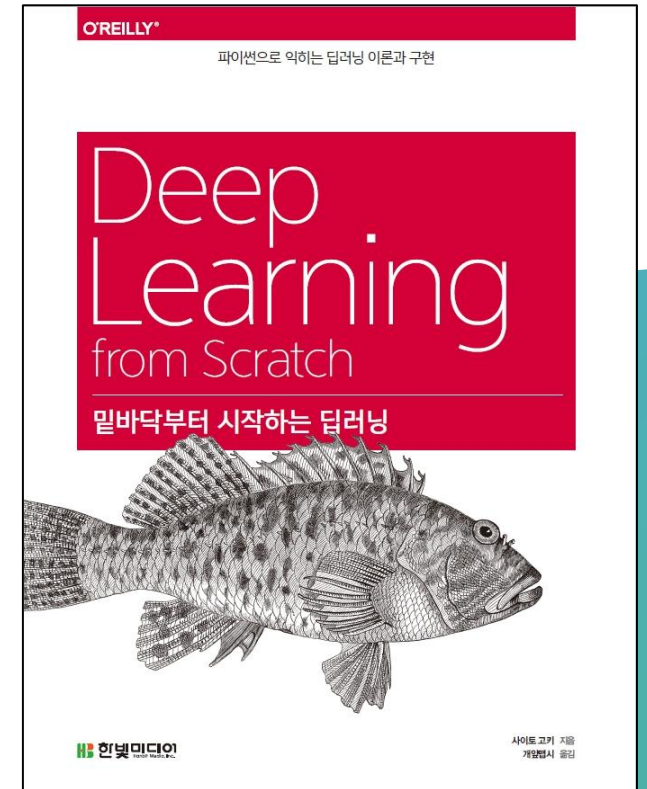


▶ 밑바닥부터 시작하는 딥러닝

# CHAPTER 8 딥러닝



# Contents

- CHAPTER 8 딥러닝
  - 8.1 더 깊게
    - 8.1.1 더 깊은 신경망으로
    - 8.1.2 정확도를 더 높이려면
    - 8.1.3 깊게 하는 이유
  - 8.2 딥러닝의 초기 역사
    - 8.2.1 이미지넷
    - 8.2.2 VGG
    - 8.2.3 GoogLeNet
    - 8.2.4 ResNet
  - 8.3 더 빠르게(딥러닝 고속화)
    - 8.3.1 풀어야 할 숙제
    - 8.3.2 GPU를 활용한 고속화

# Contents

- CHAPTER 8 딥러닝
  - 8.3.3 분산 학습
  - 8.3.4 연산 정밀도와 비트 줄이기
  - 8.4 딥러닝의 활용
    - 8.4.1 사물 검출
    - 8.4.2 분할
    - 8.4.3 사진 캡션 생성
  - 8.5 딥러닝의 미래
    - 8.5.1 이미지 스타일(화풍) 변환
    - 8.5.2 이미지 생성
    - 8.5.3 자율 주행
    - 8.5.4 Deep Q-Network(강화학습)
  - 8.6 정리



# CHAPTER 8 딥러닝

딥러닝의 특징과 과제, 가능성, 오늘날의 첨단 딥러닝에 대해 알아보기

## SECTION 08 딥러닝



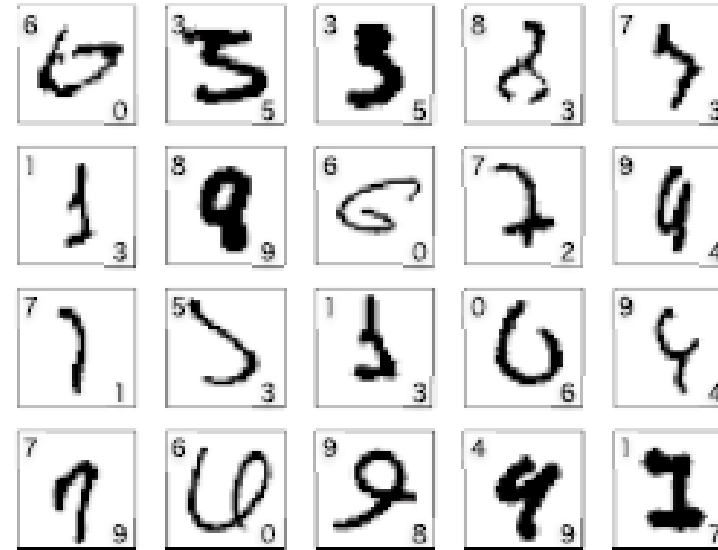
### 8.1.1 더 깊은 신경망으로

그림 8-1 손글씨 숫자를 인식하는 심층 CNN



- 3×3의 작은 필터를 사용한 합성곱 계층
- 활성화 함수는 ReLU
- 완전연결 계층 뒤에 드롭아웃 계층 사용
- Adam을 사용해 최적화
- 가중치 초기값은 'He의 초기값'

그림 8-2 인식하지 못한 이미지들 : 각 사진의 왼쪽 위는 정답 레이블, 오른쪽 아래는 이 신경망의 추론 결과



# SECTION 08 딥러닝



## 8.1.2 정확도를 더 높이려면

그림 8-3 MNIST 데이터셋에 대한 각 기법의 순위(2016년 12월 시점) <sup>83)</sup>

MNIST			
who is the best in MNIST ?			
<div><div><div>1 1 5 4 3</div><div>7 5 3 5 3</div><div>5 5 9 0 6</div><div>3 5 2 0 0</div></div><div><div>MNIST</div><div>93 results collected</div><div>Units: error %</div><div>Classify handwritten digits. Some additional results are available on the <a href="#">original dataset page</a>.</div></div></div>			
Result	Method	Venue	Details
0.21%	<a href="#">Regularization of Neural Networks using DropConnect</a>	ICML 2013	
0.23%	<a href="#">Multi-column Deep Neural Networks for Image Classification</a>	CVPR 2012	
0.23%	<a href="#">APAC: Augmented Pattern Classification with Neural Networks</a>	arXiv 2015	
0.24%	<a href="#">Batch-normalized Maxout Network in Network</a>	arXiv 2015	<a href="#">Details</a>
0.29%	<a href="#">Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree</a>	AISTATS 2015	<a href="#">Details</a>
0.31%	<a href="#">Recurrent Convolutional Neural Network for Object Recognition</a>	CVPR 2015	
0.31%	<a href="#">On the Importance of Normalisation Layers in Deep Learning with Piecewise Linear Activation Units</a>	arXiv 2015	
0.32%	<a href="#">Fractional Max-Pooling</a>	arXiv 2015	<a href="#">Details</a>

데이터 확장 <sup>data augmentation</sup>은 입력 이미지(훈련 이미지)를 알고리즘을 동원해 '인위적'으로 확장한다.

그림 8-4 데이터 확장의 예



## SECTION 08 딥러닝



### 8.1.3 깊게 하는 이유

'층을 깊게 하는 것'의 중요성에 대해서, 이를 뒷받침하는 데이터와 설명을 몇 가지 소개하겠다.

그림 8-5 5×5 합성곱 연산의 예  
입력 데이터

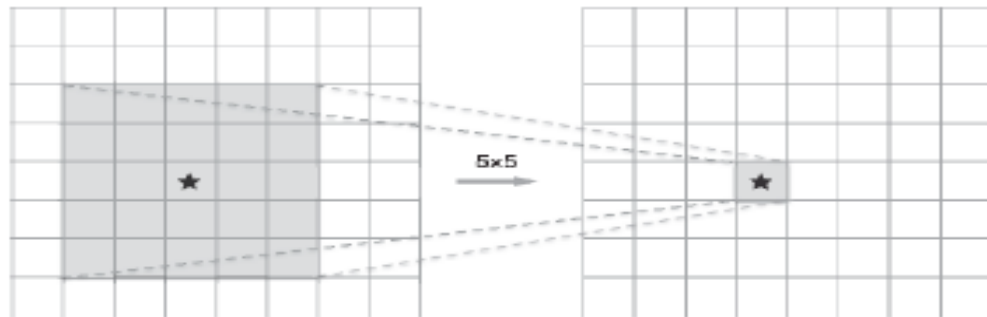
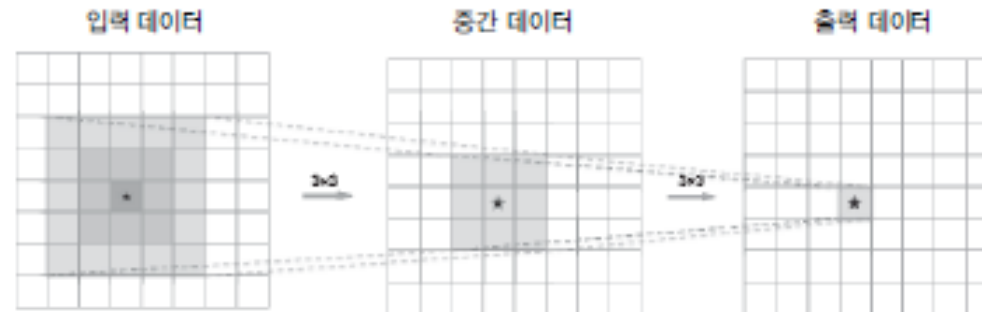


그림 8-6 3×3의 합성곱 계층을 2회 반복한 예



# SECTION 08 딥러닝



## 8.2.1 이미지넷

그림 8-7 대규모 데이터셋 ImageNet의 데이터들(24)

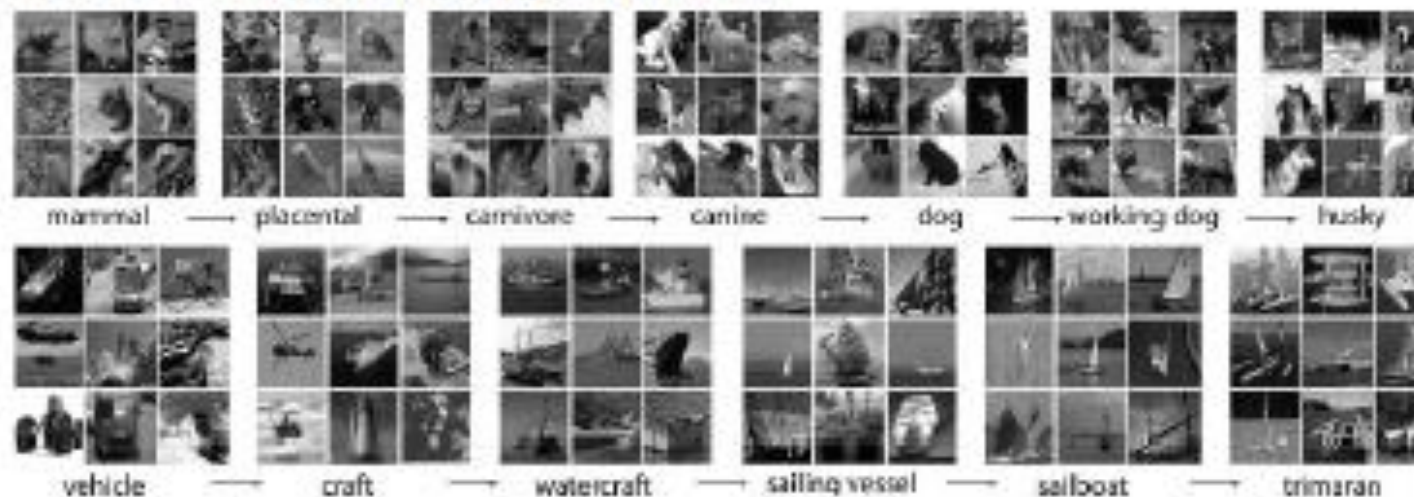
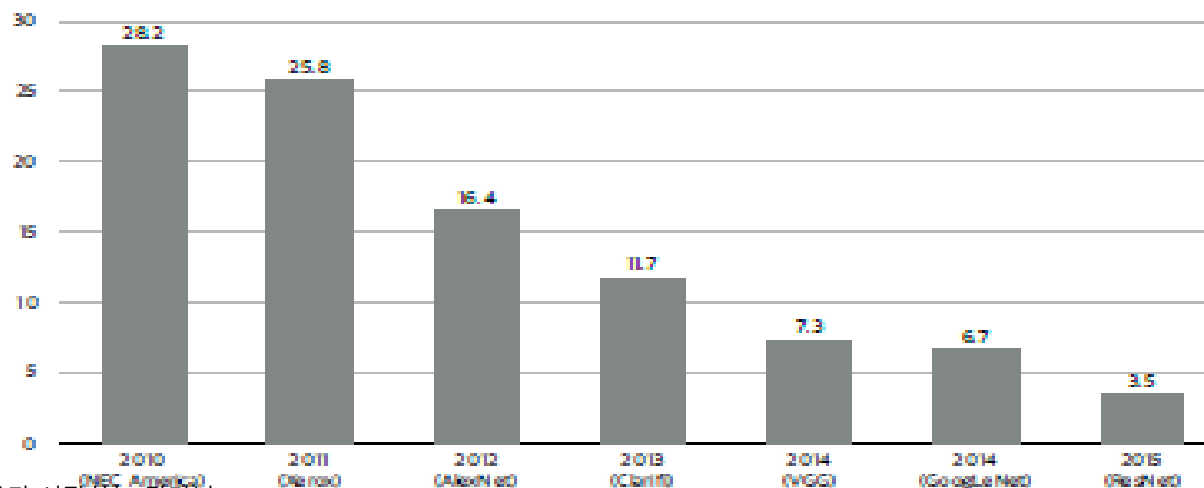


그림 8-8 ILSVRC 최우수 팀의 성적 추이 : 세로축은 오류율, 가로축은 연도, 가로축의 괄호 안은 팀 이름(또는 기법 이름)  
이미지넷 분류 톱-5 오류



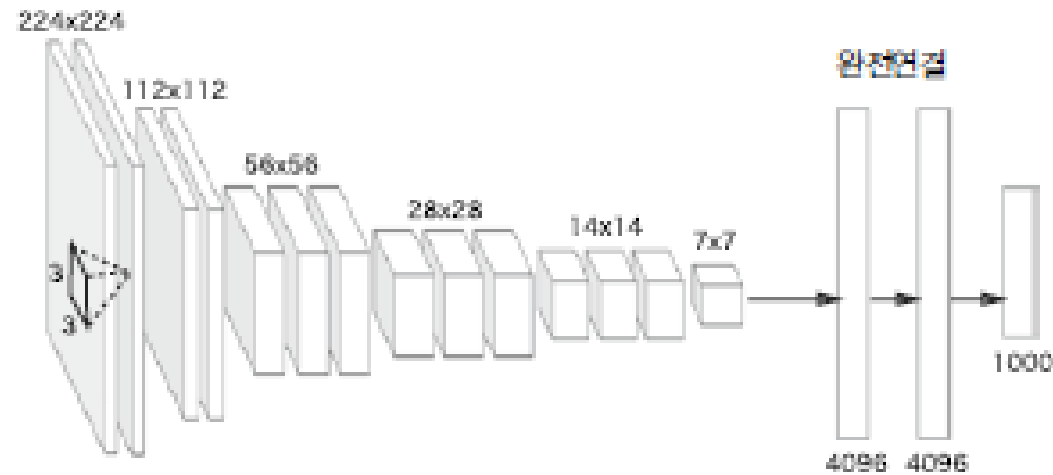


## SECTION 08 딥러닝



### 8.2.2 VGG

그림 8-9 VGG<sup>[29]</sup>



VGG에서 주목할 점은 3×3의 작은 필터를 사용한 합성곱 계층을 연속으로 거친다는 것이다.

## SECTION 08 딥러닝



### 8.2.3 GoogLeNet

그림 8-10 GoogLeNet의

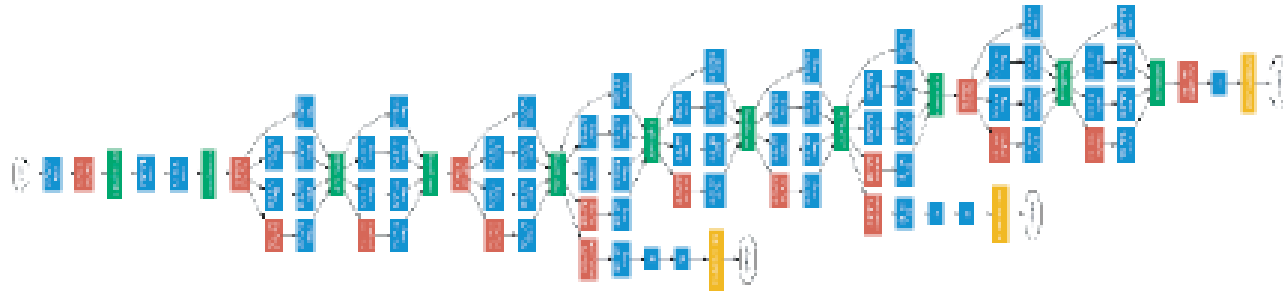
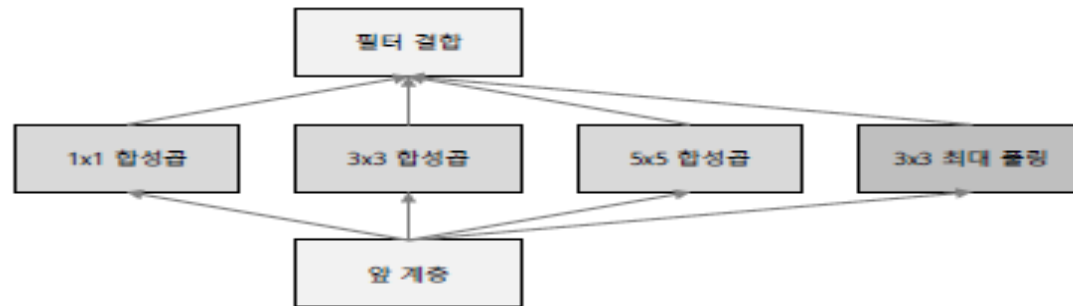


그림 8-11 GoogLeNet의 인셉션 구조



## 8.2.4 ResNet

그림 8-12 ResNet의 구성요소<sup>[24]</sup>: 'weight layer'는 합성곱 계층을 말한다

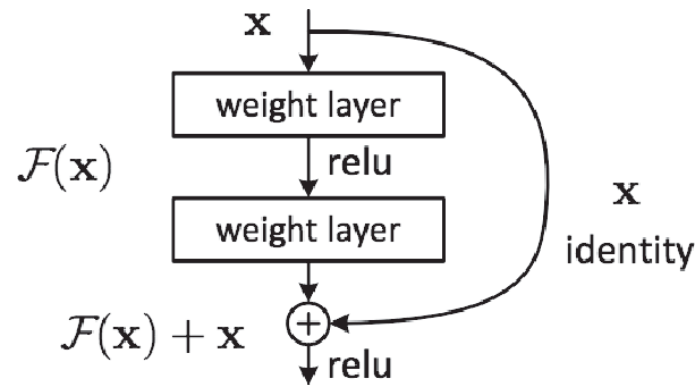
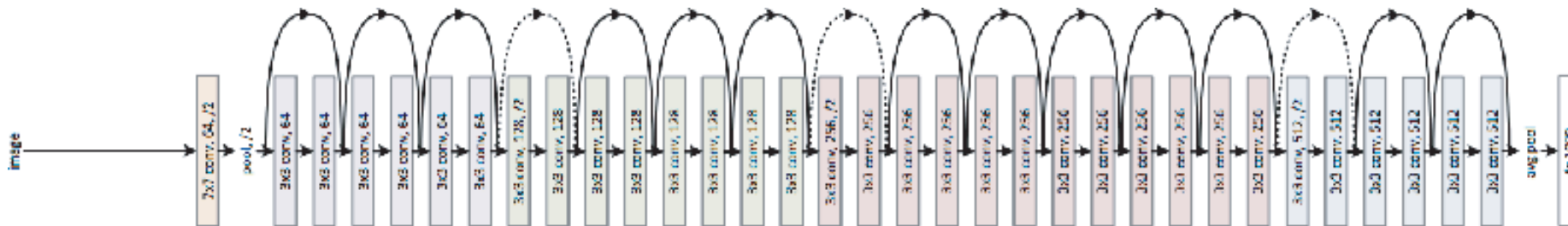


그림 8-13 ResNet<sup>[24]</sup>: 블록이 3×3인 합성곱 계층에 대응. 층을 건너뛰는 스킵 연결이 특징이다.



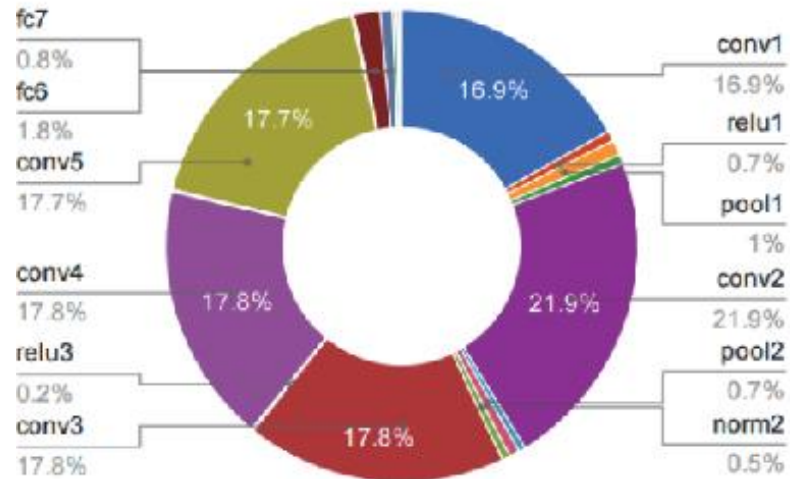
## SECTION 08 딥러닝



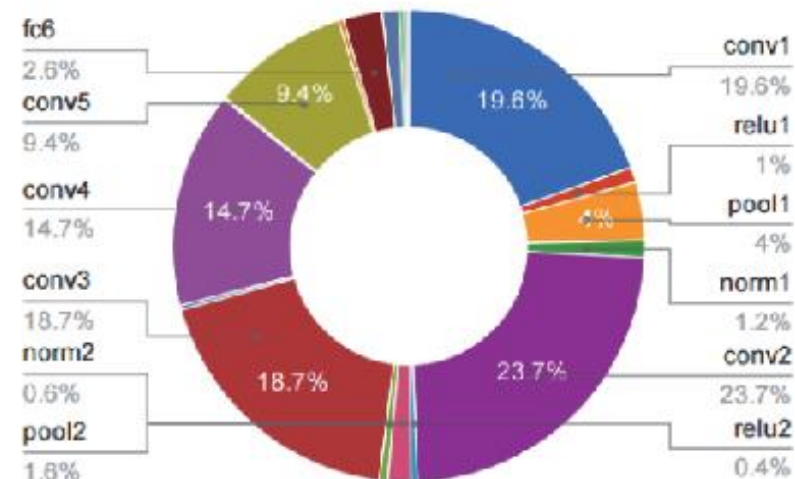
### 8.3.1 풀어야 할 숙제

그림 8-14 AlexNet의 forward 처리 시 각 층의 시간 비율 : 왼쪽이 GPU, 오른쪽이 CPU를 사용한 경우. 'conv'는 합성곱 계층, 'pool'은 풀링 계층, 'fc'는 완전연결 계층, 'norm'은 정규화 계층이다. [26]

GPU Forward Time Distribution



CPU Forward Time Distribution

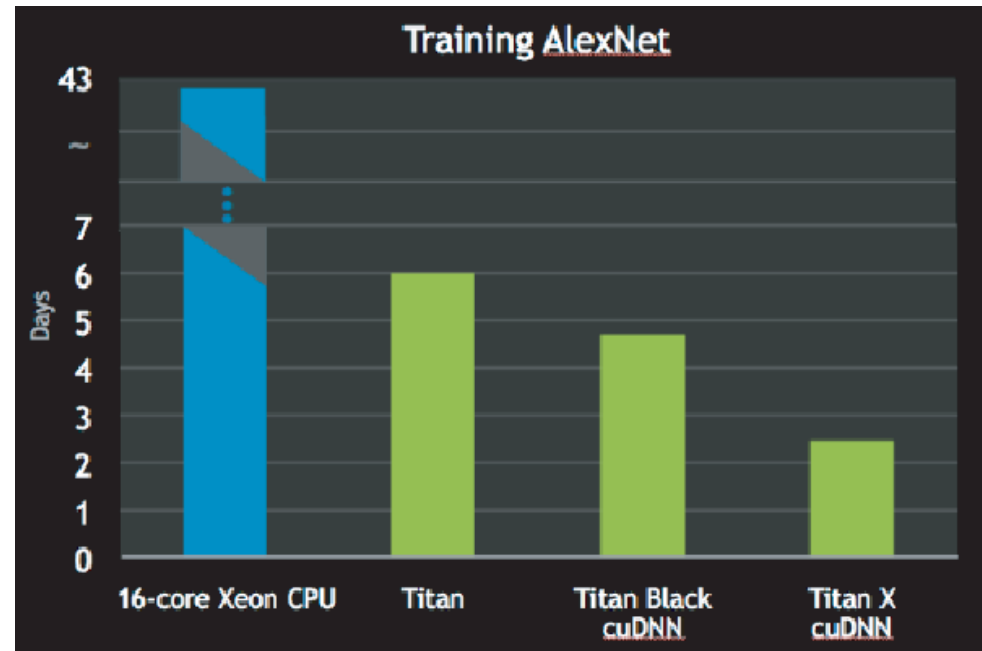


## SECTION 08 딥러닝



### 8.3.2 GPU를 활용한 고속화

그림 8-15 AlexNet의 학습 시간을 '16코어 제온 CPU'와 엔비디아 '타이탄 GPU'에서 비교한 결과<sup>[27]</sup>



엔비디아의 GPU 컴퓨팅용 통합 개발 환경인 CUDA를 사용하기 때문.

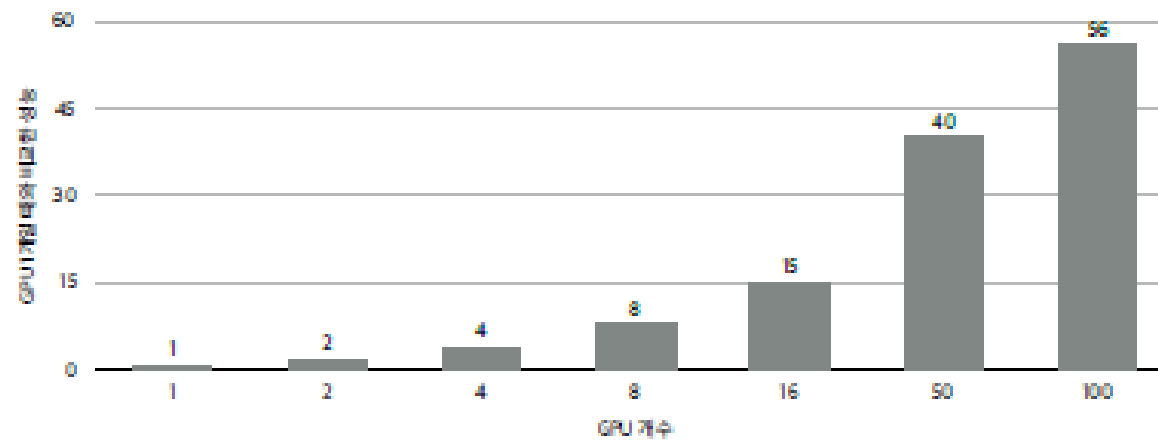
[그림 8-15]에 등장하는 cuDNN은 CUDA 위에서 동작하는 라이브러리로, 딥러닝에 최적화된 함수 등이 구현되어 있다

## SECTION 08 딥러닝



### 8.3.3 분산 학습

그림 8-16 텐서플로의 분산 학습 성능 : 가로는 GPU의 수, 세로는 GPU 1개일 때와 비교한 비율(%)



[그림 8-16]에서 보듯 GPU 수가 늘어남에 따라 학습도 빨라진다.



### 8.3.4 연산 정밀도와 비트 줄이기

계산 능력 외에도 메모리 용량과 버스 대역폭 등이 딥러닝 고속화에 병목이 될 수 있다

버스 대역폭 면에서는 GPU(혹은 CPU)의 버스를 흐르는 데이터가 많아져한계를 넘어서면 병목이 된다. 이러한 경우를 고려하면 네트워크로 주고받는 데이터의 비트 수는 최소로 만드는 것이 바람직하다

다행히 딥러닝은 높은 수치 정밀도(수치를 몇 비트로 표현하느냐)를 요구하지 않는다. 이는 신경망의 중요한 성질 중 하나로, 신경망의 견고성에 따른 특성이다.

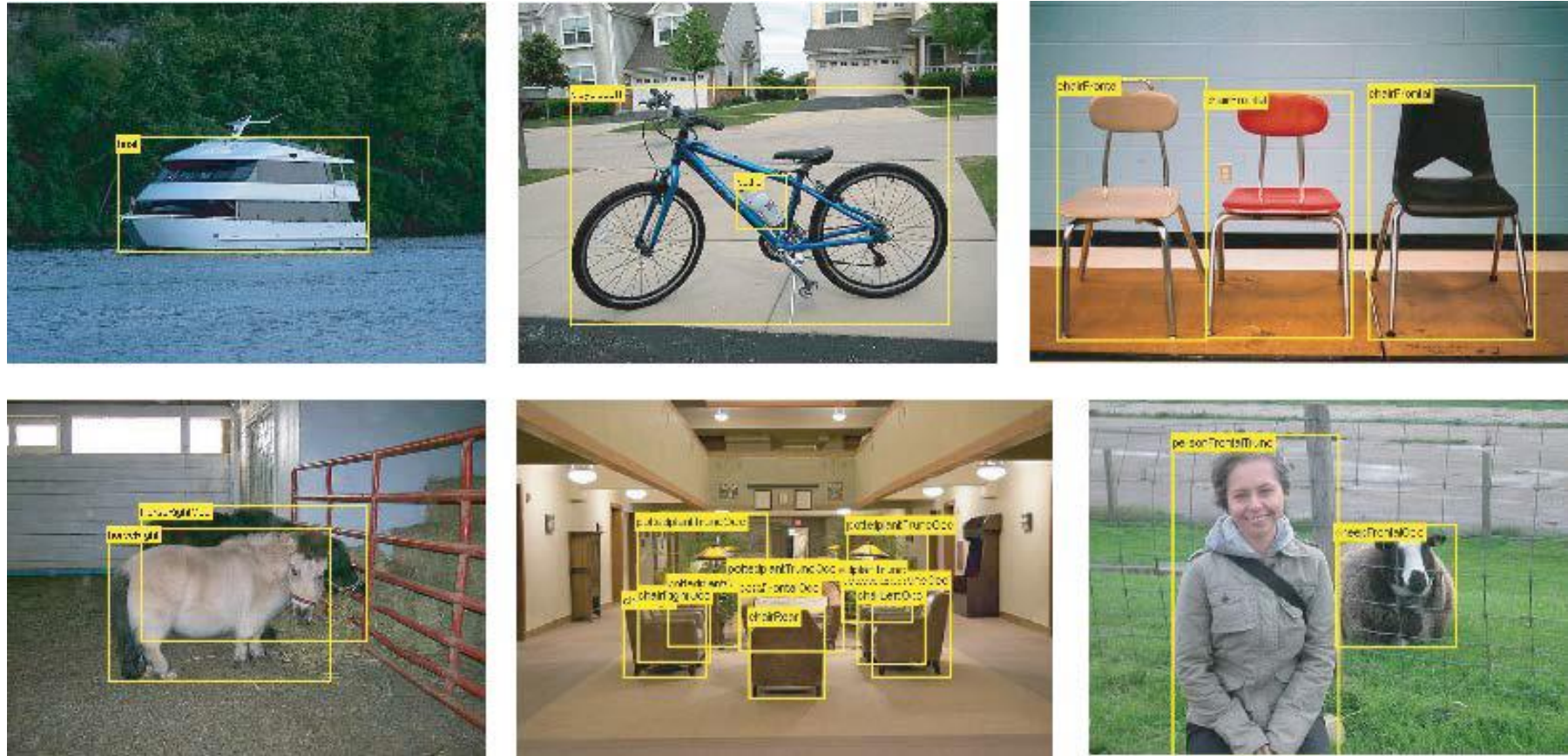
예를 들어 신경망은 입력 이미지에 노이즈가 조금 섞여 있어도 출력 결과가 잘 달라지지 않는 강건함을 보여준다

## SECTION 08 딥러닝



### 8.4.1 사물 검출

그림 8-17 사물 검출의 예<sup>[34]</sup>

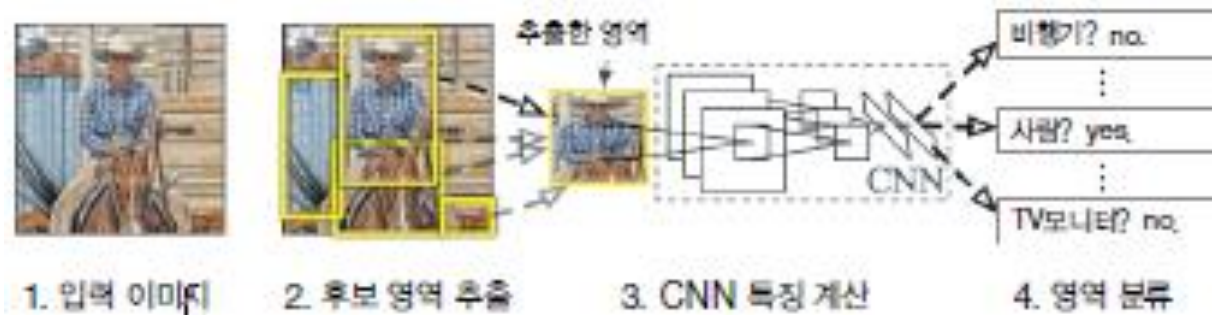




### 8.4.1 사물 검출

CNN을 이용하여 사물 검출을 수행하는 방식은 몇 가지가 있는데, 그중에서도 R-CNN<sub>Regions with Convolutional Neural Network</sub><sup>[35]</sup>이 유명하다. [그림 8-18]은 R-CNN의 처리 흐름이다.

그림 8-18 R-CNN의 처리 흐름



## SECTION 08 딥러닝



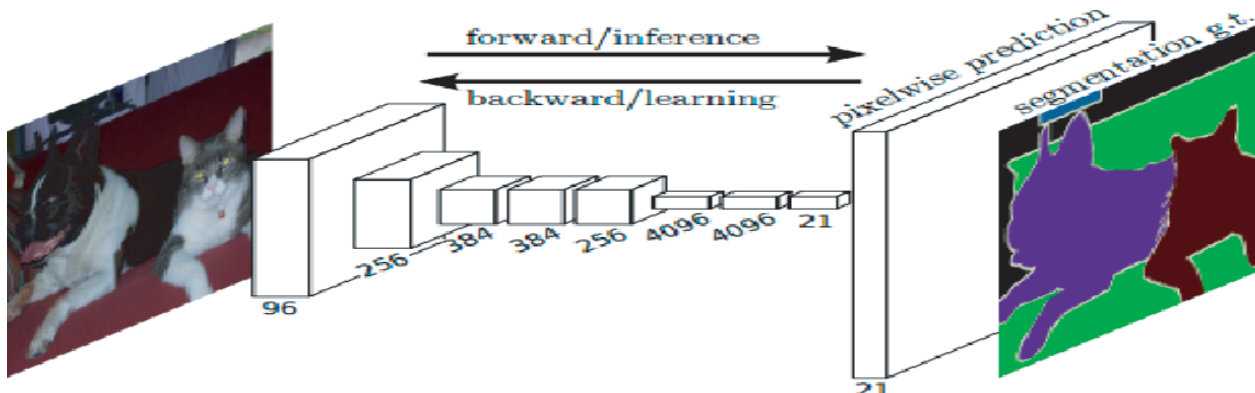
### 8.4.2 분할

분할<sub>segmentation</sub>이란 이미지를 픽셀 수준에서 분류하는 문제이다

그림 8-19 분할의 예 : 왼쪽이 입력 이미지, 오른쪽이 지도용 이미지



그림 8-20 FCN의 전체 그림<sup>[37]</sup>



## 8.4.3 사진 캡션 생성

그림 8-21 딥러닝으로 사진 캡션을 생성하는 예



[그림 8-22]와 같이 심층 CNN과 자연어를 다루는 순환 신경망(Recurrent Neural Network, RNN)으로 구성된다.

- RNN은 순환적 관계를 갖는 신경망으로 자연어나 시계열 데이터 등의 연속된 데이터를 다룰 때 많이 활용한다

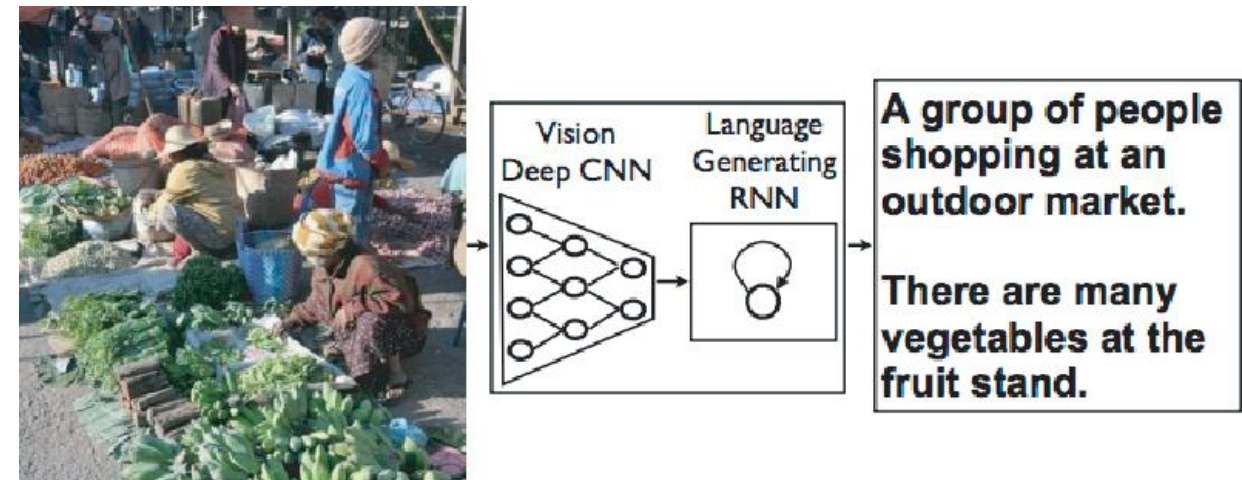


그림 8-22 NIC의 전체 구성

## SECTION 08 딥러닝



### 8.5.1 이미지 스타일(화풍) 변환

그림 8-23 「A Neural Algorithm of Artistic Style」 논문을 구현해 적용한 예 : 왼쪽 위가 '스타일 이미지', 오른쪽 위가 '콘텐츠 이미지', 아래가 새로 생성한 이미지<sup>4)</sup>





## SECTION 08 딥러닝



### 8.5.2 이미지 생성

딥러닝으로 '침실' 이미지를 무<sub>無</sub>로부터 생성하는 게 가능하다.  
[그림 8-24]의 이미지는 DCGAN<sub>Deep Convolutional Generative Adversarial Network</sub> 기법<sup>[41]</sup>으로 생성한 침실 이미지들이다.

그림 8-24 DCGAN으로 새롭게 생성한 침실 이미지들<sup>[41]</sup>



## SECTION 08 딥러닝



### 8.5.3 자율 주행

SegNet<sup>[42]</sup>이라는 CNN 기반 신경망은 [그림 8-25]와 같이 주변 환경을 정확하게 인식해낸다.

그림 8-25 딥러닝을 활용한 이미지 분할의 예 : 도로, 차, 건물, 인도 등을 정확하게 인식한다.<sup>[42]</sup>



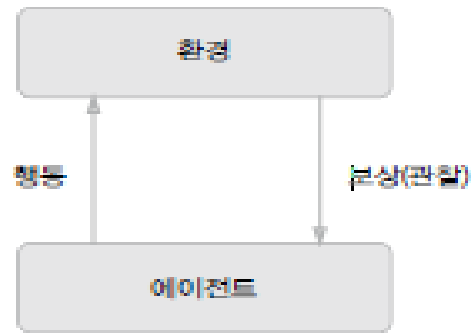
## SECTION 08 딥러닝



### 8.5.4 Deep Q-Network(강화학습)

이는 '가르침'에 의존하는 '지도 학습'과는 다른 분야로,  
강화학습(reinforcement learning)이라 한다.

그림 8-26 강화학습의 기본 틀: 에이전트는 더 좋은 보상을 받기 위해 스스로 학습한다.



딥러닝을 사용한 강화학습 중 Deep Q-Network(일명 DQN)<sup>[44]</sup>라는 방법이 있다.

그림 8-27 Deep Q-Network로 비디오 게임 조작을 학습한다. 비디오 게임 영상을 입력받아 시행착오를 거쳐 프로 게이머 뽀치는 게임 컨트롤을 학습한다.<sup>44)</sup>

