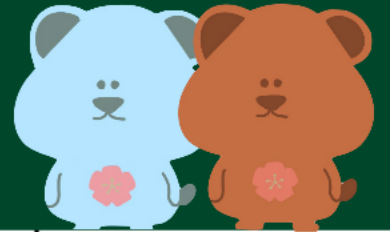


컴퓨터공학과 이주원(2276242)



15주차 발표

컴퓨터비전 ~ 입체형 이미지에서의 합성곱

목차

#01 Computer Visions

#02 CNN

#03 Convolutions over Volumes

#04 퀴즈 리뷰



1.1 Computer Vision Problems

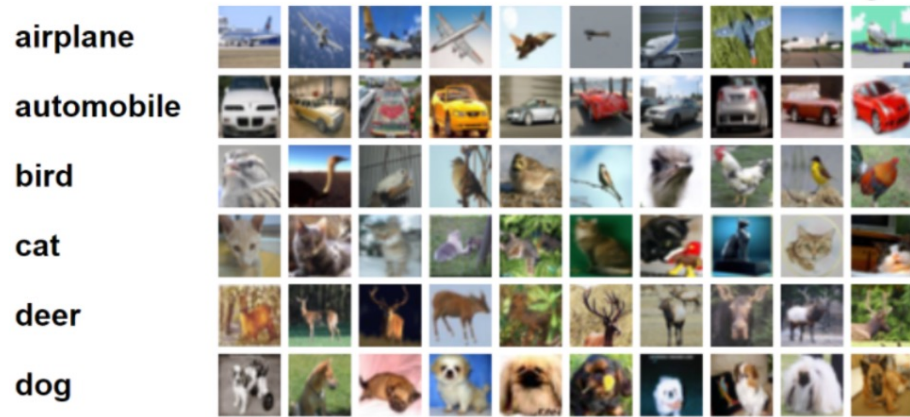
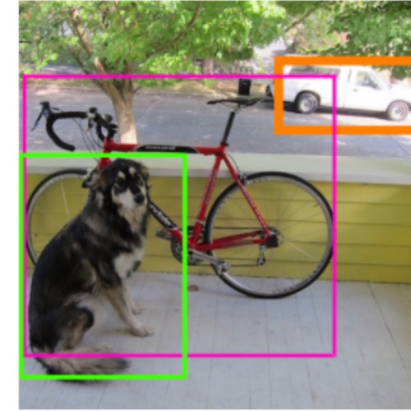


Image Classification



Object detection

content image



Ancient city of Persepolis

style image



The Starry Night (Van Gogh)

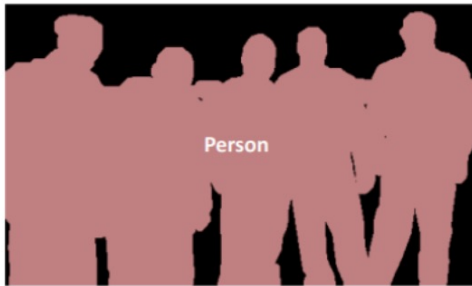
generated image



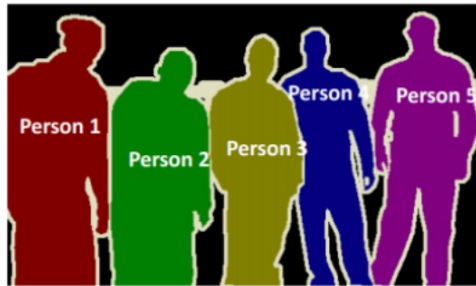
Persepolis
in Van Gogh style

Neural Style Transfer

1.1 Computer Vision Problems

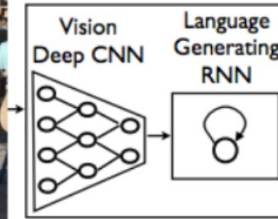


Semantic Segmentation



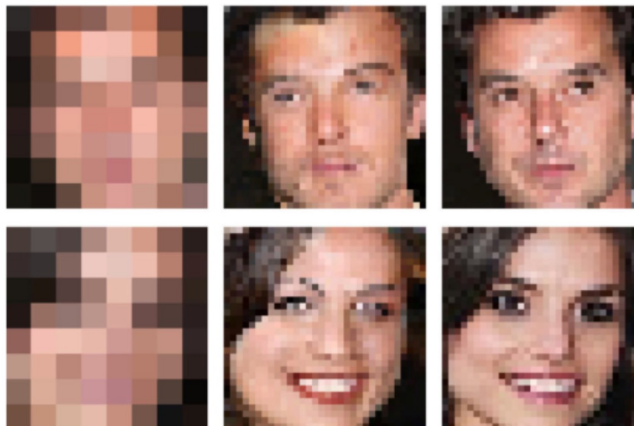
Instance Segmentation

Image Segmentation



A group of people shopping at an outdoor market.
There are many vegetables at the fruit stand.

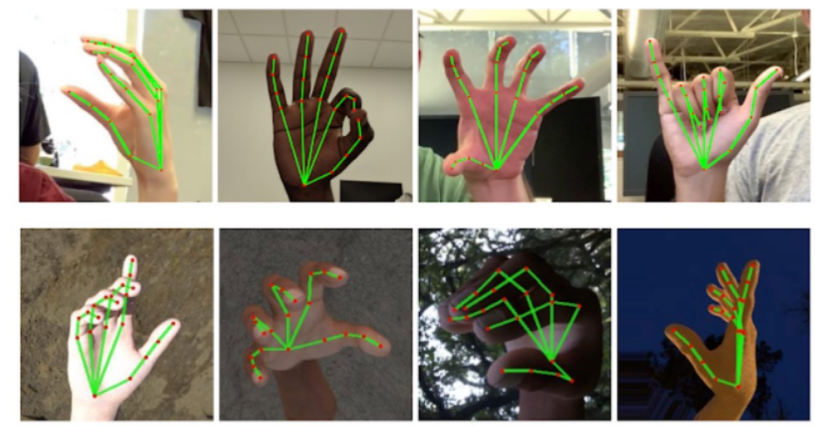
Image Captioning



Super Resolution

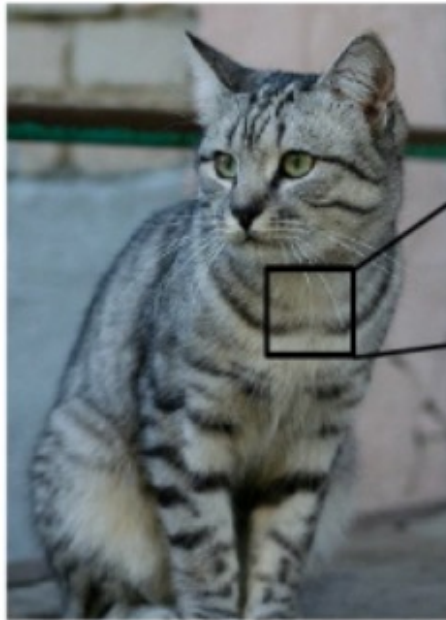


Text Detection & OCR



Keypoint Detection

1.2 Deep Learning on Large images



This image by Nikita is
licensed under CC-BY 2.0

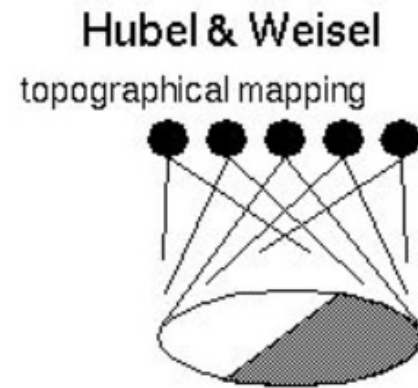
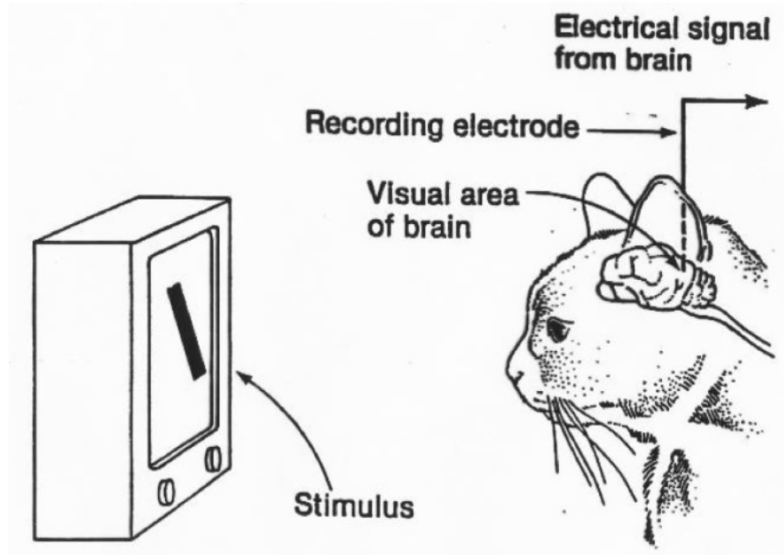
| | | | | | | | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| [| 185 | 112 | 180 | 111 | 184 | 99 | 186 | 99 | 96 | 183 | 112 | 119 | 184 | 97 | 93 | 87] |
| [| 91 | 98 | 182 | 186 | 184 | 79 | 88 | 183 | 99 | 185 | 123 | 136 | 118 | 185 | 84 | 85] |
| [| 76 | 85 | 98 | 185 | 128 | 186 | 87 | 96 | 95 | 99 | 155 | 112 | 186 | 183 | 99 | 89] |
| [| 99 | 81 | 81 | 83 | 128 | 131 | 127 | 188 | 95 | 98 | 182 | 99 | 96 | 93 | 181 | 94] |
| [| 188 | 91 | 81 | 84 | 89 | 91 | 88 | 85 | 181 | 187 | 189 | 98 | 75 | 84 | 96 | 95] |
| [| 114 | 188 | 85 | 55 | 55 | 69 | 64 | 54 | 64 | 87 | 152 | 129 | 98 | 74 | 84 | 91] |
| [| 133 | 137 | 147 | 183 | 85 | 81 | 88 | 85 | 52 | 54 | 74 | 84 | 182 | 93 | 85 | 82] |
| [| 128 | 132 | 144 | 148 | 189 | 96 | 86 | 78 | 82 | 86 | 83 | 83 | 88 | 73 | 86 | 181] |
| [| 125 | 133 | 148 | 137 | 119 | 121 | 137 | 94 | 85 | 79 | 88 | 85 | 54 | 84 | 72 | 98] |
| [| 127 | 125 | 131 | 147 | 133 | 127 | 126 | 131 | 111 | 96 | 88 | 75 | 61 | 64 | 72 | 84] |
| [| 115 | 114 | 189 | 123 | 158 | 148 | 131 | 118 | 113 | 189 | 188 | 92 | 74 | 85 | 72 | 78] |
| [| 88 | 83 | 98 | 87 | 188 | 147 | 131 | 118 | 113 | 114 | 113 | 189 | 186 | 95 | 77 | 88] |
| [| 87 | 77 | 86 | 81 | 77 | 79 | 182 | 123 | 117 | 115 | 117 | 125 | 125 | 138 | 115 | 87] |
| [| 82 | 85 | 82 | 89 | 78 | 71 | 88 | 181 | 124 | 126 | 118 | 181 | 187 | 114 | 111 | 119] |
| [| 83 | 85 | 75 | 88 | 83 | 71 | 82 | 81 | 128 | 138 | 125 | 185 | 81 | 98 | 118 | 118] |
| [| 87 | 85 | 71 | 87 | 186 | 95 | 89 | 45 | 76 | 138 | 126 | 187 | 92 | 84 | 181 | 112] |
| [| 118 | 97 | 82 | 86 | 117 | 123 | 116 | 86 | 41 | 91 | 95 | 93 | 89 | 95 | 182 | 187] |
| [| 184 | 148 | 112 | 88 | 82 | 128 | 124 | 184 | 76 | 48 | 45 | 86 | 88 | 181 | 182 | 189] |
| [| 157 | 178 | 157 | 128 | 93 | 86 | 114 | 132 | 112 | 97 | 89 | 55 | 78 | 82 | 98 | 94] |
| [| 118 | 128 | 114 | 181 | 139 | 188 | 189 | 118 | 121 | 114 | 114 | 87 | 86 | 53 | 88 | 88] |
| [| 128 | 112 | 86 | 117 | 158 | 144 | 128 | 125 | 186 | 187 | 182 | 93 | 87 | 81 | 72 | 79] |
| [| 123 | 187 | 96 | 86 | 83 | 112 | 153 | 148 | 122 | 189 | 184 | 76 | 88 | 187 | 112 | 99] |
| [| 122 | 125 | 182 | 88 | 82 | 86 | 84 | 117 | 145 | 148 | 153 | 182 | 58 | 78 | 92 | 187] |
| [| 122 | 184 | 148 | 183 | 71 | 56 | 78 | 83 | 93 | 183 | 119 | 139 | 182 | 61 | 89 | 84] |
|] | | | | | | | | | | | | | | | |] |

What the computer sees

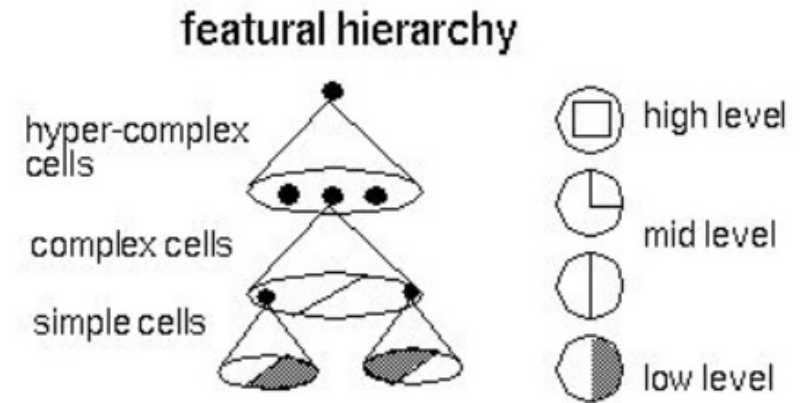
An image is just a big grid of numbers between [0, 255]:

e.g. 800 x 600 x 3
(3 channels RGB)

2.1 CNN Introduction



Topographical Mapping

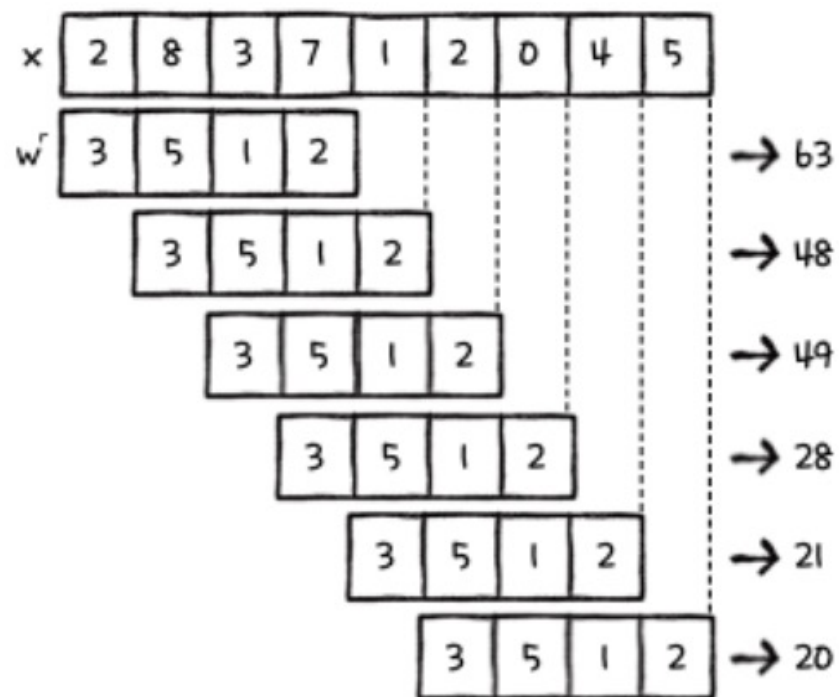
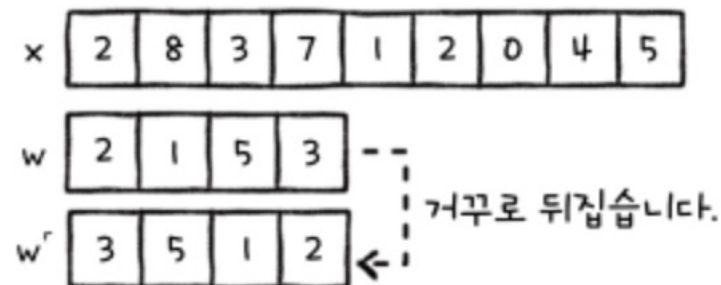


Hierarchical Organization

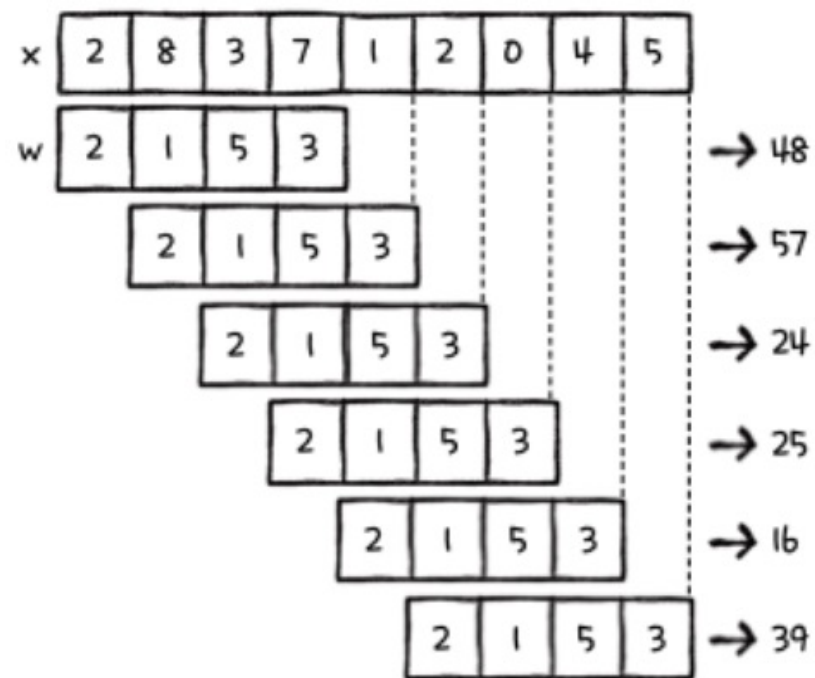


CNN

2.2 CNN - Convolution

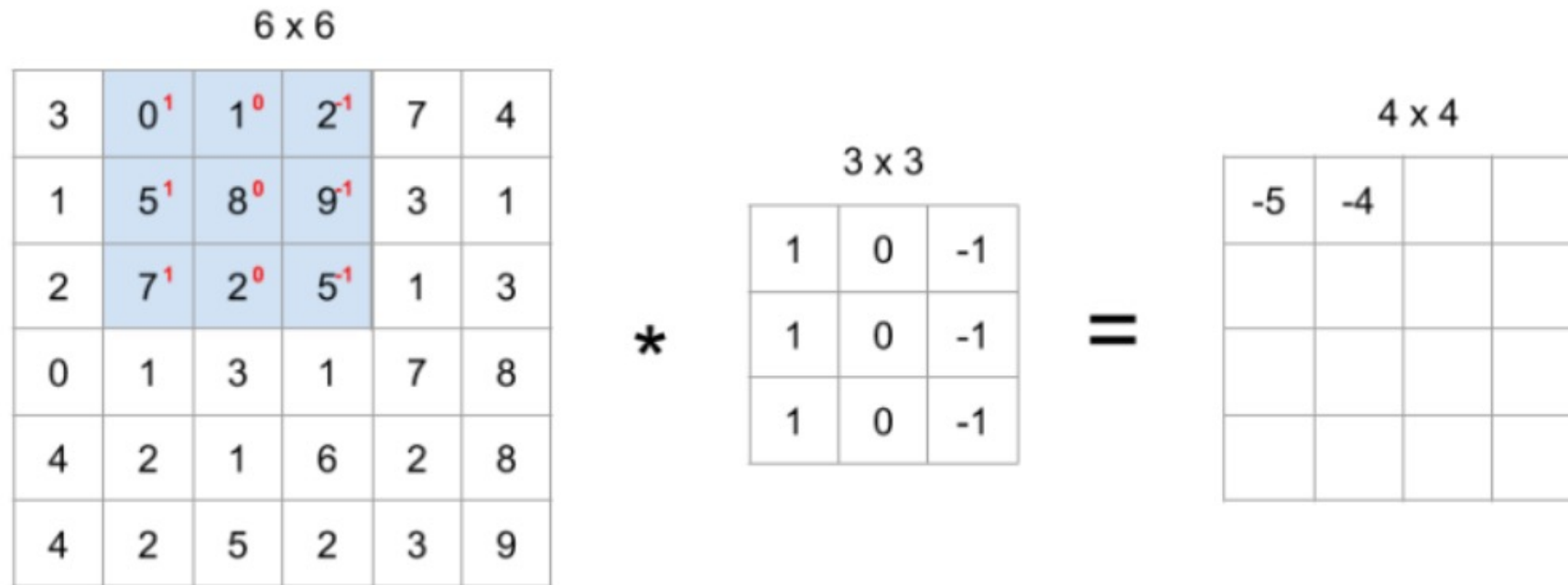


합성곱



교차상관

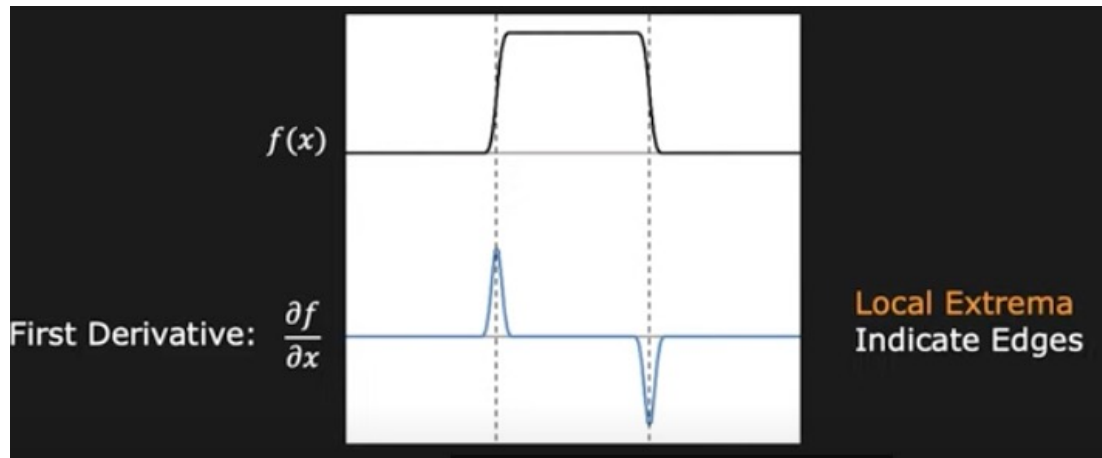
2.2 CNN - Convolution



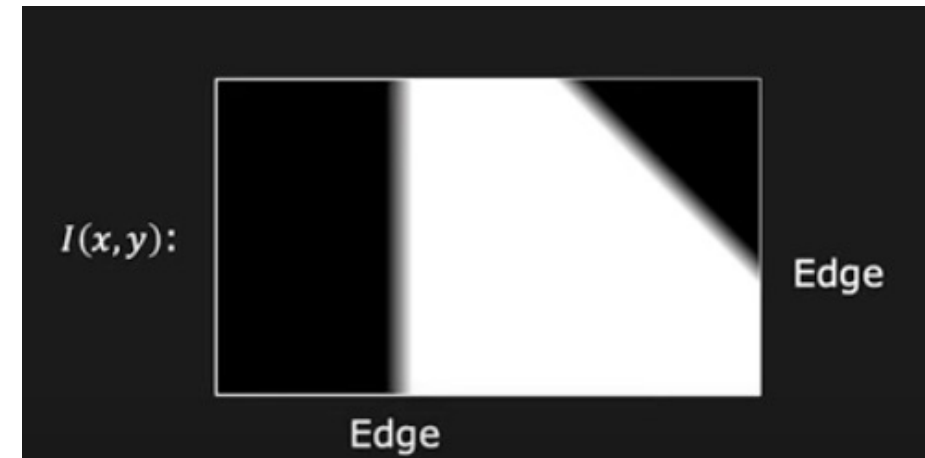
$$0 \times 1 + 1 \times 0 + 2 \times -1 + 5 \times 1 + 8 \times 0 + 9 \times -1 + 7 \times 1 + 2 \times 0 + 5 \times -1 = -4$$

2.3 CNN Filter - Edge Detection

수학적 이해



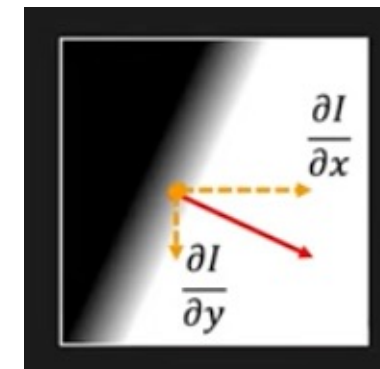
1D Edge Detection



2D Edge Detection

Gradient Magnitude $S = \|\nabla I\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$

Gradient Orientation $\theta = \tan^{-1}\left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x}\right)$



Gradient as Edge Detector

2.3 CNN Filter - Edge Detection

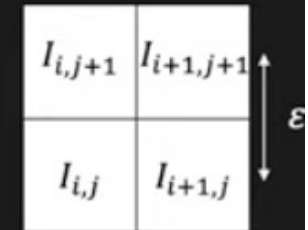
$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

유한 차분 근사 기법

Finite difference approximations:

$$\frac{\partial I}{\partial x} \approx \frac{1}{2\varepsilon} \left((I_{i+1,j+1} - I_{i,j+1}) + (I_{i+1,j} - I_{i,j}) \right)$$

$$\frac{\partial I}{\partial y} \approx \frac{1}{2\varepsilon} \left((I_{i+1,j+1} - I_{i+1,j}) + (I_{i,j+1} - I_{i,j}) \right)$$



Can be implemented as Convolution!

$$\frac{\partial}{\partial x} \approx \frac{1}{2\varepsilon} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$$

$$\frac{\partial}{\partial y} \approx \frac{1}{2\varepsilon} \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}$$

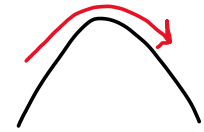
Prewitt Filter

2.3 CNN Filter - Edge Detection

| Gradient | Roberts | Prewitt | Sobel (3x3) | | | | | | | | | | | | | | | | | | | | | | |
|---------------------------------|--|---------|-------------|----|----|---|----|---|---|----|---|---|----|----|----|---|----|---|---|----|---|---|----|----|----|
| $\frac{\partial I}{\partial x}$ | <table><tr><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td></tr></table> | 0 | 1 | -1 | 0 | <table><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table> | -1 | 0 | 1 | -1 | 0 | 1 | -1 | 0 | 1 | <table><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-2</td><td>0</td><td>2</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table> | -1 | 0 | 1 | -2 | 0 | 2 | -1 | 0 | 1 |
| 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | |
| -1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | |
| -1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | |
| -1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | |
| -1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | |
| -1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | |
| -2 | 0 | 2 | | | | | | | | | | | | | | | | | | | | | | | |
| -1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | |
| $\frac{\partial I}{\partial y}$ | <table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>-1</td></tr></table> | 1 | 0 | 0 | -1 | <table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></table> | 1 | 1 | 1 | 0 | 0 | 0 | -1 | -1 | -1 | <table><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>-1</td><td>-2</td><td>-1</td></tr></table> | 1 | 2 | 1 | 0 | 0 | 0 | -1 | -2 | -1 |
| 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | -1 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | |
| -1 | -1 | -1 | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 2 | 1 | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | |
| -1 | -2 | -1 | | | | | | | | | | | | | | | | | | | | | | | |



2.3 CNN Filter - Edge Detection



Vertical edge detection

| | | | | | |
|----|----|----|---|---|---|
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |



*

| | | |
|---|---|----|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |



=

| | | | |
|---|----|----|---|
| 0 | 30 | 30 | 0 |
| 0 | 30 | 30 | 0 |
| 0 | 30 | 30 | 0 |
| 0 | 30 | 30 | 0 |



| | | | | | |
|---|---|---|----|----|----|
| 0 | 0 | 0 | 10 | 10 | 10 |
| 0 | 0 | 0 | 10 | 10 | 10 |
| 0 | 0 | 0 | 10 | 10 | 10 |
| 0 | 0 | 0 | 10 | 10 | 10 |
| 0 | 0 | 0 | 10 | 10 | 10 |
| 0 | 0 | 0 | 10 | 10 | 10 |



*

| | | |
|---|---|----|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

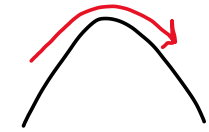


=

| | | | |
|---|-----|-----|---|
| 0 | -30 | -30 | 0 |
| 0 | -30 | -30 | 0 |
| 0 | -30 | -30 | 0 |
| 0 | -30 | -30 | 0 |

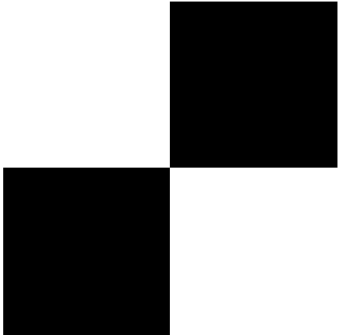


2.3 CNN Filter – Edge Detection



Vertical and Horizontal edge detection

Original Image with Vertical Edge



| | | | | | |
|----|----|----|----|----|----|
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 0 | 0 | 0 | 10 | 10 | 10 |
| 0 | 0 | 0 | 10 | 10 | 10 |
| 0 | 0 | 0 | 10 | 10 | 10 |

*

| | | |
|---|---|----|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

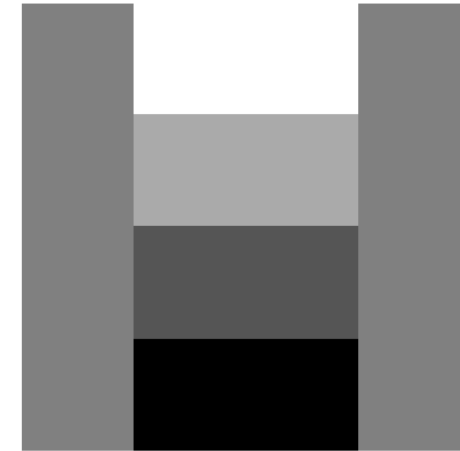
Vertical

*

| | | |
|----|----|----|
| 1 | 1 | 1 |
| 0 | 0 | 0 |
| -1 | -1 | -1 |

Horizontal

Vertical Edge Detection



Horizontal Edge Detection



2.3 CNN Filter - Edge Detection

다양한 필터들

| | | |
|---|---|----|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

vertical

| | | |
|---|---|----|
| 1 | 0 | -1 |
| 2 | 0 | -2 |
| 1 | 0 | -1 |

sobel

| | | |
|----|---|-----|
| 3 | 0 | -3 |
| 10 | 0 | -10 |
| 3 | 0 | -3 |

scharr

| | | |
|----|----|----|
| 1 | 1 | 1 |
| 0 | 0 | 0 |
| -1 | -1 | -1 |

horizontal

| | | |
|----|----|----|
| 1 | 2 | 1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

sobel

| | | |
|----|-----|----|
| 3 | 10 | 3 |
| 0 | 0 | 0 |
| -3 | -10 | -3 |

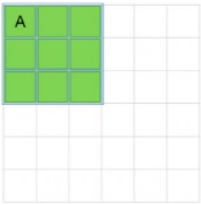
scharr

적합한 필터 탐색

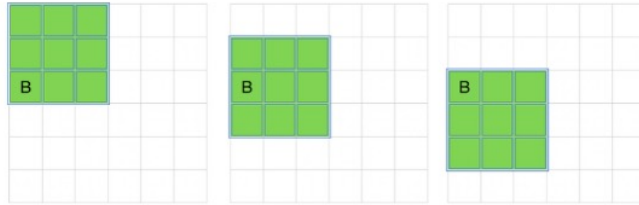
| | | |
|-------|-------|-------|
| w_1 | w_2 | w_3 |
| w_4 | w_5 | w_6 |
| w_7 | w_8 | w_9 |

2.4 CNN – Padding

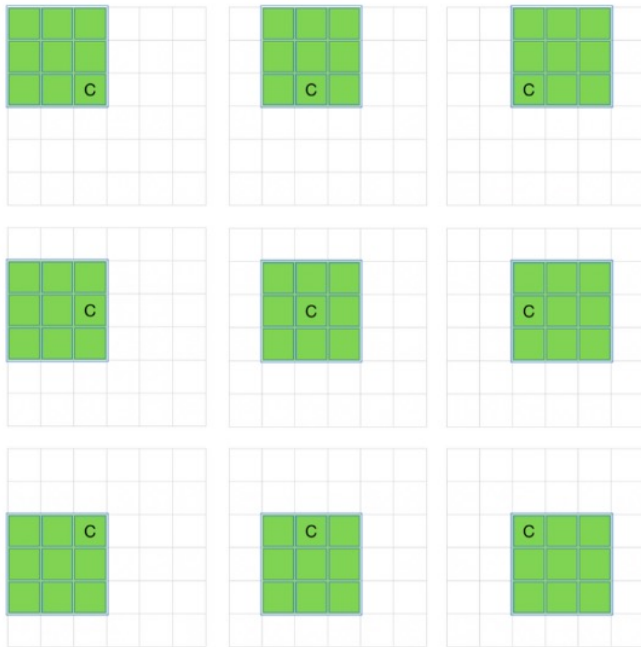
Corner Pixel



Edge Pixel



Middle Pixel



< 기존 방식 단점 >

- 1) 합성곱 연산을 반복하면 이미지가 축소
- 2) 가장자리 픽셀 -> 단 한번 사용



Padding

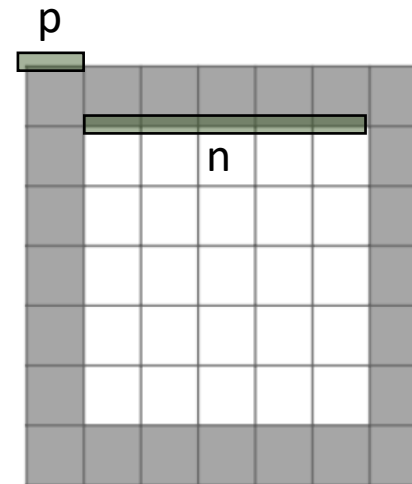
2.4 CNN – Padding

zero padding of size 2

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | | | | | | 0 | 0 |
| 0 | 0 | | | | | | 0 | 0 |
| 0 | 0 | | | | | | 0 | 0 |
| 0 | 0 | | | | | | 0 | 0 |
| 0 | 0 | | | | | | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Padding의 종류

1. Valid Padding : padding하지 않는 것
2. Same Padding : output image 크기 = input image의 크기



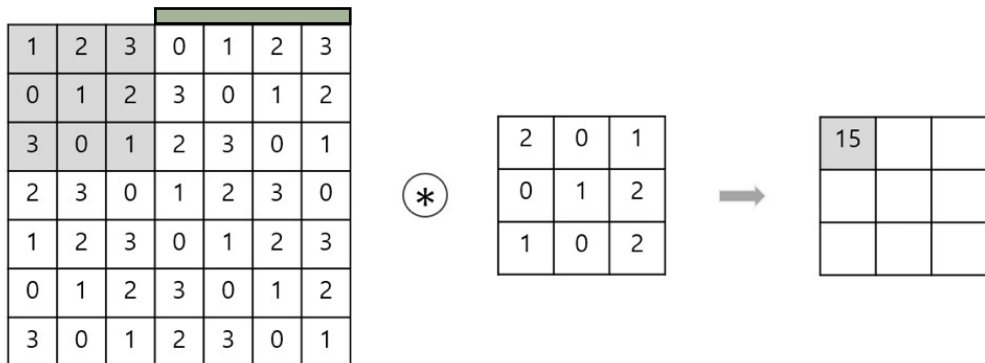
n: 입력 이미지 한변 길이
f: 필터의 크기
p: 패딩의 크기

$$n = n + 2p - f + 1$$

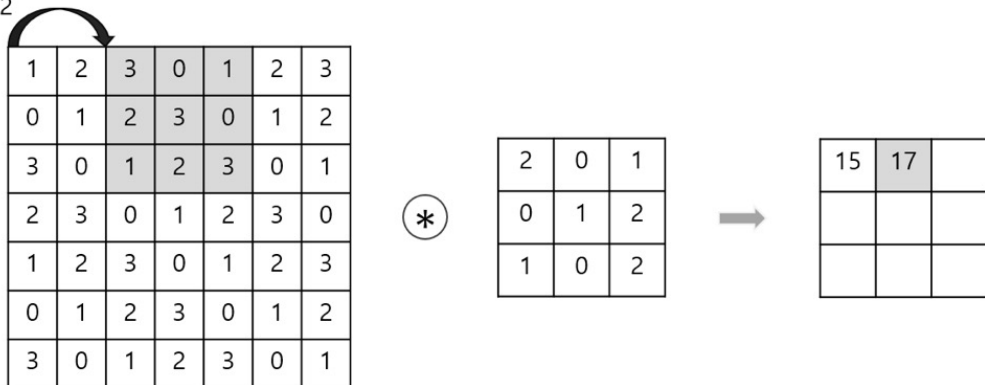
$$p = \frac{f - 1}{2}$$

2.5 CNN - Stride

Stride = 2인 합성곱 연산



stride: 2



$n \times n$ image

$f \times f$ filter

padding p

stride s

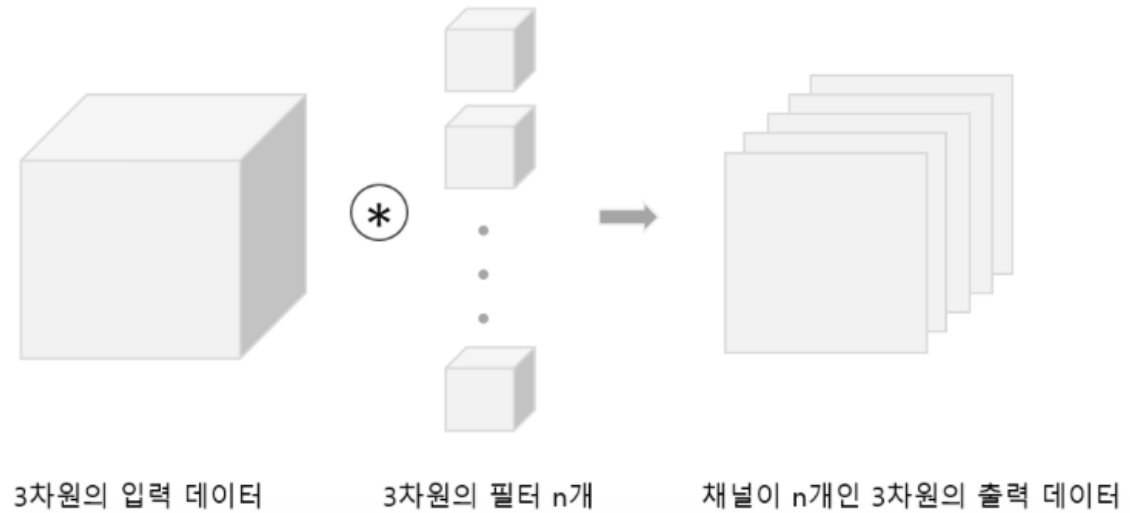
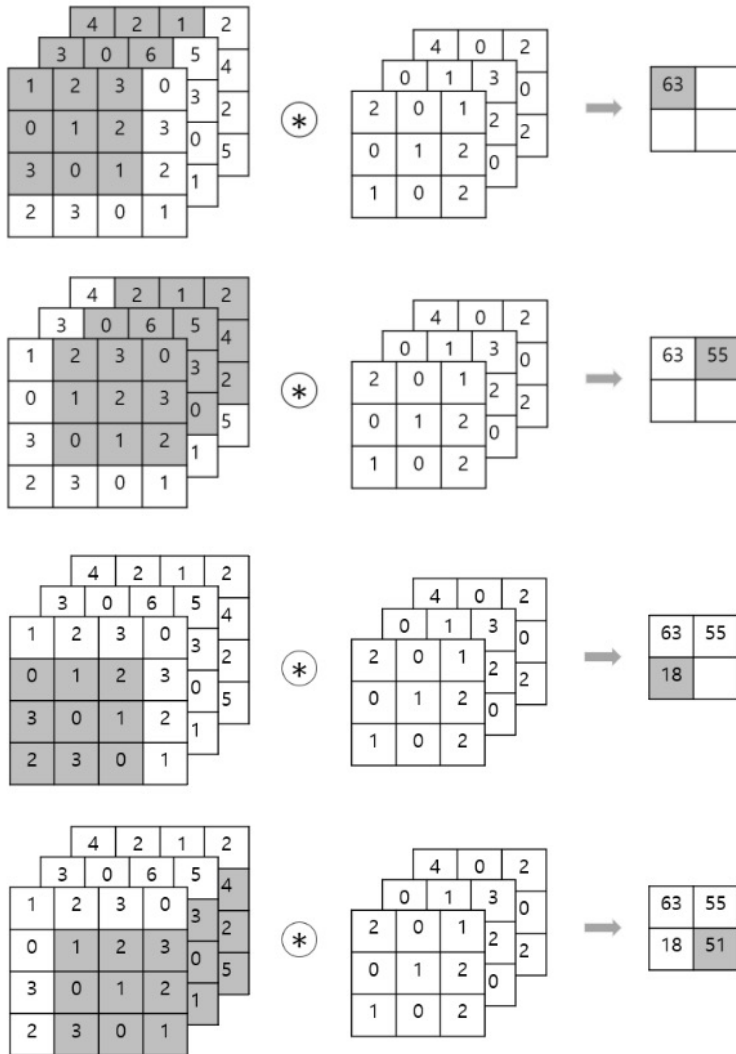


$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

×

$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

3. CNN - Convolutions Over Volumes



$(n \times n \times n_c) * (f \times f \times n_c) = (n - f + 1) \times (n - f + 1) \times n_c$ 형태가 됩니다.

- n : 이미지의 크기
- n_c : 채널의 개수
- f : 필터의 크기
- n_c : 사용된 필터의 개수

4. 퀴즈 리뷰

1. 합성곱 연산에 대한 설명 중 틀린 것은 무엇인가요? *

- ☐ 필터와 이미지의 각 픽셀에 대해 곱하고 모두 더해 결과를 얻습니다.
- ☐ 합성곱 연산은 주로 이미지의 특징을 추출하는 데 사용됩니다.
- ☒ 작은 필터는 큰 특징을, 큰 필터는 작은 특징을 찾는 데 도움이 됩니다.

2. 패딩을 사용하는 이유는 무엇인가요? *

- ☐ 이미지를 더 작게 만들기 위해서
- ☒ 이미지의 가장자리 정보를 유지하기 위해서
- ☐ 연산 속도를 높이기 위해서

THANK YOU

