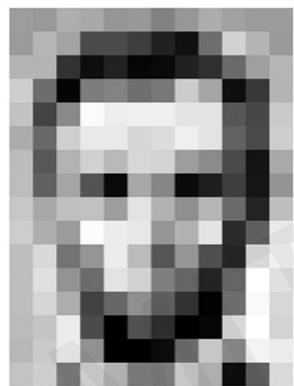


What Computers see?

→ Images = Matrix of Numbers



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	53	17	110	210	180	154
180	180	50	14	54	6	10	93	48	106	159	181
205	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	226	227	67	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	64	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	176	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	158	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	9	0	12	104	209	138	243	236
195	206	123	297	177	121	123	200	176	13	96	218

What the computer sees

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	53	17	110	210	180	154
180	180	50	14	54	6	10	93	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	64	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	176	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	158	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	9	0	12	104	209	138	243	236
195	206	123	297	177	121	123	200	176	13	96	218

An image is just a matrix of numbers [0,255]!

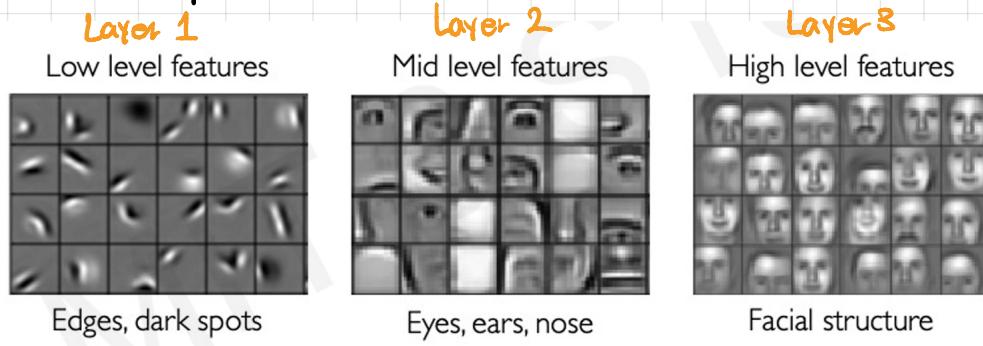
→ Computer Vision은 1. Image Classification 2. Image regression 이 주제로 있다.
↳ 학습 등...

In classification

→ Images에는 Pattern, feature는 찾아야 한다.

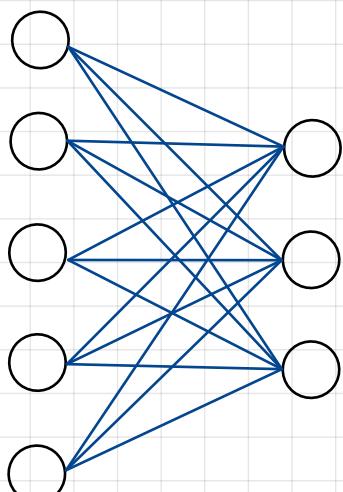
2가지) 1. Domain knowledge 2. Define features 3. Detect features to classify

Features Hierarchy



Neural network은 low-level features를 high-level features로
단계별로 특성화된 feature set을 형성해 나간다.

Fully-Connected Neural Network



설명: 각 Hidden Layer의 속한 각 유닛들이
각 Layer의 모든 유닛과 연결된다.

* 같은 Layer의 모든 유닛은 같은 입력을 받는다.
하지만 각 유닛은 다른 weight를 갖고 있으므로
각 유닛은 서로 다른 특성을 갖게 되며,
각 유닛은 같은 특성을 갖는다.

Image Analysis Using Fully-Connected Neural Networks

1. Pixel, Image는 2D - Array 이다. 1D - Vector-2 flatten 한다.

→ 이 과정에서 2가지 문제점이 생긴다.

a) Spatial Information이 제거됨

→ 같은 영역(표정)의 정보가 제거될 수 있다.

b) Too many Neurons and Parameters

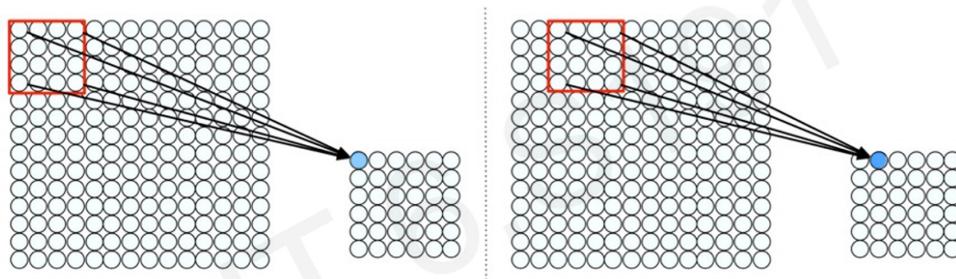
→ $2D \in R^{N \times N}$ 이라면 Input $\in R^{N^2 \times 1}$

→ 비효율적이다.

Fully-Connected Neural Networks가

2D Input을 사용하지 않기 때문이다.

2. Reducers of Spatial Information 문제 해결



Pixel의 유통인 patch를
다음 Layer의 Neuron에
전달한다.

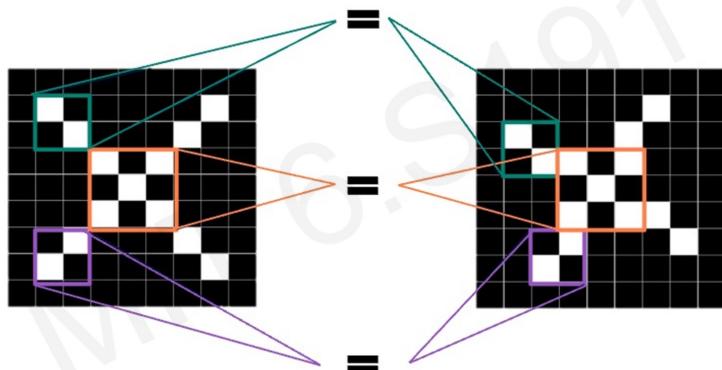
Patch = Filter \rightarrow Set of weights : Filter 자체가 weight이다.

1. 각 filter는 local features를 Extract 한다.

2. 여러 개의 filter를 사용하면 다른 features를 Extract할 수 있다.

3. 한 이미지의 다른 공간에 있는 같은 특징은 같은 filter(parameter)를 공유해야 한다.

Example



Filter를 통해 추출한 같은 feature가
겹친다. 때, 같은 위치의 다른 filter는
같은 결과를 출력한다.

Element-wise multiplication : 같은 크기의 행렬에서 같은 위치의 Element끼리 곱하는 것

1	1	1
0	1	1
0	0	1

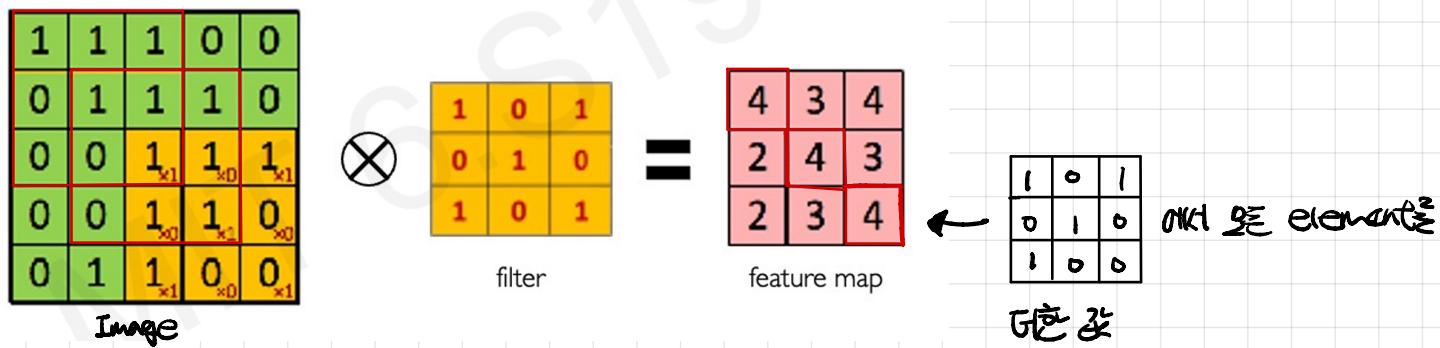
•

1	0	1
0	1	0
1	0	1

=

1	0	1
0	1	0
0	0	1

Convolution Operations

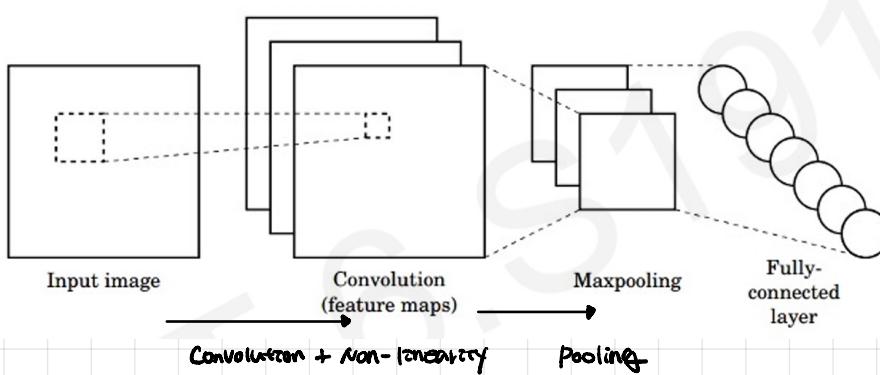


1. 홀수 행단위로 시동하여 Element-wise multiplication 수행 후 합을 구하고 저장
 - 2.Stride 만큼 한 칸씩 이동하여 (1) 과 같은 방식
- Filter는 어떤가 구성되어서 앞에서에 따라 다른 특성을 나타낼 수 있다.

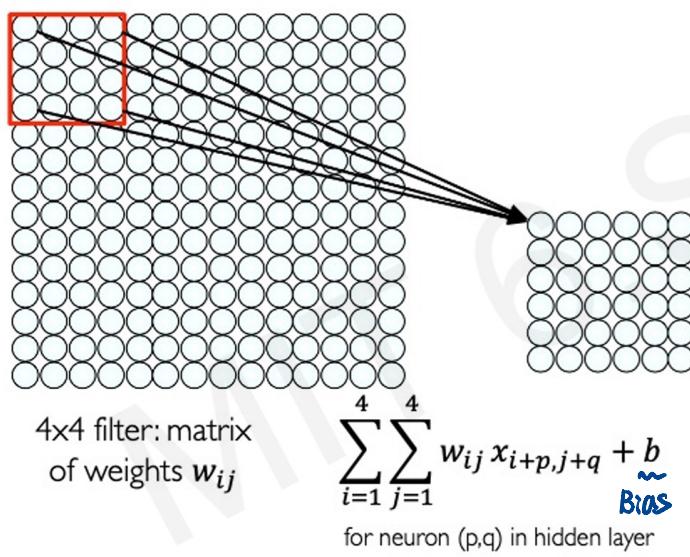
Convolution Neural Networks (CNN)

Filter는 Weight로 생각해보면, 사람이 직접 Filter의 값을 정하기보다 일반적인 Neural Networks처럼 컴퓨터가 스스로 Weight를 학습할 수 있도록 할 수 있는 것 같은 생각은 생각에서 나온 Neural Network

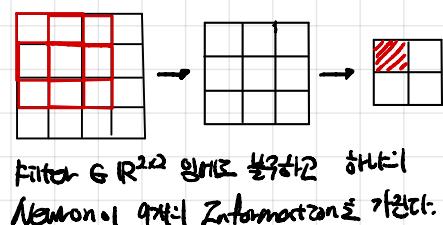
CNNs for Classification



1. Convolution + Non-linearity



* Convolution은 계산마다 높은 하나의 Neuron이 여러 Spatial Information을 나타내게 된다.

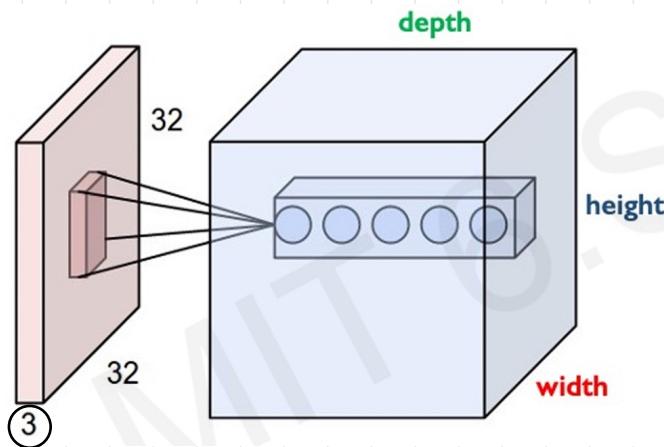


Filter는 $R^{2.2}$ 원래도 높은하고 다른 Neuron의 차례로 Information을 가진다.

- Linear Combination
 - Element-wise Multiplication + bias
- Linear Combination + non-linear Activation function 적용

$$h\left(\sum_{i=1}^4 \sum_{j=1}^4 w_{ij} \cdot x_{i+p,j+q} + b\right)$$

Additional 1) Spatial Arrangement of Output Volume



Layer Dimensions:

$h \times w \times d \rightarrow$ feature \times

where h and w are spatial dimensions
 d (depth) = number of filters

Stride:

Filter step size

$h \times w$

\rightarrow feature map size

Receptive Field:

Locations in input image that a node is path connected to

이미지에 담겨 있는 feature \times ex) RGB

이 뭘까?

$\begin{cases} R \text{ Channel} \rightarrow \text{Red} \text{의 } \times \text{정도} \\ G \text{ Channel} \rightarrow \text{Green} \text{의 } \times \text{정도} \\ B \text{ Channel} \rightarrow \text{Blue} \text{의 } \times \text{정도} \end{cases}$

하나의 채널을 선택하면
전부다 선택된다.

h, w 같은 뜻

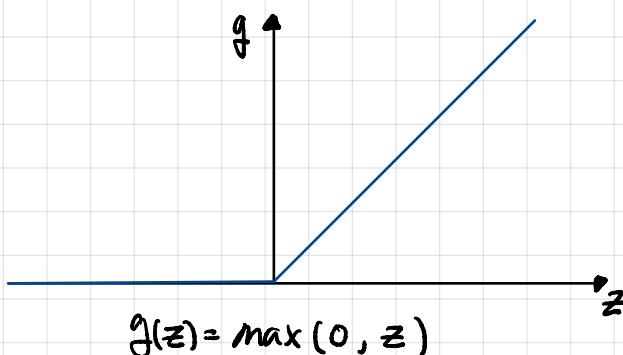
$$R^{\text{Input}} = R^{\text{Filter}}, \text{ Input size} = N \times N \times 3, \text{ Filter size} = k \times k \times 3, \text{ Stride} = S$$

$$h = w = \frac{N - k}{S} + 1 \quad \leftarrow \text{Input, Filter의 row, column size가 동일하기 때문에.}$$

Output layer dimension : $h \times w \times d$

Additional 2) ReLU: Introducing Non-Linearity

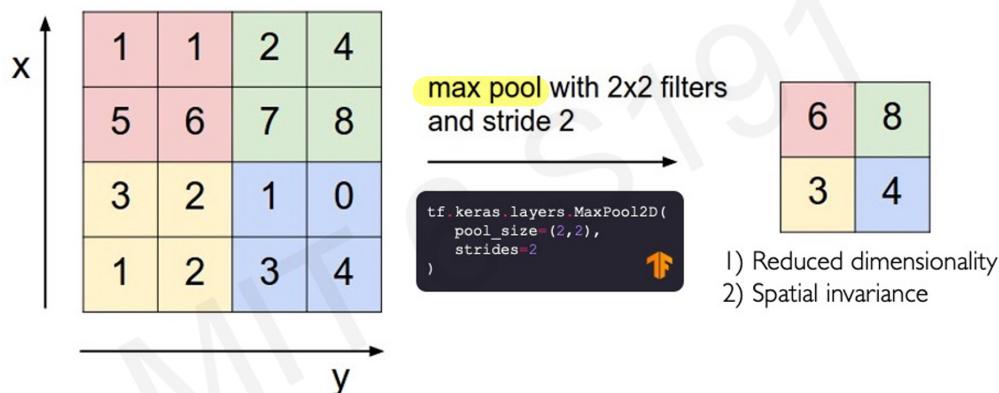
Rectified Linear Unit



- Negative Value $\rightarrow 0$

만약 값이 음수가 있다면.

2. Pooling



- 풀링을 사용하는 등 다른 방법도 있다.

장점) 1. Reduce dimensionality

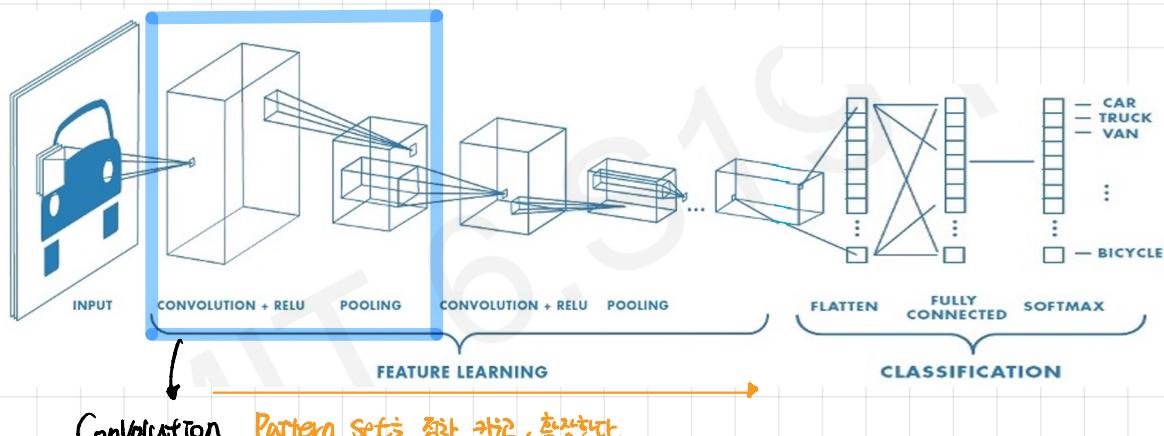
2. Spatial Invariance : 같은 특성이 다른 위치에 있어도 탐색할 수 있는 능력

- 같은 특성이어도 다른 위치에 있으면 두번 Pixel 때문에 구별이 안될 수도 있다.
- Max Pool은 최대값만 저장하므로 이 경우에 빠짐은 흐름이다.

3. DownSampling

- 이미지 크기를 줄여 더 많은 정보를 넘긴다.

CNNs for classification: Class probabilities



1. Convolution Layer

- Convolution + Non-linearity + Pooling을 반복
- 보통 2~5번의 Convolution Layer로 이루어져 있다.
- 레이어를 차례로 Low-level → Mid-Level → High-Level
→ 차례로 디테일한 특징(엣지, 유크)에서 점진적으로 넓은 영역(物体, 대상) 특징

2. Fully-Connected Layer

- High-level input을 받아 Fully-Connected Layer로 분류한다.
- Softmax(y_i) = $\frac{e^{y_i}}{\sum_j e^{y_j}}$ 을 이용해서 Class에 따른 probability를 계산한다.

$$0 \leq \sim \leq 1$$

Applications of CNN

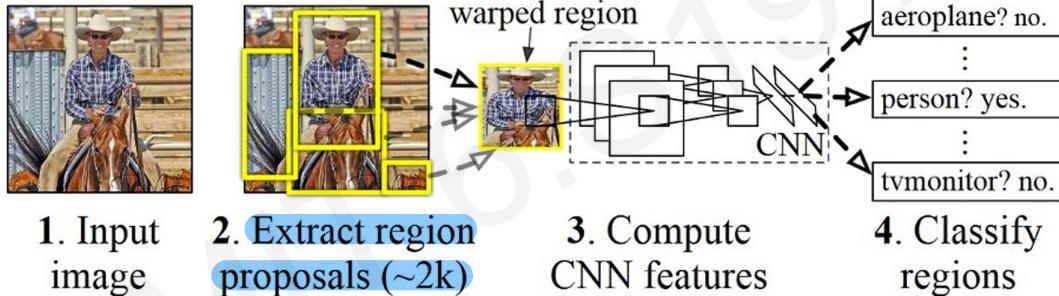
1. Classification

→ 으로 영상 분석

2. Object detection

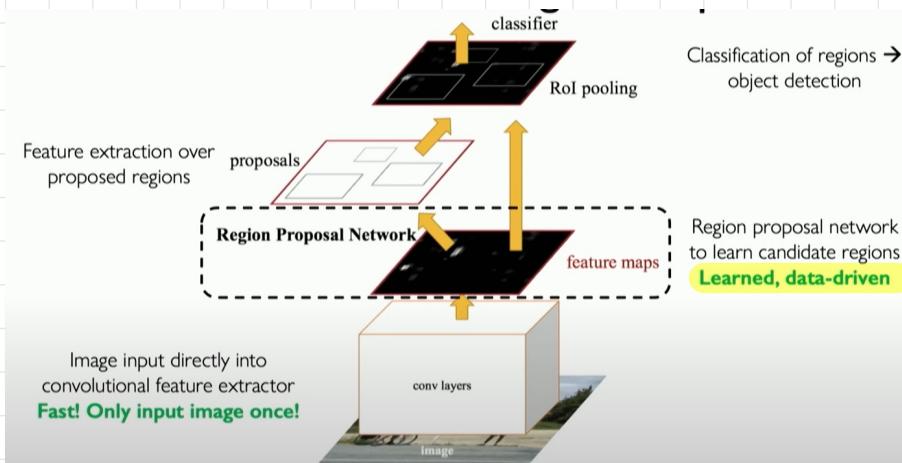
- R-CNN (Regions with CNN)

R-CNN algorithm: Find regions that we think have objects. Use CNN to classify.



↳ object가 있는 부위의 region을 추출하고 각 region에 대해
각각 CNN을 통한 Feature Vector를 추출
Ex: Action Extracting에선 2300개를 추출.
다수의 Region마다 각각 CNN을 수행해야 한다.

- Faster R-CNN



→ 각 Region마다 CNN을 실행하는 대신 객체가 스스로 학습하도록 하여
Region Proposal을 찾도록 한다.

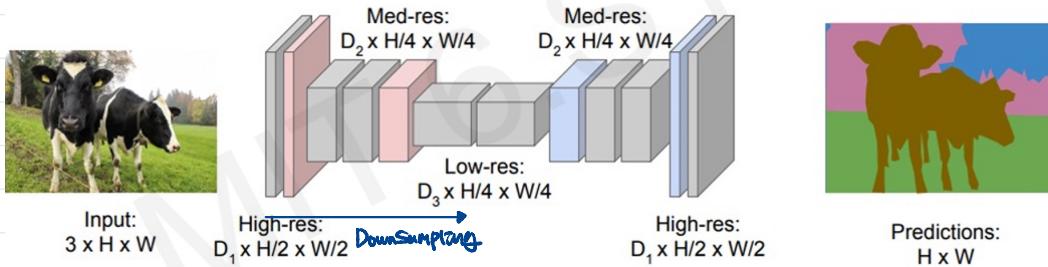
1. 전체 이미지를 CNN에 입력 → 전체 이미지에 대한 feature map을 만든다.
2. 각자가 있는 가능성이 높은 영역 (Region Proposal)을 찾는다.
3. ROI (Region of Interest) Pooling을 통해 고정된 크기로 변환
4. Fully-Connected Layer로 전달하여 각자를 분류한다.

3. Semantic Segmentation: Fully Convolutional Networks

Pixel-level Classification

FCN: Fully Convolutional Network.

Network designed with all convolutional layers,
with **downsampling** and **upsampling** operations



Downsampling: 이미지의 **높은 특성을** 빼는 과정

Upsampling: Downsampling 과정에서 생긴 Feature Map을 **증가시켜** 원래의 **특성을** 되돌려놓는 과정
- 학습도

4. Continuous Control

Ex) Navigation from Vision

Entire model is trained end-to-end **without any human labelling or annotations**

