

# CSC321 Lecture 12: Image Classification

## Object recognition

- 주어진 이미지에 어떤 object category가 있는지 알아내는 것
- 어려운 점:
  - 사물마다 위치, 크기, 모양, 생김새 등이 다름
  - 겹치는 부분(occlusion), 빛, 변화 등
- 활용 분야: 이미지 검색, object detection (어떤 사물이 나타나는 모든 곳 탐지, e.g., 자율주행 자동차가 행인이나 정지 표시 발견하는 것)

## Recognition datasets

- 머신러닝 시스템을 훈련, 테스트 하기 위해서는 데이터셋이 필요함 -> 데이터셋의 디자인 중요!
  - 데이터셋 고려사항: 어떤 카테고리를 포함할건지? 어디에서 이미지를 가져올건지? 몇 장의 이미지를 수집할건지? 어떻게 이미지를 정규화(전처리) 할건지?

## Image classification

- Conv nets(=Convolutional Neural Networks)는 이미지 분류 방법 중 하나임 -> 지난 5년 동안 가장 성공적이었음
- 지난 20년동안 가장 큰 업적: 1) 큰 데이터셋 (디지털 카메라, 인터넷), 2) 빠른 컴퓨터 (GPU는 CPU보다 훈련 속도가 30배 정도 빠름) => 그 결과 점점 큰 신경망을 피팅할 수 있게 됨

## MNIST dataset

- 손글씨 숫자 데이터셋
  - 카테고리: 10개 숫자 카테고리(0~9)
  - 소스: 봉투에 적힌 손글씨 우편번호 스캔본
  - 크기: 트레이닝 60000장, 테스트 10000장 & grayscale & 28x28 크기
  - 정규화: 이미지 내 가운데 위치 고정, 일정한 크기로 조정
    - 숫자 인식이 어떤 이미지 처리 파이프라인의 한 부분이라는 가정
- 1998년 LeNet(LeCun et al.) 개발 -> 98.9%의 테스트 정확도로 숫자 분류
  - 수표에 적힌 숫자를 자동으로 읽는 시스템 개발 가능해짐
- 잘 만들어짐!

## Caltech101

- 2003년 수집된 object recognition 데이터셋
  - 카테고리: 101개 사물 카테고리
    - 선정방식: 사전의 랜덤 페이지를 펼쳐서 이미지와 연관된 명사 고름
  - 소스: Google 이미지 검색에서 후보자 물색 후 직접 이미지 선정
  - 이미지 수: 카테고리당 대체로 40~800장의 이미지 보유 -> 총 9146장 이미지
    - 얼굴같이 보편적이고 인기있는 카테고리의 이미지 수가 더 많은 경향
  - 정규화: 300 pixels 넓이로 크기 조정(300x200), 방향성을 가진 사물은 같은 방향을 향하도록 조정(e.g., 비행기, 오토바이), 세로로 된 사물은 회전 (알고리즘이 세로로 된 사물을 활용할 수 없어서; e.g., 빌딩)



- 데이터셋의 단점 (\*\*Caltech101만 사용한 논문은 학술지 게재 거절)
  - 데이터가 지나치게 깔끔함
    - 실제 상황에서의 사물과 다름
  - 한정된 사물 카테고리 수
  - 몇몇 카테고리의 이미지 수가 너무 적음
    - e.g., 31장
  - 이미지 보정에 의한 일그러짐, artifact
  - => 각 카테고리의 평균 이미지를 보면, 그 이미지가 실제로 마주하는 사물을 대표하기 어려움을 알 수 있음



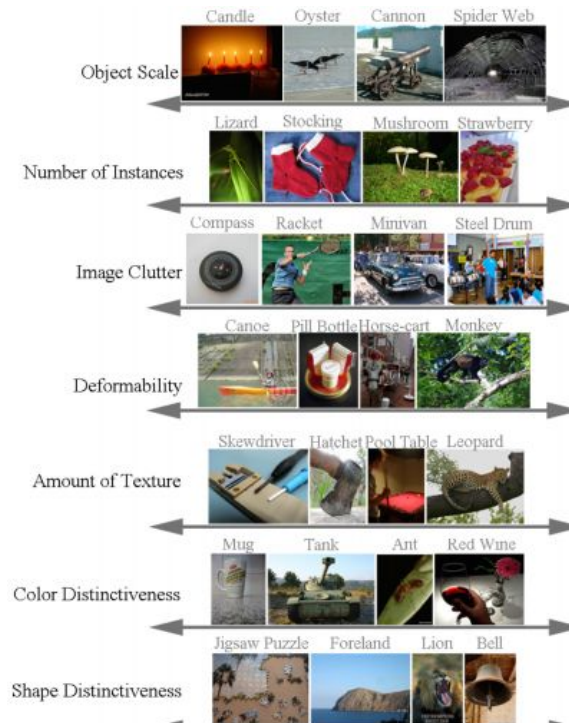
## ImageNet

- 사물 인식 데이터셋 (2009) -> 사물 인식 분야 발전에 큰 역할을 함



- ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 데이터셋
- 데이터 설명

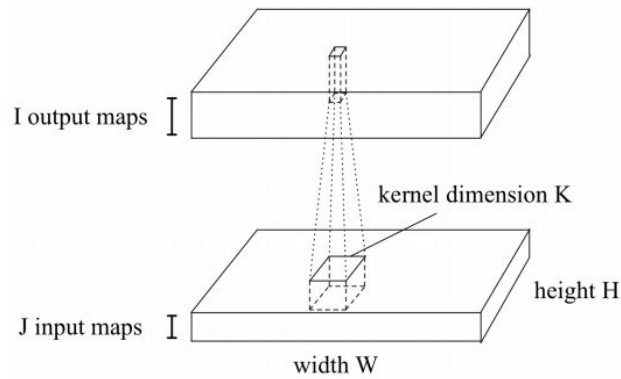
- 카테고리: WordNet 데이터베이스에서 착안
  - 2010년 기준 22000개
  - ILSVRC에 1000개 카테고리 사용
  - 구체적임 -> e.g., 강아지 몇백 종류
- 크기: 120만개 full-size 이미지 (ILSVRC)
- 소스: 검색 엔진, 사람 레이블링
- 정규화: 없음 & 대회 참가자들이 원하는대로 전처리 가능
- 이미지 & 사물 카테고리가 여러 방향으로 다름



- 파일 사이즈: MNIST (60MB) vs. ImageNet (50GB)

### Conv Net 크기

- 네트워크 사이즈 재는 법
  - unit 갯수 -> 훈련 중간에 활성화 내용들이 메모리에 저장되어야 하기 때문 (e.g., backprop)
  - weight 갯수 -> weight들이 메모리에 저장되기 때문 & 파라미터 갯수가 오버피팅 정도를 결정
  - connection 갯수 -> connection마다 3 이상의 계산이 이뤄짐 (1번 forward pass, 2번 backward pass)
- Fully connected layer가 M개의 input units와 N개의 output units가 있다면 -> MN connections, MN weights 생성
  - 예시:

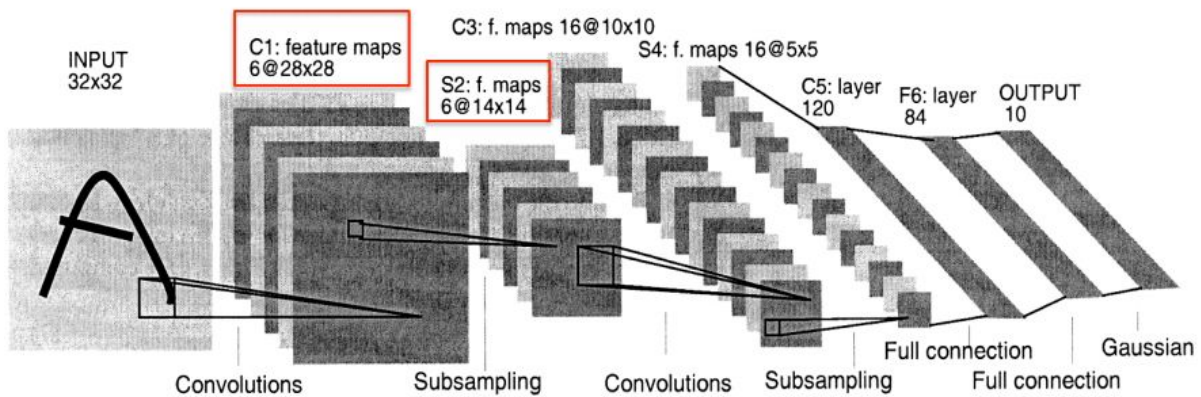


	fully connected layer	convolution layer
# output units	WHI	WHI
# weights	$W^2 H^2 I J$	$K^2 I J$
# connections	$W^2 H^2 I J$	$WHK^2 IJ$

- 중요 포인트
  - 대부분의 unit과 connection은 convolutional layer에 있음
  - 대부분의 weight는 fully connected layer에 있음
- 만약 레이어를 크게 만들 경우 리소스 한계에 부딪힐 수 있음 - 계산 시간, 메모리
- 1998년 이후 conv net의 크기는 아주아주 커짐

#### LeNet (1998)

- MNIST 데이터에 활용
- 구조



Sizes of layers in LeNet:

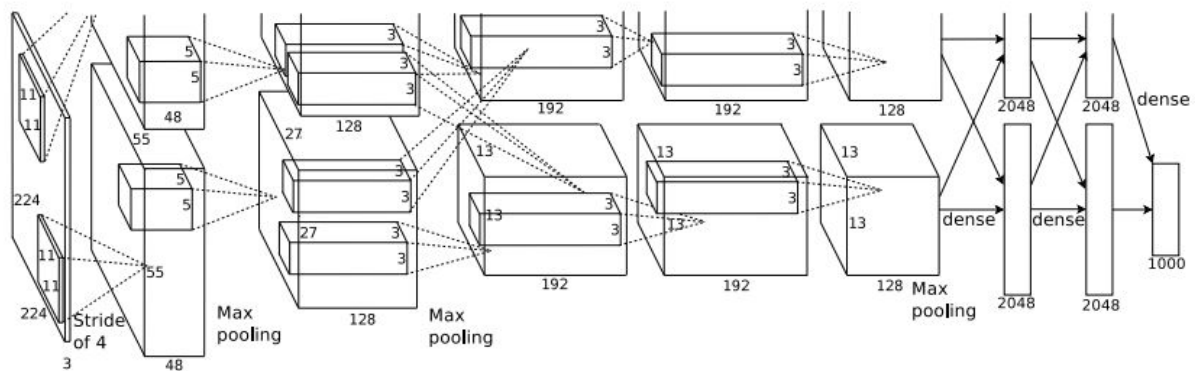
Layer	Type	# units	# connections	# weights
C1	convolution	4704	117,600	150
S2	pooling	1176	4704	0
C3	convolution	1600	240,000	2400
S4	pooling	400	1600	0
F5	fully connected	120	48,000	48,000
F6	fully connected	84	10,080	10,080
output	fully connected	10	840	840

## Conv net 크기 비교

	LeNet (1989)	LeNet (1998)	AlexNet (2012)
<b>classification task</b>	digits	digits	objects
<b>categories</b>	10	10	1,000
<b>image size</b>	$16 \times 16$	$28 \times 28$	$256 \times 256 \times 3$
<b>training examples</b>	7,291	60,000	1.2 million
<b>units</b>	1,256	8,084	658,000
<b>parameters</b>	9,760	60,000	60 million
<b>connections</b>	65,000	344,000	652 million
<b>total operations</b>	11 billion	412 billion	200 quadrillion (est.)

## AlexNet (2012)

- 8개 weight layers, 16.4% top-5 error (i.e., 네트워크가 카테고리를 맞추려면 5번 시도)



(Krizhevsky et al., 2012)

- ReLU, weight decay, 데이터 증식, SGD with momentum, dropout 사용

## GoogLeNet (2014)

- 22 weight layers
- fully convolutional -> fully connected layers 없음
- convolution들이 작은 convolutions들로 쪼개짐
- ImageNet에서 6.6% 테스트 에러

