컴퓨터비전(AI응용)

CNN 기반 분류 앱

00

수업목표

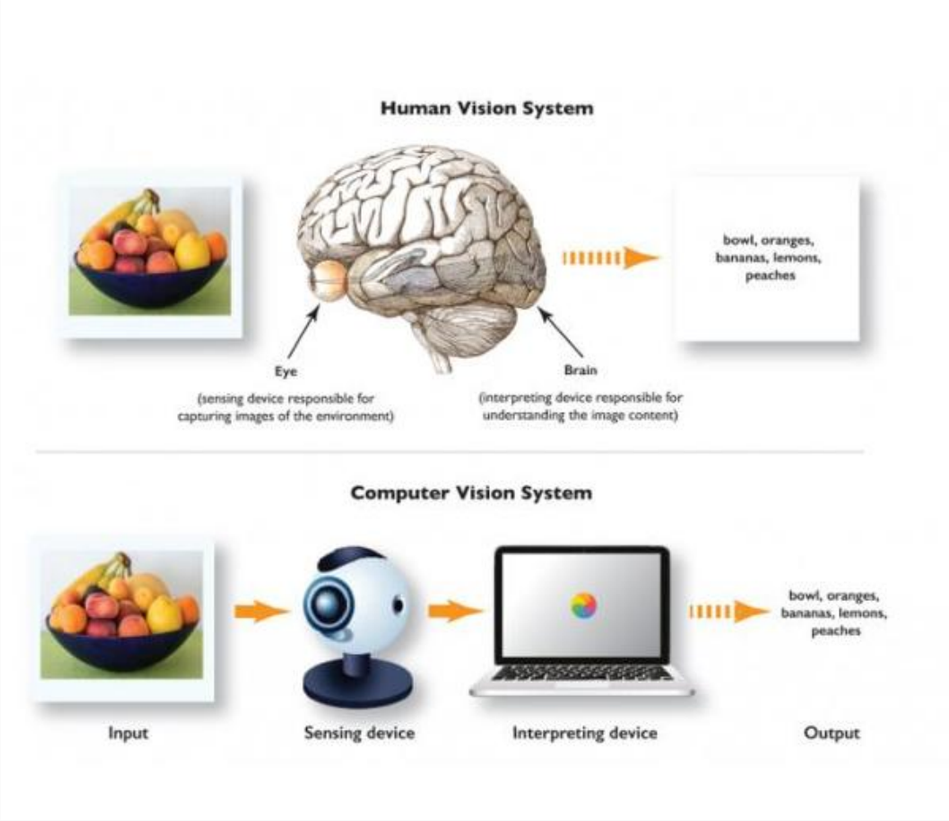
CNN의 이해
CNN의 특징과 장점을 명확히 이해하고 설명할 수 있다.

MobileNet 구조 분석
MobileNet의 경량화 구조와 동작 원리를 파악한다.

Custom Dataset 활용
Custom Dataset을 활용하여 나만의 이미지 분류 모델을 직접 학습시킬 수 있다.

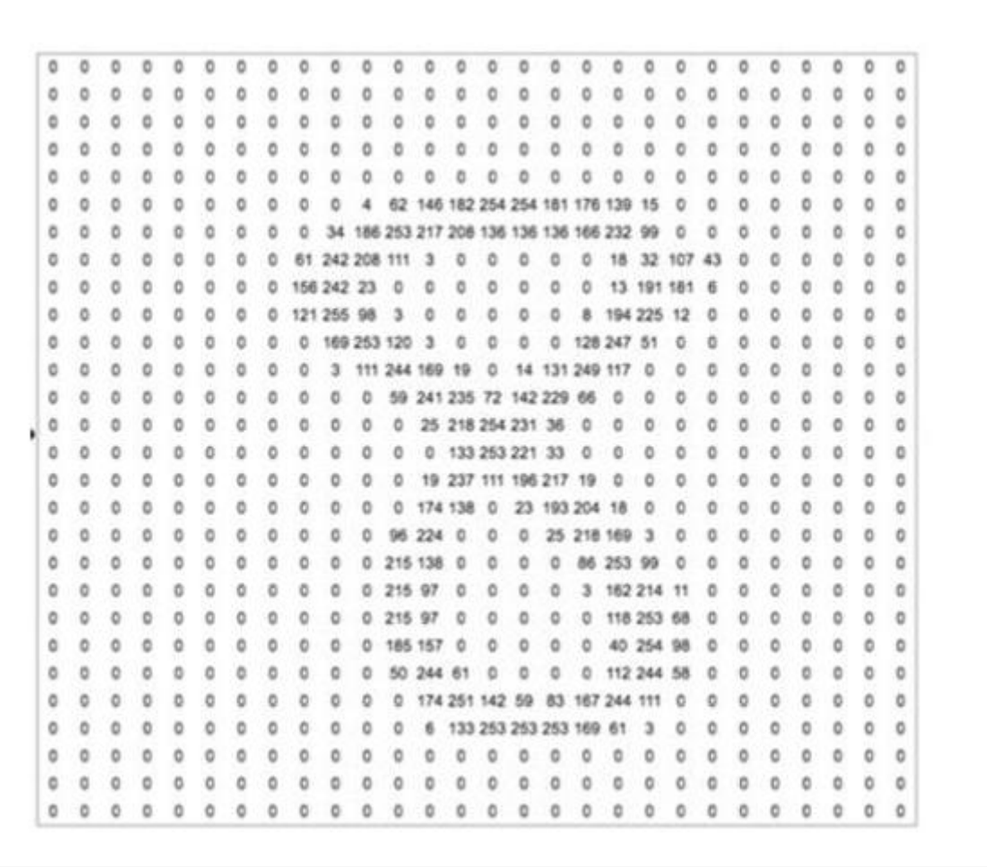
Transfer Learning 적용
전이 학습(Transfer Learning)을 적용하여 사전 학습된 모델을 새로운 데이터에 최적화한다.

01 컴퓨터비전이란



- 컴퓨터 비전은 시각적 데이터(이미지나 동영상)를 처리 및 분석하고 이해할 수 있는 디지털 시스템을 만드는 데 중점을 둔 컴퓨터 과학 분야
- 컴퓨터 비전의 개념은 컴퓨터가 이미지의 픽셀 단위로 처리하고 이해하도록 가르치는 것에 기반합니다. (인간이 하는 방식과 동일하게)

02 이미지 데이터 특성



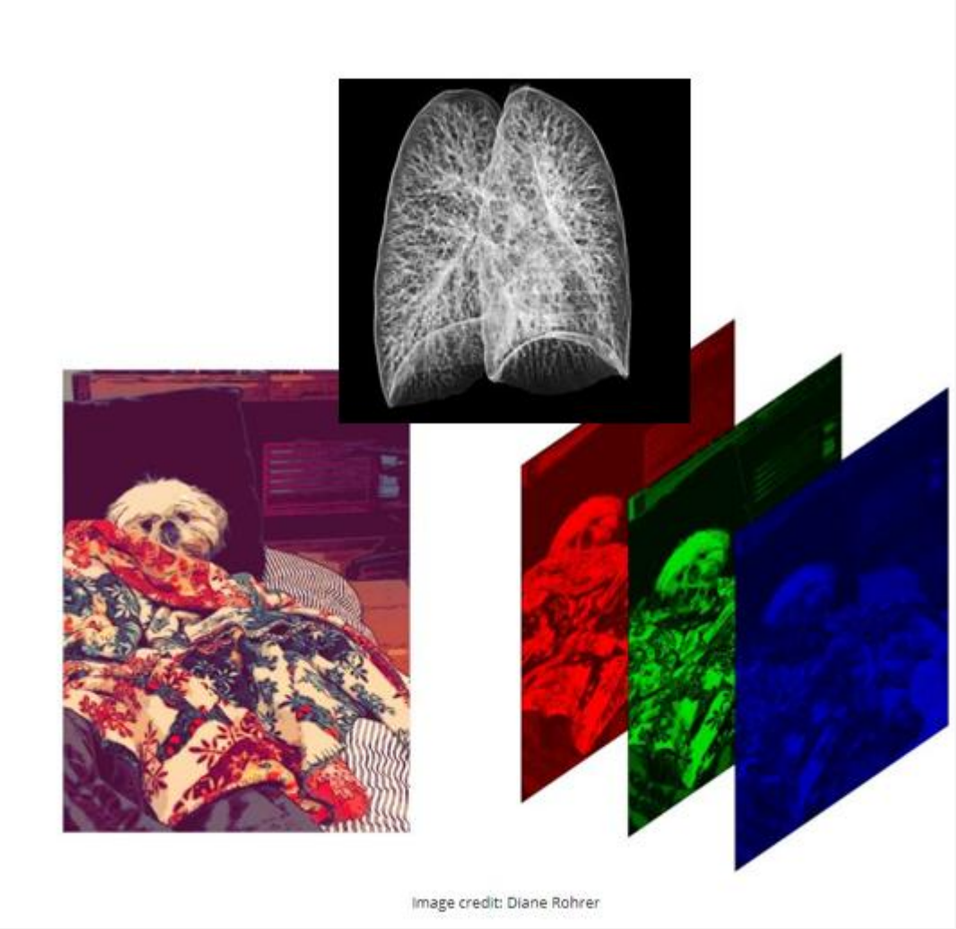
Matrix

- 숫자로 이루어진 2차원 배열
- 행(Row)과 열(Column)로 이미지의 크기가 결정되는 직사각형 형태
- Matrix 크기가 클수록, 필요 저장공간과 처리하는데 소요되는 리소스가 비례하여 증가

Pixels

- Matrix를 이루고 있는 숫자 데이터
- 각 숫자 (discrete value)값들은 밝기를 나타냄
- Matrix 크기가 클수록, spartial resultion 이 좋음

02 이미지 데이터 특성



Bit depth

- 픽셀이 갖는 밝기 값의 범위
- 범위가 클수록 이미지가 표현하는 밝기가 다양해지며, 필요로 하는 저장공간은 커짐

Channel

- 각 픽셀이 갖고 있는 숫자로 이루어진 벡터
- RGB(red, green, blue)에서 red channel의 의미? 이미지 데이터에서 red 밝기에 대한 값 들만 모아놓은 것

03

CNN의 특징

Why CNN is Powerful

가중치 공유

필터(커널)의 가중치를 전체 이미지에 동일하게 적용하여
학습해야 할 매개변수(Parameter)를 효율적으로 줄임

자동 특징 추출

특징 추출(Feature Extraction) 단계를 여러 개 배치하여
CNN이 데이터로부터 특징 표현을 스스로 학습하게 함

계층적 학습

다층 구조를 통해 **저수준(점, 선)부터 고수준(형태, 객체) 특징까지 단계적으로 학습**하여 복잡한 패턴 인식 가능

시공간 상관관계 탐색

이미지 내 인접 픽셀 간의 공간적 관계나
비디오 데이터의 시간적 상관관계를 효과적으로 탐색하고 활용 가능

Convolution

Convolutional Kernel

Image Classifier

If positive, “\”

If negative, “/”

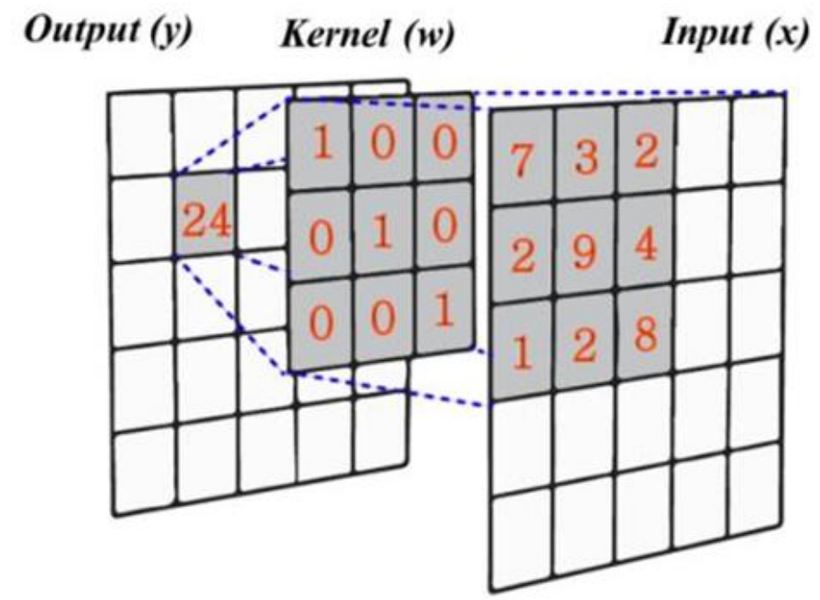
Convolution

- 필터(또는 커널)와 입력의 요소들 사이의 내적 연산 (dot product)
- 2D데이터(이미지)의 공간적 특성을 반영할 수 있음

$$y[i,j] = (x * w)[i,j]$$

$$= \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} x[m-i, n-j] w[m,n]$$

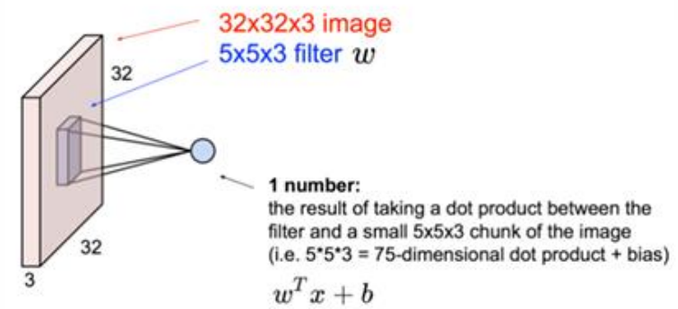
Cross-Correlation in 2D

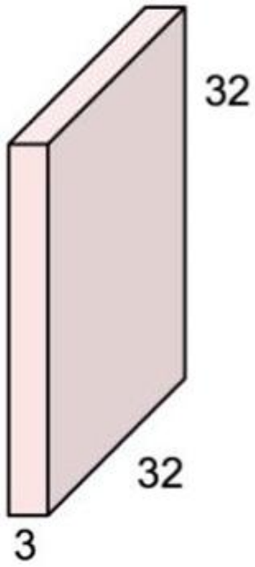


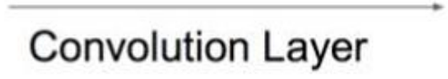
04

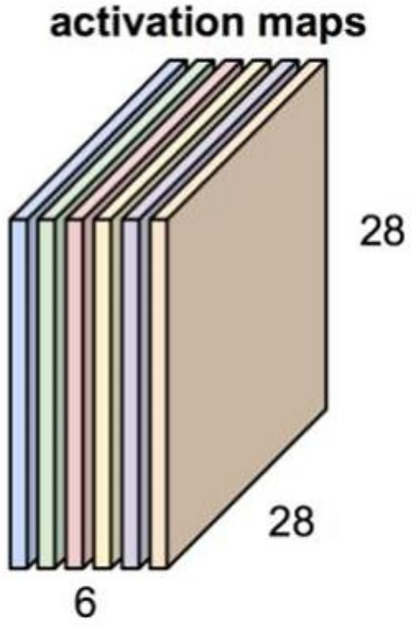
Convolution

- 직관적으로 설명하면, 입력의 특정 위치의 특정 패턴에 대해 반응하는 (activate) 필터를 학습하는 것
- 이런 Activation map을 depth 차원을 따라 쌓은 것이 곧 출력





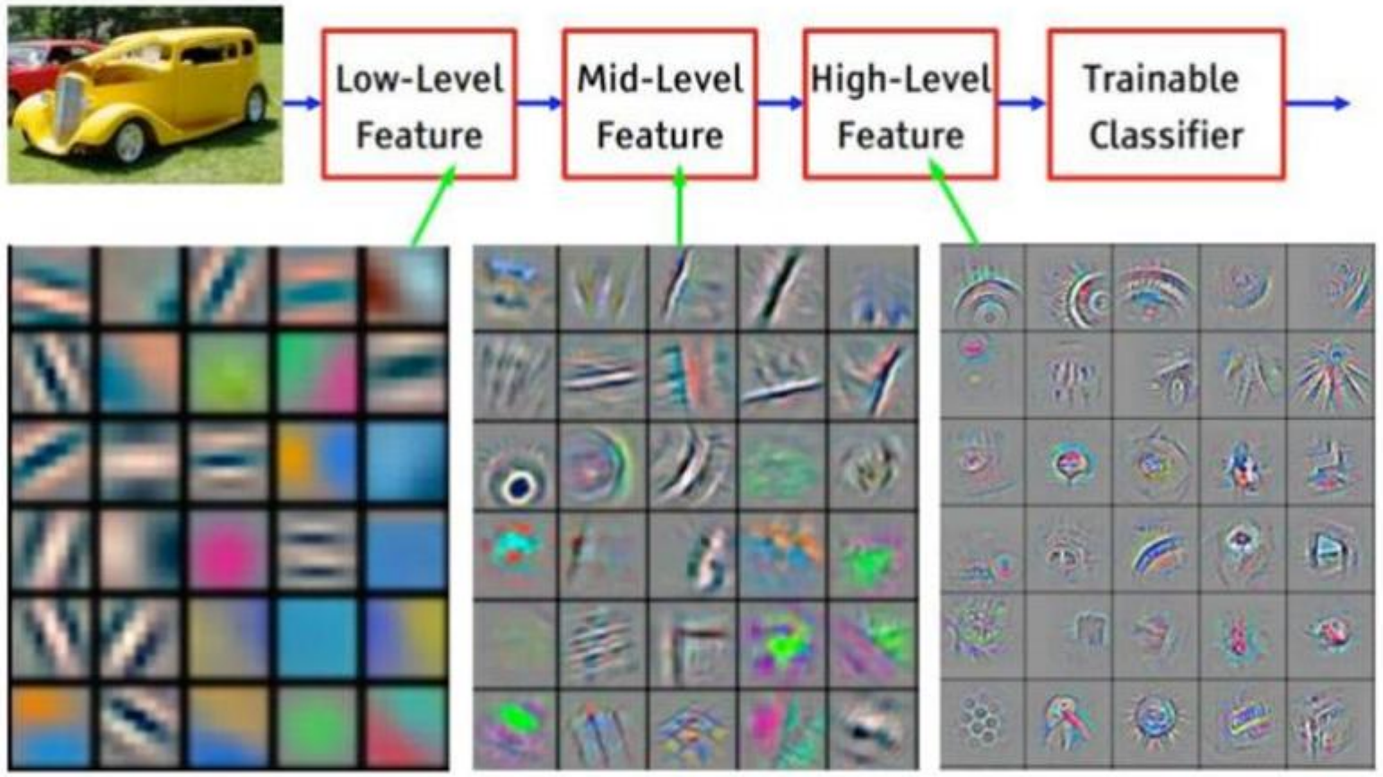




04

Convolution

- Convolution을 거치면서 이미지는 입력 이미지의 특징(feature)를 찾아내고 이 특징들을 종합하여 최종적으로 입력된 이미지가 무엇인지 알아내는 등의 일을 수행하는 것

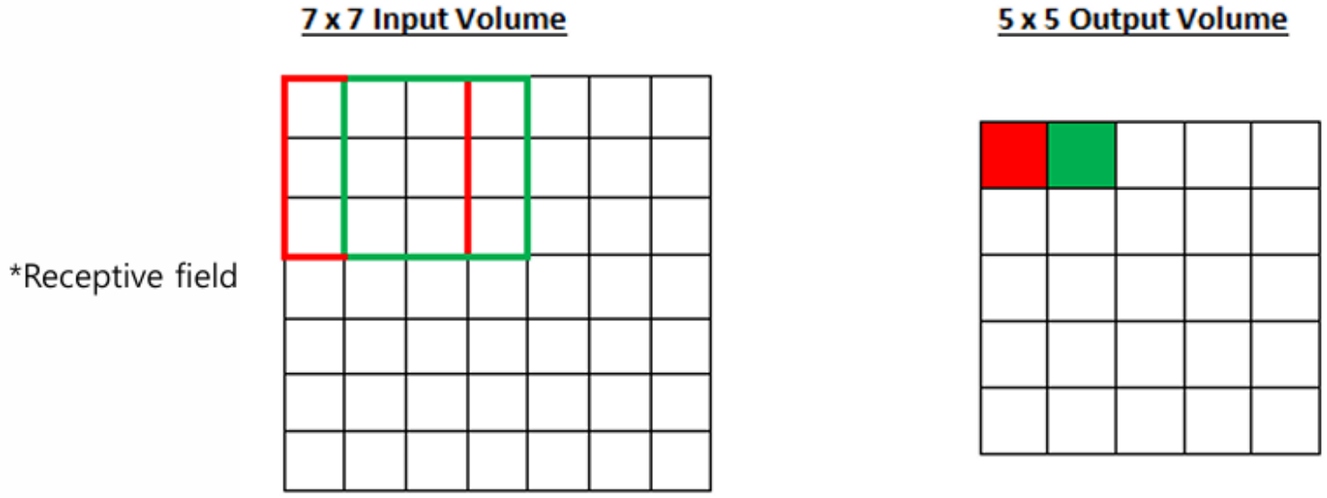


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Convolution

Stride

- 어떤 간격 (가로/세로의 공간적 간격) 으로 컬럼을 할당할 지를 의미
- 만약 stride가 1이라면, 컬럼을 1칸마다 할당
- 이럴 경우 각 깊이 컬럼들은 receptive field 상 넓은 영역이 겹치게 되고, 출력 볼륨의 크기도 커짐
- 반대로, 큰 stride를 사용한다면 receptive field끼리 좁은 영역만 겹치게 되고 출력 볼륨도 작아짐

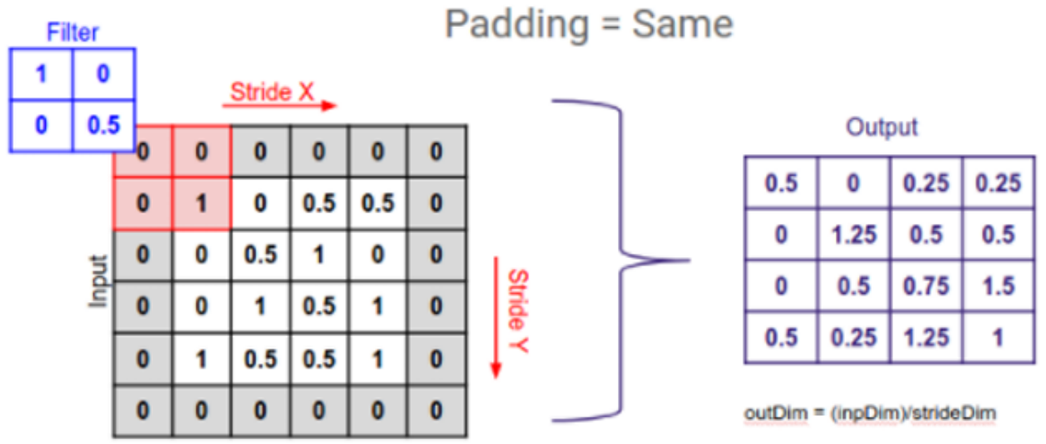


receptive field : 빨간 영역, 녹색 영역

Convolution

Zero-Padding

- 입력 볼륨의 가장자리를 0으로 채우는 것
- zero-padding을 사용할 때의 장점은, 출력 볼륨의 공간적 크기(가로/세로)를 조절할 수 있다는 것



출력 볼륨의 크기 (가로/세로)는 입력 볼륨 크기 (W), CONV 레이어의 receptive field 크기(F)와 stride (S), 그리고 제로 패딩 (zero-padding) 사이즈 (P) 의 함수로 계산할 수 있다.

$$(W-F+2P)/S+1$$

(W = 227, P = 0, F =11, S = 4) => [55 x 55]

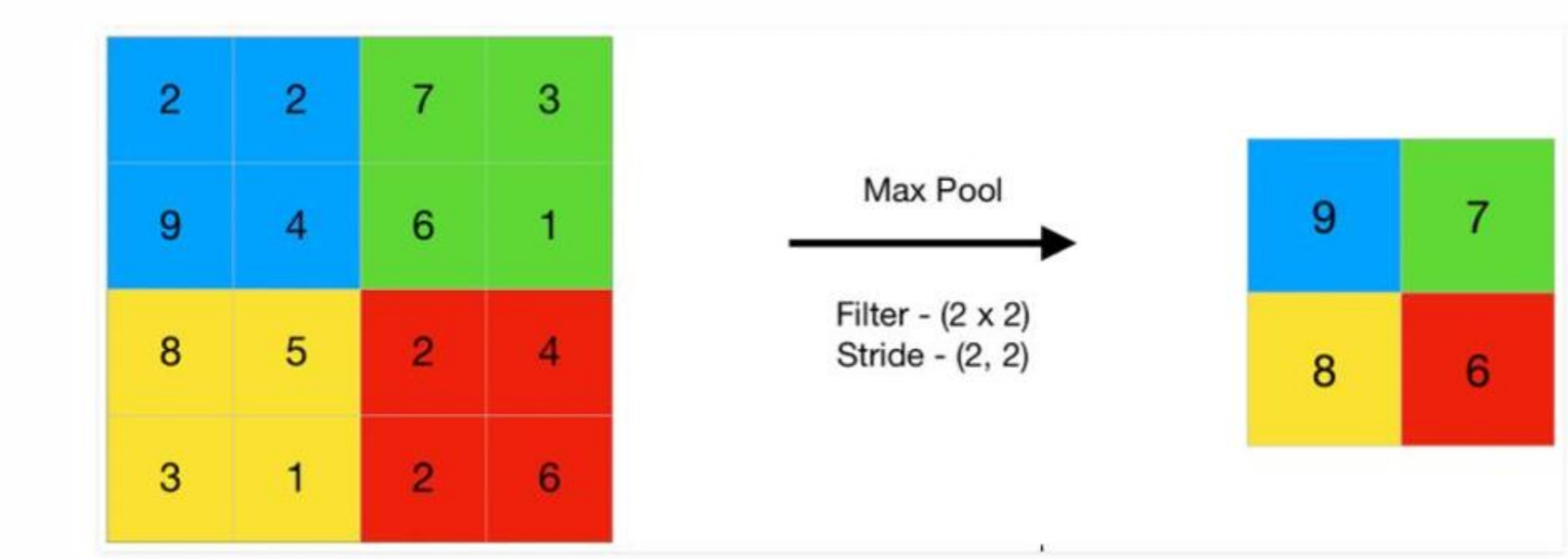
정수가 되도록 조정

05

Pooling

MAX – POOLING

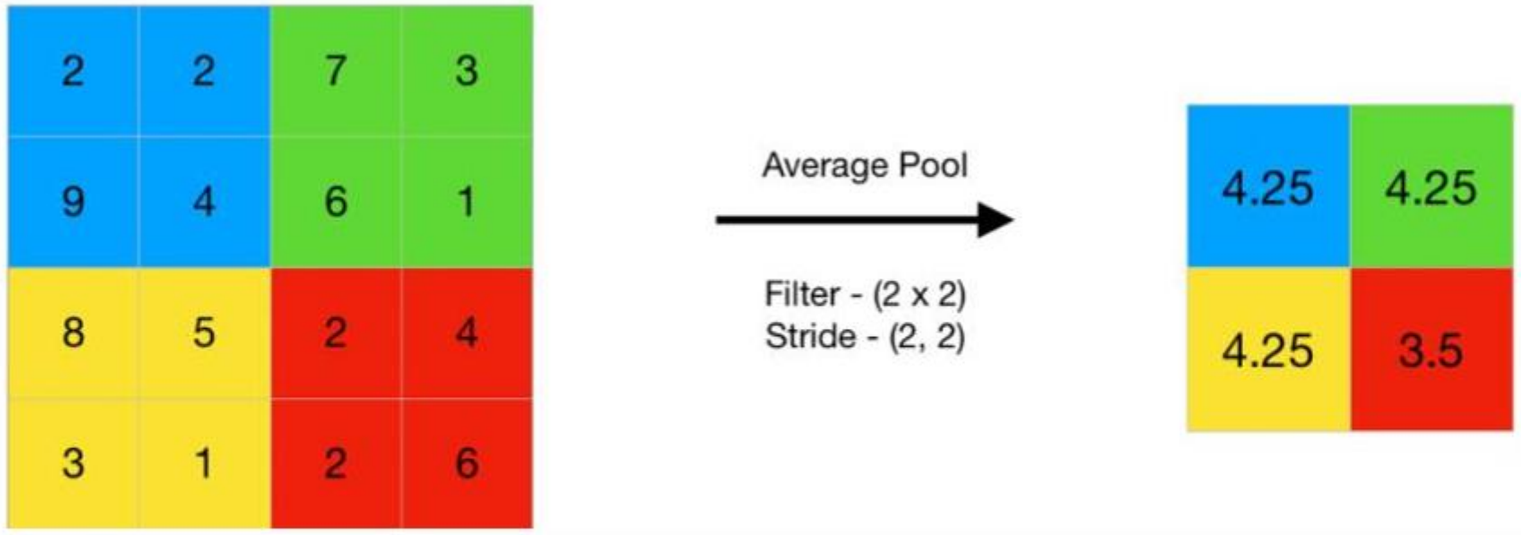
필터가 덮고 있는 Feature Map 영역에서 최대 요소를 선택합니다.
따라서 Max Pooling 층 이후의 출력은 이전 피처 map 의 가장 두드러진 특징을 포함하는 Feature Map이 됩니다. Feature Map 의 특정 패치에서 가장 두드러진 특징을 제공합니다



Pooling

Average – POOLING

필터가 덮고 있는 Feature Map 영역에 존재하는 요소들의 평균을 계산합니다.



06

CNN의 아키텍쳐

Convolutonal Layer

필터(Filter)가 이미지 위를 지나가며 한 번에 몇 픽셀(pixel)씩 스캔하고 각 특징이 속하는 클래스를 예측하는 특징 맵(Feature Map)을 만듭니다

Pooling Layer

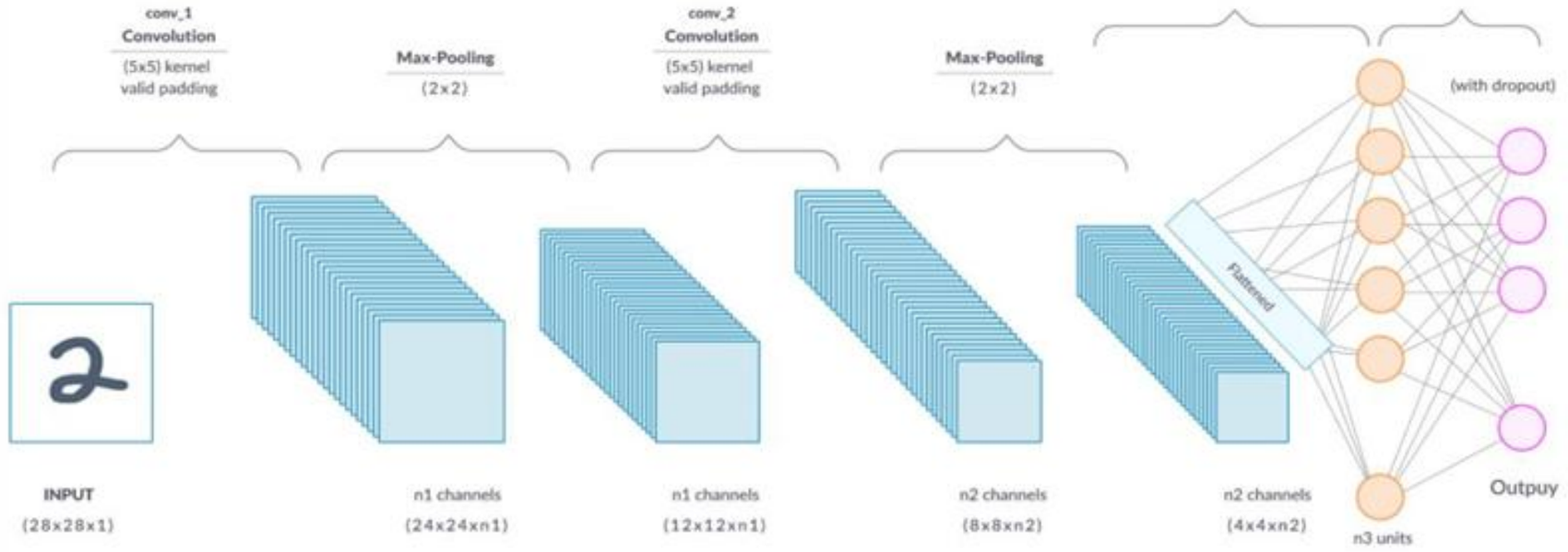
합성곱 계층에서 얻은 각 특징(Feature) 의 정보를 줄이면서 가장 중요한 정보를 유지합니다

Fully Connected Layer (FC)

완전 연결 계층 / 이전 계층의 출력을 받아 '평탄화'(flatten) 한 후 단일 벡터로 변환합니다

06

CNN의 아키텍쳐


conv_1
Convolution
(5x5) kernel
valid padding
Max-Pooling
(2x2)
conv_2
Convolution
(5x5) kernel
valid padding
Max-Pooling
(2x2)
(with dropout)
Flattened
INPUT
(28x28x1)
n1 channels
(24x24xn1)
n1 channels
(12x12xn1)
n2 channels
(8x8xn2)
n2 channels
(4x4xn2)
n3 units
Outpuy

07

소프트맥스(Softmax)

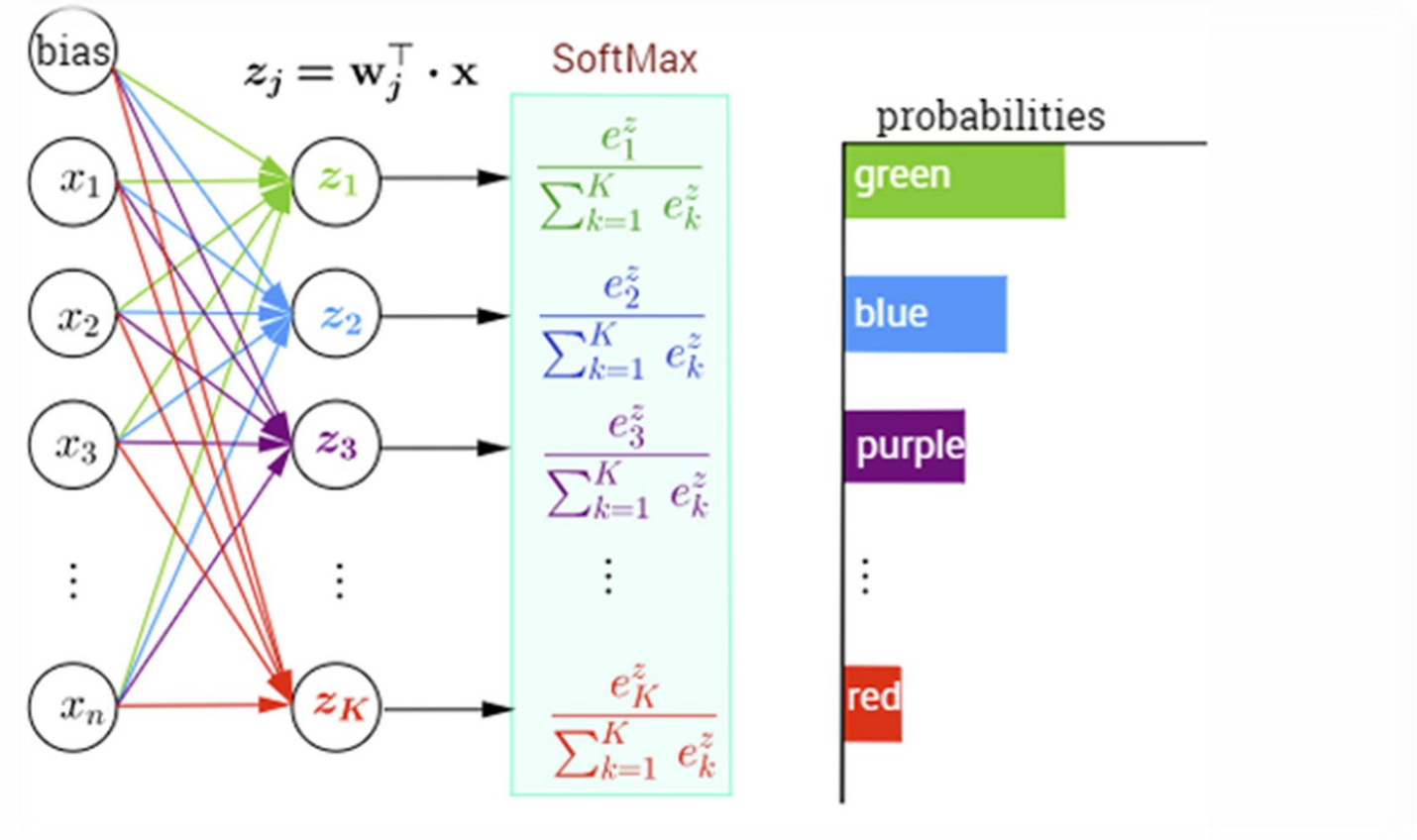
- 마지막 은닉층은 벡터를 형성하는 출력 값을 생성하며, 출력 신경층은 K개 중에서 분류하도록 설계됨
- 소프트맥스는 R의 벡터를 K 요소로 매핑함
- 소프트맥스의 속성(모든 출력 값이 (0, 1) 범위에 있으며
- 합이 1.0)은 기계 학습에서 매우 유용한 확률적 해석에 적합함
- 소프트맥스 정규화는 데이터 세트에서
  데이터 포인트를 제거하지 않고,
  극단 값이나 이상치의 영향을 줄이는 방법임

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_K \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^\top \\ \mathbf{w}_2^\top \\ \mathbf{w}_3^\top \\ \vdots \\ \mathbf{w}_K^\top \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

$$\sigma(j) = \frac{\exp(\mathbf{w}_j^\top \mathbf{x})}{\sum_{k=1}^{K} \exp(\mathbf{w}_k^\top \mathbf{x})} = \frac{\exp(z_j)}{\sum_{k=1}^{K} \exp(z_k)}$$

07

소프트맥스(Softmax)



08

CNN의 아키텍쳐 진화

1998년부터 현재까지, 더 깊고 강력한 신경망을 향한 여정

1998~2012

연구 공백기

큰 데이터셋의 부재로
심층 CNN 기능 입증에 어려움.

Early Era

2012

AlexNet 등장

ImageNet 기반 성능 입증.
모듈식 합성곱 계층 설계 유행의 시작.

ReLU | Dropout

2014

VGG & Inception

'매우 깊은 CNN'의 발명.

VGG: 깊이에 집중
Google LeNet: 모듈화

2015

ResNet

Skip-connection 도입으로
기울기 소실 문제 해결,
심층 네트워크 가능.

2017~

DenseNet

모든 계층을 서로 연결하여 정보 흐름
개선 및 파라미터 효율성 극대화.

Major Milestone
Research Gap

09

LeNet-5

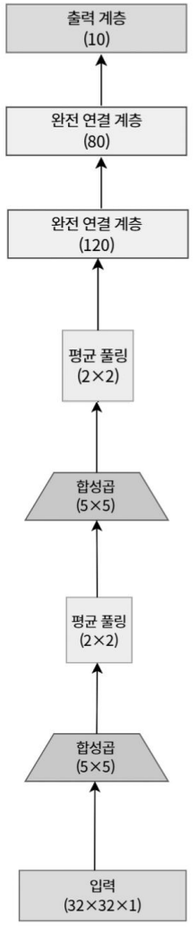


그림 3.6 LeNet 아키텍처

모델 구조

2개의 합성곱(CNN) 계층과 3개의 완전 연결(FCN) 계층으로 구성 초기 형태의 CNN 아키텍처입니다.

경량 파라미터

약 6만 개의 파라미터로 구성되어 있어,
현대 모델 대비 매우 가볍고 효율적인 구조를 가집니다.

강건성 (Robustness)

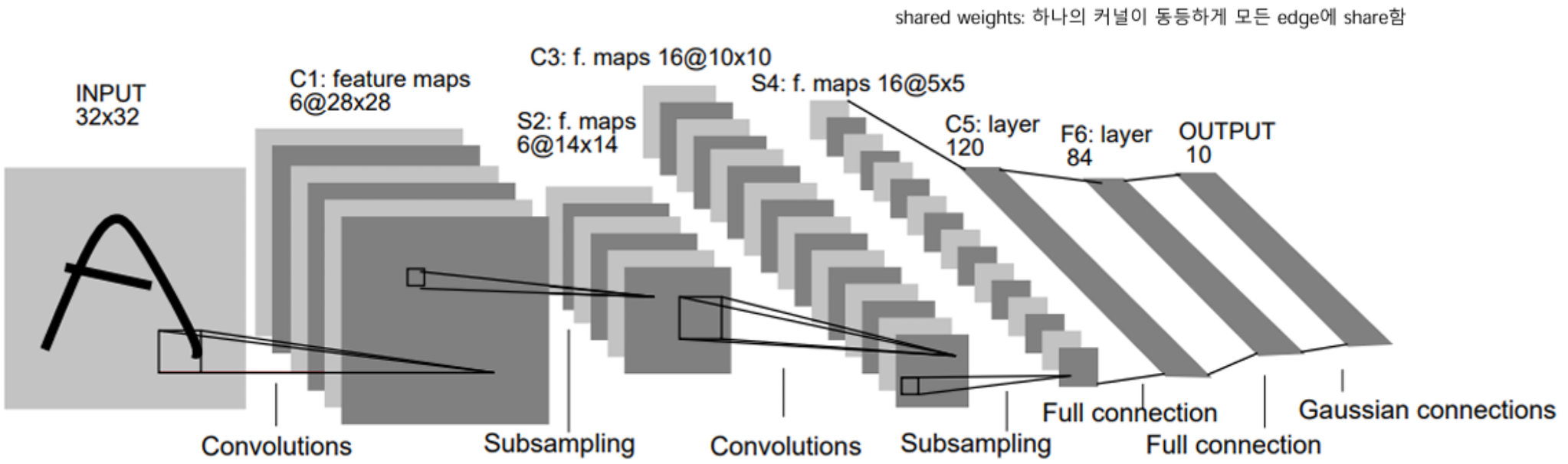
이미지의 회전, 위치 변화, 크기 변화에 대한 불변성을 가지며
왜곡에도 강한 특성을 보입니다.

서브샘플링 (Pooling)

공간 크기를 줄여 연산량을 감소시킵니다.
특이하게 '훈련 가능한 가중치'를 가진 **평균 풀링을 사용**

09

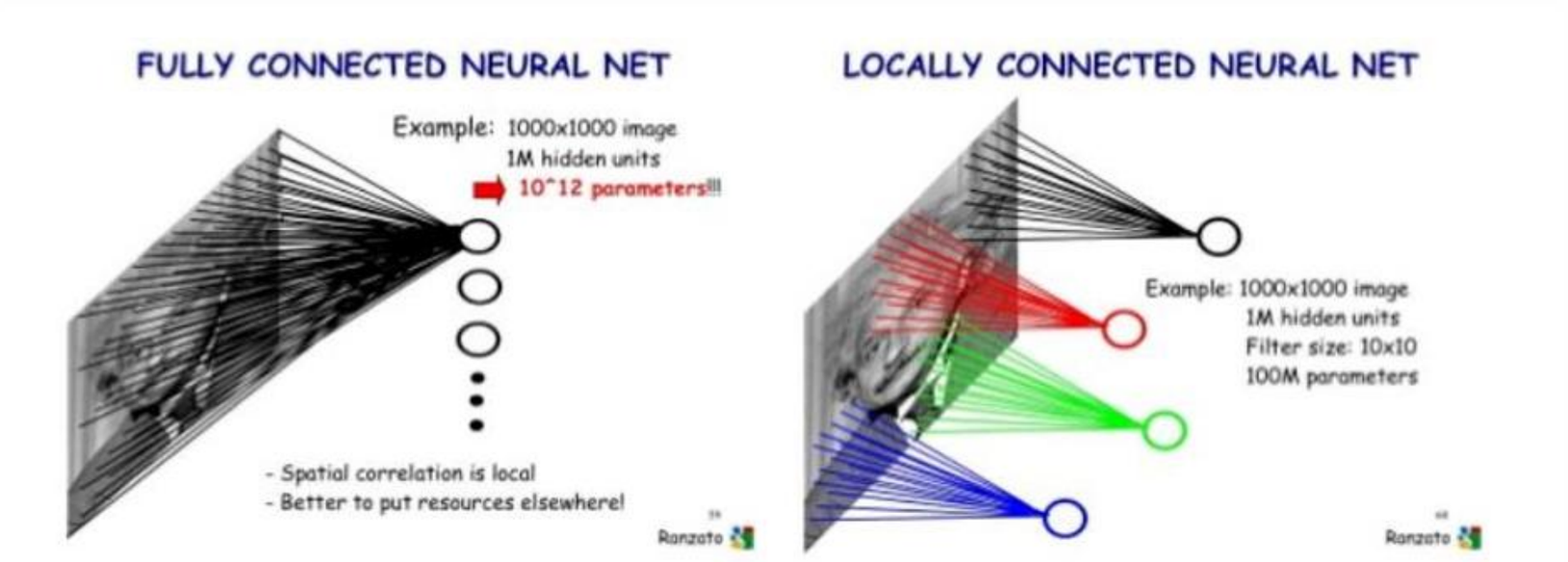
LeNet-5


shared weights: 하나의 커널이 동등하게 모든 edge에 share함
INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

LeNet-5

Local connectivity

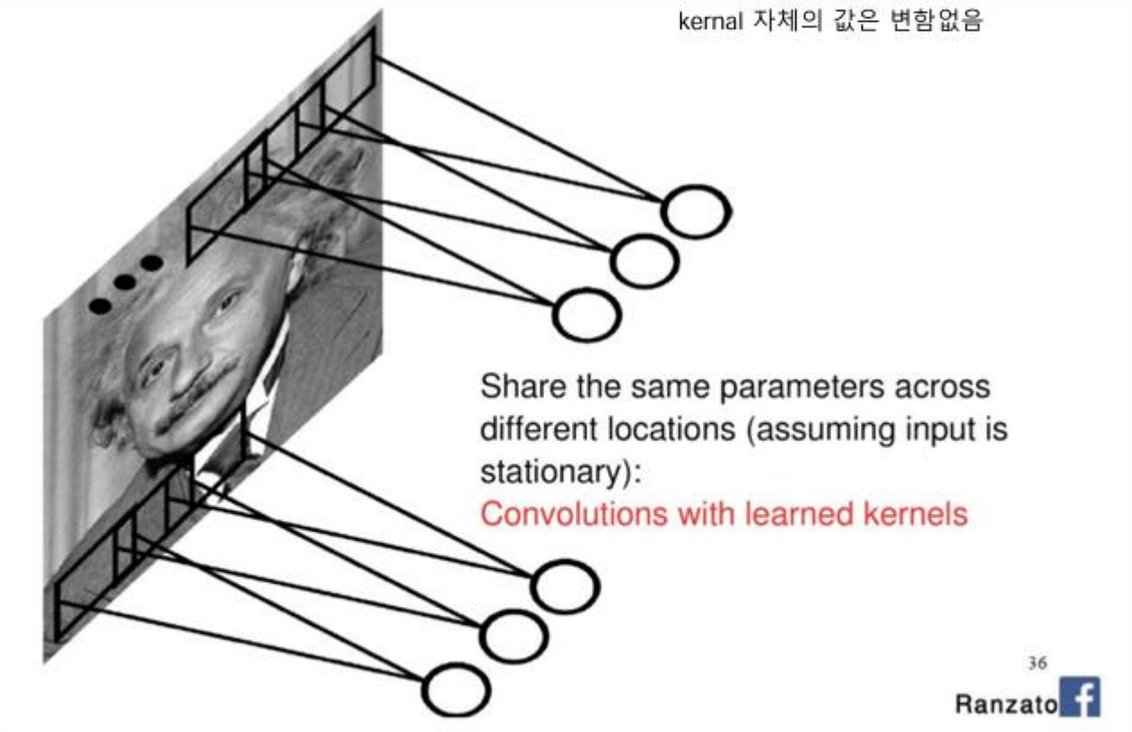
- 이미지와 같은 고차원 입력에서 현재 레이어의 한 노드를 이전 볼륨의 모든 노드들과 연결하는 것은 비효율적
- 대신, 레이어의 각 노드들을 입력 볼륨의 로컬한 영역(local region)에만 연결
- 이 영역을 **receptive field**라고 함



LeNet-5

Shared weights (or weight replication)

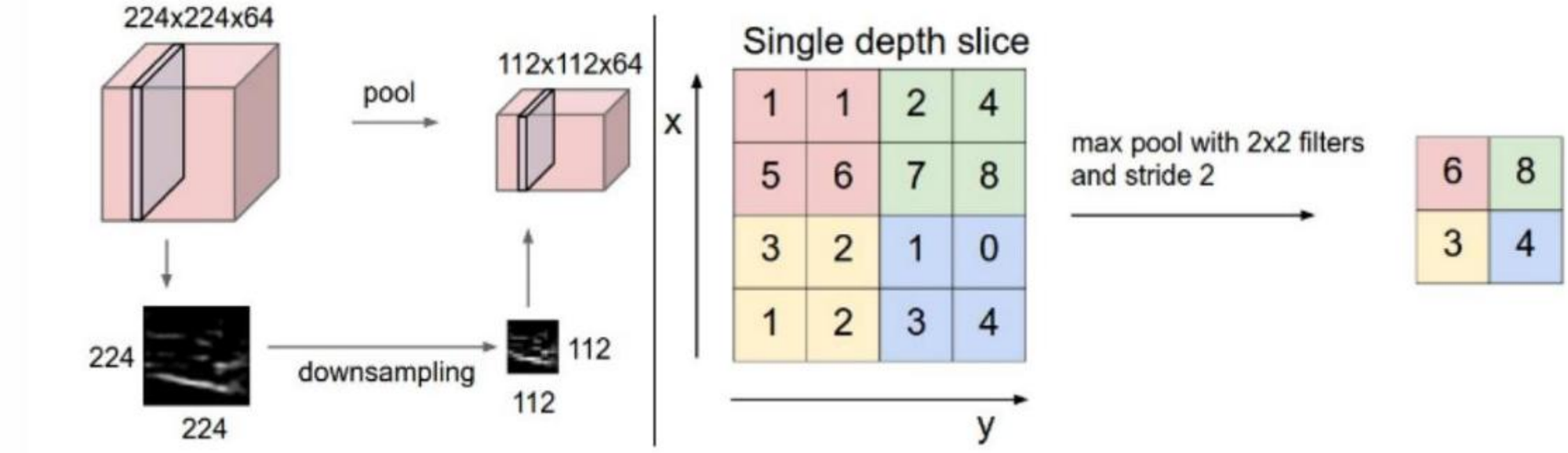
- convolution을 위해 filter를 적용할 때, filter가 적용된 결과(local receptive field)는 계속 변경되지만, 적용하는 filter(weight) 값은 변하지 않는 것을 의미
- 한마디로 동일한 weight가 convolution할 때, 동일하게 적용(shared)되는 것을 의미



LeNet-5

Spatial or temporal sub-sampling (Pooling)

- 추출한 local feature로 부터 입력된 데이터의 translation, distortion에 관계없이 topology(분류체계)에 영향을 받지 않는 global feature를 추출하기 위함



04

AlexNet 미세 조정하기

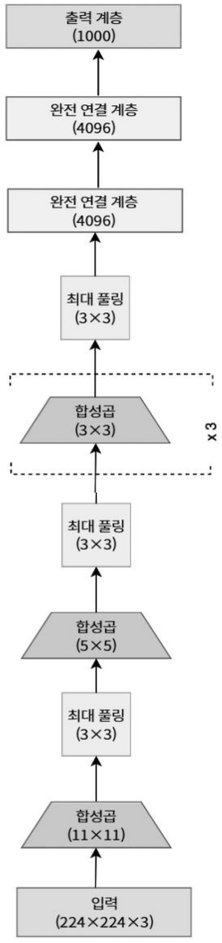


그림 3.9 AlexNet 아키텍처

LeNet의 진화형

LeNet 아키텍처를 기반으로 보완하여 만든 후속 모델로, 딥러닝 붐을 일으킨 핵심 모델입니다.

8개 계층 구조

총 8개의 깊은 계층으로 구성되어 있습니다. **(5개의 합성곱(CNN) 계층 + 3개의 완전 연결(FCN) 계층)**

파라미터 및 풀링

약 6천만 개의 파라미터로 구성되었으며, 평균 풀링 대신 최대 풀링(Max Pooling) 방식을 채택했습니다.

강력한 특징 추출

대규모 ImageNet 데이터셋을 기반으로 훈련되어 이미지의 특징 추출에 더욱 강력한 성능을 보여줍니다.

05

VGGNet 실행하기

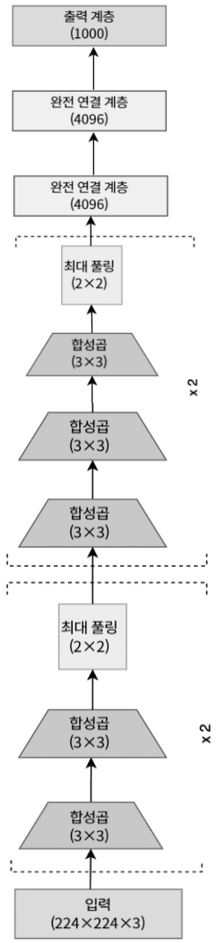


그림 3.12 VGG16 아키텍처

13개 이상의 깊은 계층 구조

기본적으로 10개의 합성곱(Conv) 계층과 3개의 완전연결(FC) 계층 등 총 13개 이상의 층으로 구성되어 있다.

대규모 파라미터

약 1억 3천 8백만 개의 파라미터로 구성되어 있으며, 이를 통해 **이미지의 복잡한 특징을 효과적으로 학습**한다.

'깊이(Depth)'에 집중한 설계

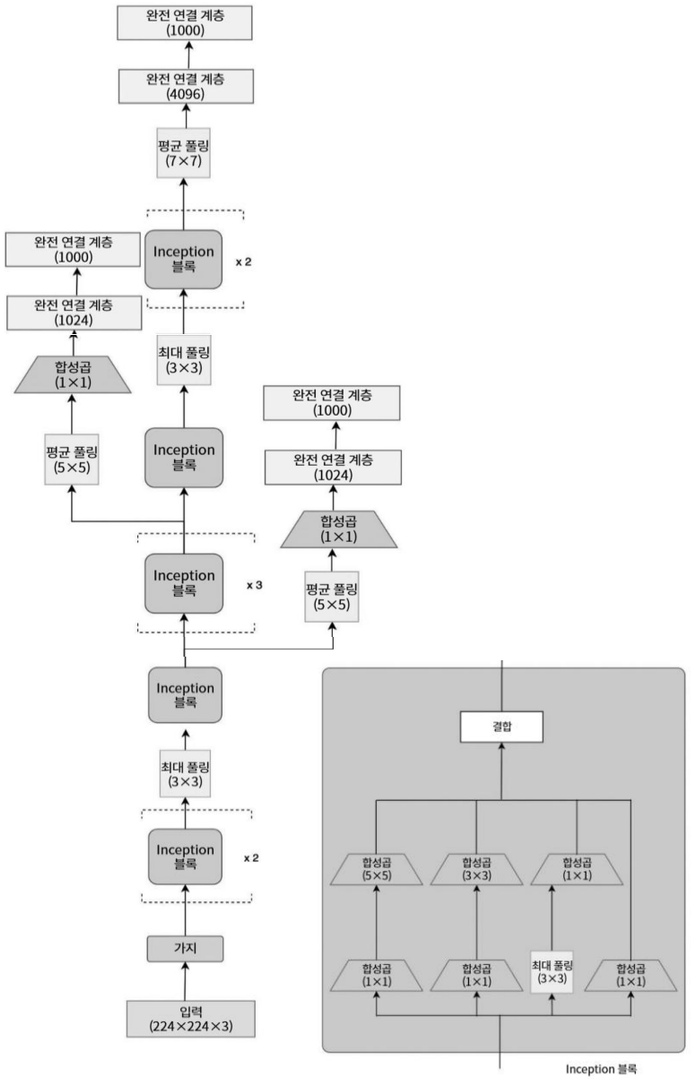
AlexNet보다 더 깊게 층을 쌓고, 3x3 크기의 작은 합성곱 커널을 사용하여 비선형성을 높이고 깊이에 집중했다.

다양한 모델 변형

계층 깊이에 따라 VGG13, VGG16, VGG19 등으로 나뉘며, 배치 정규화(BN)를 추가한 변형 모델도 존재한다.

06

Google LeNet과 Inception 모듈



Inception 모듈

병렬 합성곱 계층으로 구성된 모듈로, 다양한 크기의 필터를 동시에 적용하여 특징을 추출한다.

1x1 합성곱 활용

모델의 매개변수 개수를 획기적으로 줄이기 위해 1x1 합성곱 연산을 사용하여 차원을 축소한다.

구조적 개선 (GAP & 보조 분류기)

완전 연결 계층 대신 전역 평균 풀링(GAP)을 사용하여 과적합을 줄이고, 보조 분류기로 학습을 안정화한다.

높은 효율성

최종적으로 **22개의 깊은 계층을 사용하지만, 파라미터 수는 VGG보다 훨씬 적은 약 5백만 개로 줄였다.**