

딥러닝_영상분석 Day3



Outline

딥러닝_영상분석 Day3



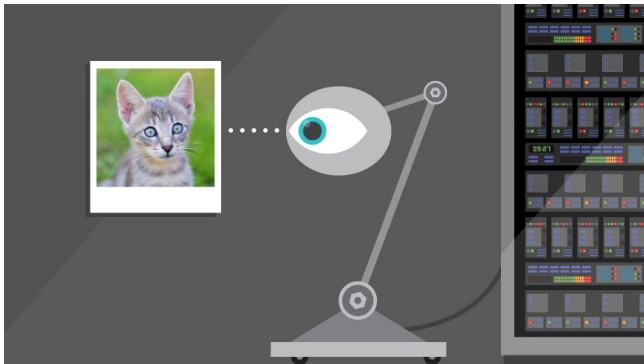
- 1 Intro. To Computer Vision
- 2 Image feature
- 3 Convolutional Neural Network
- 4 Regional CNN

Intro. To Computer Vision

Computer vision

컴퓨터비전은 컴퓨터가 디지털 영상에 대한 높은 수준 이해를 얻을 수 있도록 하는 것으로, 영상의 획득, 처리, 분석을 모두 포함하는 다학제 분야이다.

특히 인간의 시각을 통한 높은 인지 능력을 재현하는데 관심을 가지며, 물리학, 통계학, 학습 이론, 기하학 등을 기반으로 하는 영상 분석 모델을 통해 영상 정보를 분석한다.



Typical tasks in Computer vision

What Computer Vision can do?

컴퓨터비전 분야는 특히 사진 또는 동영상에 잠재된 특정 관심 정보를 자동적으로 추출하기 위한 알고리즘을 다룬다.

이러한 관심 정보는 다음을 포함한다.

- 3D 모델
- 객체 검출과 인식
- 비디오 트래킹
- 영상의 콘텐츠를 기반으로 한 검색과 그룹화
- 영상 복원과 생성

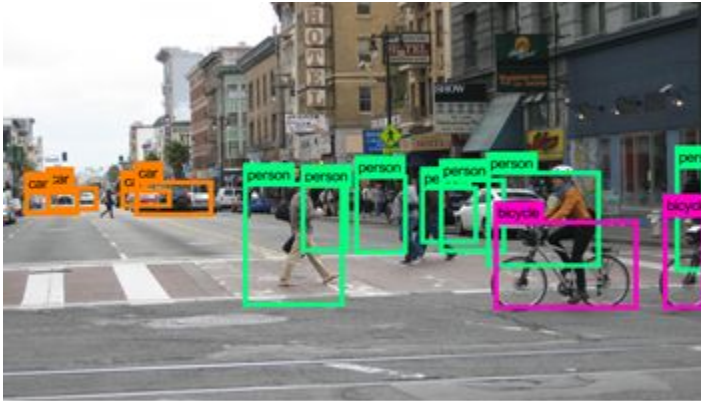
Typical tasks in Computer vision

What Computer Vision can do?

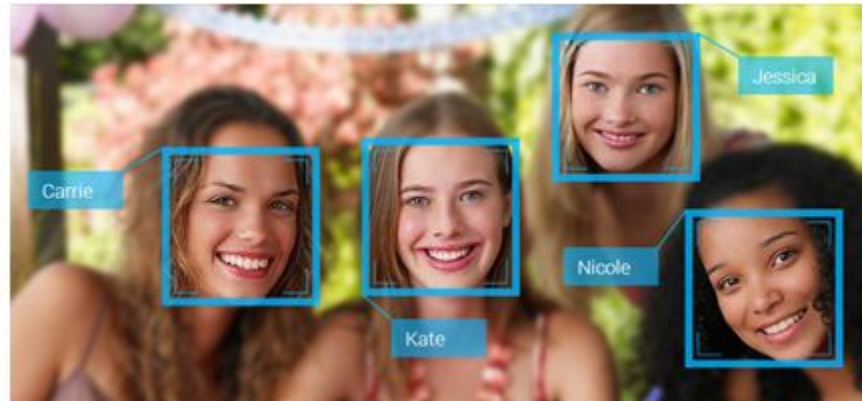


Typical tasks in Computer vision

Applications : Object detection



<Object detection>



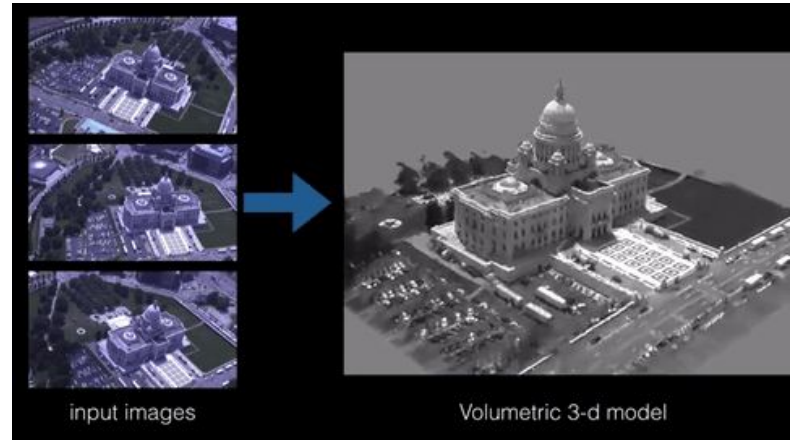
<Face recognition>

Typical tasks in Computer vision

Applications : 3D modeling

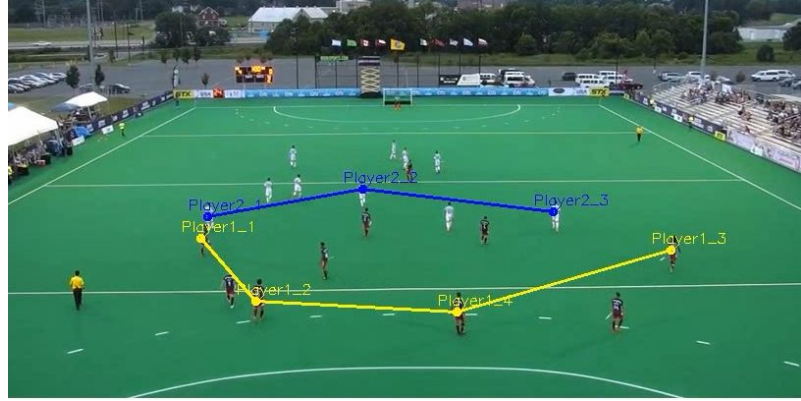


3D Reconstruction



Typical tasks in Computer vision

Applications : Video tracking



Typical tasks in Computer vision

Applications : Image captioning



A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.



A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.

Typical tasks in Computer vision

Applications : Pose estimation



Typical tasks in Computer vision

Applications : Feature matching

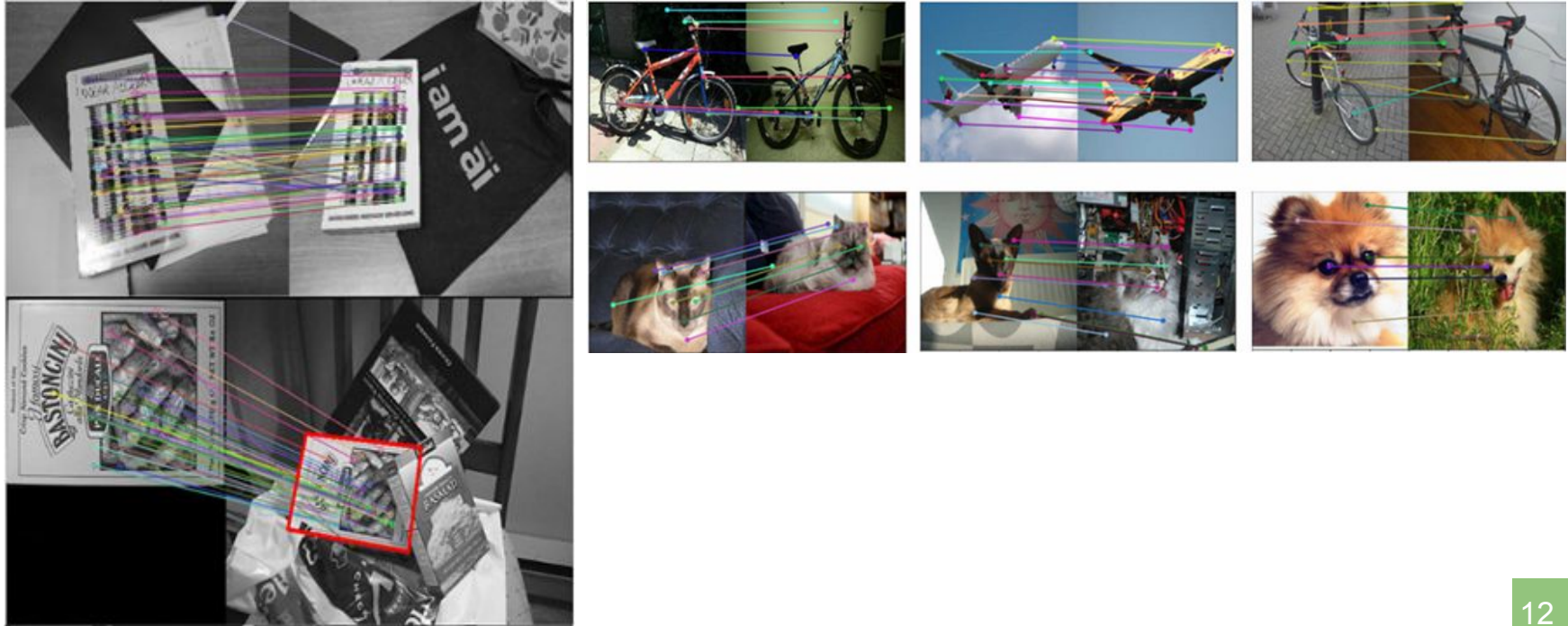
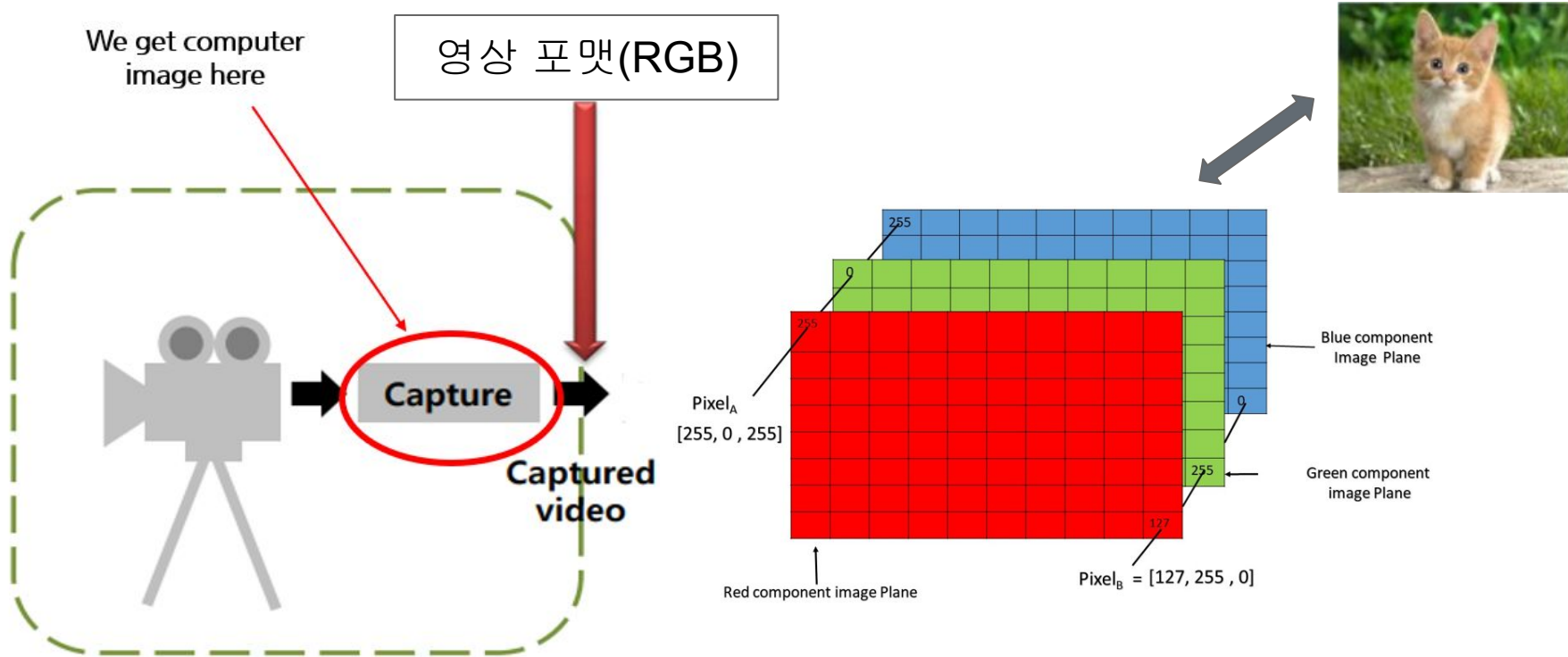
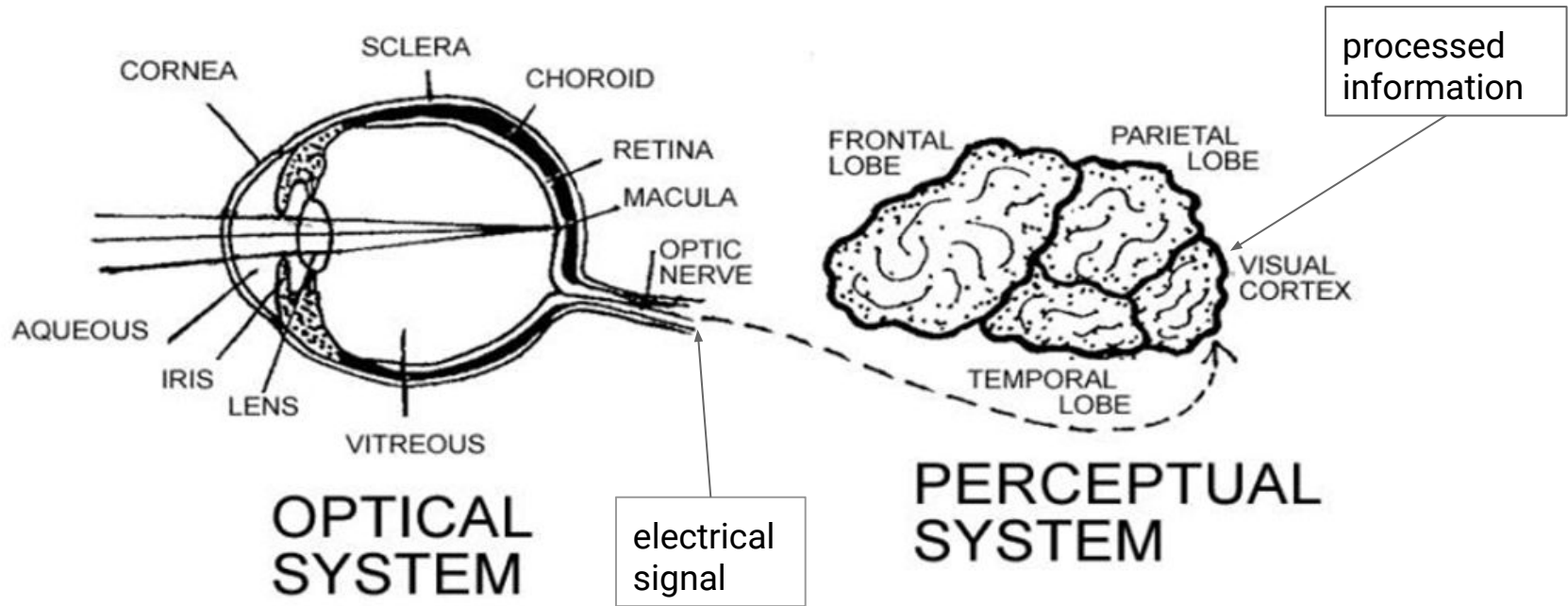


Image feature

Image representation





Why we need image features

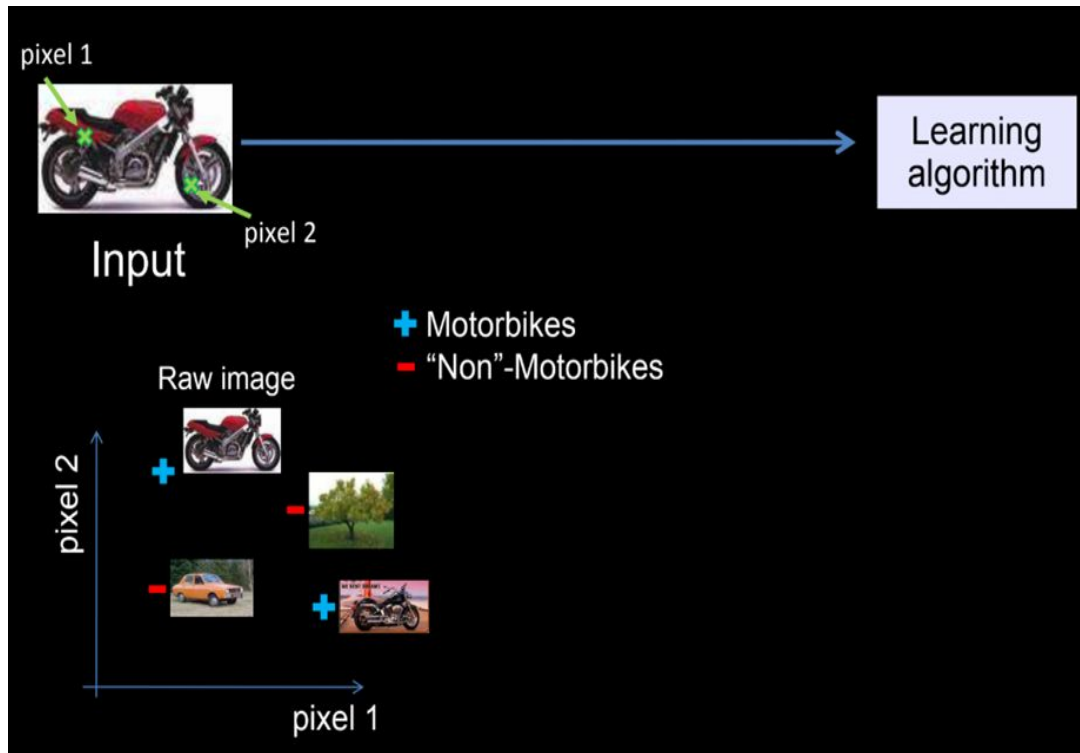
You see this:



But the camera sees this:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

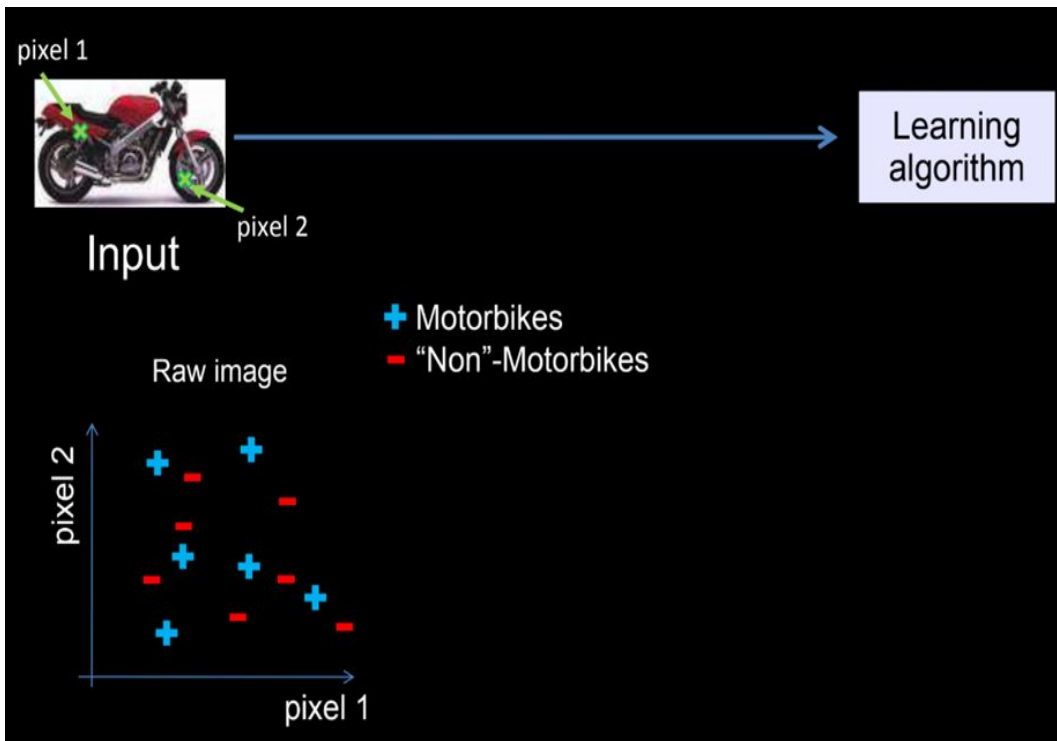
Why we need image features



기본적으로 분류는 데이터가 가지고 있는 값들을 사용한다.

영상의 경우도 그럴까?

Why we need image features

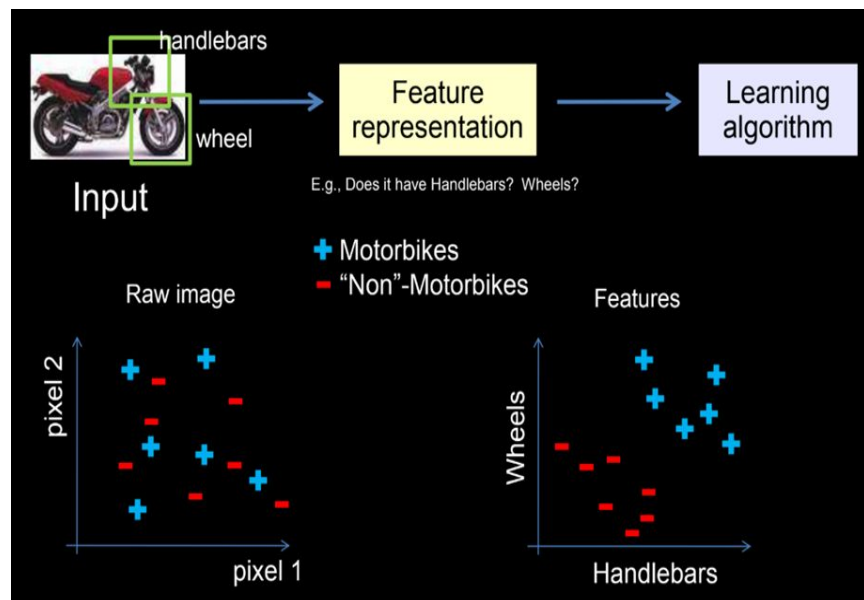
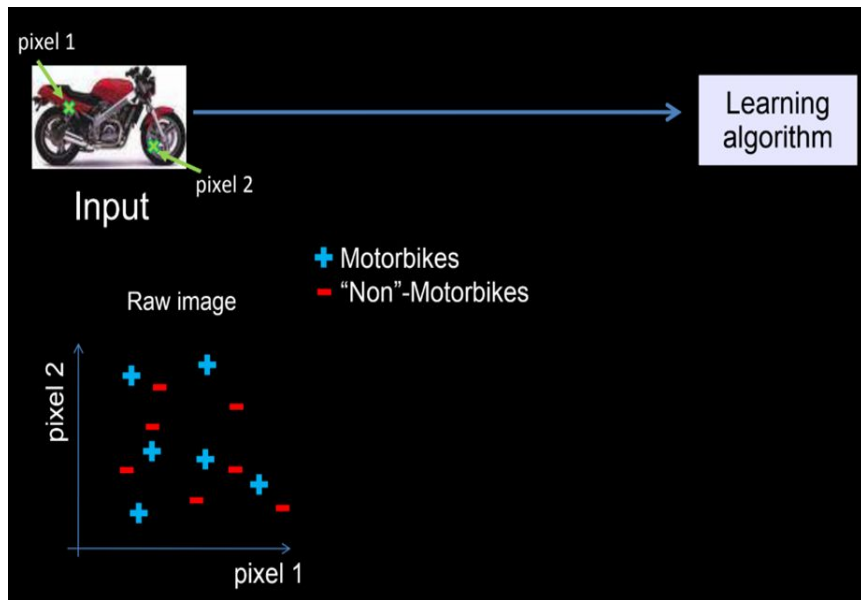


기본적으로 분류는 데이터가 가지고 있는 값들을 사용한다.

영상의 경우도 그럴까?

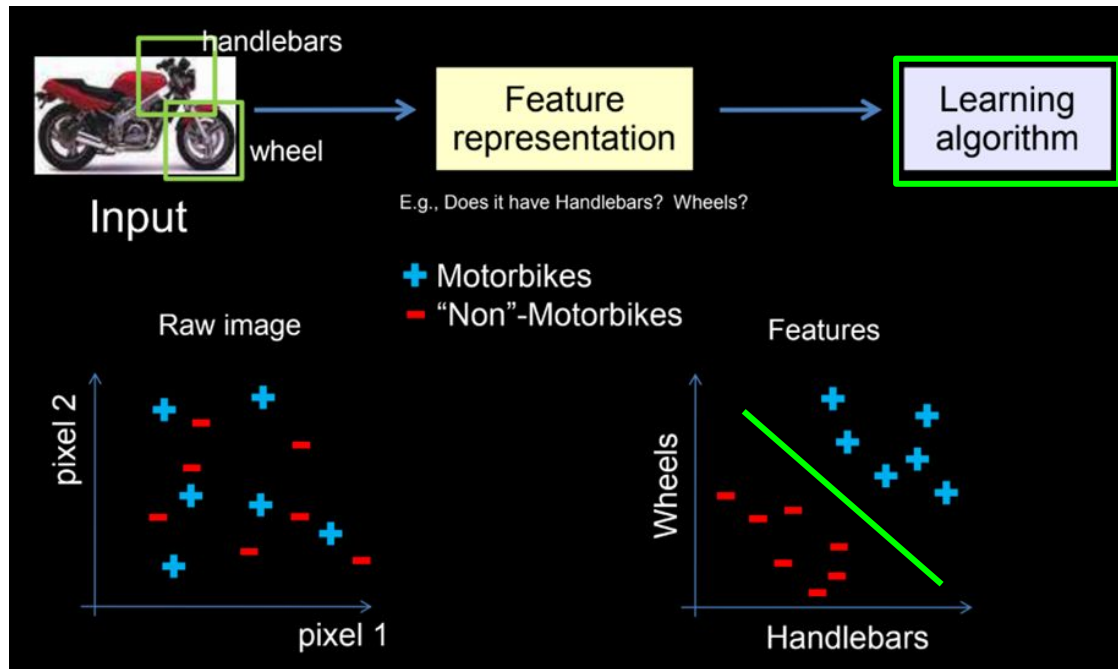
영상의 개별 픽셀값들을 통한 분류는 매우 어렵다!

Why we need image features



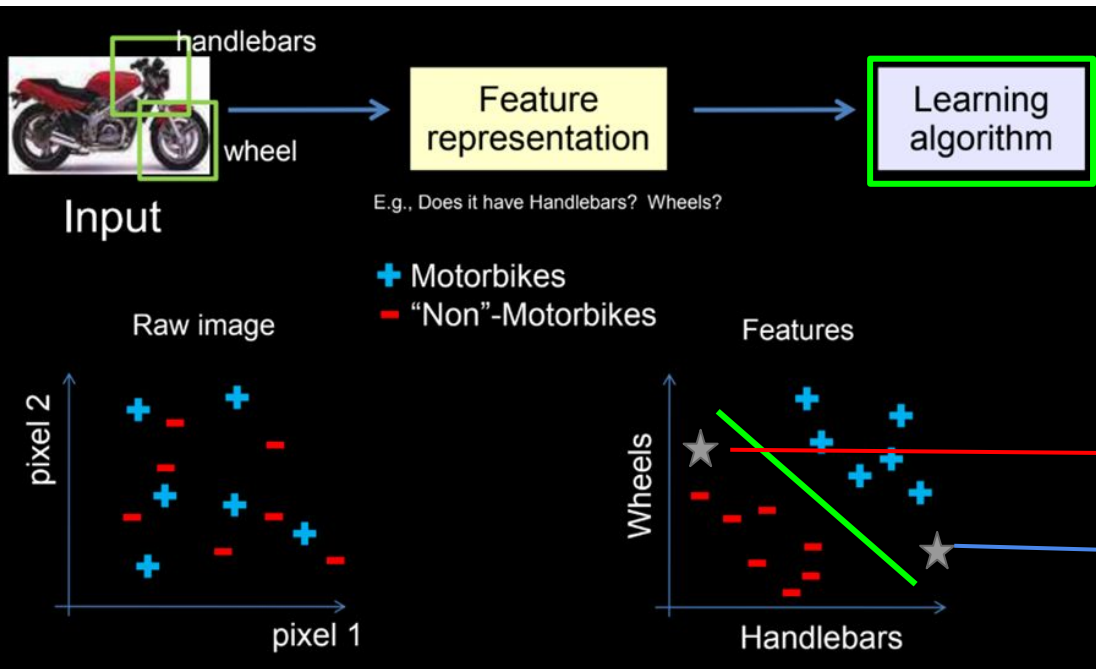
각 클래스의 객체는 공통적으로 가지고 있는 하위 개념들을 통해 구성됨(추상화 개념)
만일 분류 문제를 풀기 위한 하위 개념들을 찾아낼 수 있다면 픽셀 단위로는 해결하기
어려웠던 문제를 풀 수 있다.

Why we need image features



각 영상(data)를 하위 개념들의 조합(new representation)으로 나타낼 수 있다면, 머신러닝 알고리즘을 통해 더 나은 성능을 보이는 분류 모델을 학습할 수 있다.

Why we need image features



이러한 하위 개념을 **Feature**라고 부르며, 새로운 **representation**을 바탕으로 머신러닝 모델을 학습.

이 경우 모터바이크 여부를 판단할 수 있는 분류 모델

predict : It's not a motorbike

predict : It's a motorbike

Convolutional filter(1)

1. **영상의 특징**을 분석하기 위해서는 픽셀 단위가 아닌 픽셀들이 이루는 상관성(패턴)을 분석할 필요가 있다.
2. 영상과 같은 행렬의 지역 상관성을 분석하는데 적합한 **컨볼루션 필터(Convolutional Filter)**가 유용한 분석 도구가 될 수 있음.
3. 컨볼루션 필터는 **가중치 공간 필터(Weighted spatial filter)**로 가중치 값에 따라 다양한 영상처리가 가능함.

Convolutional filter(2)

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



Averaging filter
Blurring

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



Opposite filter
sharpening

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

Horizontal

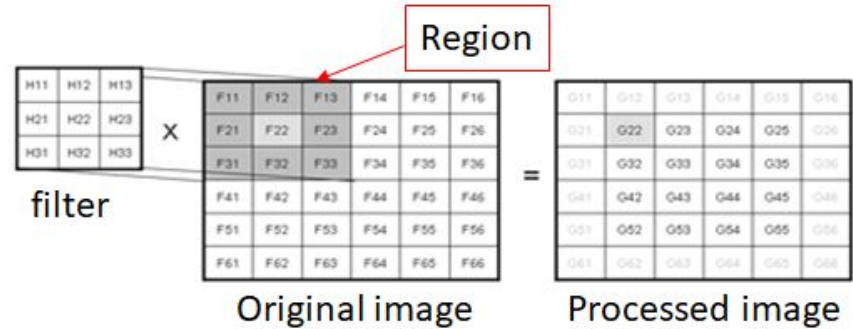
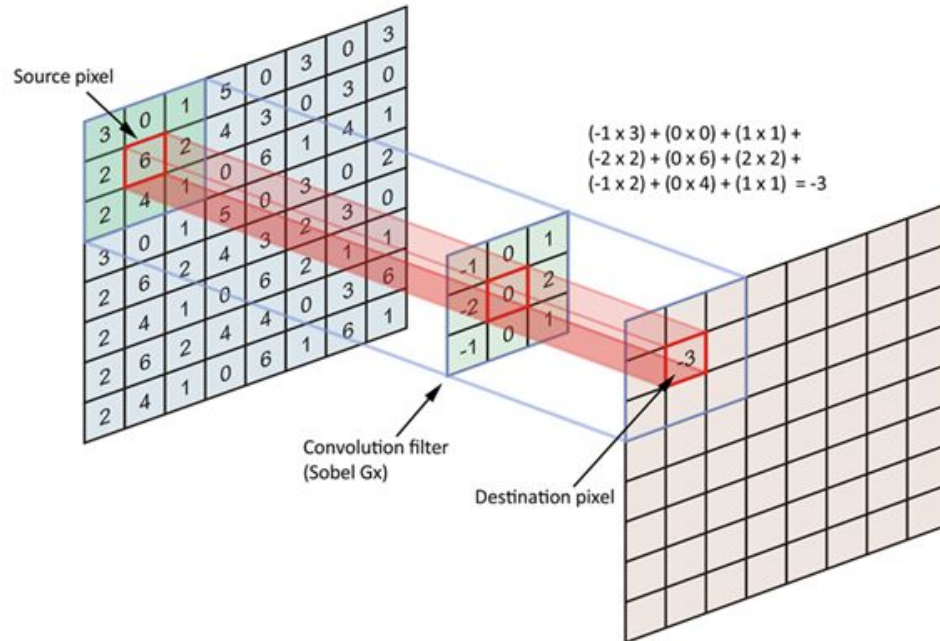
-1	0	1
-2	0	2
-1	0	1

Vertical



Sobel filter
Edge detecting

Calculation of Convolutions(1)



Region moves on a original image and make a processed new image. The movement of Region is called window sliding.

Calculation of Convolutions(2)

3_0	3_1	2_2	1	0
0_2	0_2	1_0	3	1
3_0	1_1	2_2	2	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Example : Sobel filter

Original image



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

G_x



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

G_y



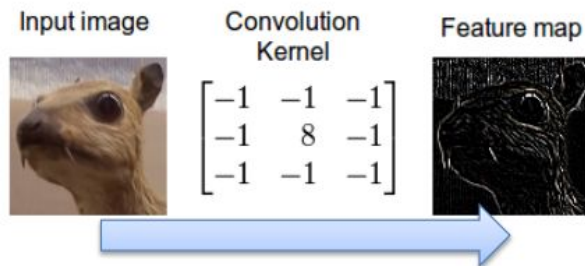
Result of Sobel operator

Convolutional filter(3)

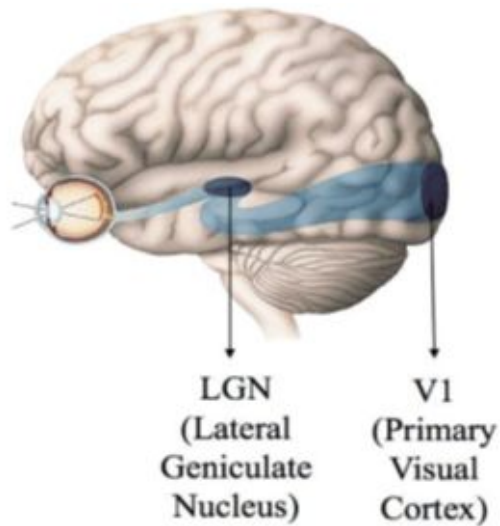
컨볼루션의 다양한 이름

- 영상으로부터 불필요한 정보를 제거할 때 : **Filter**
- 주어진 문제에 적합한 정보를 얻기 위할 때 : **Feature**
- 동일한 사이즈의 영상 패치(patch)를 하나의 값으로 치환할 때 : **Kernel**

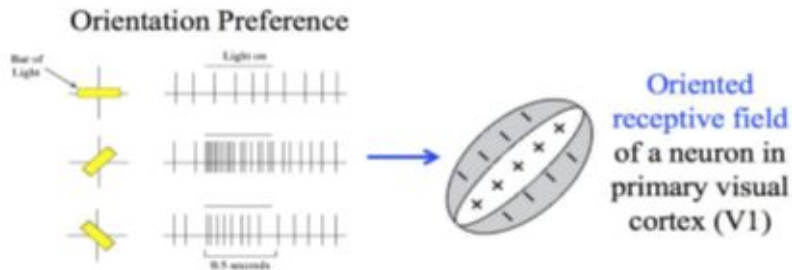
올바른 컨볼루션 필터의 가중치를 선택하는 것은 성공적인 분석에 가장 중요함 => 어떻게 최적의 가중치를 찾을까?

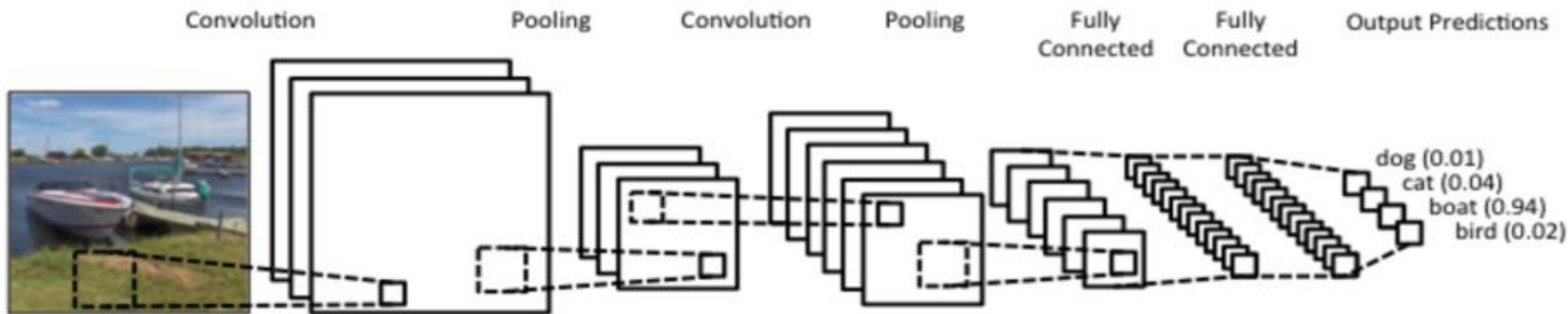


Convolutional Neural Network



Work by Hubel and Wiesel in the 1950s and 1960s showed that cat and monkey visual **cortexes** contain neurons that individually respond to small regions of the **visual field**.
the region of visual space within which visual stimuli affect the firing of a single neuron is known as its **receptive field**

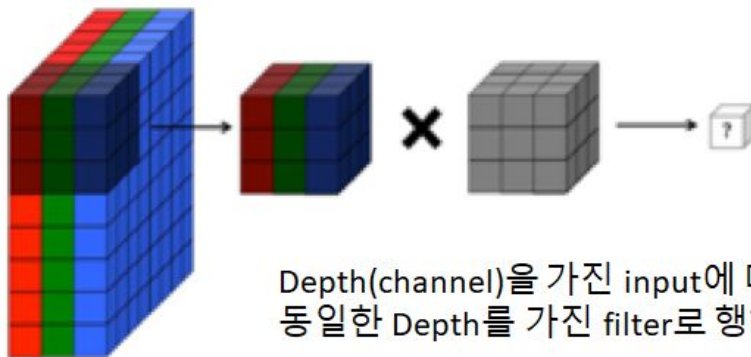




1. CNN(Convolutional Neural Network)는 Convolutional Layer와 Pooling Layer 그리고 FCN(Fully Connected Network)으로 구성되어있음.
2. 각 Convolutional Layer는 다수의 Convolutional Kernel로 구성되어 있고, 마지막에는 Activation function이 있다.
3. Pooling Layer는 feature map을 다운사이징하면서 일종의 정보 요약의 역할을 한다.
4. Fully Connected Network는 영상 특징 정보(Image feature information)를 토대로 비선형적 분류를 가능하게 한다.

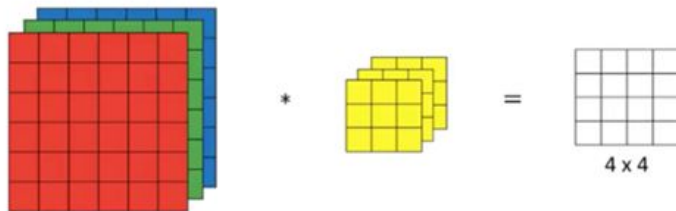
CNN의 구조

1. Convolutional kernel : RGB 영상 연산 방법



3차원 텐서에 대한 컨볼루션 필터는 마찬가지로 3차원 공간필터가 된다(Region이 3차원이 되므로)

Convolutions on RGB image

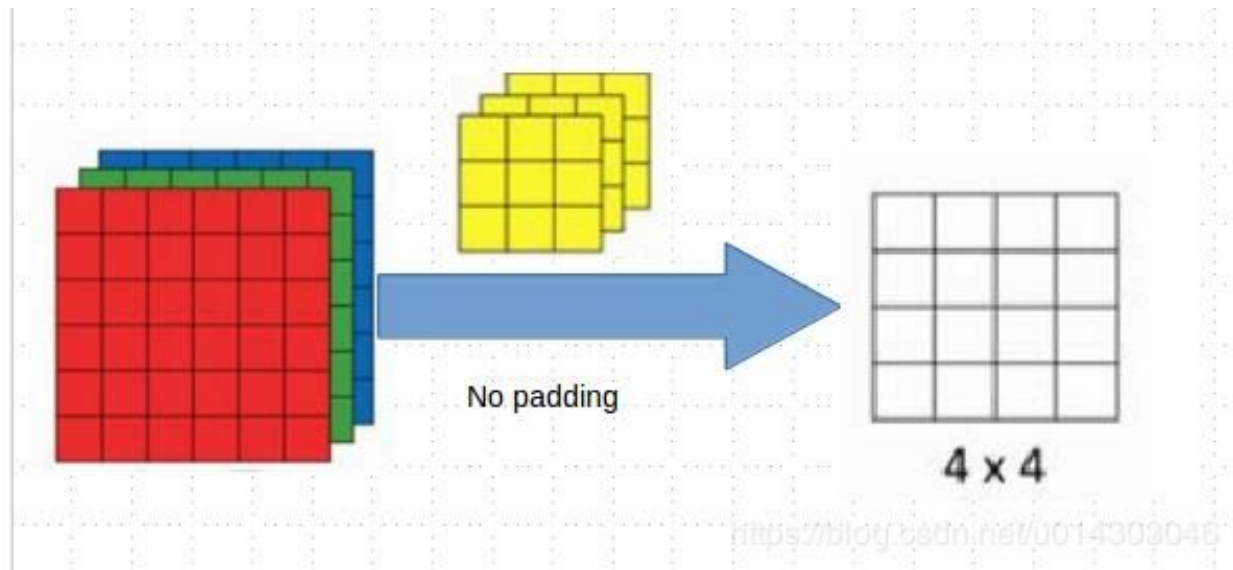


Input volume이 $H \times W \times N$ 차원이라면?
마찬가지로 $K \times K \times N$ 의 Kernel을 사용해준다.

(H, W : 가로, 세로. K : kernel의 너비 및 높이)

CNN의 구조

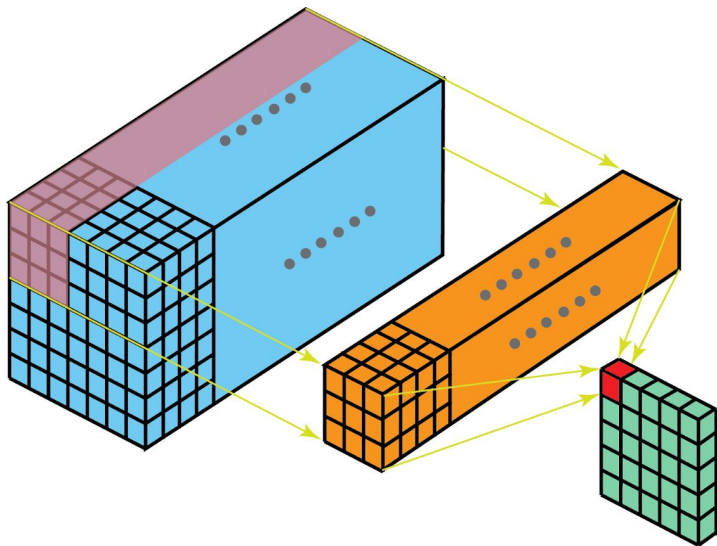
1. Convolutional kernel : RGB 영상 연산 방법



Padding이 있다면 feature map은 입력 텐서와 같은 너비와 깊이를 갖는다(깊이는 동일하게 1)

CNN의 구조

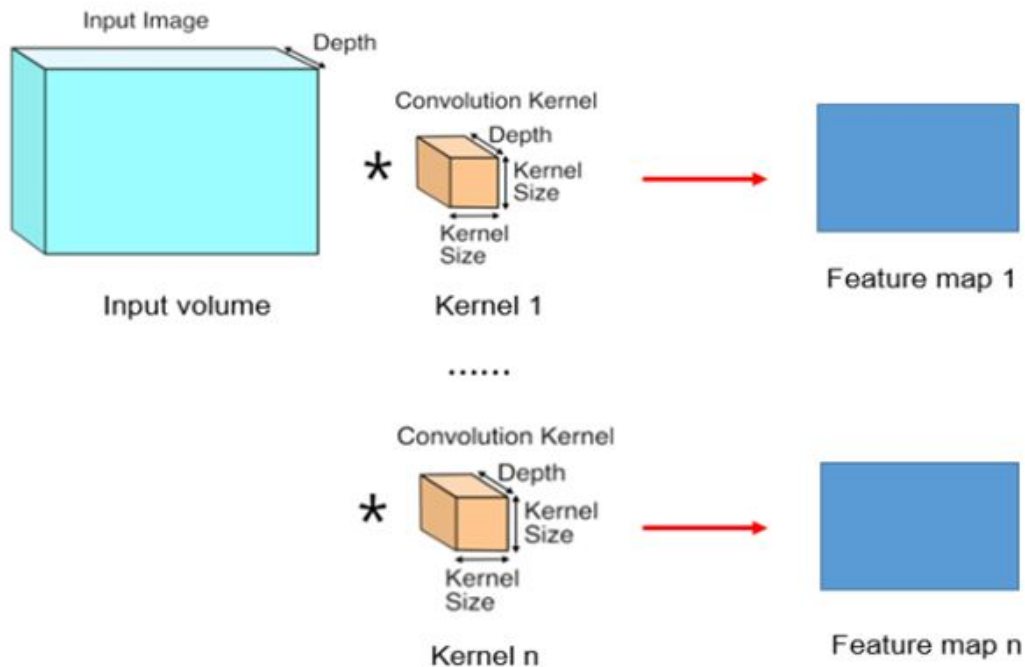
1. Convolutional Kernel : Depth가 N인 텐서에 대한 연산 방법



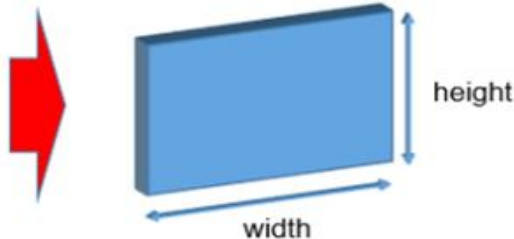
첫 번째 레이어 이후의 N (이전 레이어의 필터 갯수)개 만큼의 Depth를 가진 특징 정보는 동일한 Depth의 필터로 컨볼루션 연산을 수행한다.

CNN의 구조

2. Convolutional Layers : 다중 필터의 feature map

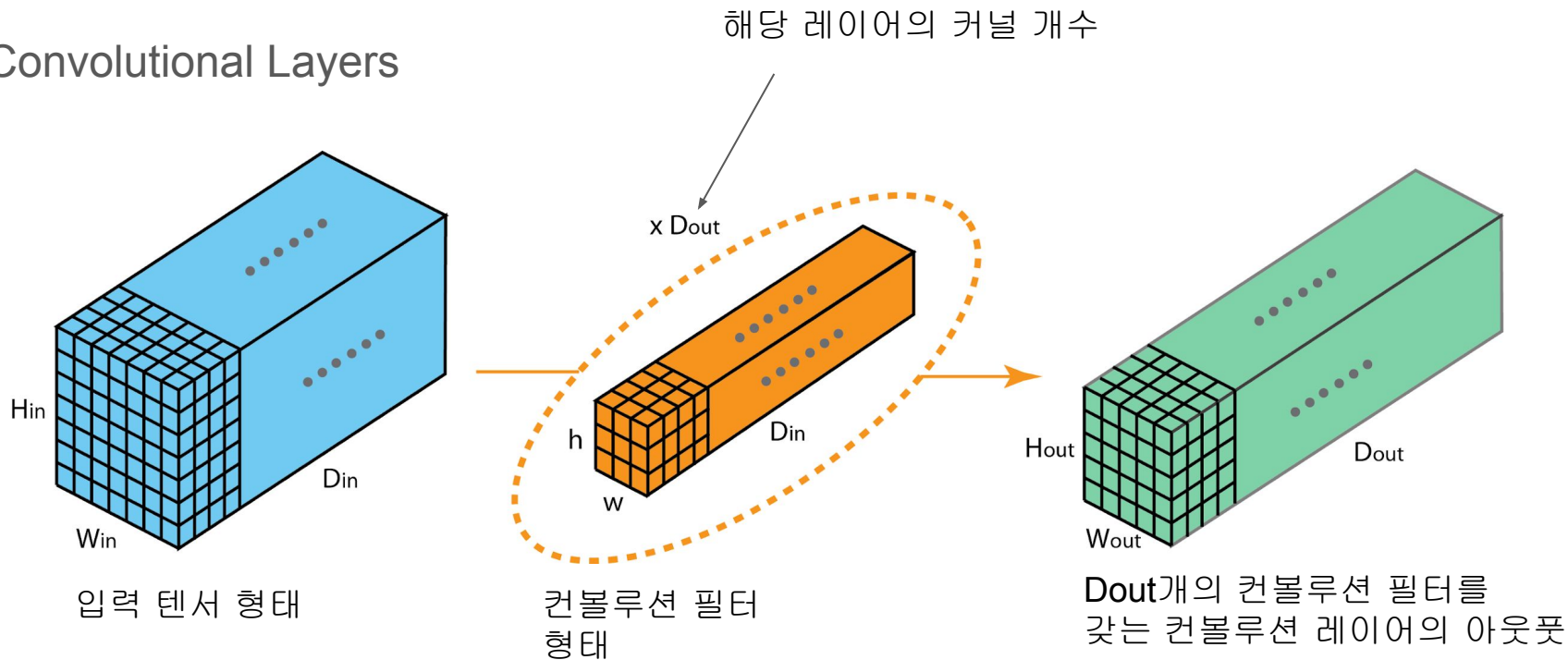


Convolutional layer의 각 Kernel 별로 생성된 feature map이 쌓여 하나의 3차원 텐서를 만든다. 텐서의 shape는 Height x Width x n



CNN의 구조

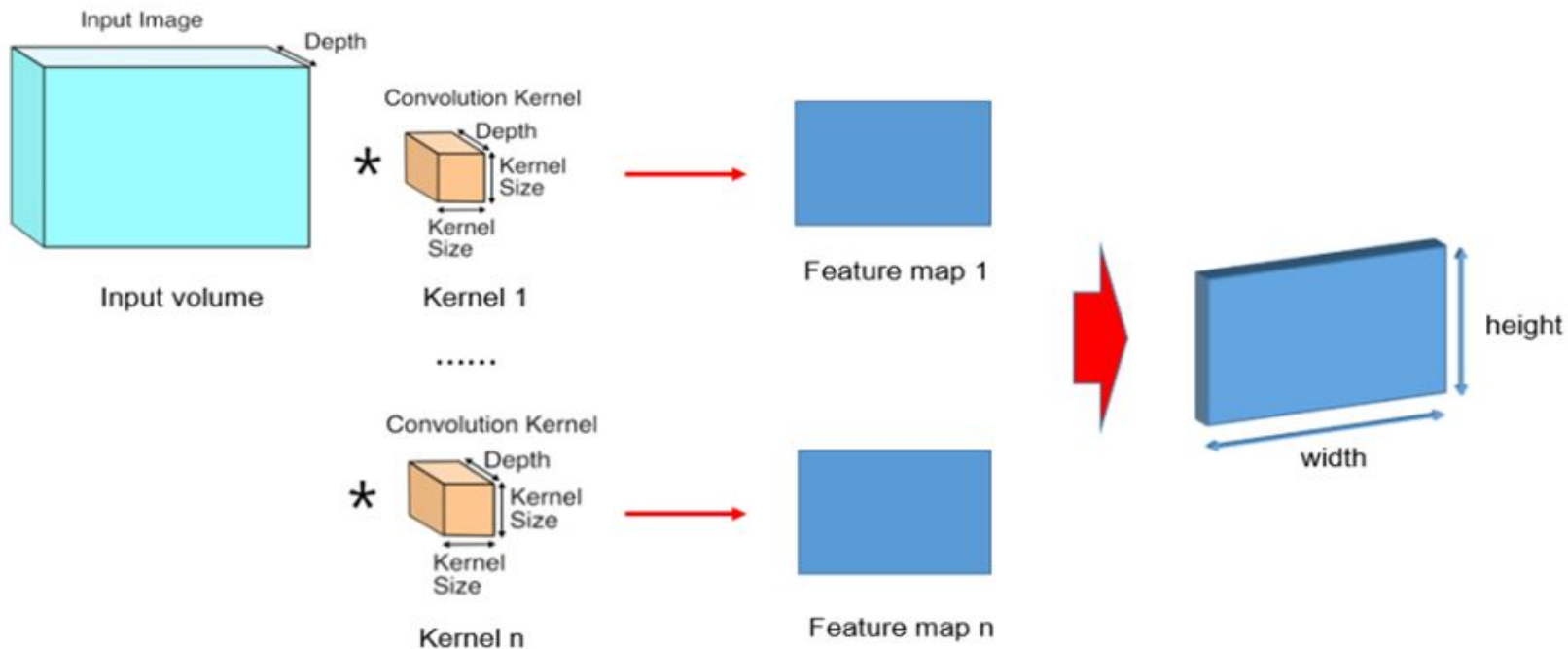
2. Convolutional Layers



첫 번째 레이어 이후의 N (이전 레이어의 필터 갯수)개 만큼의 Depth를 가진 특징 정보는 동일한 Depth의 필터로 컨볼루션 연산을 수행한다.

CNN의 구조

2. Convolutional Layers



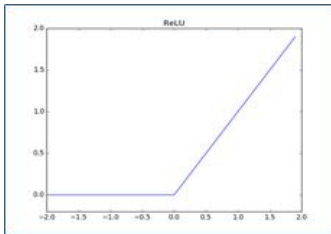
CNN의 구조

2. Activation functions in Convolutional Layer

ReLU Layer

Filter 1 Feature Map

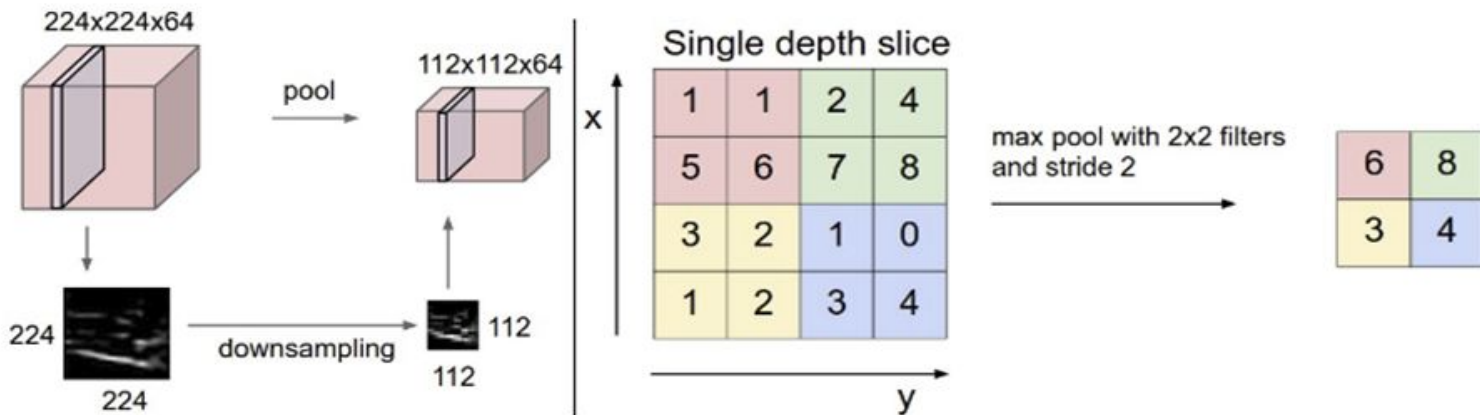
9	3	5	-8
-6	2	-3	1
1	3	4	1
3	-4	5	1



9	3	5	0
0	2	0	1
1	3	4	1
3	0	5	1

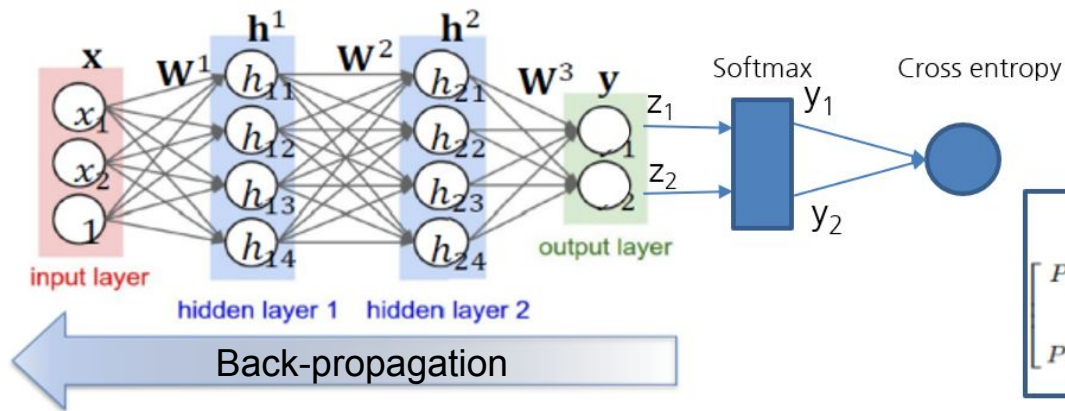
CNN의 구조

3. Pooling Layer



CNN의 구조

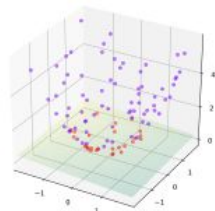
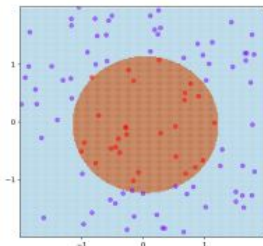
4. FCN



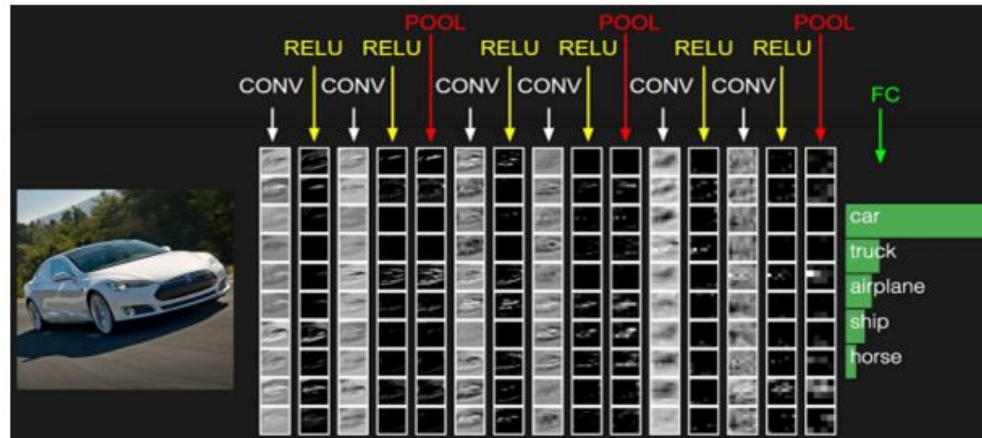
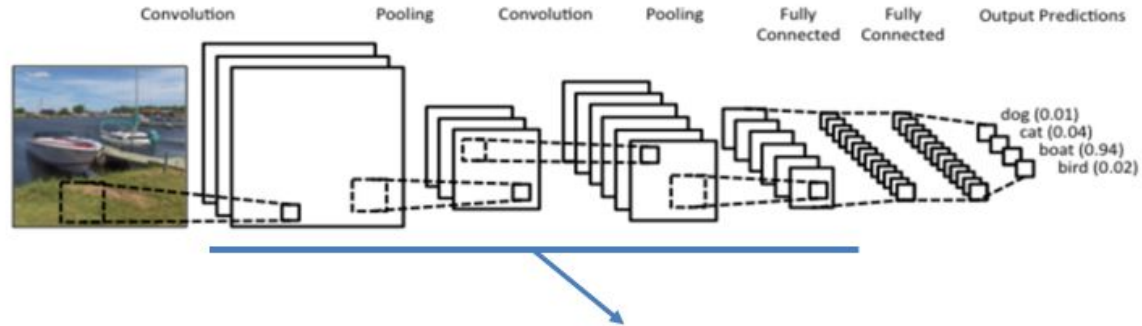
Softmax

$$\begin{bmatrix} P(t=1|\mathbf{z}) \\ \vdots \\ P(t=C|\mathbf{z}) \end{bmatrix} = \begin{bmatrix} \varsigma(\mathbf{z})_1 \\ \vdots \\ \varsigma(\mathbf{z})_C \end{bmatrix} = \frac{1}{\sum_{d=1}^C e^{z_d}} \begin{bmatrix} e^{z_1} \\ \vdots \\ e^{z_C} \end{bmatrix}$$

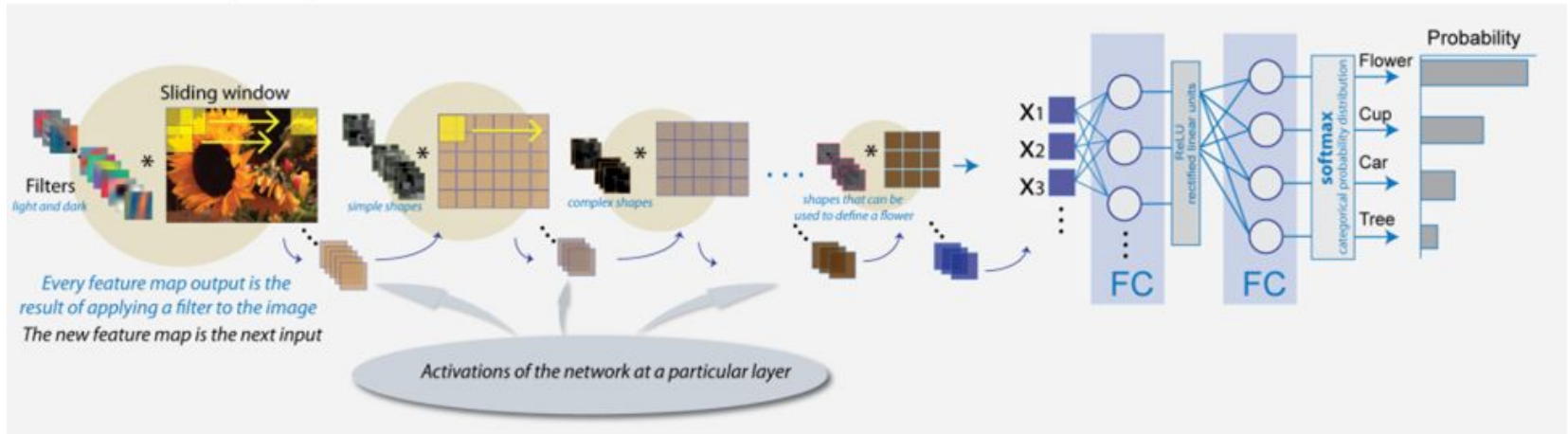
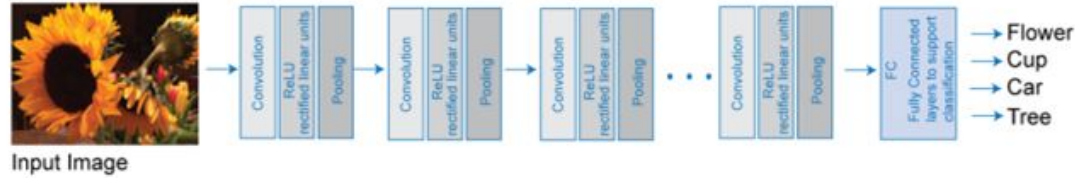
Cross entropy : $-\sum t_i * \log y_i$



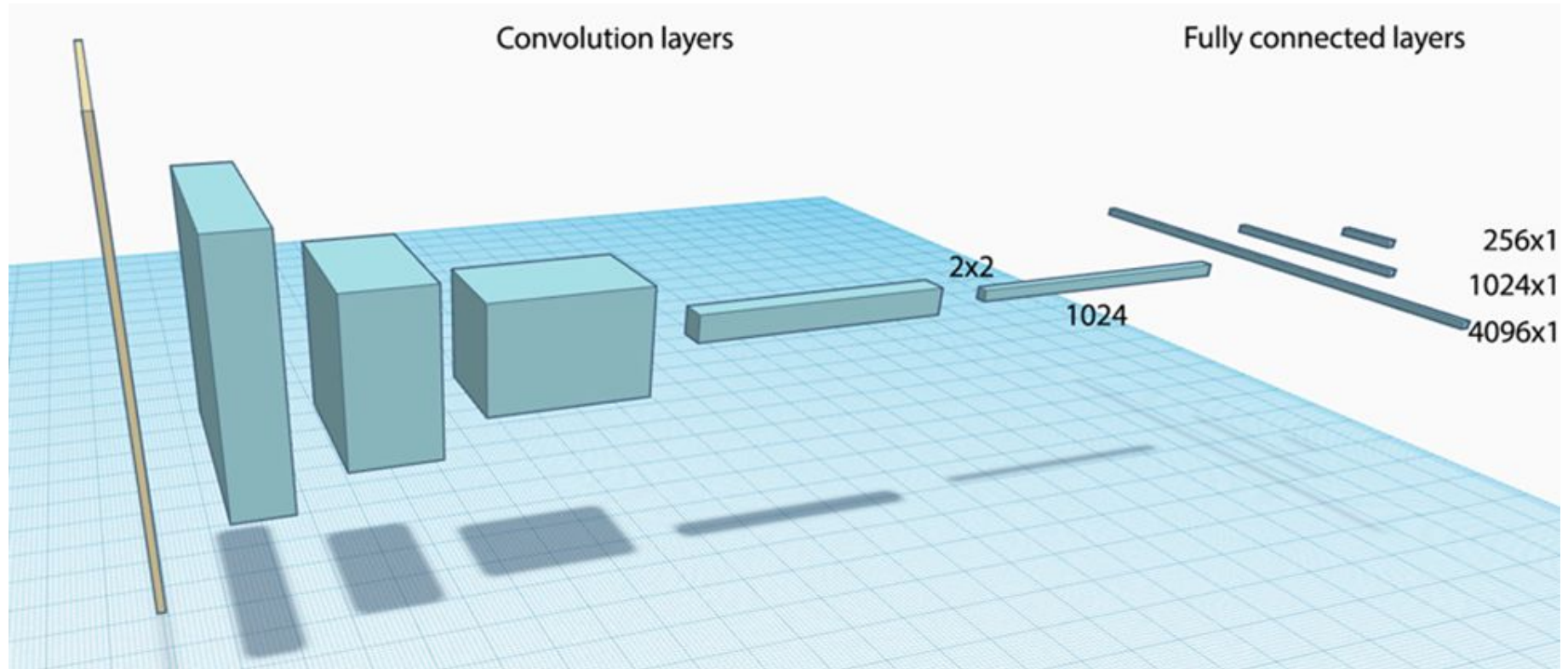
Overview of CNN(1)



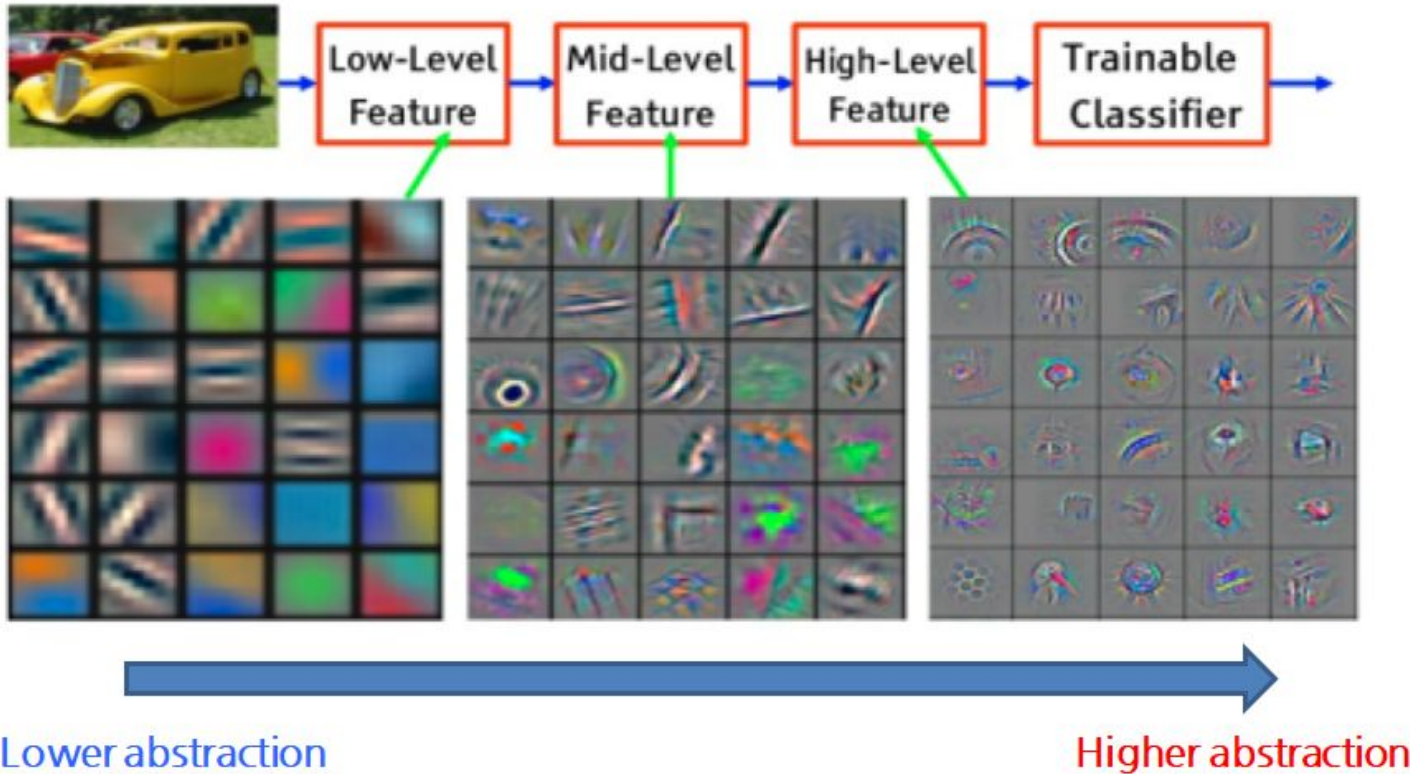
Overview of CNN(1)



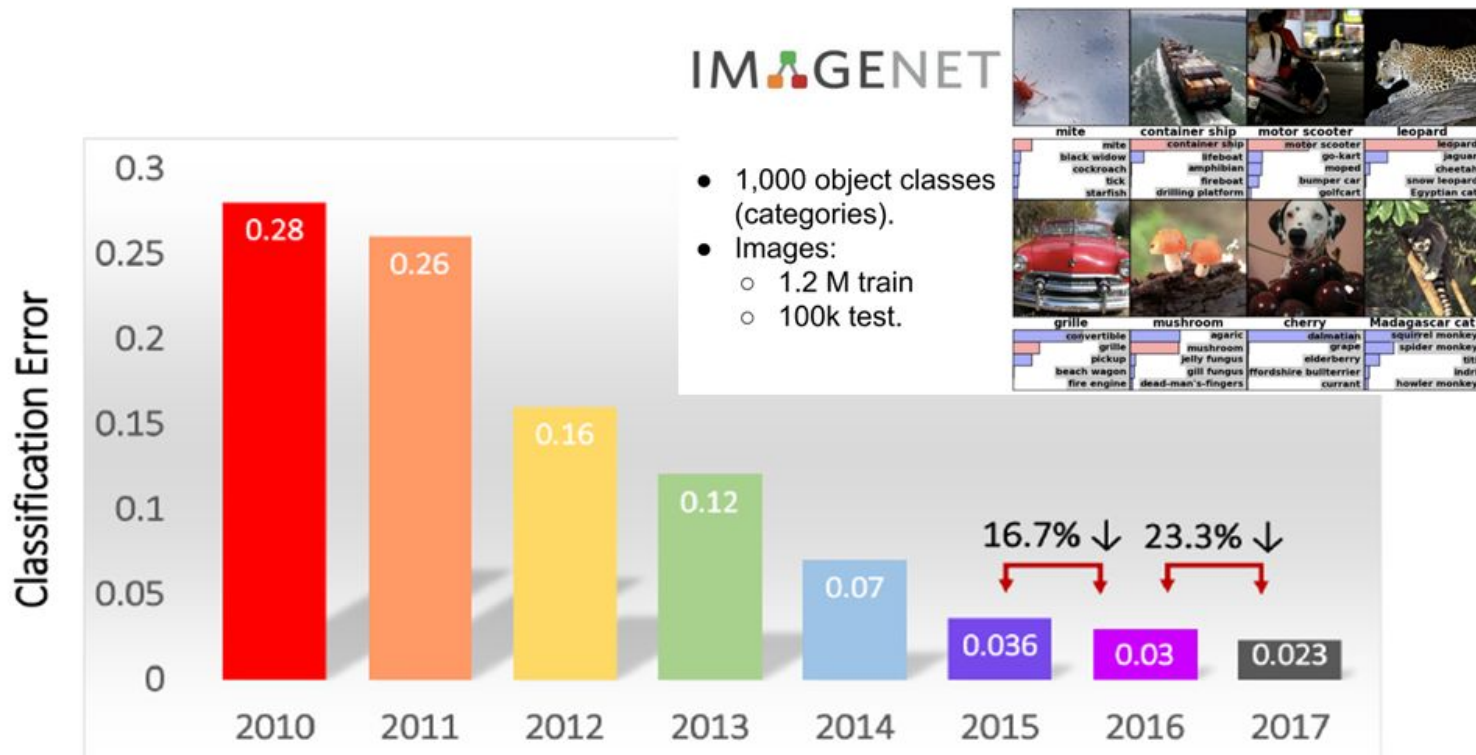
Overview of CNN(3)



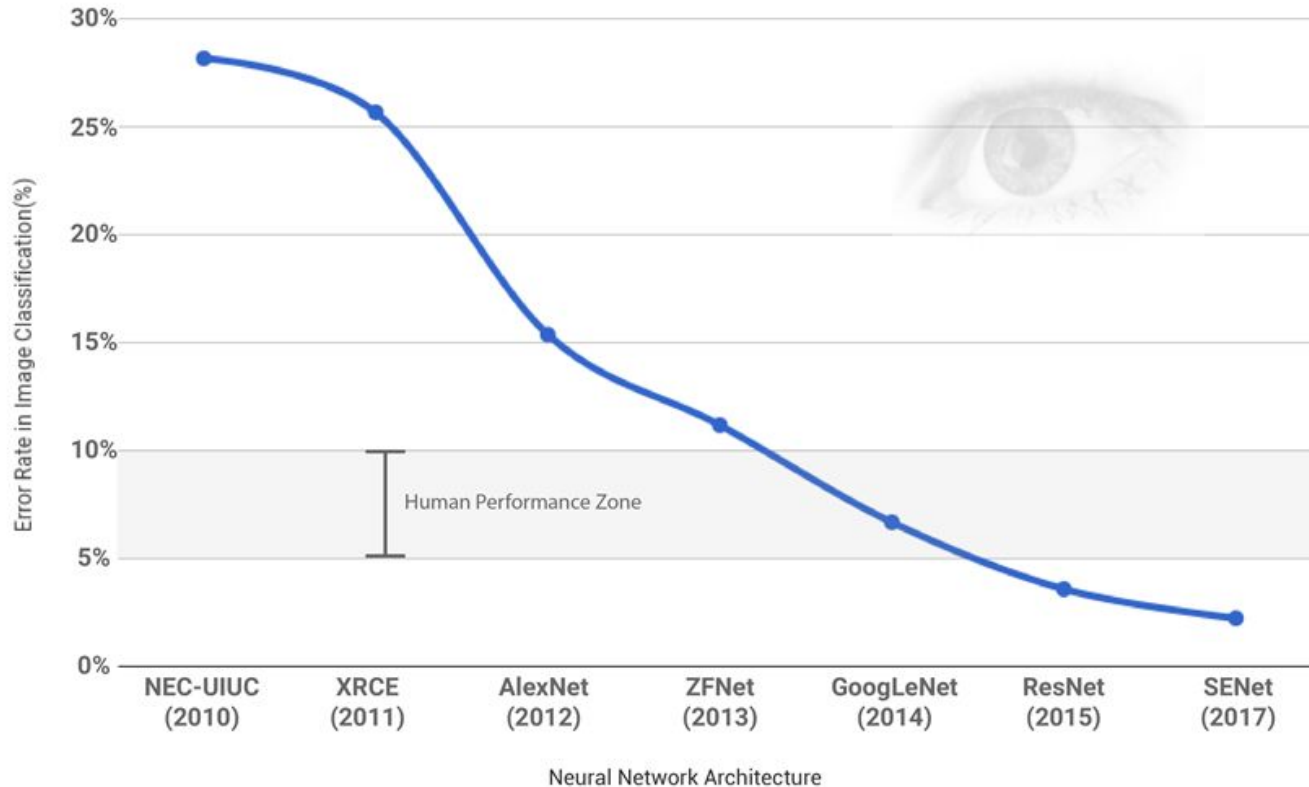
Hierarchical Features of CNN



Imagenet Classification Competition

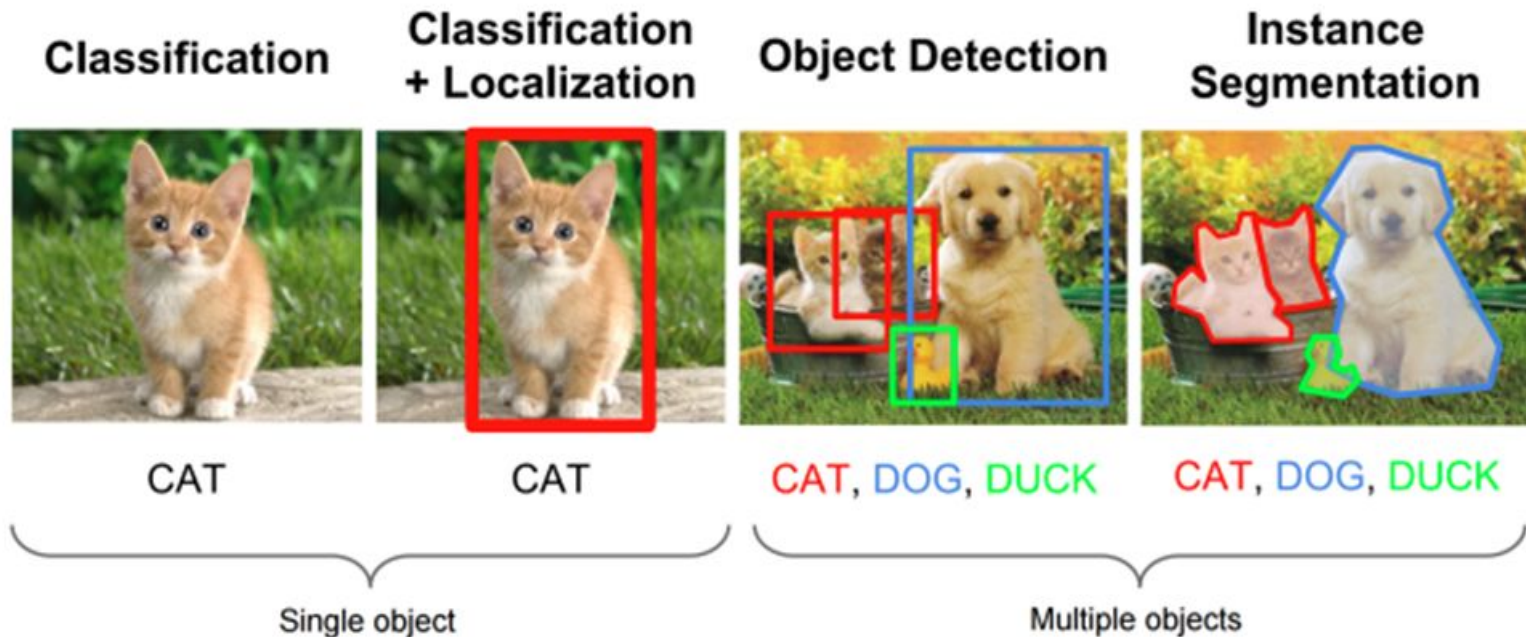


Imagenet Classification Competition

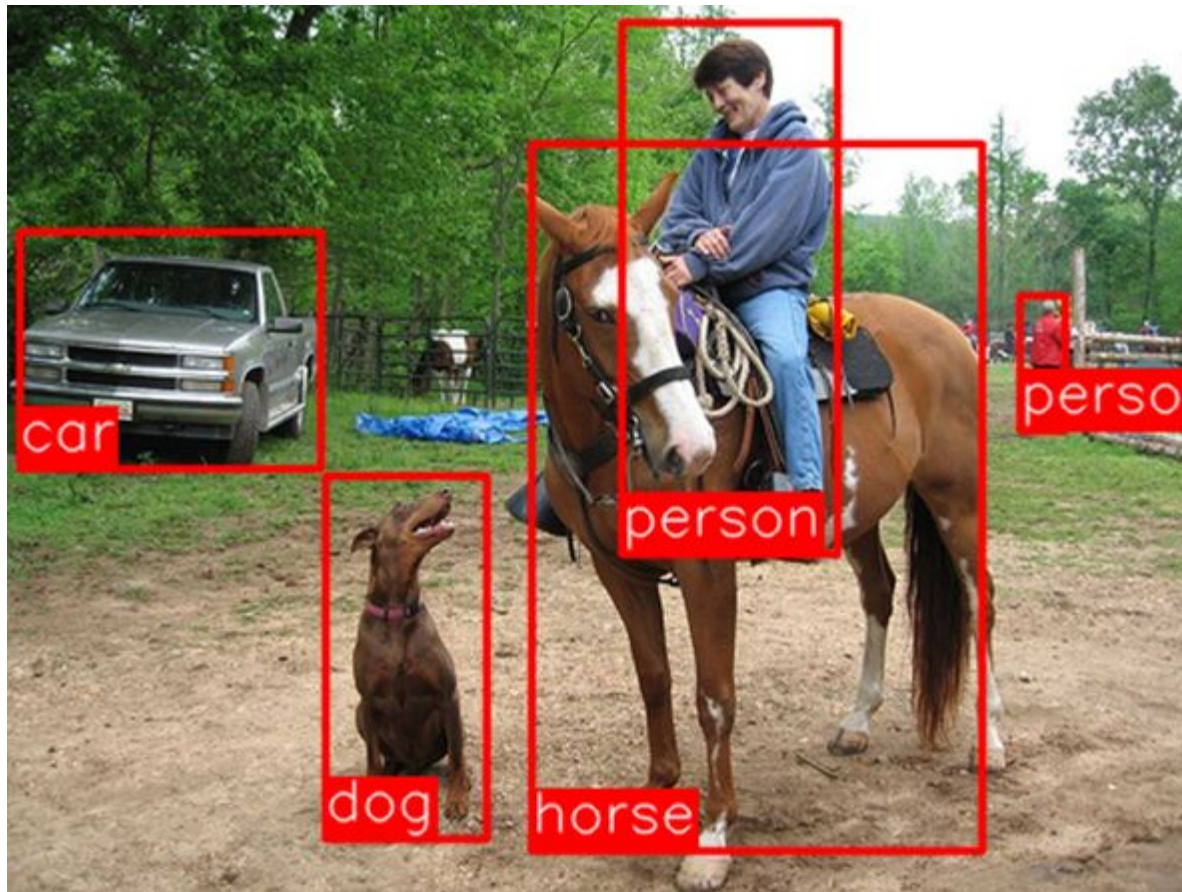


Regional CNN

객체검출(Object detection)

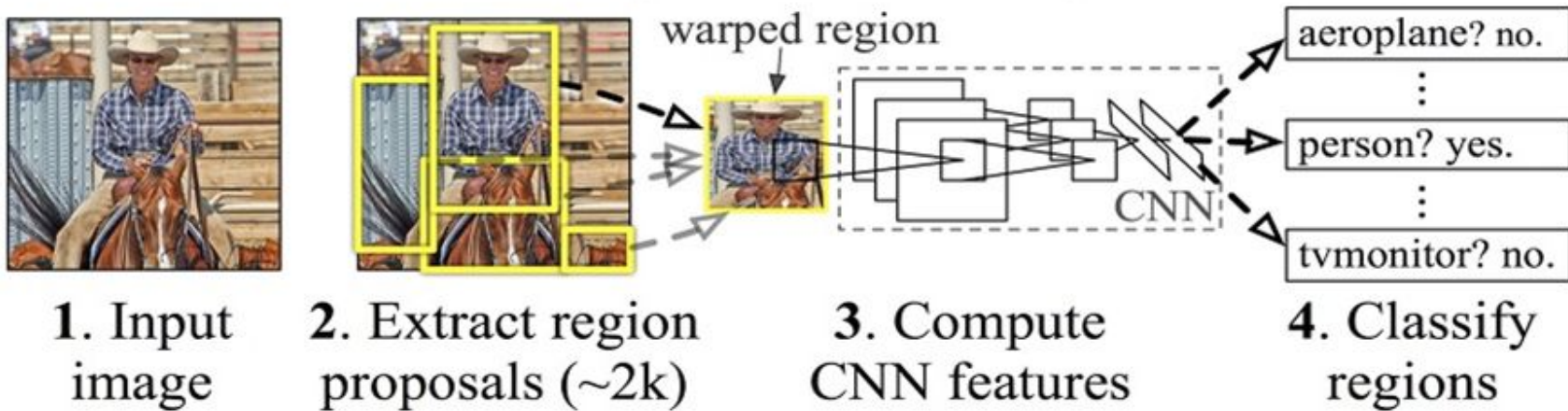


객체검출(Object detection)



RCNN

R-CNN: *Regions with CNN features*



Step 2. CNN을 분류 문제에 대해 사전 지도학습(Domain specific classification learning)

Step 1. 이미지 입력

Step 2. Selective search 알고리즘을 통해 region proposal 수행

Step 3. 각 region을 동일한 사이즈로 warping하여 CNN에 입력하여 특징 검출

Step 4. 분류기를 통해 각 region을 분류

Step 5. 마지막으로 Non-maximum suppression 알고리즘을 통해 지역을 확정함.

RCNN - Selective search

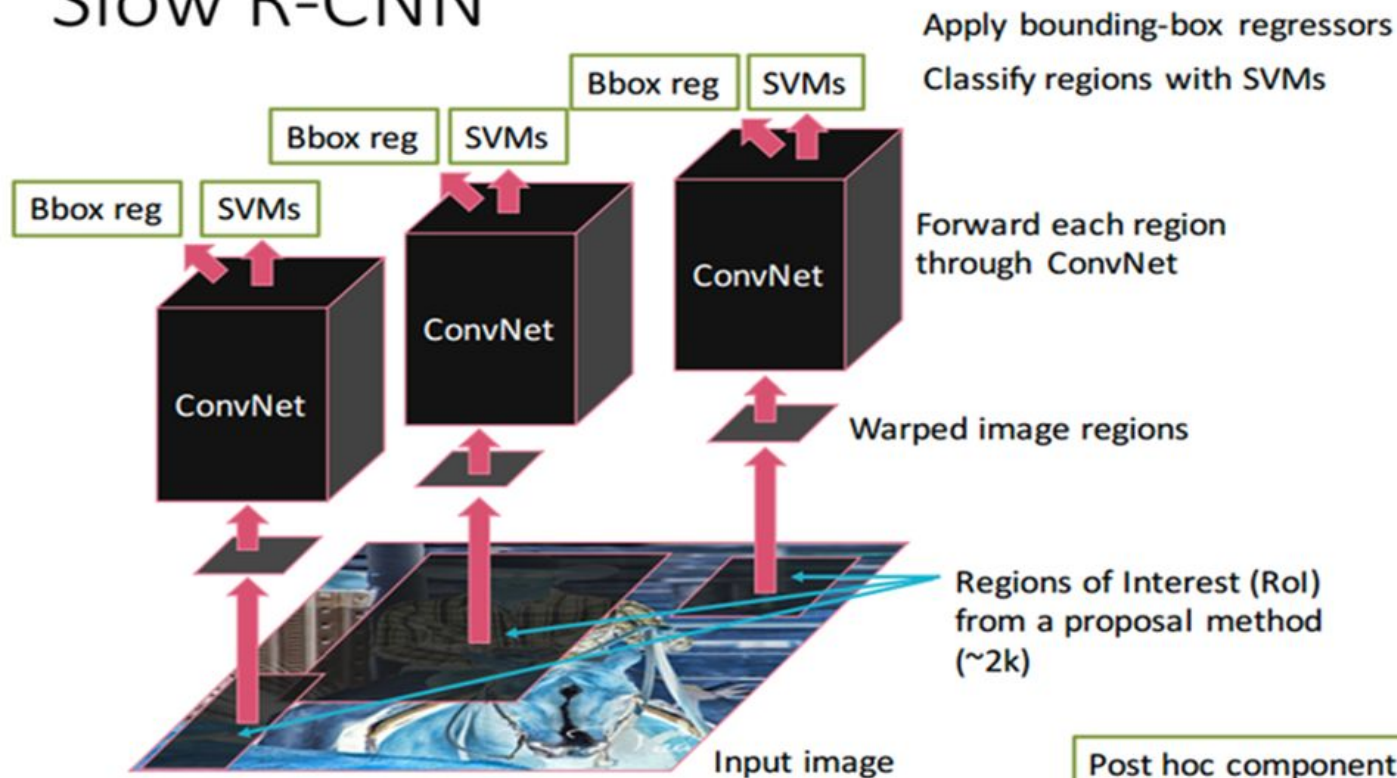


Figure 2: Two examples of our selective search showing the necessity of different scales. On the left we find many objects at different scales. On the right we necessarily find the objects at different scales as the girl is contained by the tv.

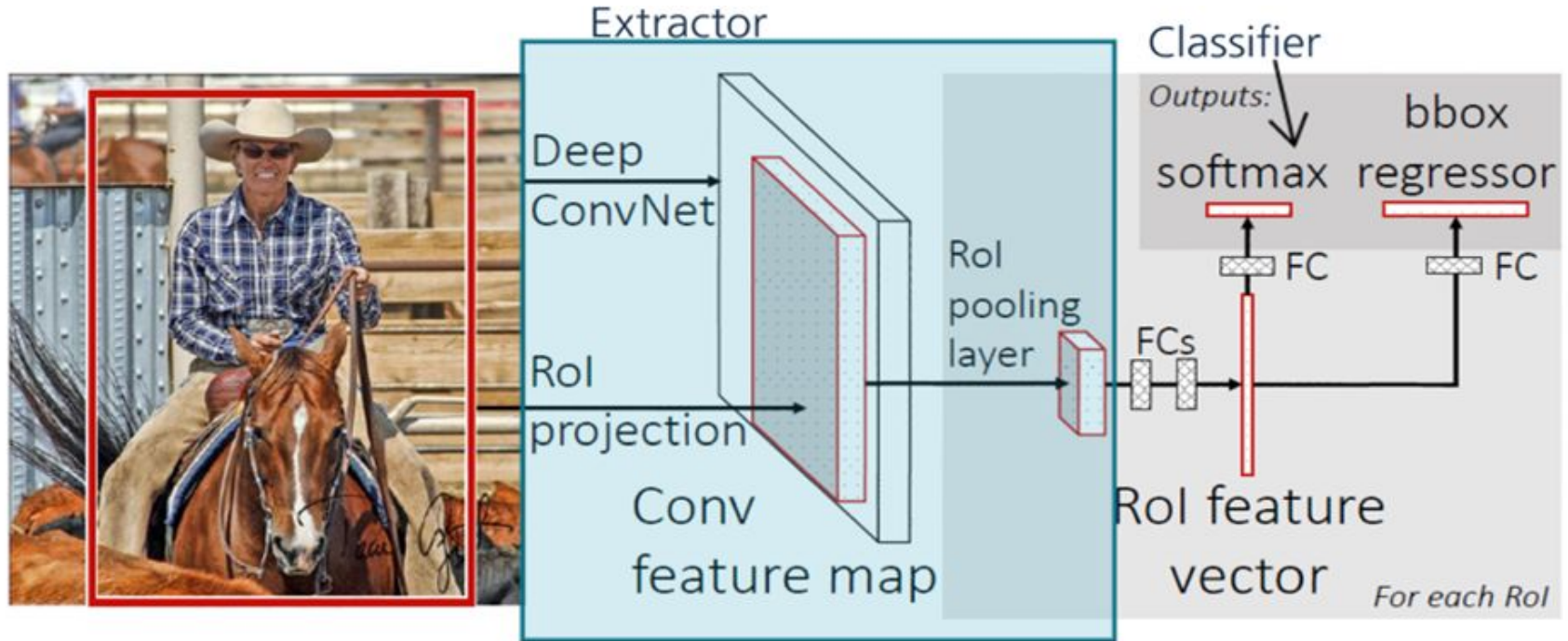
- Group regions with some rules(intensity, color..)
- In experience, S.S resulted in an average of 2403 region proposals per image with a 91.6% recall of all ground-truth bounding boxes(0.5 IoU threshold)

RCNN

Slow R-CNN

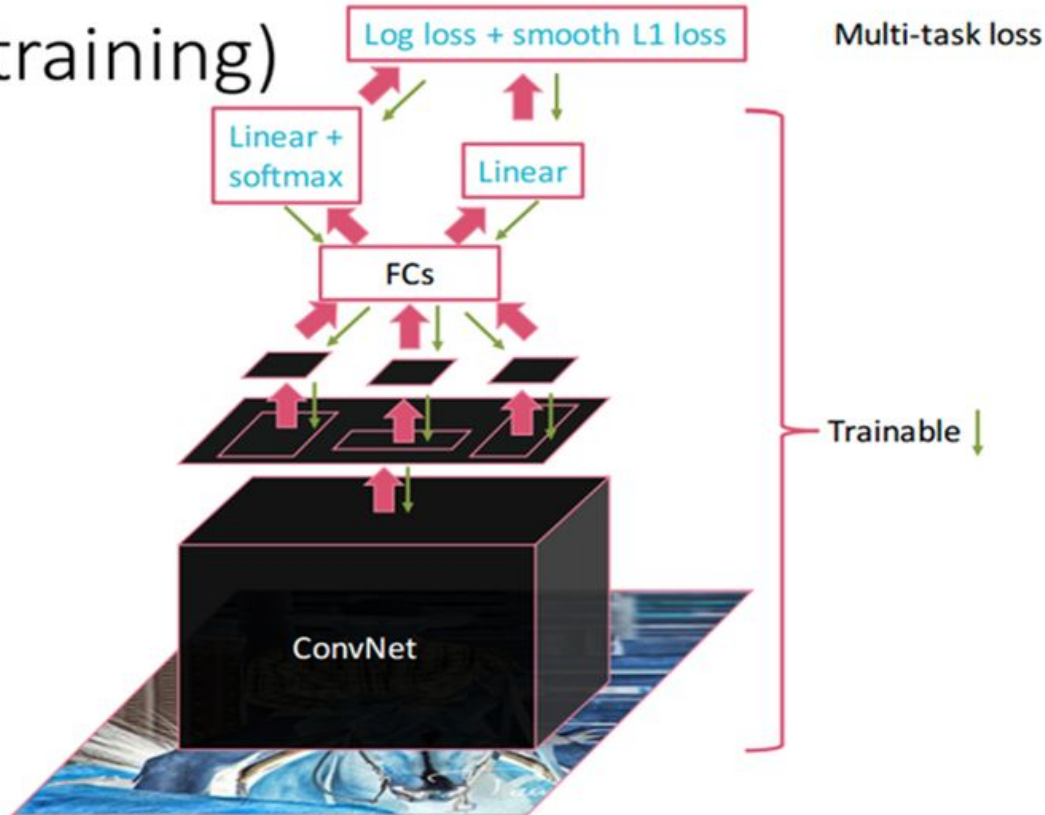


Fast RCNN : Overview



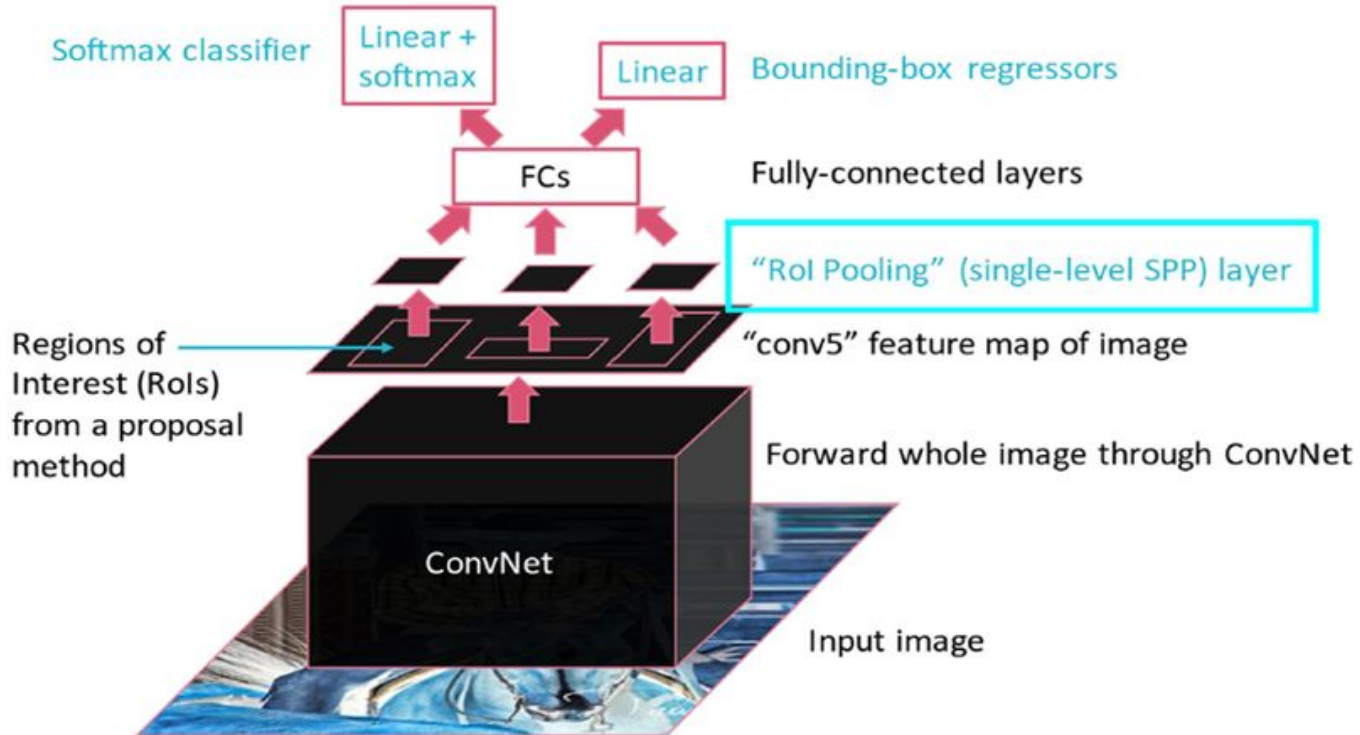
Fast RCNN : training process

Fast R-CNN
(training)

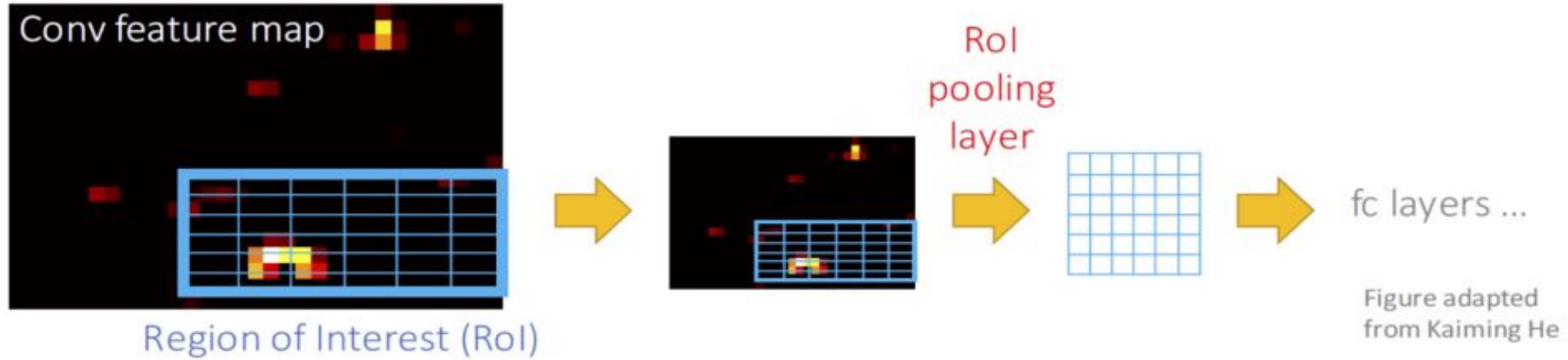


Fast RCNN : testing process

Fast R-CNN (test time)



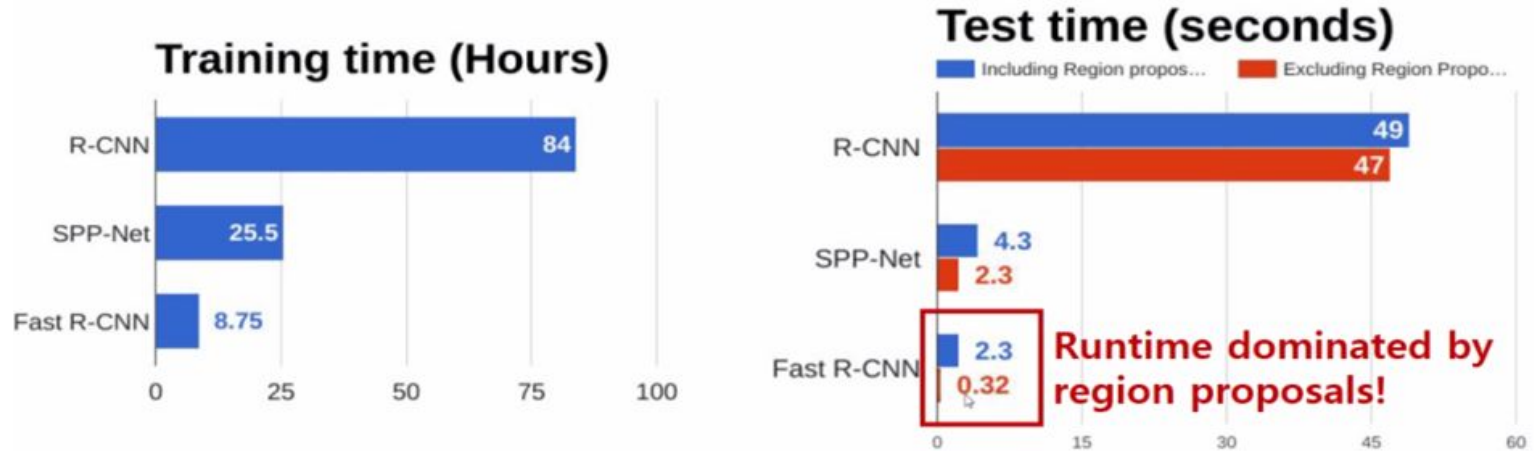
Fast RCNN : RoI pooling layer



RoI in Conv feature map : $21 \times 14 \rightarrow 3 \times 2$ max pooling with stride(3, 2) \rightarrow output : 7×7
RoI in Conv feature map : $35 \times 42 \rightarrow 5 \times 6$ max pooling with stride(5, 6) \rightarrow output : 7×7

Fast RCNN : Weak point

R-CNN vs SPP-net vs Fast R-CNN



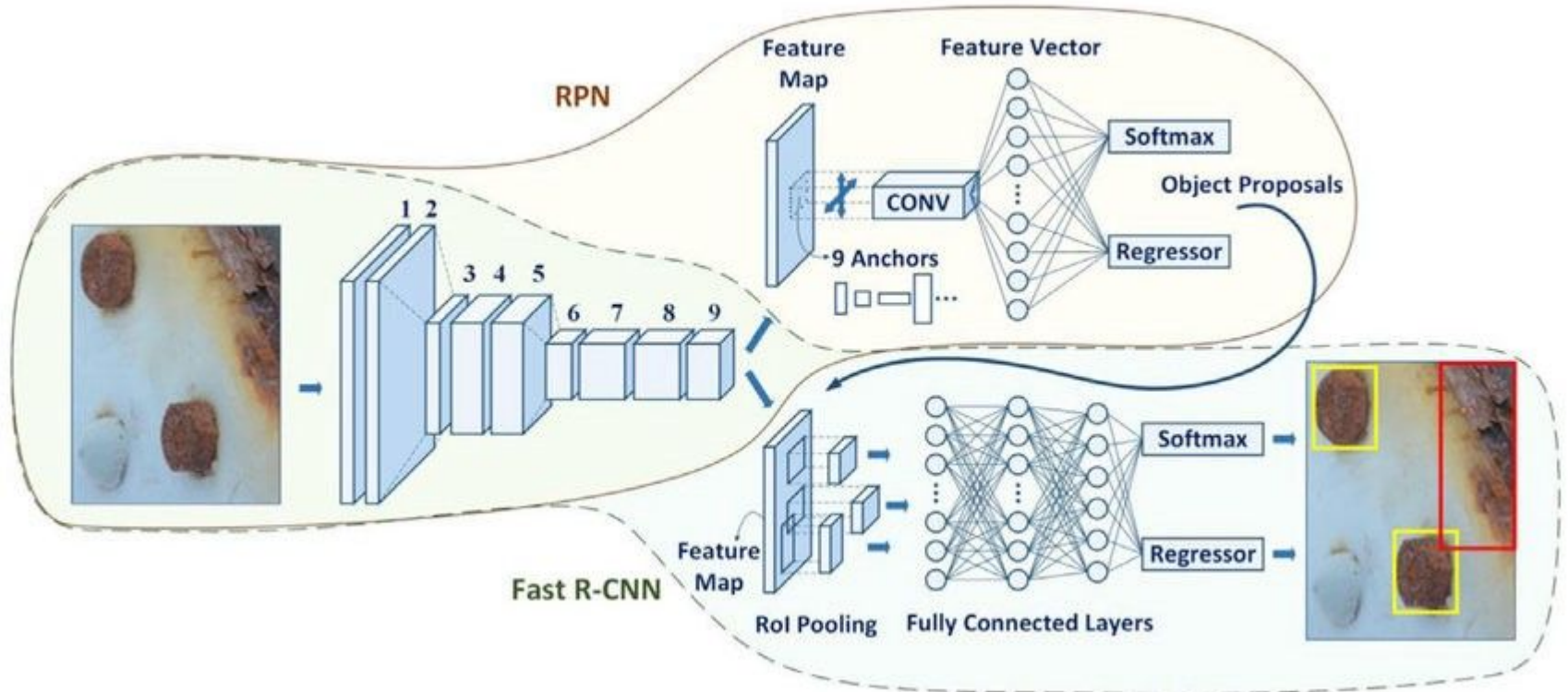
Faster RCNN

Faster R-CNN



Fast RCNN의 병목(Bottle-neck) 부분이었던 **Selective search**를 신경망 구조로 변경

Faster RCNN





Free Google Slides Templates