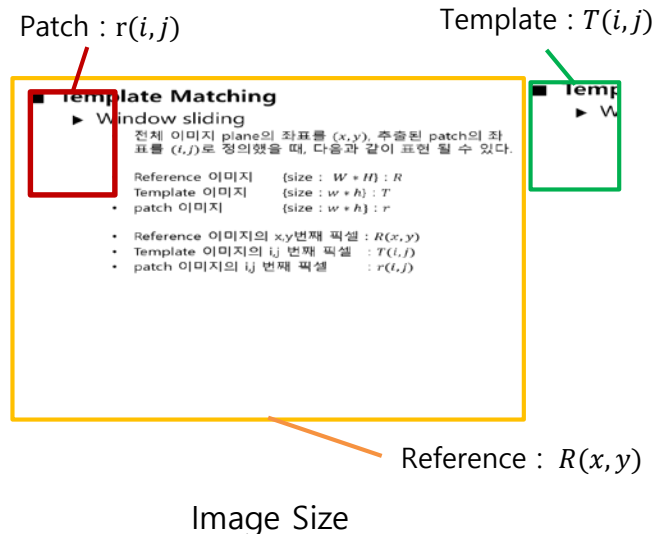
## ■ Template Matching

### ▶ Window sliding

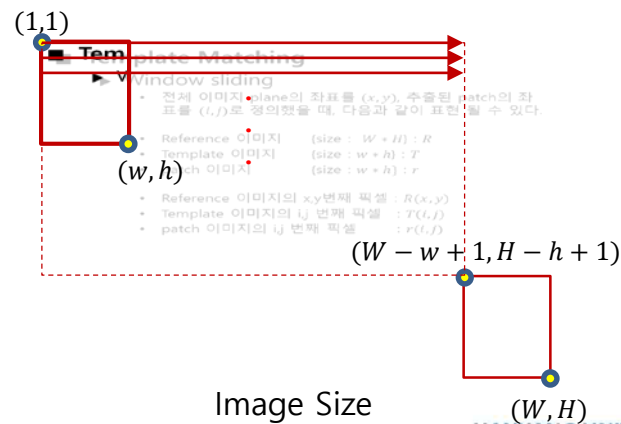
- 전체 이미지 plane의 좌표를  $(x,y)$ , 추출된 patch의 좌표를  $(i,j)$ 로 정의했을 때, 다음과 같이 표현 될 수 있다.

- Reference 이미지      {size :  $W * H$ } :  $R$
- Template 이미지      {size :  $w * h$ } :  $T$
- patch 이미지          {size :  $w * h$ } :  $r$

- Reference 이미지의  $x,y$ 번째 픽셀 :  $R(x,y)$
- Template 이미지의  $i,j$  번째 픽셀 :  $T(i,j)$
- patch 이미지의  $i,j$  번째 픽셀 :  $r(i,j)$



Window Sliding Process





# Template matching : window sliding & patch distance

## ■ Template Matching

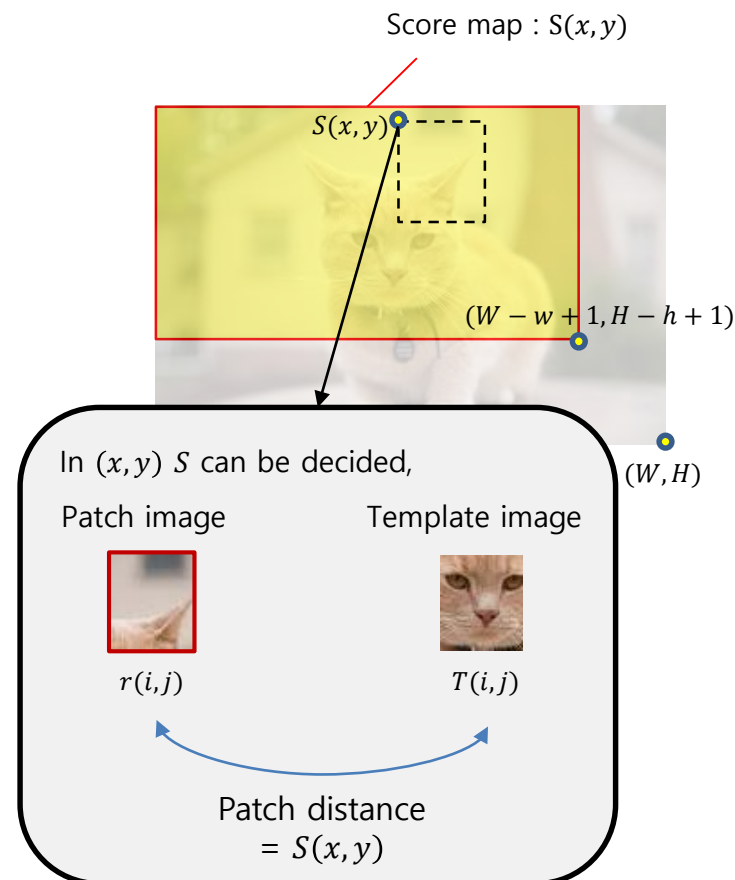
### ▶ Patch distance

- Patch는 Reference 이미지의 Window Slide로 얻을 수 있다.

$$r = WindowSlide(R, x, y)$$

- Window slide 방식을 통해 얻어낸 각 patch는 Template image와 다음 식과 같이 비교될 수 있다.

$$S(x, y) = dist(r, T)$$



# Template matching : window sliding & patch distance

## Distance Measure

▶ Mean-squared error (MSE)

- 평균 제곱 오차는 각 patch 이미지와 Template 이미지의 픽셀 오차 제곱에 평균을 취하여 얻을 수 있다.

$$S(x, y) = dist_{mse}(r, T) = \frac{1}{w * h} \sum_{i,j}^{w,h} \{R(x + i, y + j) - T(i, j)\}^2$$

- Ex)

1	2
3	4

Template ( $w = 2, h = 2$ )

1	1	2	1
2	3	4	1
1	2	1	1

Reference ( $W = 4, H = 2$ )

0.75	0	3	
2.5	4.5	5.75	

MSE Score ( $W = 3, H = 3$ )

$$S(1,1) = \frac{1}{2 * 2} \sum_{i,j}^{2,2} \{R(1 + i, 1 + j) - T(i, j)\}^2$$

$$= \frac{1}{4} ((1 - 1)^2 + (2 - 1)^2 + (3 - 2)^2 + (4 - 3)^2) = 0.75$$

$$S(1,2) = \frac{1}{2 * 2} \sum_{i,j}^{2,2} \{R(1 + i, 2 + j) - T(i, j)\}^2$$

$$= \frac{1}{4} ((1 - 1)^2 + (2 - 2)^2 + (3 - 3)^2 + (4 - 4)^2) = 0$$

⋮

$$S(2,3) = \frac{1}{2 * 2} \sum_{i,j}^{2,2} \{R(1 + i, 2 + j) - T(i, j)\}^2$$

$$= \frac{1}{4} ((1 - 4)^2 + (2 - 1)^2 + (3 - 1)^2 + (4 - 1)^2) = 5.75$$

# Patch distance : MSE, PSNR

## Distance Measure

$$r = \text{window\_slide}(R, x, y)$$

### ▶ Peak signal noise-to-ratio (PSNR)

- 최대 신호 대 잡음비로 영상, 이미지의 화질 손실 정보를 평가할 때 사용되며 MSE를 이용해서 계산할 수 있다.
- $MAX_r$ 은 patch image  $r$ 의 최대 픽셀 값으로,  $n$ 비트 표현 이미지일 경우  $MAX_r = 2^n - 1$ 로 표현할 수 있다.

$$\begin{aligned} S(x, y) &= \text{dist}_{psnr}(r, T) \\ &= 10 \log_{10} \left( \frac{(MAX_r)^2}{\text{dist}_{mse}(r, T)} \right) \\ &= 10 \log_{10} \left( \frac{(MAX_r)^2}{\frac{1}{wh} \sum_{i,j}^{w,h} \{R(x+i, y+j) - T(i,j)\}^2} \right) \end{aligned}$$

- Ex) when, Image is uint8 format,  $MAX_r = 2^8 - 1$

0.75	0	3	
2.5	4.5	5.75	

MSE Score ( $W = 3, H = 3$ )

49.4	inf	43.4	
44.2	41.6	40.5	

PSNR Score ( $W = 3, H = 3$ )

$$\begin{aligned} S(1,1) &= 10 \log_{10} \left( \frac{(MAX_r)^2}{\text{dist}_{mse}(r, T)} \right) \\ &= 10 * \log_{10} \left( \frac{255^2}{0.75} \right) = 49.4 \\ &\vdots \\ S(2,3) &= 10 \log_{10} \left( \frac{(MAX_r)^2}{\text{dist}_{mse}(r, T)} \right) \\ &= 10 * \log_{10} \left( \frac{255^2}{5.75} \right) = 40.5 \end{aligned}$$

## Patch distance : CC, NCC

### Distance Measure

#### ▶ Cross Correlation

- 교차 상관으로 Cross Correlation은 Convolution으로 부터 유래된다.
- Cross Correlation의 장점은 빠른 속도의 연산이라는 장점이 있다.

$$\begin{aligned} S(x, y) &= \text{dist}_{CC}(r, T) = r \otimes T \\ &= \sum_{i,j}^{w,h} T(i, j) R(x + i, y + j) \end{aligned}$$

#### ▶ Normalized Cross Correlation

- Reference 이미지와 Template 이미지를 정규화 하여 Cross Correlation을 산출하도록 한다.
- $\mu_T$ 는 Template 이미지의 평균이며  $\mu_r$ 은 patch 이미지의 평균이다.

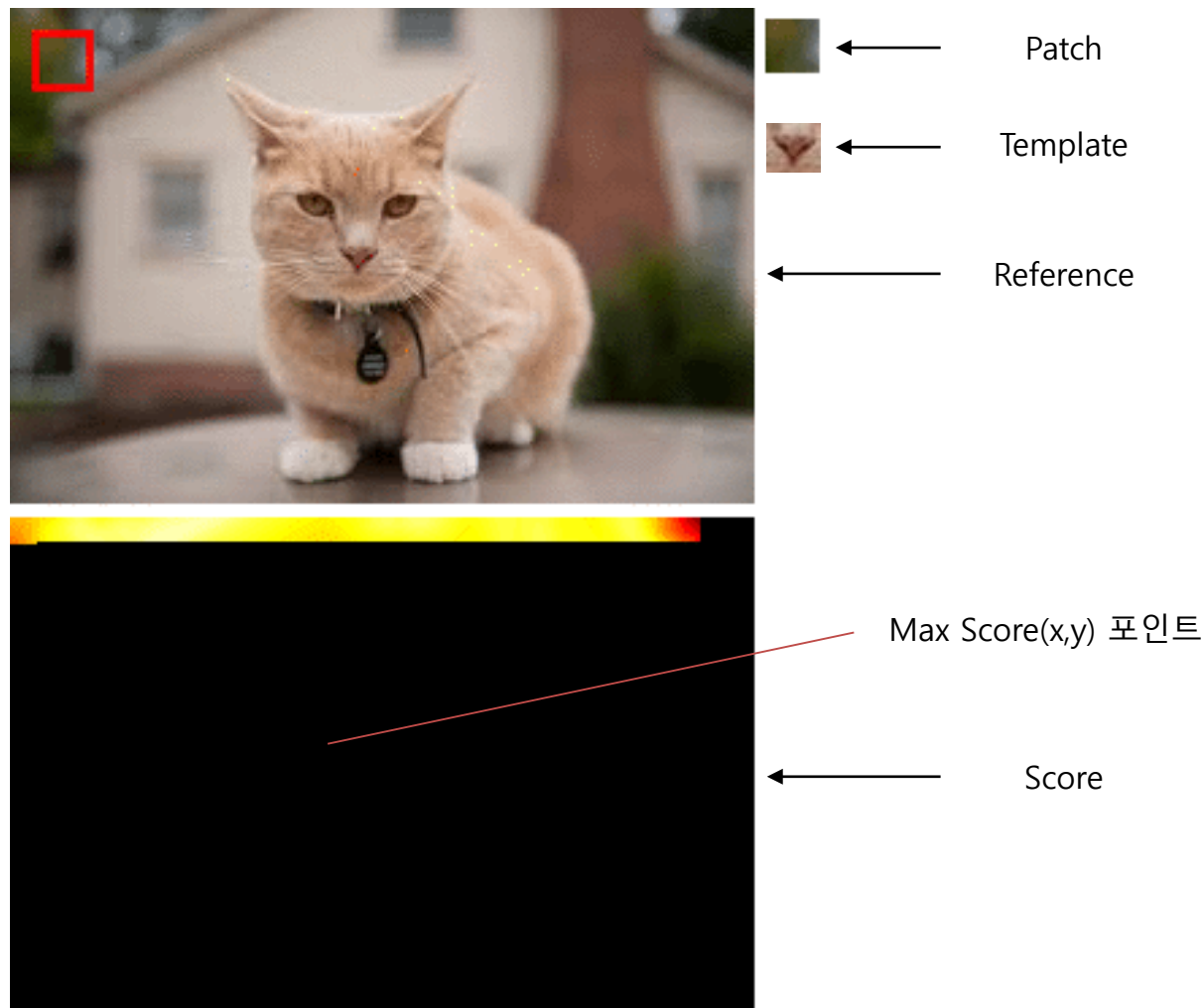
$$\begin{aligned} S(x, y) &= \text{dist}_{NCC}(r, T) \\ &= \frac{\sum_{i,j}^{w,h} [T(i, j) - \mu_T][R(x + i, y + j) - \mu_r]}{\sqrt{\sum_{i,j}^{w,h} [T(i, j) - \mu_T]^2 \sum_{i,j}^{w,h} [R(x + i, y + j) - \mu_r]^2}} \end{aligned}$$

#### Note

Template Matching시 Cross-Correlation은 Distance-measure로부터 motivate 되었지만, **Similarity-measure**로 보는 것이 적합하다.

# Patch distance : CC, NCC

## ■ MSE Example

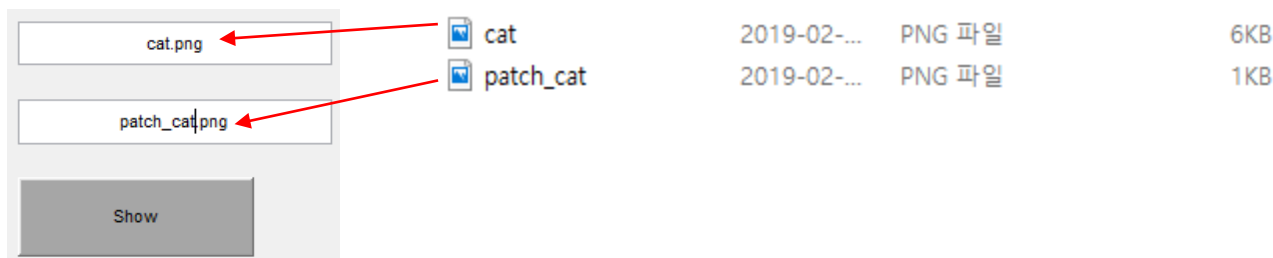


# 실습 1 : 숨은 그림 찾기

## ■ 사용법

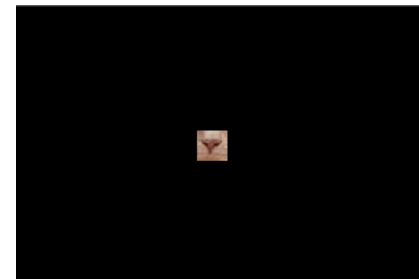
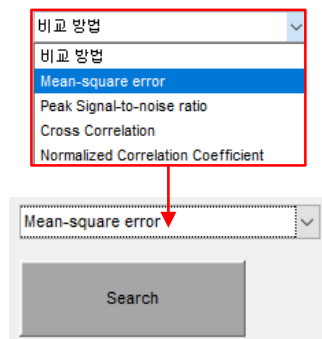
### ▶ 1. 이미지 불러오기

- "파일 이름.png", "패치 이름.png" 를 수정하고 Show를 눌러 파일을 불러오도록 한다.



### ▶ 2. 비교 방법 설정 및 탐색

- MSE, PSNR, CC, NCC 등의 측정 방식을 설정하고 Search를 통해 탐색한다.



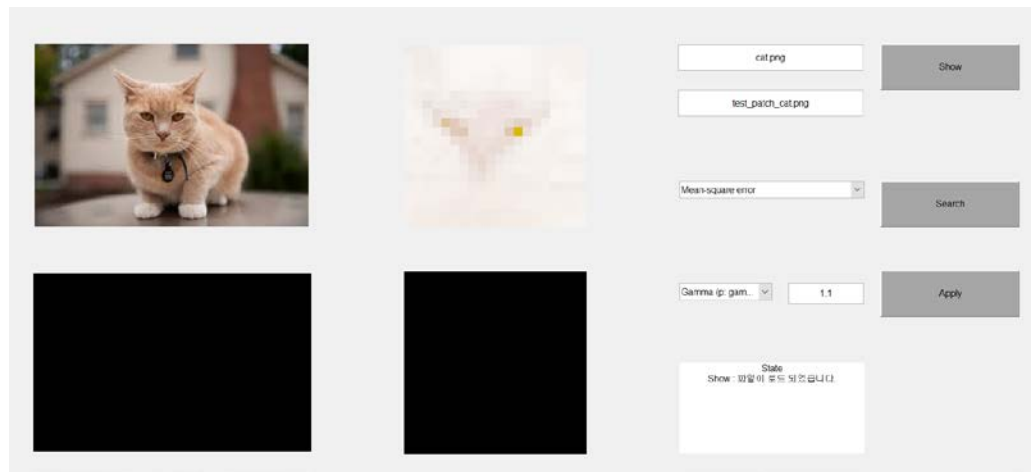
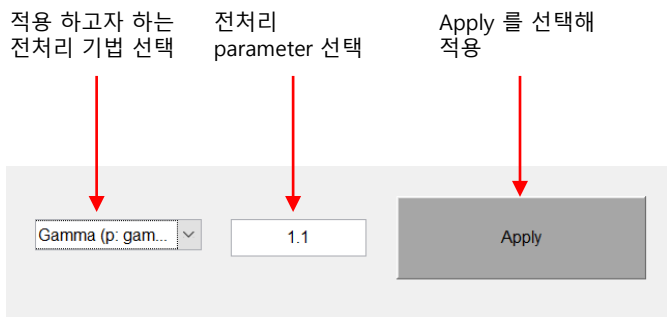
State  
Mean-square error : 1

# 실습 2 : 전처리

## ■ 사용법

### ▶ 1. 전처리 적용하기

- Test\_patch\_cat.png를 불러온 후 전처리 기법들을 이용하여 template matching 수행



HSV : template 과 이미지를 HSV color space 로 변환한다. Parameter 는 H 채널에 주어지는 가중치 이다. 0-10 사이의 값이다.

Gamma : gamma correction 을 이미지에 적용 한다. Parameter 는 gamma 값이다. 0-2 사이의 값을 사용해야 한다.

Histogram equalization : equalization 을 적용 한다. Parameter 는 bin 의 개수를 의미한다.

### 실습 3 : Low-Level 비전 이용하여 숨은 그림 찾기

#### ■ Low-Level 알고리즘과 patch-distance 선택

noise 와 outlier 등이 포함된 template에 대하여 전처리와 적합한 patch-distance 을 선택하여 matching 을 수행해 본다. 각각의 알고리즘의 특성을 생각하여 알고리즘과 파라미터를 설정하는 것이 필요.

#### ■ 오차 측정

참 값과 구해진 값의 위치의 차이로 오차를 측정

