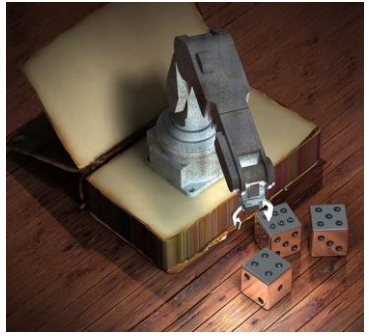
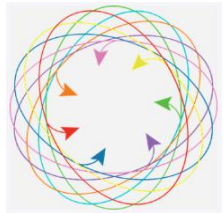
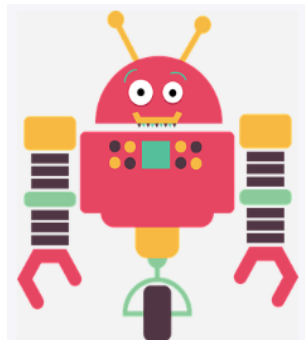
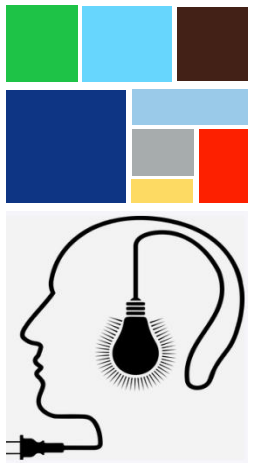








In the future, an AI agent will know that you are at work and have ten minutes free, and then help you accomplish something that is high on your to-do list.

딥러닝 알아보기



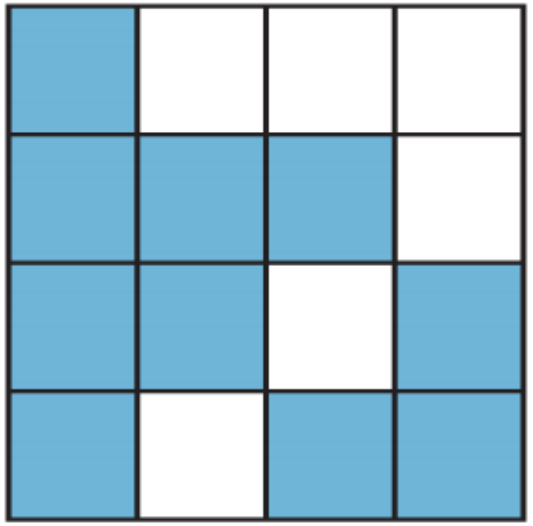

PART1. CNN 대한 이해

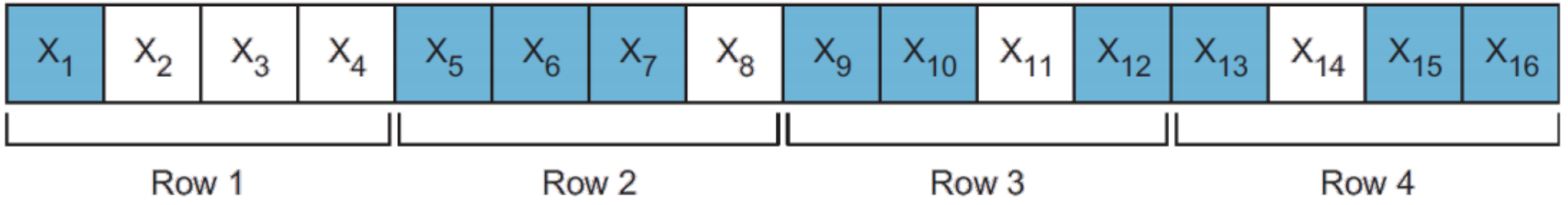
Enjoy your possibility

Tensorflow-keras 로 CNN 알고리즘 구현하기

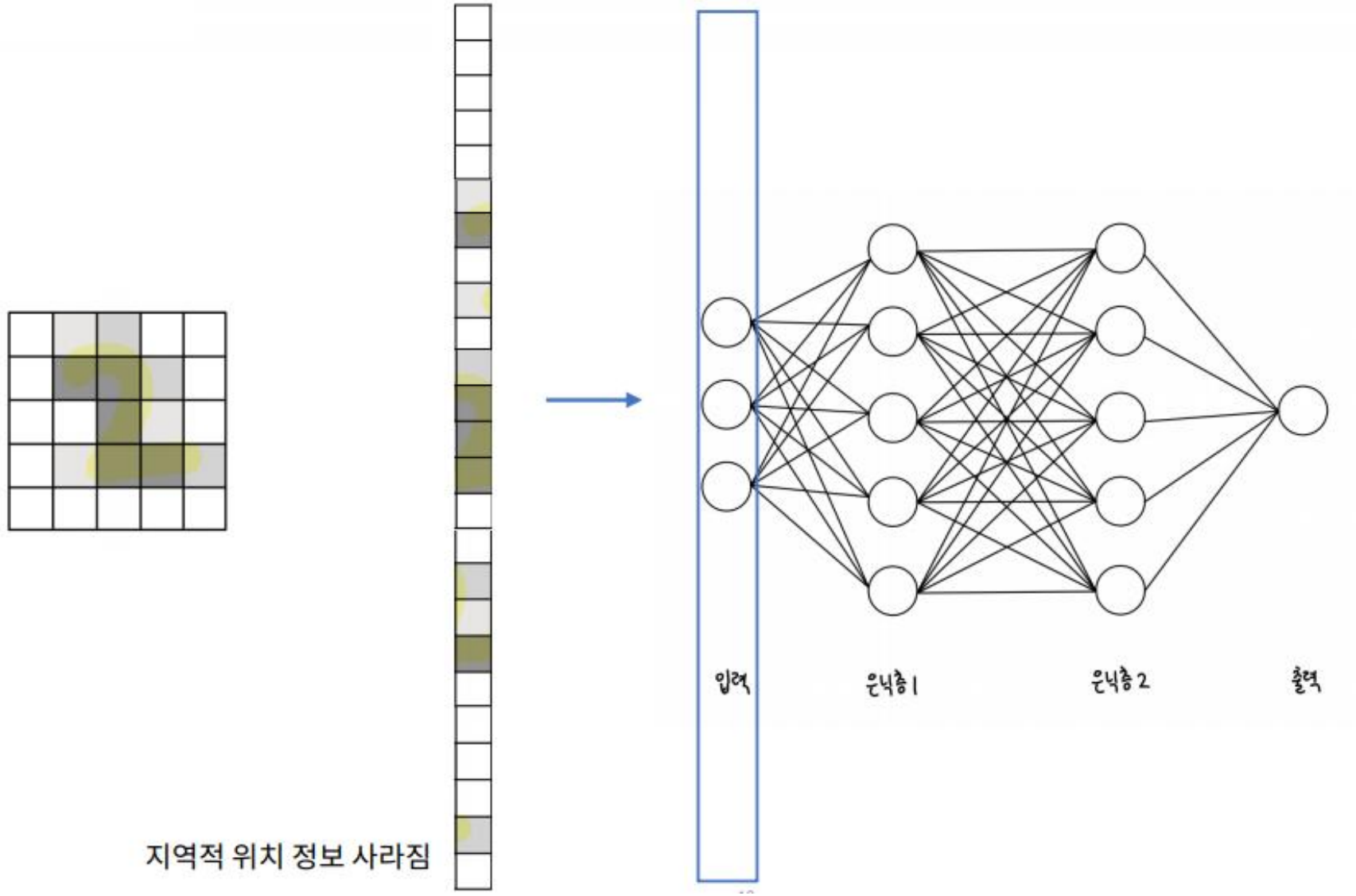
왜 CNN인가?

- 이미지를 1D 벡터 입력으로 평면화하면 2D 이미지의 공간적 특징이 손실됨



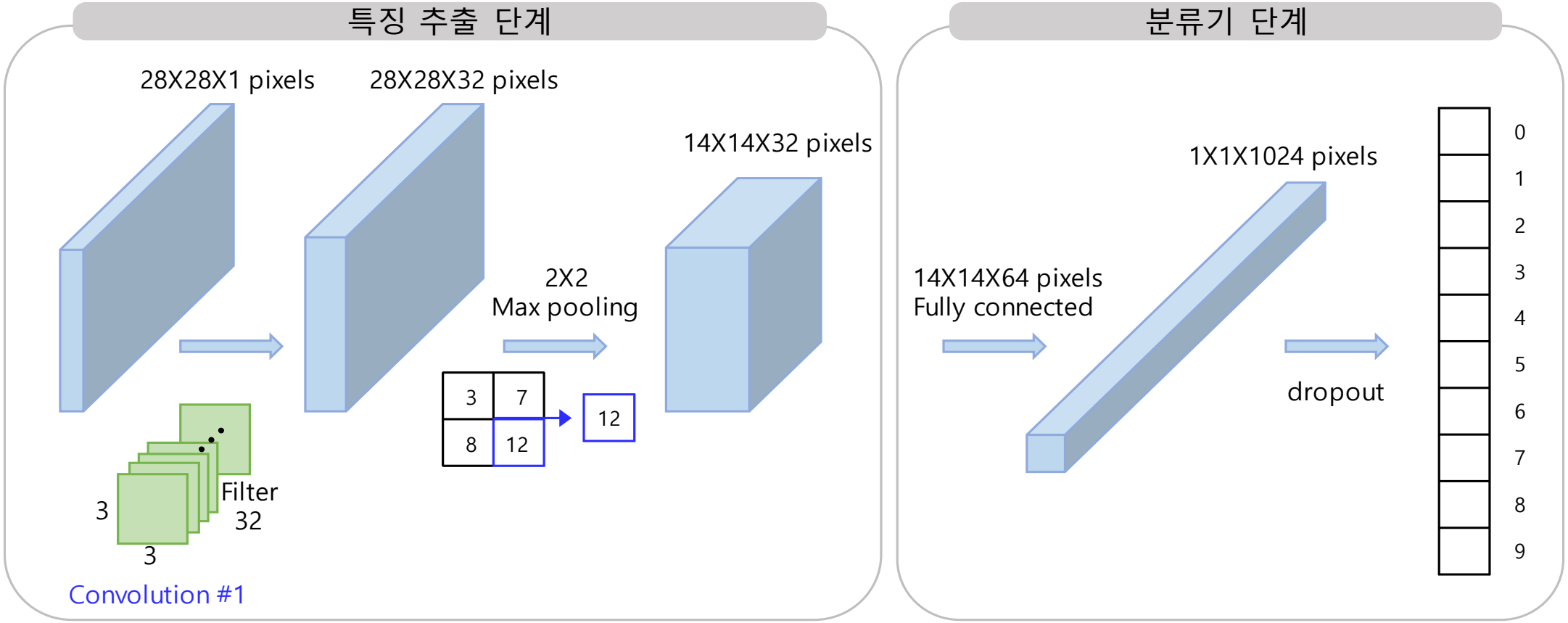


왜 CNN인가?



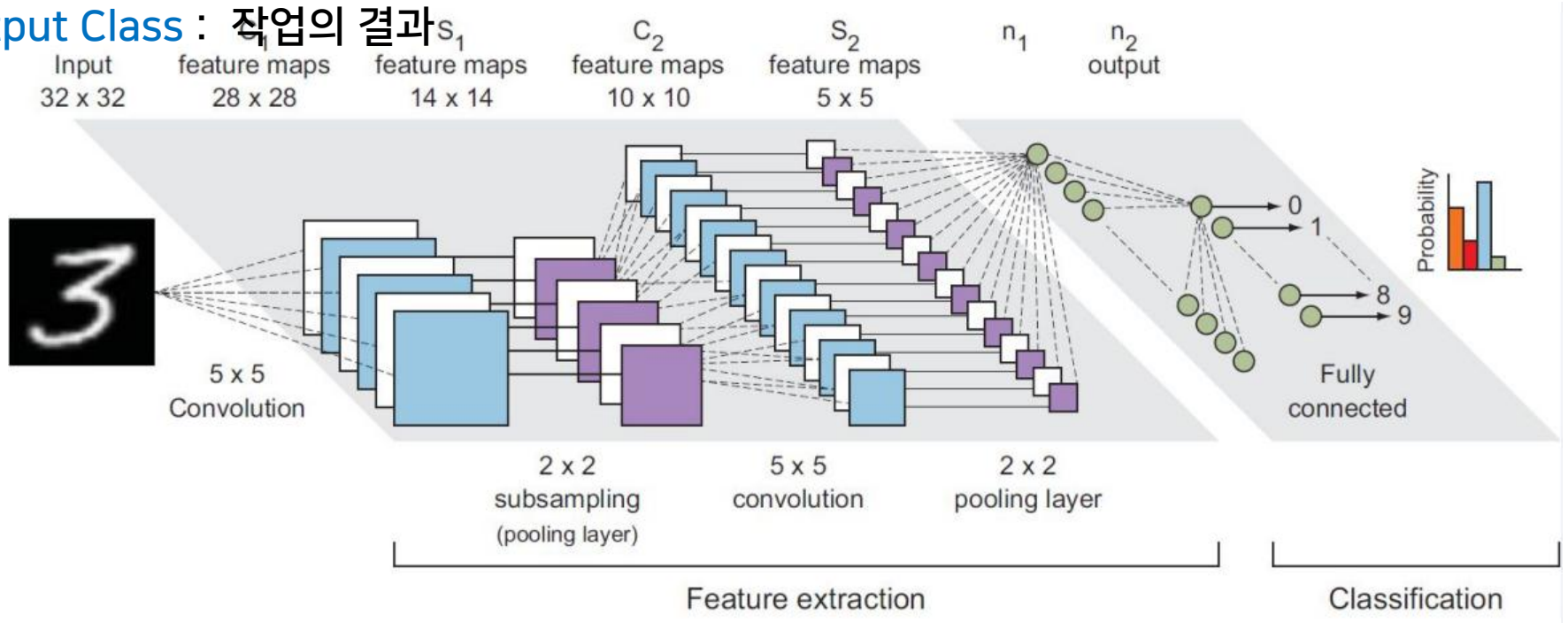
CNN(Convolution Neural Network, 합성곱신경망)

- CNN 구조



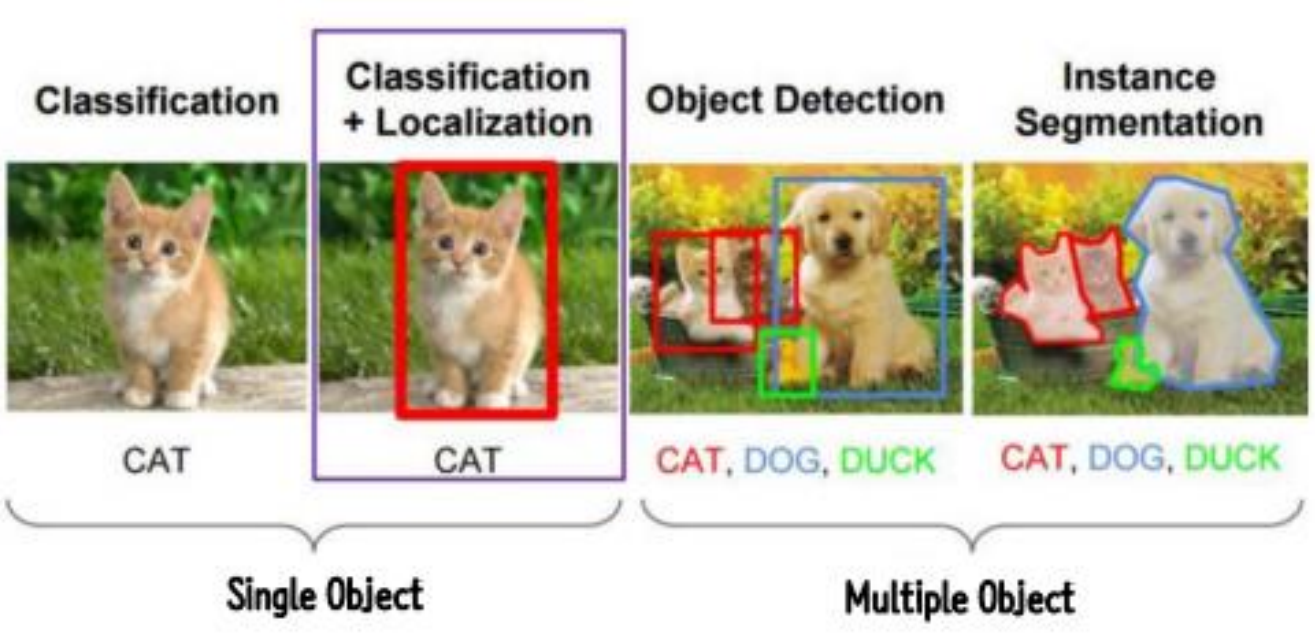
CNN 개요

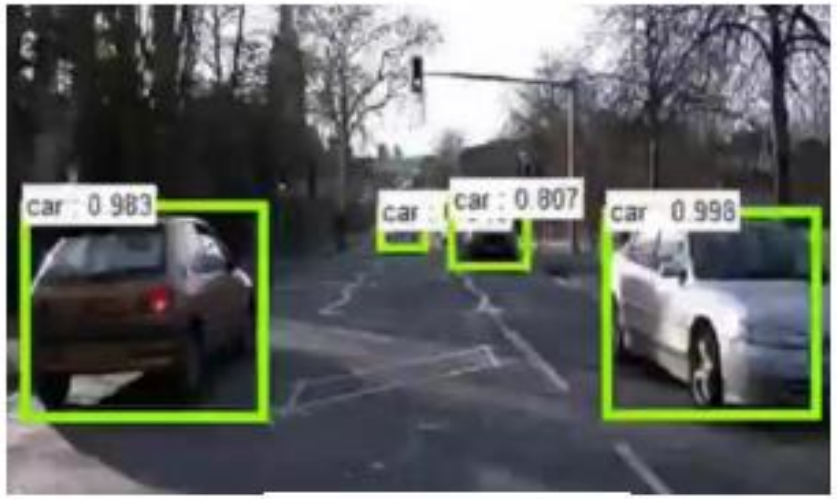
- Input Image : 이미지를 하나의 입력으로 취함
- Convolution Layers : Feature Extraction을 수행하는 Layer
  - ✓ Convolution Layer + ReLU : Feature 추출, 의미 없는 특징을 zero화
  - ✓ Pooling Layer : Feature 개수 축소, 중요한 Feature 만 유지 (선택적 작업)
- Fully-Connected Layer : 비선형 조합 학습 및 분류 작업 수행하는 Layer
- Output Class : 작업의 결과



CNN 활용분야

- 분류(Classification)
- 지역화(Localization)
- 이미지 세분화(Image Segmentation)
- 물체 감지(Object Detection)

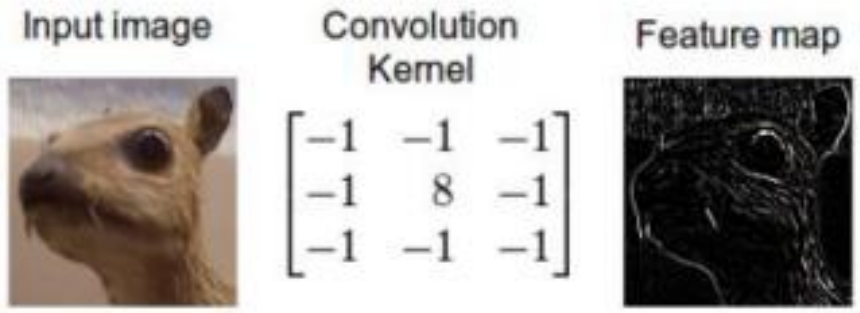


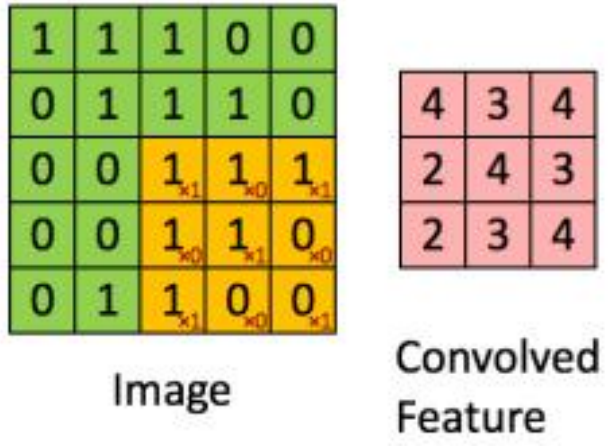


Object Detection

CNN 이해 - 필터 커널(Filter Kernel)

- Filter(Convolution Kernel Matrix)를 적용하여 입력에 대해 특정 성분에 대해서만 뽑아내는 작업
  - ✓ 예) 사선 정보, 직선 정보, 동그란 정보, 각진 정보….
  - ✓ 알고 싶은 특정 성분에 따라 Filter 의 모양이 다름
  - ✓ CNN은 Filter를 갱신하면서 학습하는 것임
- Image에 특정 Filter를 Convolution한 결과를 Feature Map 이라고 함
  - ✓ Feature Map 은 Image에 적용된 Filter 개수 만큼의 Channel을 갖게 됨
  - ✓ n개의 Filter 가 적용된 경우 n개 Channel
- Stride: Filter를 순회하는 간격, Stride가 2로 설정되면 2칸씩 이동하면서 convolution 하게됨





http://deeplearning.stanford.edu/wiki/images/6/6c/Convolution_schematic.gif

CNN 이해 - 필터 커널(Filter Kernel)

- https://setosa.io/ev/image-kernels/

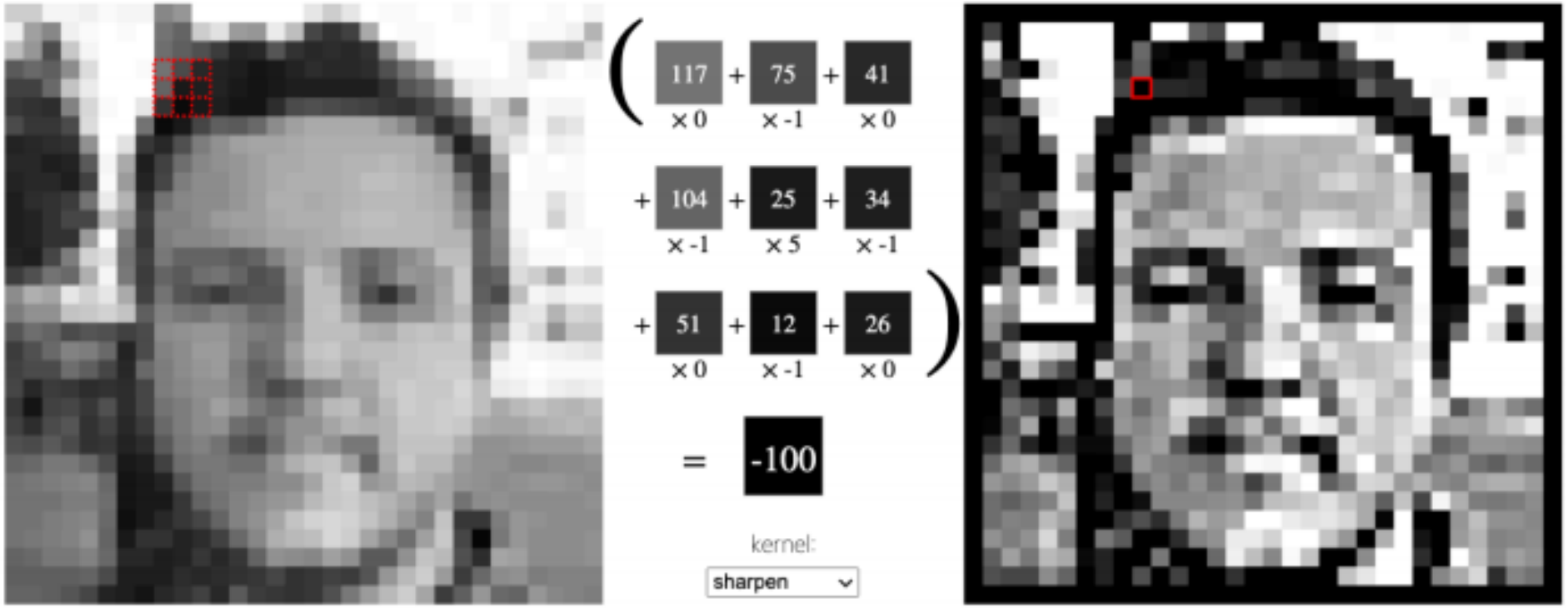
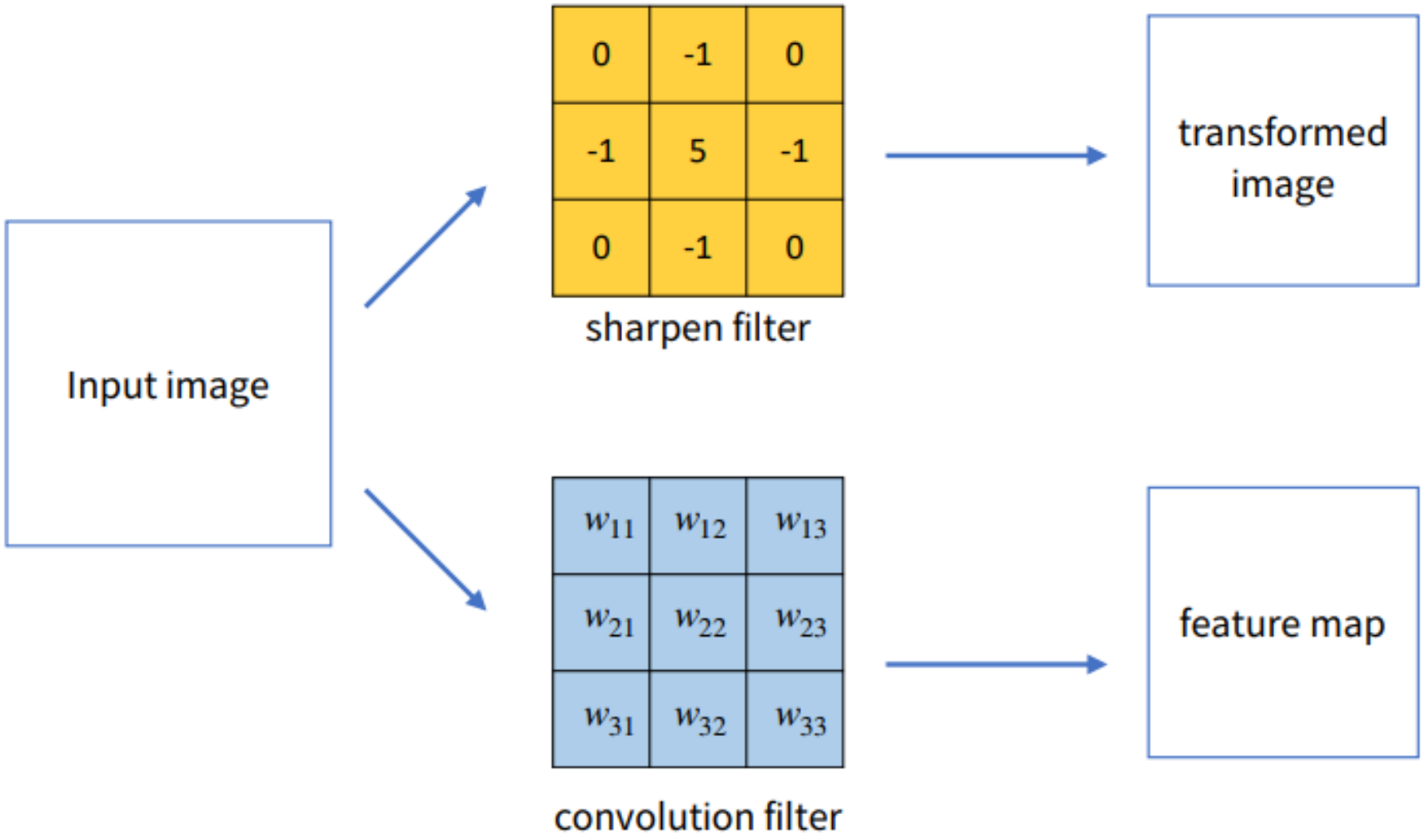


Image Kernels Explained Visually (By Victor Powell)

CNN 이해 – 필터 커널(Filter Kernel)



CNN 이해 - 필터 커널(Filter Kernel)

- 원본 이미지에 특수한 행렬로 컨볼루션을 취함
- 행렬의 특성에 따라 원본 이미지로부터 특성이 강조된 이미지를 얻을 수 있음

①

1	1	1	2	2	2
1	1	1	2	2	2
1	1	1	2	2	2
1	1	1	2	2	2

*

필터 1

-1	0	1
-1	0	1
-1	0	1

*

필터 2

-1	-1	-1
0	0	0
1	1	1

②

1	1	1	1
1	1	1	1
1	1	1	1
2	2	2	2
2	2	2	2
2	2	2	2

*

필터 1

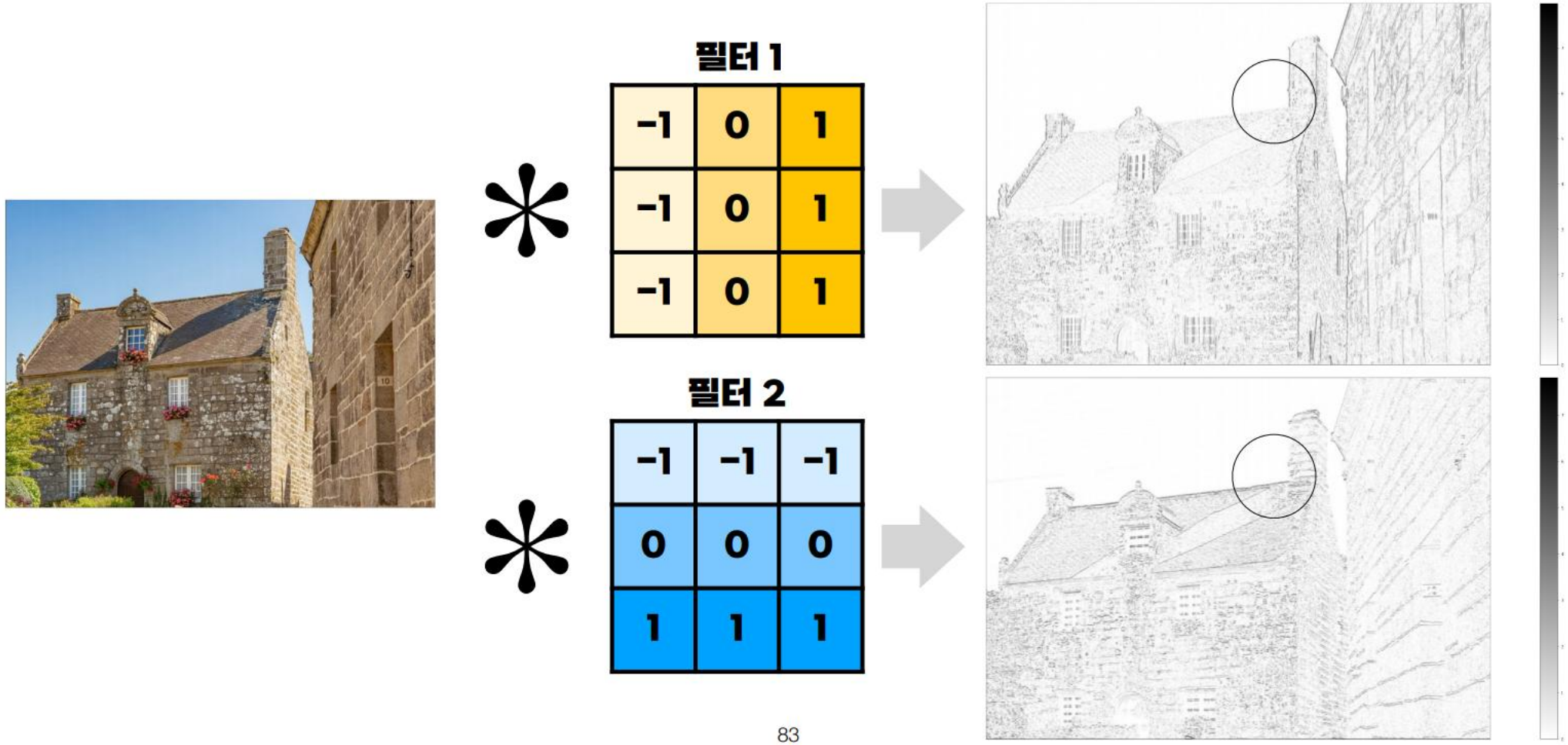
-1	0	1
-1	0	1
-1	0	1

*

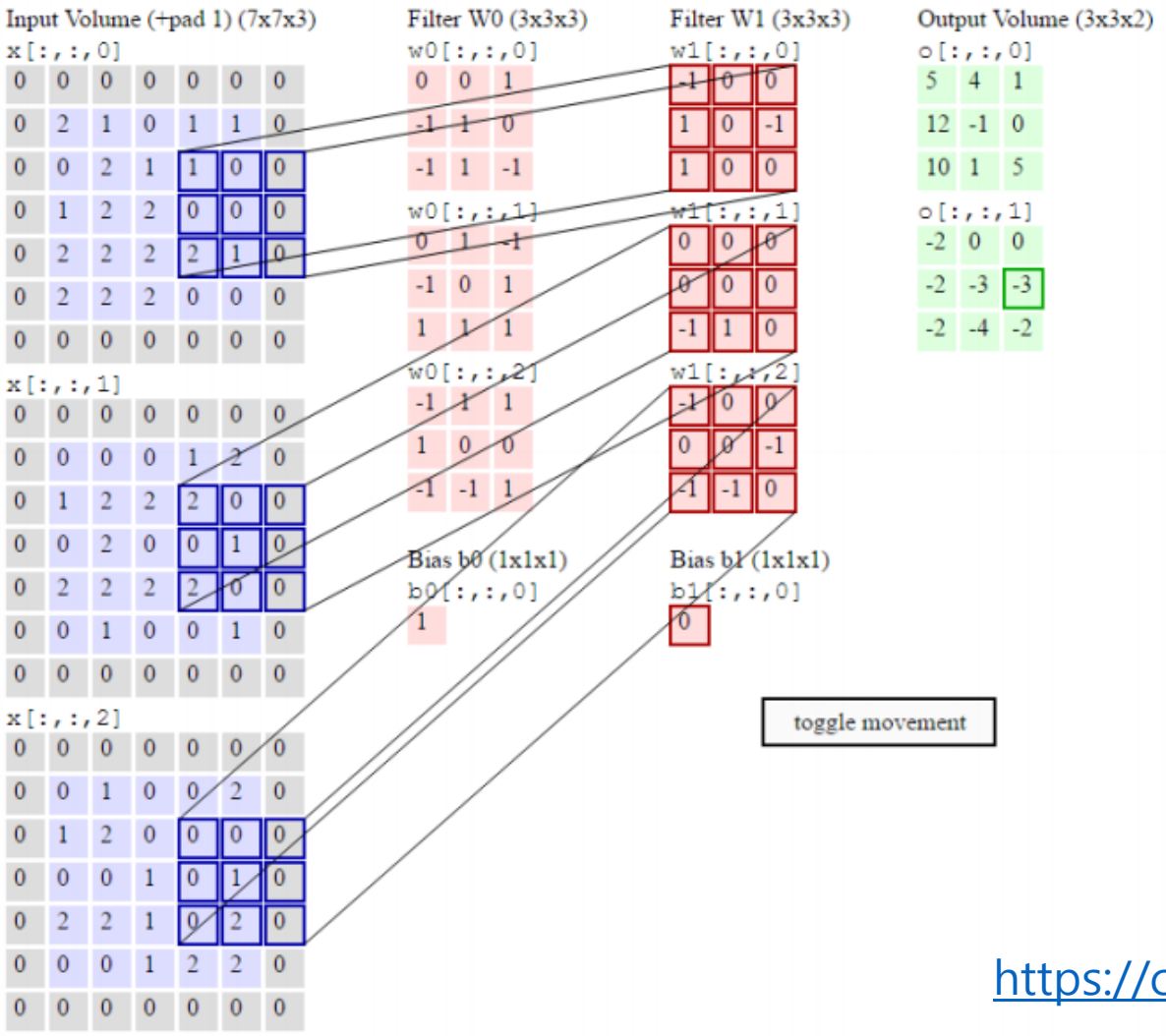
필터 2

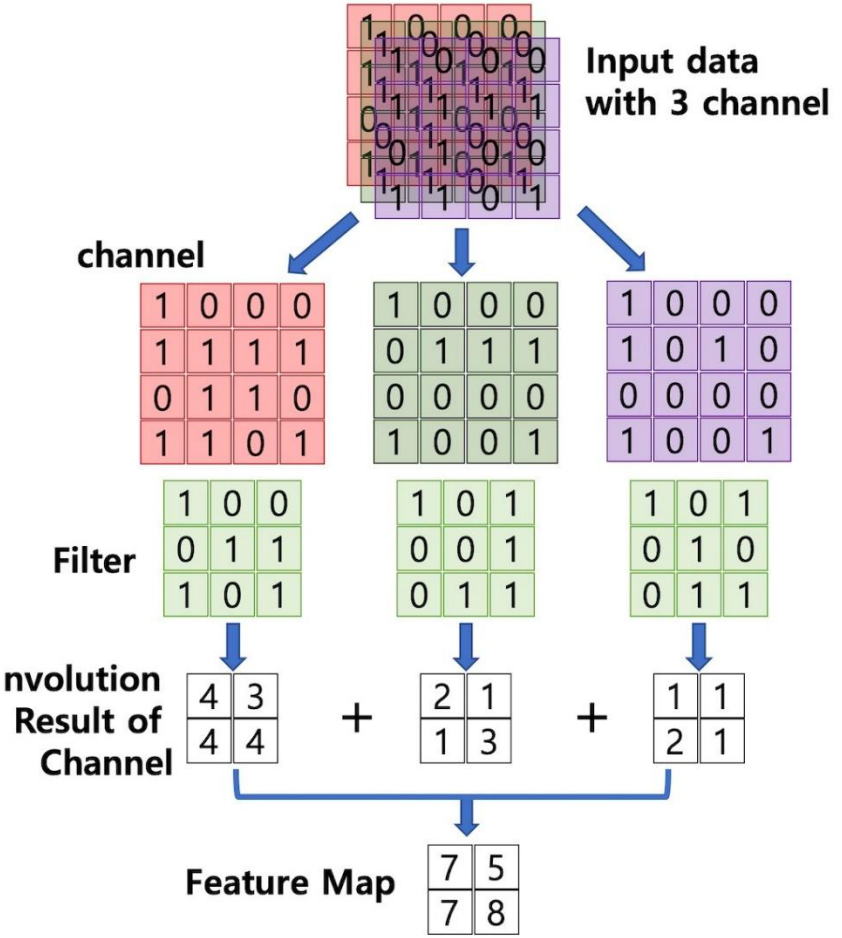
-1	-1	-1
0	0	0
1	1	1

CNN 이해 - 필터 커널(Filter Kernel)



CNN 이해- 필터 커널(Filter Kernel)



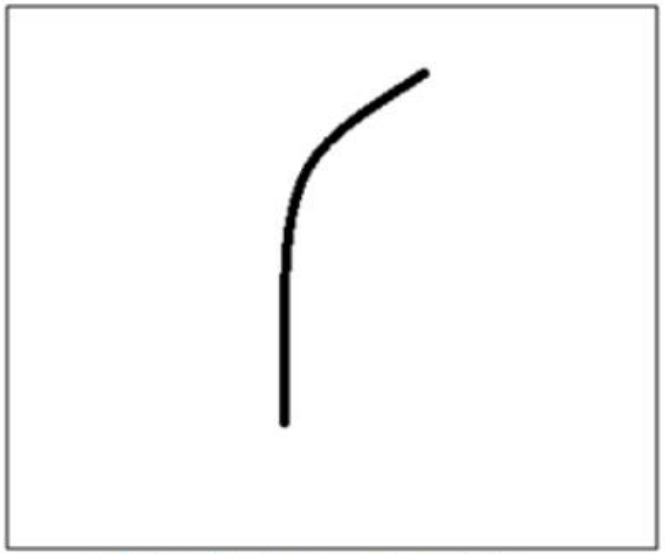


https://cs231n.github.io/convolutional-networks/

CNN 이해 - 필터 커널(Filter Kernel)

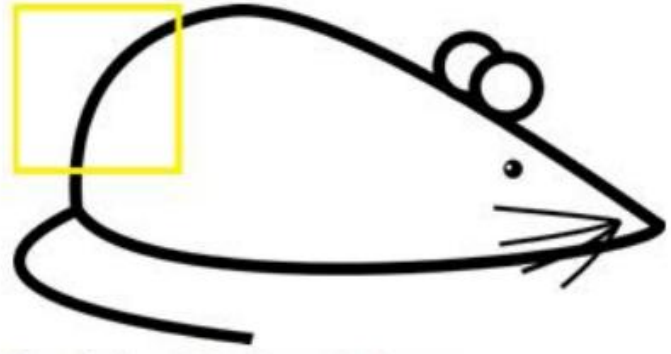
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter



Visualization of a curve detector filter

Original image



Visualization of the filter on the image

CNN 이해 - 필터 커널(Filter Kernel)

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

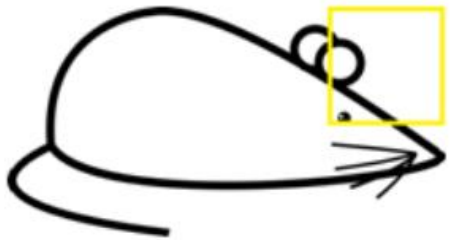
Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)



Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

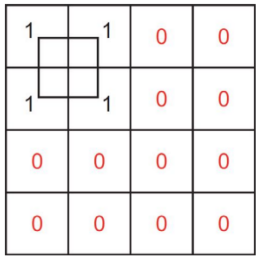
*

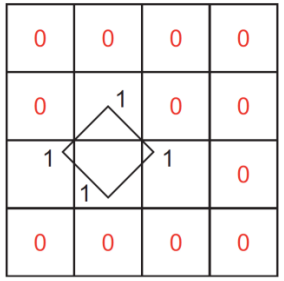
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

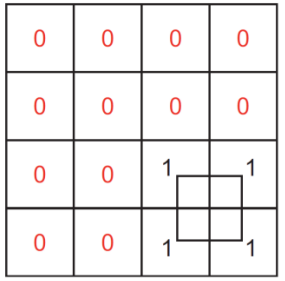
Pixel representation of filter

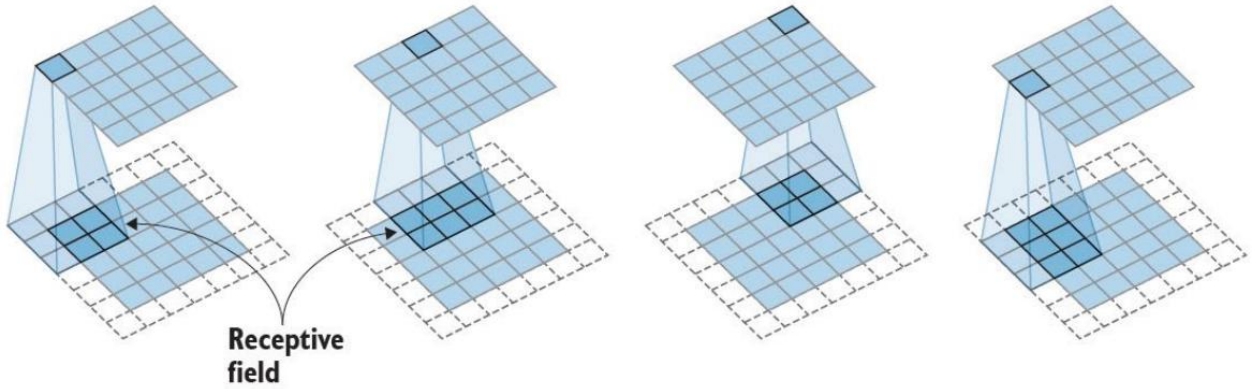
Multiplication and Summation = 0

CNN 이해





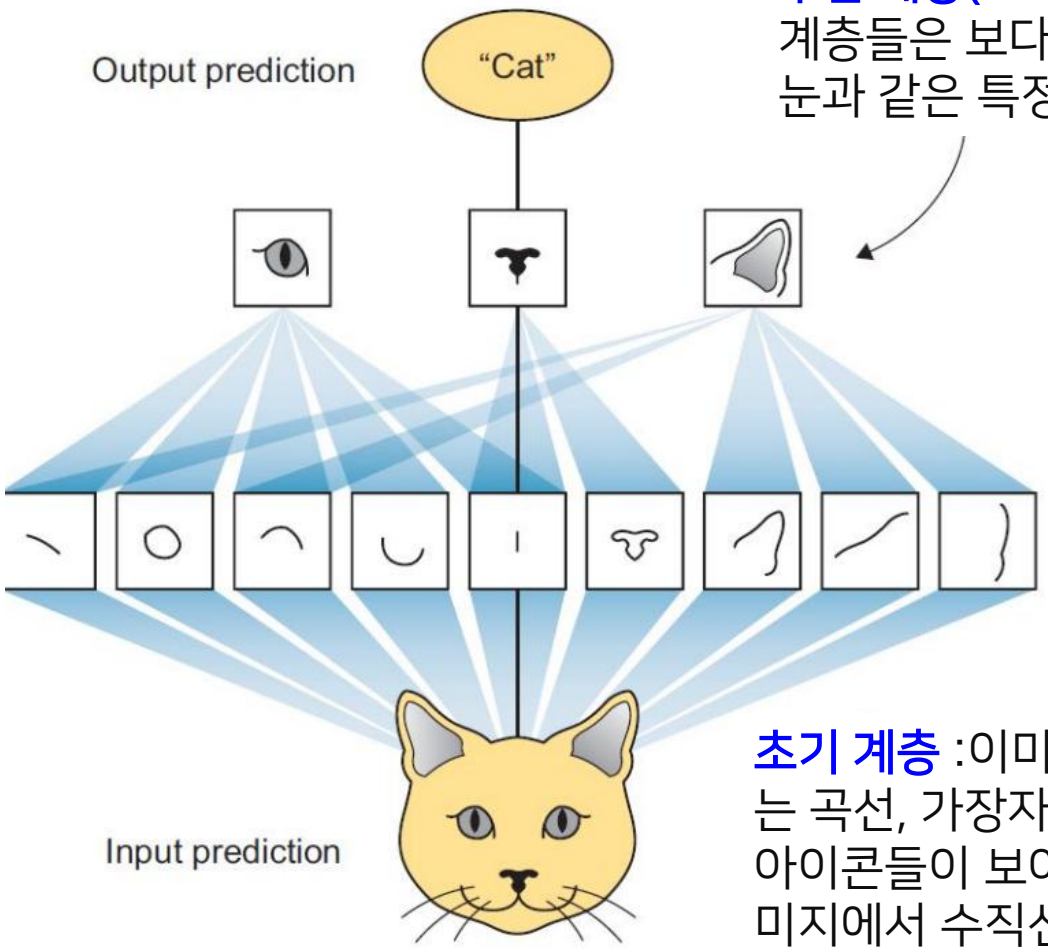




Preserves locality

https://cs231n.github.io/assets/conv-demo/index.html

CNN 이해



후반 계층(Later layers): 신경망의 깊은 부분에 위치한 이 계층들은 보다 복잡한 특징을 학습합니다. 고양이의 귀, 코, 눈과 같은 특정 객체의 일부를 인식할 수 있습니다.

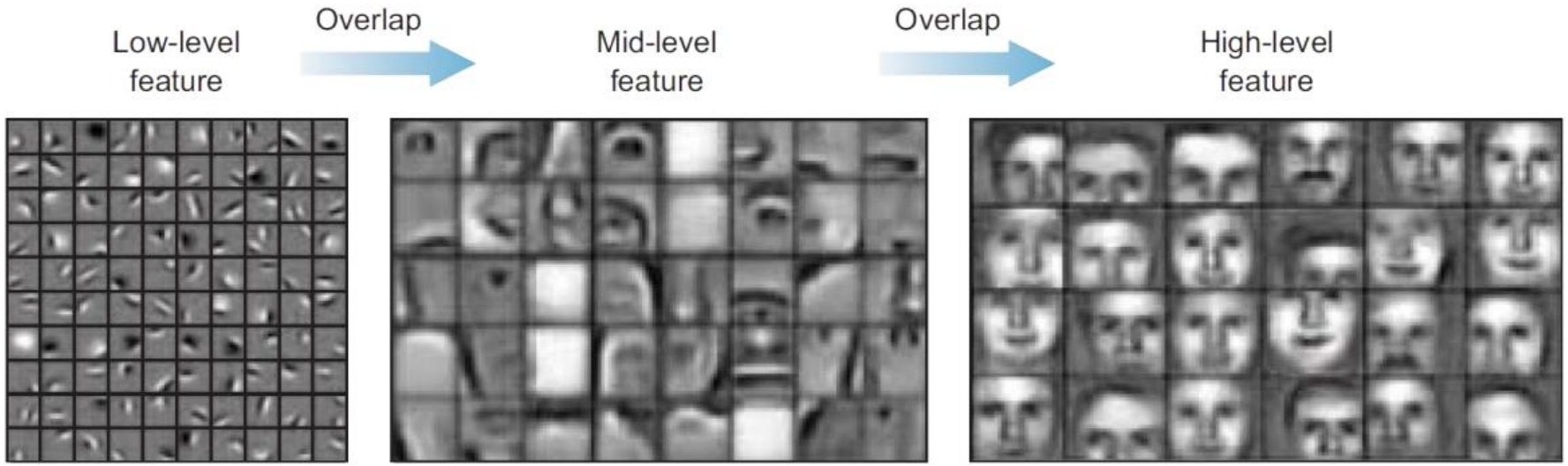
초기 계층 :이미지의 간단한 특징을 학습합니다. 여기서는 곡선, 가장자리 등 기본적인 형태와 패턴을 나타내는 아이콘들이 보여지며, 실제 CNN에서는 이 계층들이 이미지에서 수직선, 수평선 등의 기본적인 형태를 감지합니다.

CNN 이해

신경망의 높은 계층이 낮은 계층의 출력에 기반한 가중치가 적용된 합으로 형성됨

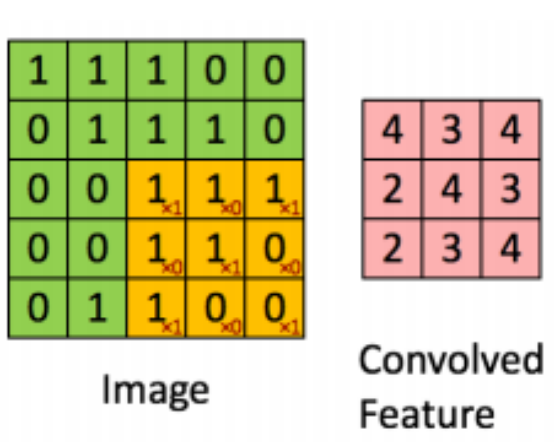
각 높은 계층의 뉴런은 낮은 계층의 특징들을 받아 이들의 가중합을 계산하고, 이를 바탕으로 더 복잡한 특징을 인식

- Transfer Learning : 이미 대규모 데이터셋에서 사전 훈련된 모델의 지식을 새로운 문제에 적용하는 방법론으로 이때 낮은 계층의 일반적인 특징은 유지되고, 높은 계층의 특징은 새로운 문제에 맞게 조정
- Capsule Network : 개별 뉴런 대신 "캡슐"이라 불리는 뉴런의 그룹을 사용하여 이미지 내 객체의 다양한 속성과 공간적인 관계를 보다 효과적으로 인식하는 신경망 구조

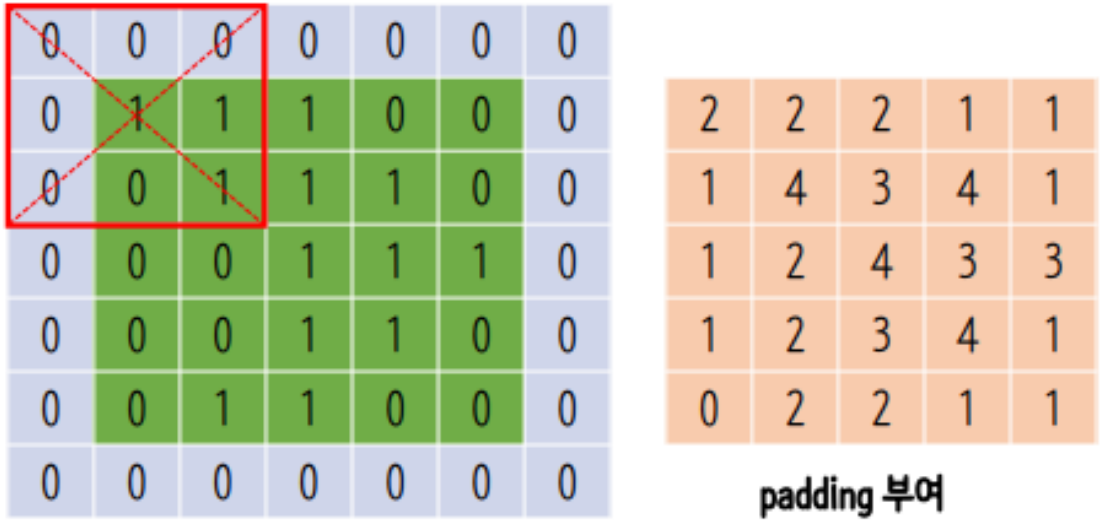


CNN 이해 - Padding

- Convolution Layer에서 Filter를 사용하여 Feature Map을 생성할 때, 이미지 크기가 작아지는 것을 막기 위해 테두리에 Filter 크기를 고려하여 특정 값(일반적으로 0)으로 채우는 작업
- 5X5 이미지에서 3X3 Filter를 사용하면 3X3 크기의 Feature Map이 만들어짐
  즉, 5X5 이미지의 둘레에 0을 채워 7X7의 이미지로 만들어서 3X3 Filter를 적용해 5X5 의 Feature Map을 얻음
- Padding 작업을 통해 인공신경망 이미지 외곽을 인식하도록 하는 효과도 있음 (필수 작업은 아님)



padding 없음

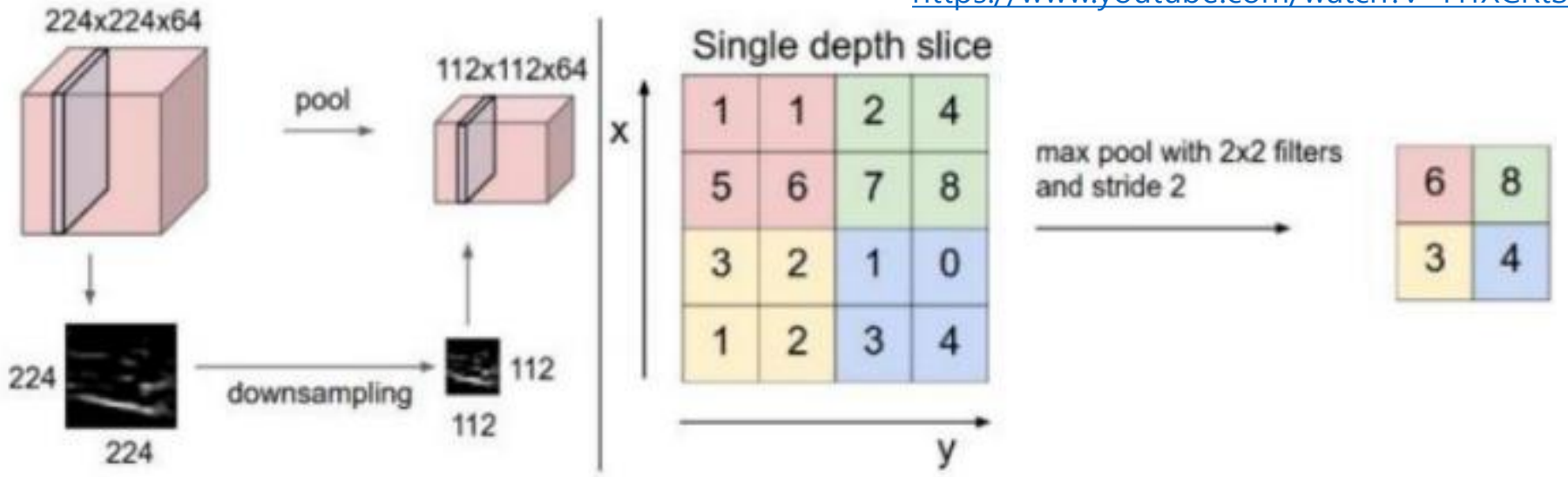


padding 부여

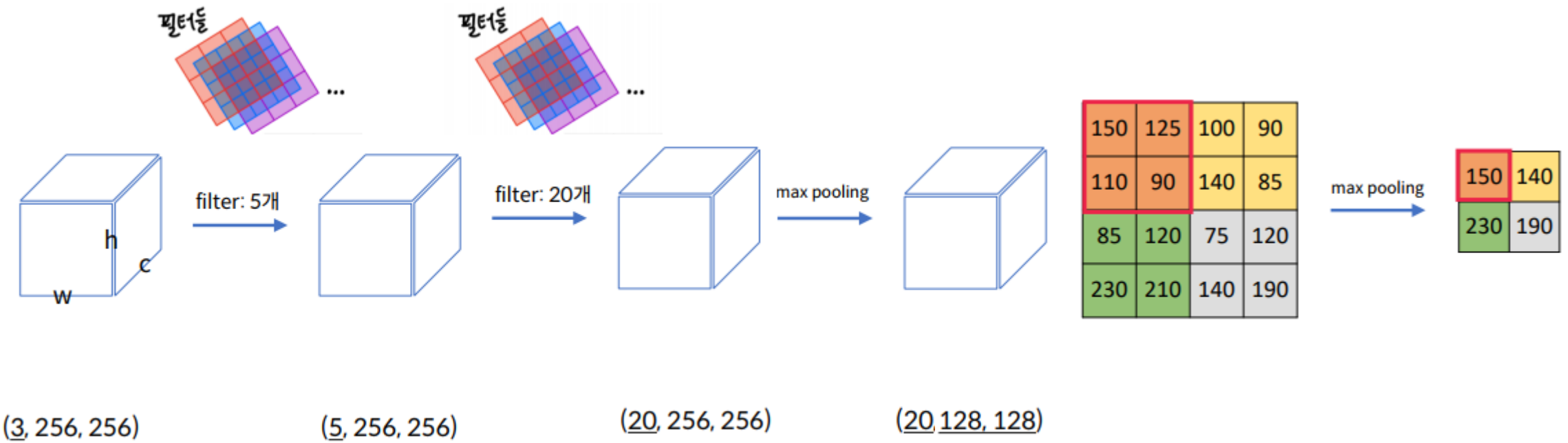
CNN 이해 - Pooling

- Convolution Layer의 Output을 Input으로 받아 Feature Map의 크기를 줄이거나 특정 데이터를 강조하는 용도로 사용
- Max Pooling, Min Pooling, Average Pooling 등의 종류가 있음
- Pooling Size를 Stride로 지정하며, 이 크기에 따라 줄어드는 양이 줄어듬
- 입력 데이터의 행, 열 크기는 Pooling 사이즈의 배수(나누어 떨어지는 수) 이어야 함

https://www.youtube.com/watch?v=U1KiC0AXhHg
https://www.youtube.com/watch?v=f1fXCRtSUWU

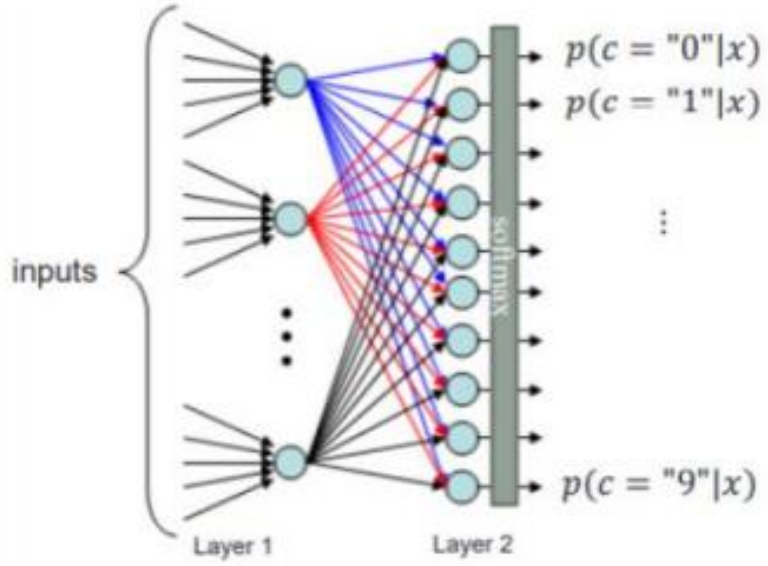


CNN 이해 – Filter & Pooling



CNN 이해 - Fully Connected Layer

- Flatten Layer : CNN의 데이터를 Fully Connected Neural Network의 형태로 변경하는 Layer
  - 입력 데이터의 Shape 변경만 수행
  - 입력 Shape이 (8, 8, 10)이면 Flatten이 적용된 Shape 은 (640, 1)이 됨
- Softmax Layer : Flatten Layer의 출력을 입력으로 사용하며, 분류 클래스에 매칭 시키는 Layer
  - 분류 작업을 실행해 결과를 얻게됨
  - 입력 Shape이 (640, 1)이고 분류 클래스가 10인 경우 Softmax가 적용된 출력 Shape은 (10,1)이 됨. 이때 weight의 shape은 (10, 640)이며 Softmax Layer의 paramete가 6,400개임



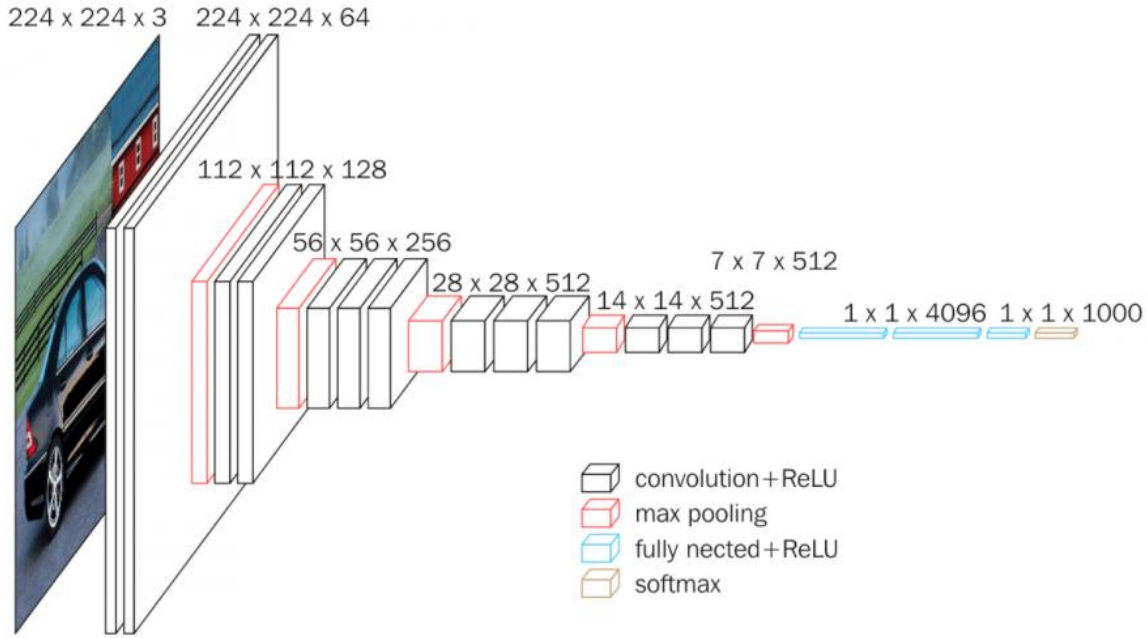



PART2. 전이학습

Enjoy your possibility

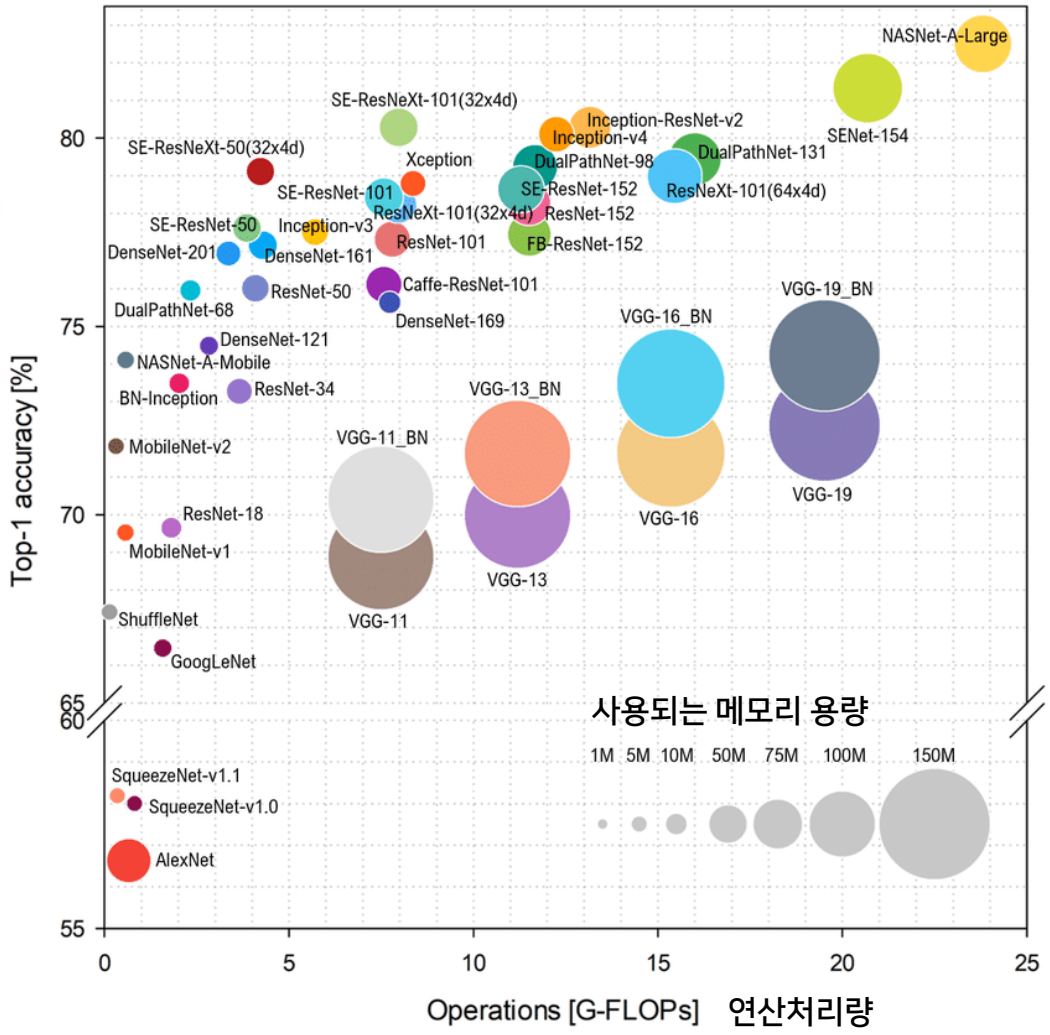
다양한 CNN 네트워크 알아보기

다양한 CNN 네트워크



VGG16 Architecture

출처 : https://neurohive.io/en/popular-networks/vgg16/



출처 : https://www.researchgate.net/figure/Ball-chart-reporting-the-Top-1-and-Top-5-accuracy-vs-computational-complexity-Top-1-and_fig1_328509150

전이학습의 필요성

- 딥러닝 모델 학습의 계산 자원과 시간
  - MNIST 흑백 이미지 인식 모델 → 최소 3개의 convolution layer + 1개의 fully connected layer 필요
  - CPU 환경에 따라 수 분~수십 분 소요됨
  - CIFAR-10 고해상도 컬러 이미지 → 최소 5개의 convolution layer + 2개의 fully connected layer 필요
  - 일반 CPU에서 수 시간~수십 시간 소요됨
  - 최신 대규모 CNN 아키텍처(ResNet-152, EfficientNet-B7 등) → 수백 GPU 시간 + 대량의 데이터 필요함

- 전이학습의 해결책
  - Pre-trained CNN 모델 활용 → 분석 데이터에 맞게 Fine-tuning
  - 학습 시간 획기적 단축됨
  - 데이터 부족 상황에서도 우수한 성능 확보 가능함

동작 원리

- 다른 도메인 간의 전이
  - 동물 사진으로 학습한 모델 → 자동차/오토바이 분류에 활용 가능함
  - 도메인 간 거리가 멀수록 효과는 감소함
  - Fine-tuning 통해 우수한 성능 확보 가능함

- 딥러닝 모델의 층별 특성
  - 초기 층 → "일반적인(general)" 특징 추출함 (에지, 색상, 텍스처 등)
  - 마지막 층 → "구체적인(specific)" 특징 추출함 (얼굴 특징, 특정 객체 부분 등)
- 시각적 특징 추출 과정
  - 초기 층 → 수평선, 수직선, 에지, 단순 텍스처 감지함
  - 중간 층 → 복잡한 패턴, 질감, 단순 형태 감지함
  - 마지막 층 → 고수준 특징(얼굴, 바퀴, 눈 등) 식별함
- 재사용 가능성
  - 초기 층 → 다른 데이터셋에도 재사용 가능함
  - 마지막 층 → 새로운 문제마다 새로 학습 필요함

전이학습 방법론

- 전이학습의 기본 단계
  - 사전 훈련된(Pre-trained) 모델 가져옴
  - 분류 헤드(Classification Head) 제거함
  - 새로운 분류 헤드 추가함
  - 사전 훈련된 모델 가중치 동결(freeze)하고 새 층만 학습시킴
  - 필요시 모델 상위 몇 개 층 함께 미세 조정(fine-tuning)함

- 적용 방식
  - 특징 추출(Feature Extraction) → 사전 훈련 모델 완전 동결 + 새 분류 헤드만 학습함
  - 미세 조정(Fine-tuning) → 사전 훈련 모델 일부(주로 상위 층) + 새 분류 헤드 함께 학습함

전이학습의 개념과 중요성

- 전이학습 정의
  - 한 도메인 데이터로 학습한 모델 → 다른 도메인에 적용하여 성능 향상시키는 방법임
- 핵심 개념
  - 대규모 데이터셋 훈련된 사전 훈련 모델 활용함
  - 새 작업에 필요한 데이터셋 크기와 훈련 시간 대폭 감소시킴
  - 라벨링된 데이터 획득 어려운 경우 특히 유용함
- 중요성
  - 적은 양의 데이터로도 우수한 성능 달성 가능함
  - 모델 훈련 시간 단축 + 컴퓨팅 자원 절약됨
  - 다양한 작업/도메인에 유연하게 적용 가능함
- 산업 적용 사례
  - 의료 영상 분석 → 제한적 의료 영상 데이터에 ImageNet 사전 훈련 모델 적용함 (X-ray, MRI 진단)
  - 자율주행 차량 → 객체 인식 + 도로 상황 분석에 활용함
  - 자연어 처리 → BERT, GPT 등 사전 훈련 모델을 특정 도메인에 맞게 조정함

사전 훈련된 모델을 사용한 전이 학습 방법

- 전이 학습 방법
  - 특징 추출(Feature Extraction)
    - 사전 훈련 모델의 마지막 분류 계층 제외한 네트워크 재사용함
    - 새 데이터셋에 대한 특징 추출에 활용함
    - 작은 데이터셋(< 1,000 샘플)에 적합함
    - 새 작업이 원본 작업과 유사할 때 효과적임
  - 미세 조정(Fine-tuning)
    - 네트워크의 상위 계층 일부를 새 데이터셋에 맞게 재훈련함
    - 더 큰 데이터셋(> 1,000 샘플)에 적합함
    - 새 작업이 원본 작업과 다소 다를 때 유용함
    - 더 맞춤화된 특징 학습 가능함

- 사전 훈련된 모델 유형

모델	발표 연도	깊이	파라미터 수	특징
VGG16/19	2014	16-19층	138-144M	간단한 구조, 메모리 요구량 높음
ResNet	2015	18-152층	11-60M	잔차 연결, 깊은 네트워크 훈련 가능
MobileNet	2017	28층	4.2M	경량화, 모바일 기기에 적합
EfficientNet	2019	다양함	5-66M	효율적인 확장 방법, 성능/효율성 균형

전이학습을 활용한 이미지 분류 실습

- 구현 단계
  - 사전 훈련된 모델 로드함
  - 기존 분류 헤드 제거 + 새 레이어 추가함
  - 기본 모델 동결함
  - 모델 컴파일 + 훈련함
  - 필요시 미세 조정 진행함

- 데이터셋 크기에 따른 전략
  - 작은 데이터셋(<1,000 샘플) → 특징 추출만 사용함
  - 중간 크기 데이터셋(1,000-10,000 샘플) → 상위 몇 개 층만 미세 조정함
  - 큰 데이터셋(>10,000 샘플) → 더 많은 층 미세 조정 또는 처음부터 훈련 고려함

전이학습의 한계와 주의사항

- 도메인 차이
  - 소스 도메인과 타겟 도메인 간 차이 클수록 전이 효과 감소함
  - 필요시 도메인 적응(Domain Adaptation) 기법 고려해야 함

- 과적합(Overfitting) 위험
  - 작은 데이터셋에서 미세 조정 시 과적합 가능성 증가함
  - 드롭아웃, 배치 정규화, 데이터 증강 등으로 방지해야 함

- 계산 리소스
  - 대규모 모델 사용 시 메모리 요구사항 증가함
  - 모바일/임베디드 환경에서는 경량화된 모델(MobileNet 등) 고려해야 함