

 CODINGO  X  posco

K-Digital Training 스마트 팩토리 3기

목차

- 컴퓨터 비전
- 이미지 처리 및 이해
- 비전 알고리즘

컴퓨터 비전

컴퓨터 비전(Computer Vision)



- 컴퓨터가 디지털 이미지나 비디오 데이터를 이해하고 해석하는 인공지능 분야
- 목표는 컴퓨터를 통해 시각적인 정보를 처리하고 이해하는 것

컴퓨터 비전(Computer Vision)

- 사용되는 기술

1. 이미지 전처리

- 이미지 데이터를 사전에 처리하여 품질을 향상시키거나 원하는 정보를 추출하는 과정
- 이미지 필터링, 크기 조정, 밝기 및 콘트라스트 조정, 노이즈 제거

2. 특징 추출

- 의미있는 정보를 추출하는 과정
- SIFT(Scale-Invariant Feature Transform), SURF(Speeded-Up Robust Features), HOG(Histogram of Oriented Gradients)

- 사용되는 기술

3. 객체 감지와 인식

- **객체 감지**(Object Detection) – 이미지나 비디오에서 특정 객체의 위치와 경계를 탐지하는 작업
- **객체 인식**(Object Recognition) – 이미지나 비디오에서 특정 객체를 식별하고 분류하는 작업
- **컨볼루션 신경망**(Convolutional Neural Network, CNN) 모델 사용

컴퓨터 비전(Computer Vision)



- 사용되는 기술

4. 세분화

- **세분화**(Segmentation)는 이미지에서 픽셀 수준에서 객체의 경계를 식별하는 작업
- 이미지를 개별적인 객체로 분할하거나 배경과 객체를 구분
- U-Net, Mask R-CNN, SAM 등의 방법이 있음

5. 추적

- 추적(Tracking)은 동영상에서 객체를 지속적으로 감지하고 추적하는 작업

이미지 처리와 이해

이미지 처리(Image Processing)

- 이미지 처리는 디지털 이미지에 대해 다양한 연산과 처리를 수행하여 이미지를 개선하거나 원하는 정보를 추출하는 작업

이미지 처리(Image Processing)

- 사용되는 기술

1. 이미지 필터링

- 이미지 필터링은 이미지에 다양한 필터를 적용하여 이미지를 부드럽게 만들거나 경계를 강조하는 등의 변형을 가하는 작업
- 대표적인 이미지 필터링 기법 - 평균 필터, 가우시안 필터, 소벨 필터

2. 이미지 노이즈 제거

- 불필요한 잡음을 제거하여 품질을 향상시키는 작업
- 대표적인 노이즈 제거 기법 - 미디언 필터, 가우시안 필터, 양방향 필터

이미지 처리(Image Processing)

- 사용되는 기술

3. 이미지 변환과 보정

- 변환 - 이미지를 회전, 확대/축소, 반전등의 변환을 가하는 작업
- 보정 - 밝기, 콘트라스트, 색상등을 보정하여 시각적인 품질을 개선하는 것

4. 이미지 세분화

- 객체의 경계를 식별하거나 개별적인 영역으로 분할하는 작업
- 세분화 알고리즘 - 임계값 처리, 분할 알고리즘, 그래프 컷

이미지 처리(Image Processing)

- 사용되는 기술

- 5. 특징 추출

- 의미 있는 정보를 추출하기 위해 특징 추출작업을 수행
 - 중요한 패턴, 형태, 색상 등을 나타내는 속성
 - 주로 텍스처, 엣지, 색상 히스토그램등의 특징을 추출

이미지 이해

- 픽셀(Pixel)
 - 화면을 구성하는 가장 기본 단위(화소)
 - 이미지는 픽셀의 집합으로 표현
 - 적색(Red), 녹색(Green), 청색(Blue)의 값을 조합하여 나타냄
 - 각 픽셀이 가질 수 있는 색의 종류는 각 픽셀에 몇 비트를 할당하느냐에 따라 달려 있음
 - 이 때 할당된 비트 수를 색상의 깊이(Depth)라고 하며 비트가 많을 수록 더 많은 색 표현 가능
 - 8비트의 경우 256가지 색상을 가질수 있음($2^8 = 256$)

이미지 이해

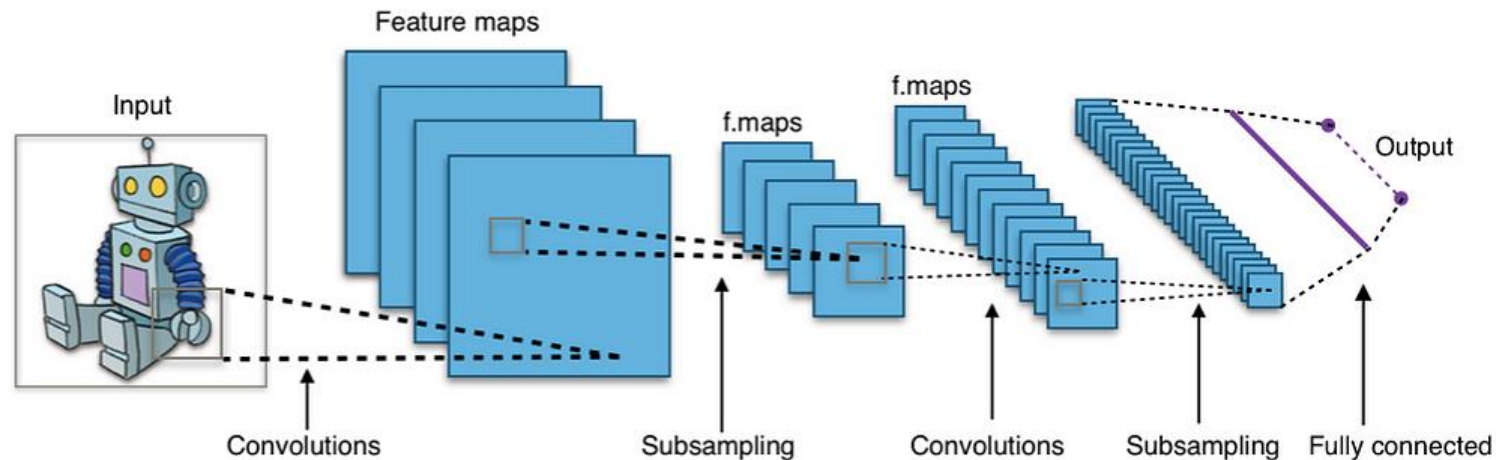
- RGB 모델

- Red, Green, Blue 모델
- 빛의 삼원색으로 불리는 적색, 녹색, 청색이 기본이 되는 컬러 모델
- 0~255의 값을 사용
- 각각의 성분은 8비트로 표현
- 0이면 해당 성분이 없음을 의미하고 255이면 성분의 강도가 가장 높음을 의미
- 색상은 세가지 성분의 강도 조합으로 표현
- (255, 0, 0)은 빨간색, (0, 255, 0)은 초록색, (0, 0, 255)은 파란색을 나타냄
- A(Alpha, 투명도)값이 추가되어서 RGBA 로도 나타냄
 - A 값이 0이면 투명하고 255이면 불투명, 그 사이의 값이면 반투명을 나타냄

비전 알고리즘

CNN(Convolutional Neural Network) 모델

- 합성곱 신경망
- 인간의 시신경 구조를 모방한 기술
 - 특정 이미지를 봤을 때 시신경의 전체가 아닌 특정 부위의 영역만 활성화 됨
- 이미지의 공간 정보를 유지한 채 학습을 하는 모델
 - 1차원 배열로 평면화 시키는 과정에서 공간정보가 손실될 수 있음
 - 2차원 배열을 이용하여 공간정보가 손실되지 않도록 함



출처: https://en.wikipedia.org/wiki/Convolutional_neural_network

Convolution

- 합성곱
- 오른쪽 그림에서 노란 영역을 filter(kernel) 라고 함 (여기서는 3x3)
- 움직이는 이동거리(step size)를 stride (여기서는 1)
- 결과 feature값은 행렬의 내적의 합
- 컨볼루션 연산을 수행한 뒤에는 filter의 개수가 레이어의 채널 크기가 된다.
- 따라서 레이어가 깊어질수록 너비와 높이는 감소하고 깊이(채널)는 증가

| | | | | |
|-----------------|-----------------|-----------------|---|---|
| 1 _{x1} | 1 _{x0} | 1 _{x1} | 0 | 0 |
| 0 _{x0} | 1 _{x1} | 1 _{x0} | 1 | 0 |
| 0 _{x1} | 0 _{x0} | 1 _{x1} | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

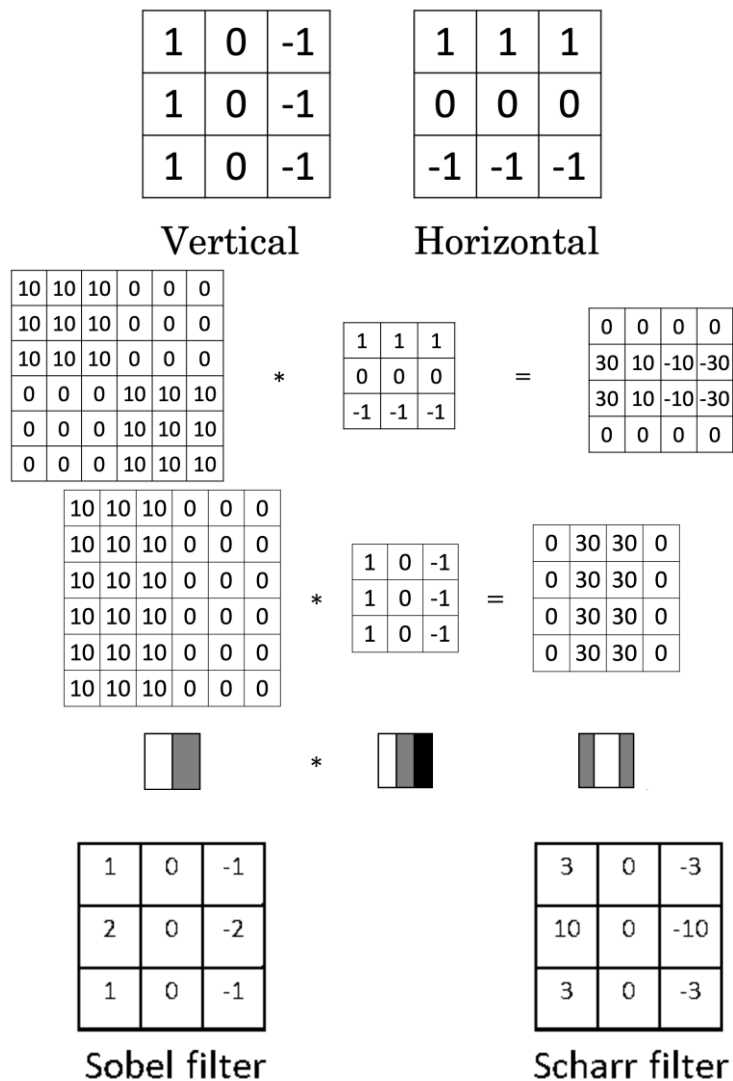
Image

| | | |
|---|--|--|
| 4 | | |
| | | |
| | | |

Convolved
Feature

Filter

- 여러가지 종류가 존재
- 수직선을 파악하는 vertical filter
- 수평선을 파악하는 horizontal filter
- 열의 중앙에 더 많은 가중치를 부여하는 sobel filter와 scharr filter



Filter

- 따라서 $n \times n$ 이미지가 $f \times f$ 필터와 컨볼루션 된 경우, 출력 이미지 크기는 $(n-f+1) \times (n-f+1)$ 이 된다.(stride 가 1 일때)
- 단점 : 컨볼루션 연산 마다 원본 이미지가 축소된다.

| | | |
|---|---|----|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

| | | |
|----|----|----|
| 1 | 1 | 1 |
| 0 | 0 | 0 |
| -1 | -1 | -1 |

Vertical

Horizontal

| | | | | | |
|----|----|----|----|----|----|
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 0 | 0 | 0 | 10 | 10 | 10 |
| 0 | 0 | 0 | 10 | 10 | 10 |
| 0 | 0 | 0 | 10 | 10 | 10 |

*

| | | |
|----|----|----|
| 1 | 1 | 1 |
| 0 | 0 | 0 |
| -1 | -1 | -1 |

=

| | | | |
|----|----|-----|-----|
| 0 | 0 | 0 | 0 |
| 30 | 10 | -10 | -30 |
| 30 | 10 | -10 | -30 |
| 0 | 0 | 0 | 0 |

| | | | | | |
|----|----|----|---|---|---|
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |

*

| | | |
|---|---|----|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

=

| | | | |
|---|----|----|---|
| 0 | 30 | 30 | 0 |
| 0 | 30 | 30 | 0 |
| 0 | 30 | 30 | 0 |
| 0 | 30 | 30 | 0 |



*



| | | |
|---|---|----|
| 1 | 0 | -1 |
| 2 | 0 | -2 |
| 1 | 0 | -1 |

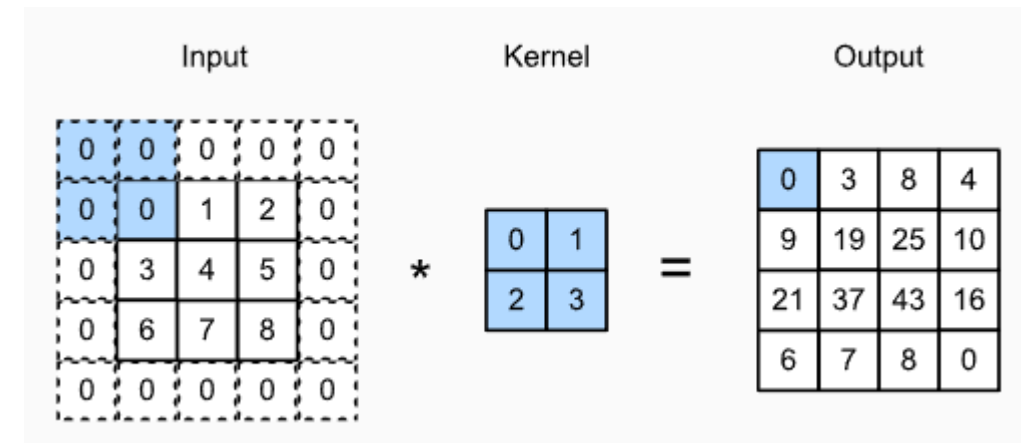
Sobel filter

| | | |
|----|---|-----|
| 3 | 0 | -3 |
| 10 | 0 | -10 |
| 3 | 0 | -3 |

Scharr filter

Padding

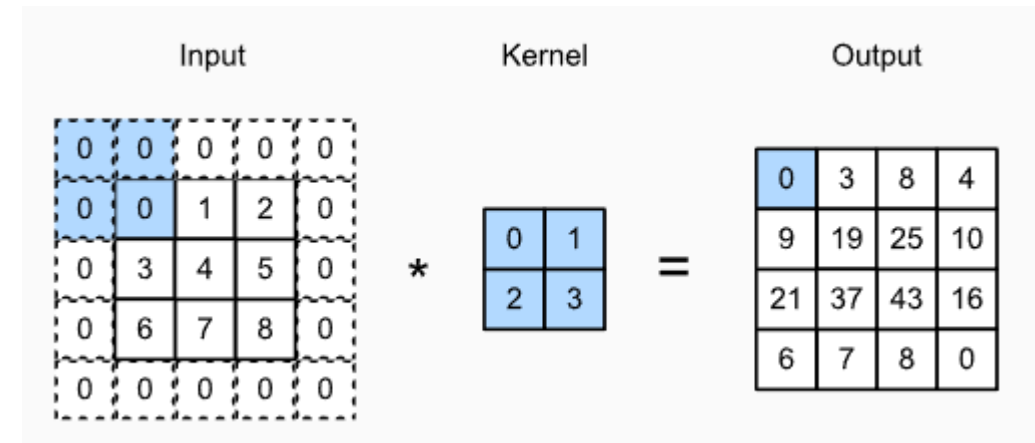
- 컨볼루션의 두 가지 문제
 - 연산마다 원본 이미지가 축소
 - 이미지의 모서리에 있는 정보는 한 번만 사용되고 가운데에 있는 픽셀은 여러 번 사용, 이미지의 모서리에 있는 정보가 약해진다.
- 이 두 가지 문제를 해결하기 위해 padding 개념 도입
- 외각에 0의 padding을 추가하여 원본 이미지의 크기를 유지



출처: <https://www.codingninjas.com/studio/library/convolution-layer-padding-stride-and-pooling-in-cnn>

Padding

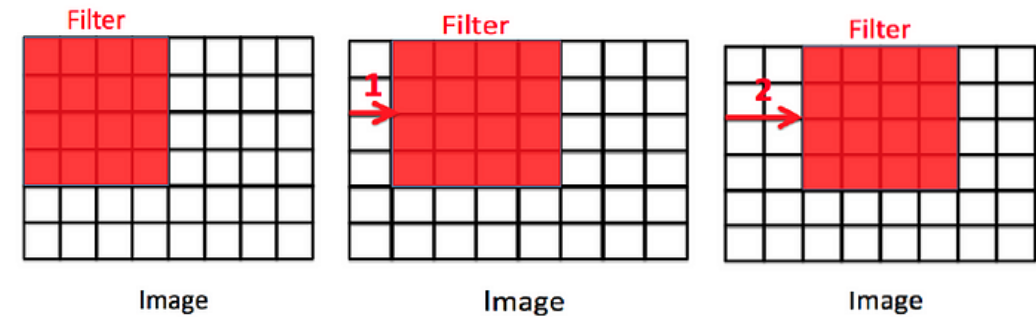
- 따라서 $n \times n$ 이미지에 p 만큼 패딩 후 $f \times f$ 필터와 컨볼루션 된 경우, 출력 이미지 크기는 $(n+2p-f+1)^2$ 이 된다.(stride 가 1 일때)



출처: <https://www.codingninjas.com/studio/library/convolution-layer-padding-stride-and-pooling-in-cnn>

Stride

- 필터를 적용할 때 얼마나 이동할지 의미
- 1이면 한 칸씩 이동
- 2이면 두 칸씩 이동
- 따라서 $n \times n$ 이미지에 p 만큼 패딩 후 $f \times f$ 필터로 s 만큼 이동하며 컨볼루션 된 경우, 출력 이미지 크기는 $((n+2p-f)/s+1)^2$ 이 된다.



출처: <https://www.codingninjas.com/studio/library/convolution-layer-padding-stride-and-pooling-in-cnn>

Pooling

- 이미지의 크기를 계속 유지한채 FC(Fully Connected) 레이어로 가게 되면 연산량이 기하급수적으로 늘어남
- 크기도 줄이고, 특정 feature를 강조하는 역할

Pooling

- 종류
 - Max Pooling – 주로 사용
 - 가장 큰 값만 이용
 - Average Pooling
 - 평균 값 이용
 - Min Pooling
 - 가장 작은 값만 이용
- 보통 하이퍼 파라미터를 이용하여 설정

| | | | |
|-----|----|----|----|
| 13 | 20 | 30 | 0 |
| 8 | 12 | 3 | 0 |
| 34 | 70 | 33 | 5 |
| 111 | 80 | 10 | 23 |

Activation Map

| | |
|-----|----|
| 20 | 30 |
| 111 | 33 |

Max Pooling

| | |
|----|----|
| 13 | 8 |
| 66 | 18 |

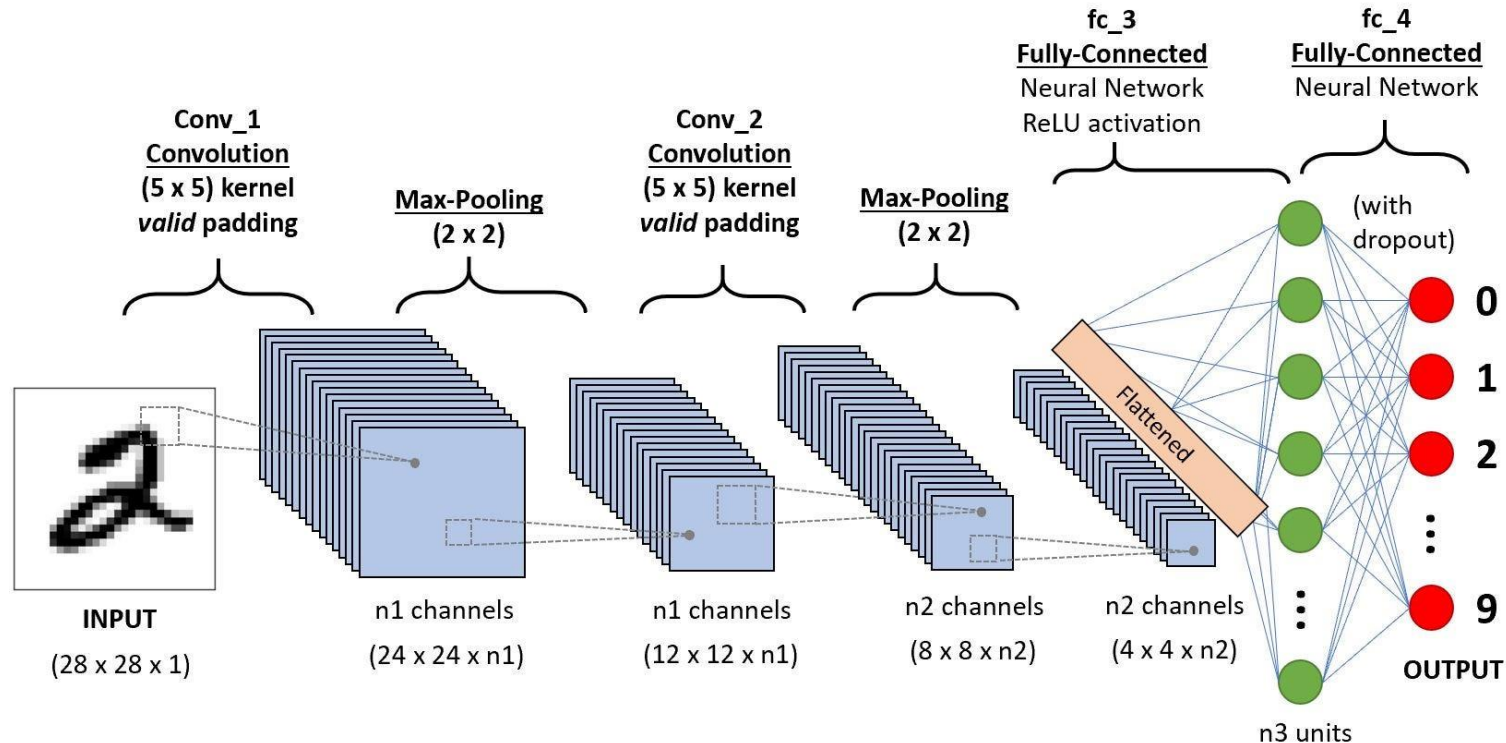
Average Pooling

| | |
|----|---|
| 8 | 0 |
| 34 | 5 |

Min Pooling

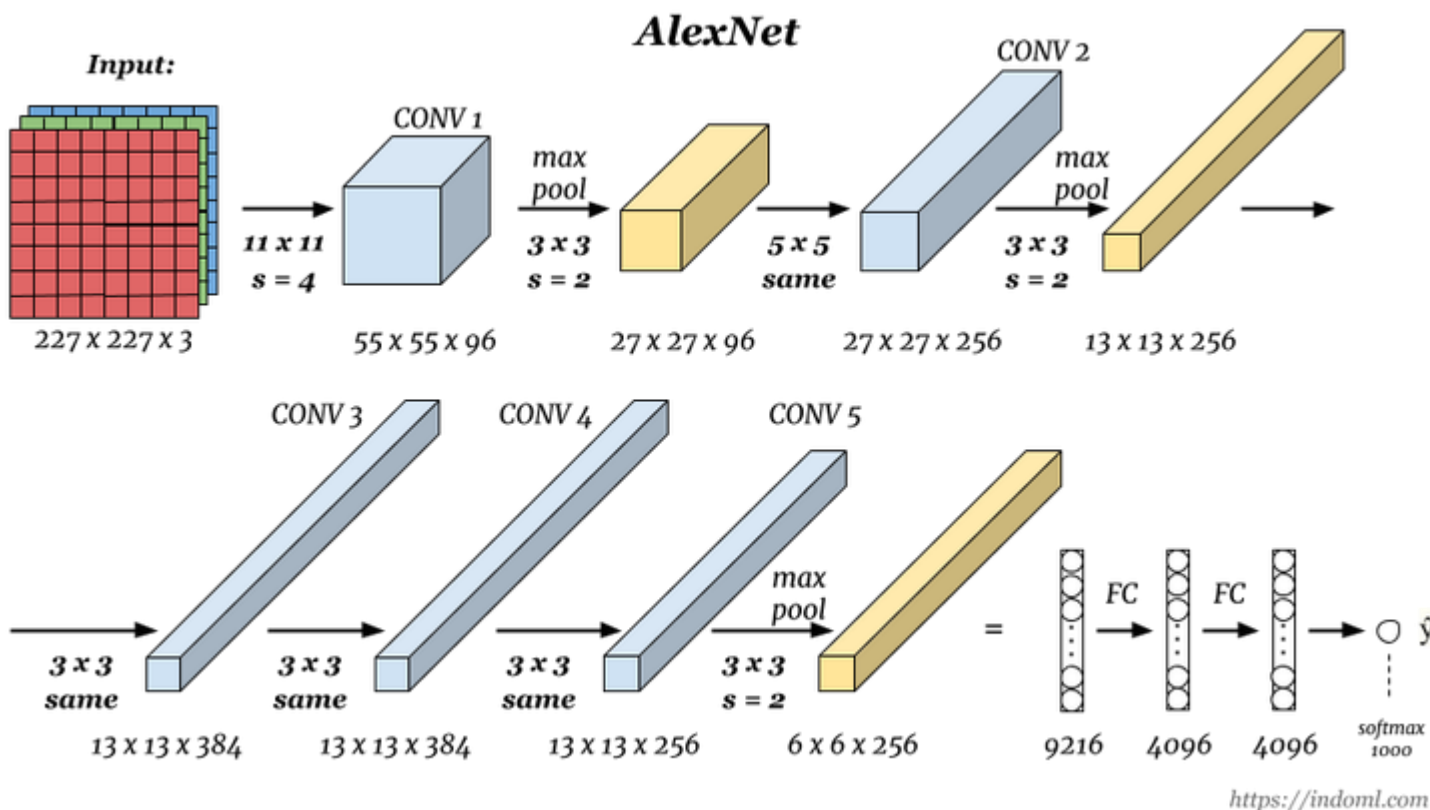
CNN network

- 하나의 예
- Network는 마음대로 구성 가능



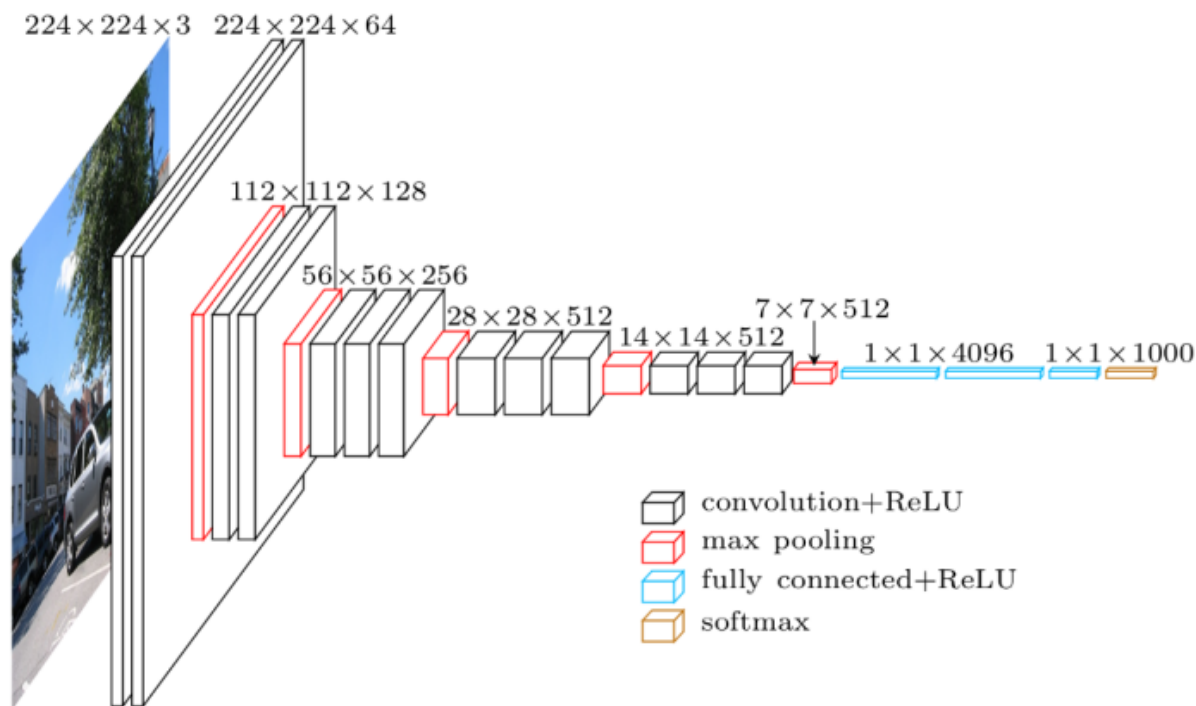
AlexNet

- 이미지 분류(Image Classification)에서 CNN이 큰 주목을 받게 해 준 네트워크



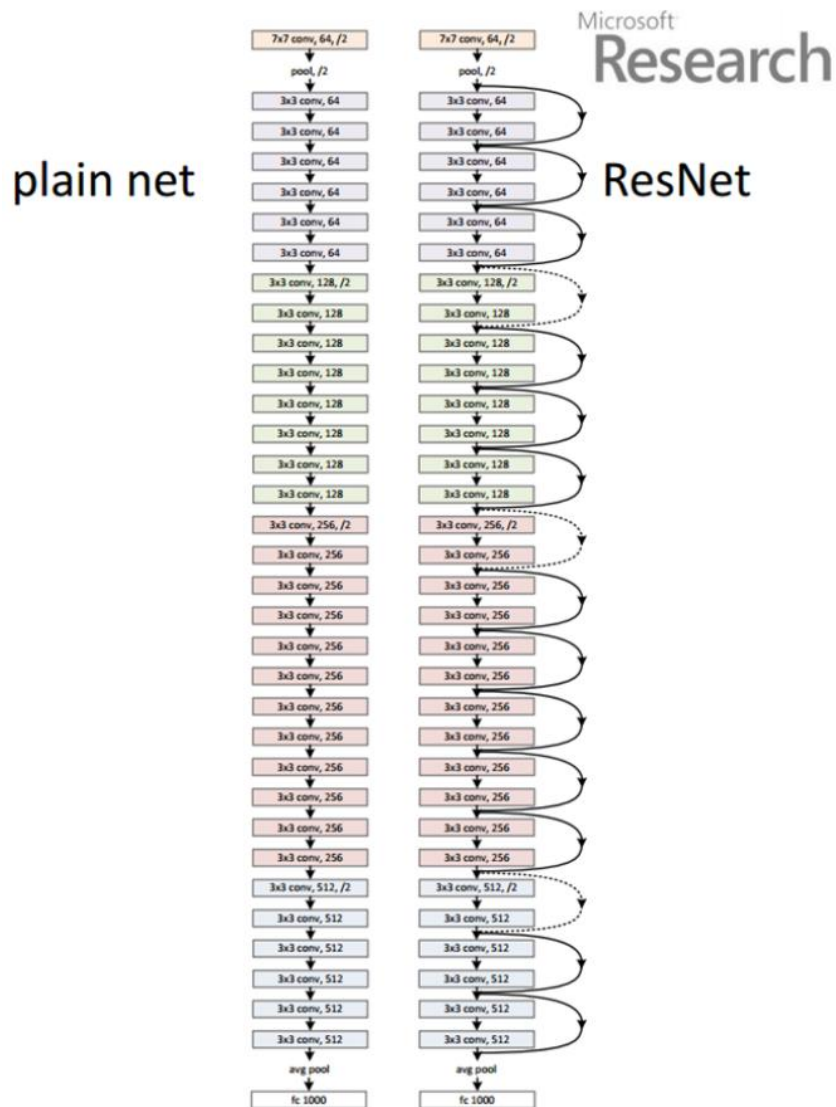
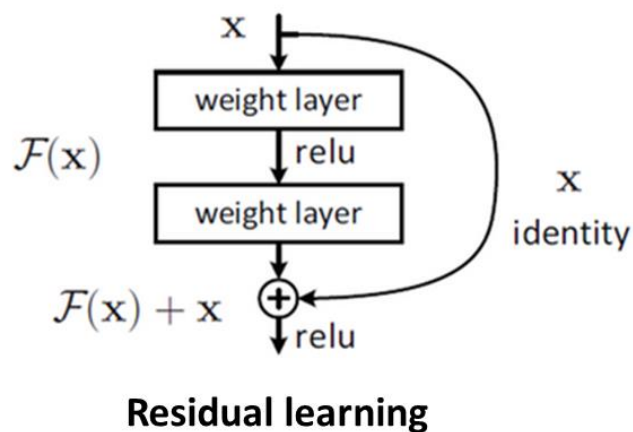
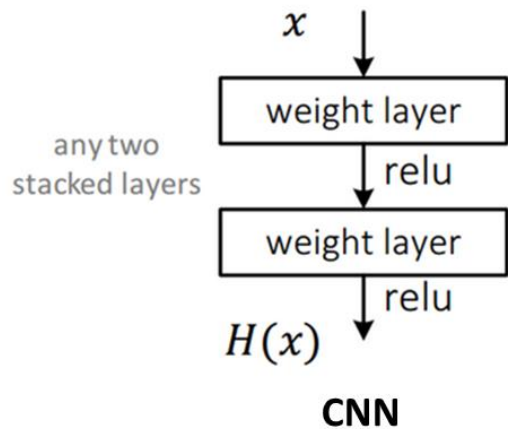
VGGNet

- ILSVRC 2014에서 2등을 한 네트워크
- 레이어를 깊게 쌓았을 때 더 좋은 성능이 나올 수 있다는 것을 보여줌



ResNet

- ILSVRC 2015에서 1등을 한 네트워크
- Skip connection(shortcut connection)과 배치 정규화(batch normalization)를 사용
- 오차 역전파 시 기울기가 쉽게 전파된다는 장점



- <https://paperswithcode.com/sota/image-classification-on-imagenet>
- 2023년 7/28 기준 SOTA(State-of-the-Art)는 BASIC-L
- 사용한 optimizer는 Lion(EvoLved Sign Momentum.
 - Google Brain에서 만든 optimizer, Adam(w)보다 더 잘 동작한다고 알려져 있음)
- Transformer 모델이 대세