



빅데이터와 인공지능의 이해

10.컴퓨터비전

조창제

강의자료

2025.03.

목차

CONTENTS



I

고전 컴퓨터 비전 기술

이미지처리

II

다양한 컴퓨터 비전

연구분야

III

특성맵을 활용한

모델 해석





1.이미지

1) 이미지는 width와 height, channel의 array형태

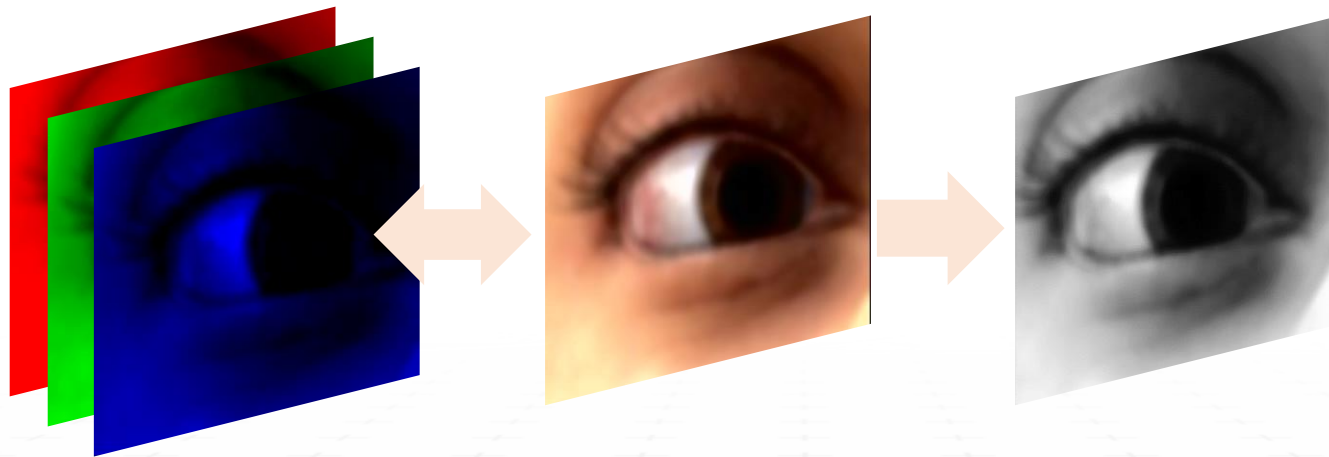
① (width, height, channel) \rightarrow (256, 256, 3)

- array의 범위는 2^8 인 256개의 값을 가지며, 0부터 시작하므로 0~ 255의 범위를 가짐

2) 흑백 변환

① $\text{Gray} = 0.299 R + 0.587 G + 0.114 B$

- 인간이 느끼는 밝기 기여도 기준





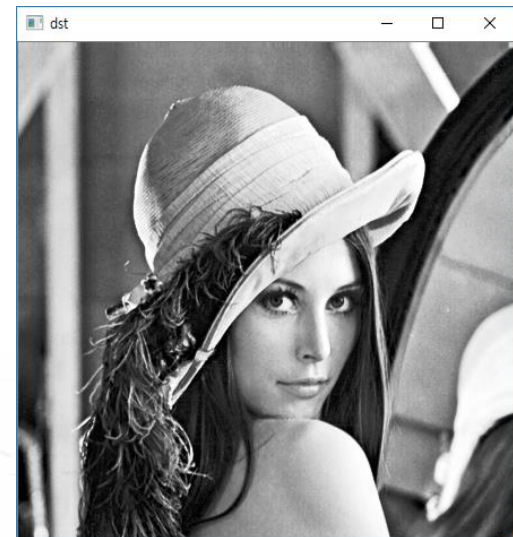
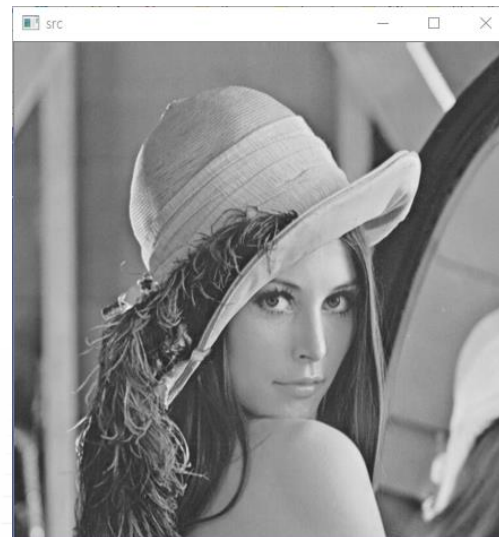
1.이미지 변환

1) 이미지 변환

- ① Histogram Equalization: 이미지의 밝기 범위를 균등하게 분포시키는 방법
- ② Blurring: 이미지를 흐리게 만들어 노이즈를 제거하거나 이미지를 부드럽게 만드는 방법
- ③ Affine Transform: 회전, 크기조정, 이동

2) 특징 추출

- ① Edge detection: 이미지에서 객체의 경계를 감지하는 방법
 - 소벨필터, 캐니 엣지 검출 등





1.분류 모델의 이해

1) CNN의 동작원리

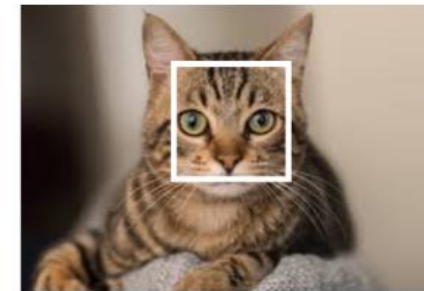
① 사람의 인지 과정

- 사람이 사물을 판단할 때는 거시적으로 형태를 파악하고
판별이 어려울 때에는 자세한 부분을 파악

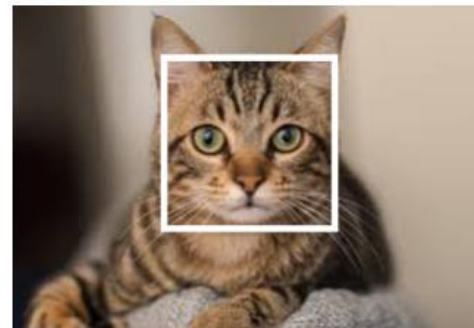
② CNN의 인지 과정

- CNN의 인지 영역은 동일한데 레이어가 깊어질수록
이미지 자체를 작게 인식
- 따라서, 초반 레이어는 털과 같은 텍스처를 인지하고
후반 레이어는 형태를 파악함

③ 데이터의 특성에 따라 레이어 깊이를 선택하는 것이 중요



사람의 인지과정



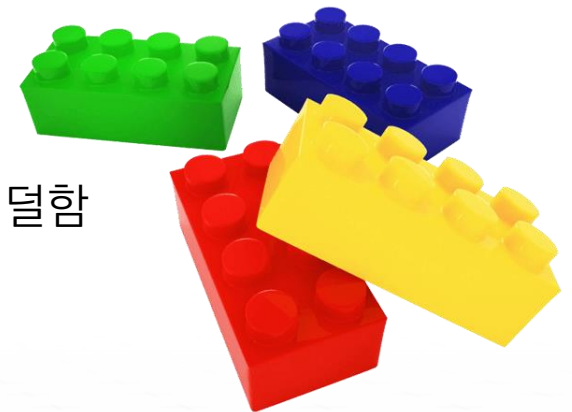
CNN의 인지과정



1.분류 모델의 이해

1) 좋은 레이어 구성(Conv>Norm>Activation>Dropout>Pool 순)

- ① 주로 Conv, Norm, Activation 혹은 Norm, Activation, Conv 순으로 사용
- ② Norm > Dropout
 - Dropout > Norm 순으로 구성하면 0으로 대체된 값들로 인해 Norm이 유의미한 통계 값을 얻는데 방해
- ③ Activation > Dropout
 - Dropout > Activation 순으로 구성하면 활성화 함수가 0을 입력 받음
 - 활성화 함수의 출력 값이 특정 값으로 반환되므로 모델에 안 좋은 영향을 줌
- ④ Dropout > Pool
 - Pool 과정 이후 Dropout 시 정보를 요약한 다음 뉴런을 비활성화
 - Dropout 이후 Pool 시 뉴런을 비활성화 한 후 정보를 요약하므로 정보의 손실이 덜함



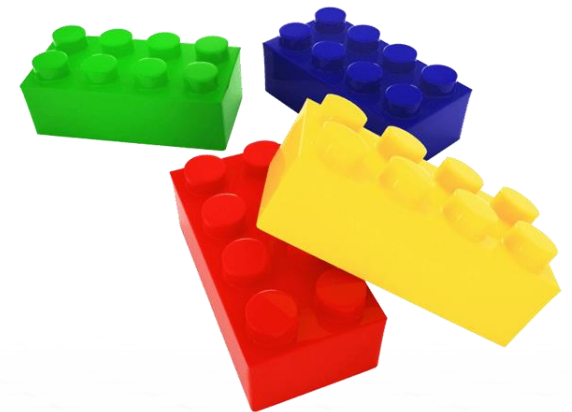


1.분류 모델의 이해

1) 안 좋은 레이어 구성의 예

① Conv > Norm > Act > Pool > Norm > Conv > Act 순

- Max Pool은 최대값만을 취하므로 분포에 변화를 주지 않음
- Pool을 수행한 후 다시 Norm을 통해 정규화 한 후 것은 정규화를 두 번 수행하는 것과 유사하므로 비효율적
- Conv > Norm > Act > Pool > Conv > Norm > Act순으로 변환하는 것이 바람직





1.개요

1) 아이디어

- ① 한 도메인에서 학습한 지식을 다른 도메인으로 전이하여 새로운 작업을 수행하는 방법
 - 유사한 도메인에서 효과적으로 동작
- ② 아는 게 많은 사람은 뭘 배워도 빠르게 배운다는 점을 착안하여 만들어진 학습방식
 - 잘 만들어진 모델을 가져와 특성(Feature)를 추출하면 **작은 데이터에서도 효율적으로 학습가능**

2) CNN모델의 발전과정

- ① 잘 만들어진 모델이라는 점에서 **Classification 모델의 발전과정**과 일맥상통
- ② 잘 만들어진 모델의 기준
 - 1,000개의 클래스를 분류하는 image net 데이터셋에서 우수한 성과를 보인 모델 활용



ILSVRC2012_val_00000001.JPEG



ILSVRC2012_val_00000002.JPEG



ILSVRC2012_val_00000003.JPEG



ILSVRC2012_val_00000004.JPEG



ILSVRC2012_val_00000005.JPEG



ILSVRC2012_val_00000006.JPEG



ILSVRC2012_val_00000007.JPEG



ILSVRC2012_val_00000008.JPEG



ILSVRC2012_val_00000009.JPEG



ILSVRC2012_val_00000010.JPEG



ILSVRC2012_val_00000011.JPEG



ILSVRC2012_val_00000012.JPEG



ILSVRC2012_val_00000013.JPEG



ILSVRC2012_val_00000014.JPEG



ILSVRC2012_val_00000015.JPEG



ILSVRC2012_val_00000016.JPEG



ILSVRC2012_val_00000017.JPEG



ILSVRC2012_val_00000018.JPEG

1.발전과정

1) 모델 발전

① LeNet > AlexNet > VGGNet > InceptionNet > ResNet > DenseNet > MobileNet > EffcientNet

2) LeNet

① Convolutional Layer와 Pooling의 조합을 반복하는 현대적인 CNN구조를 제안한 모델

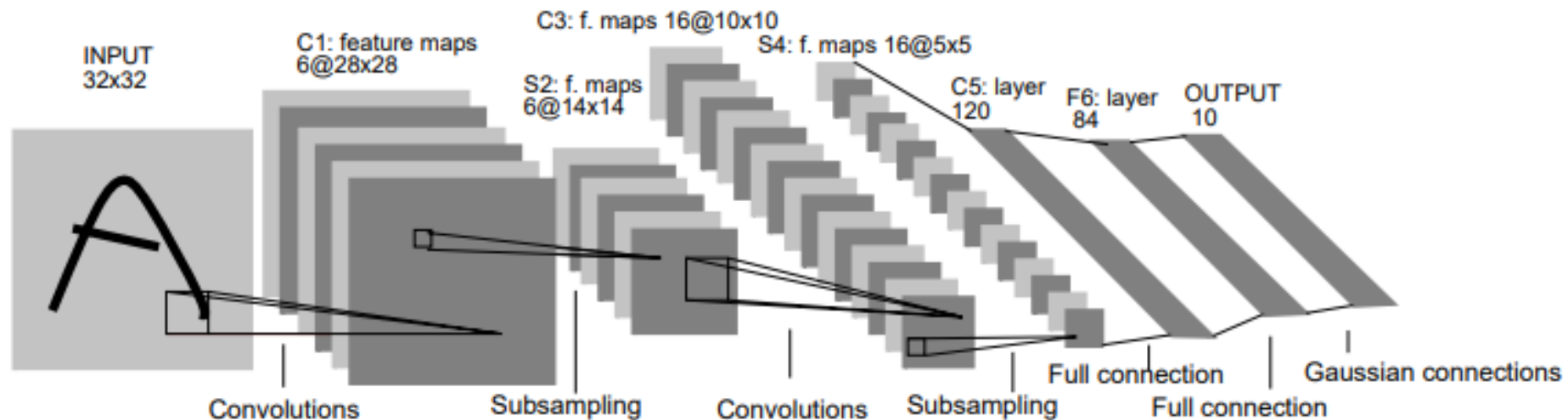


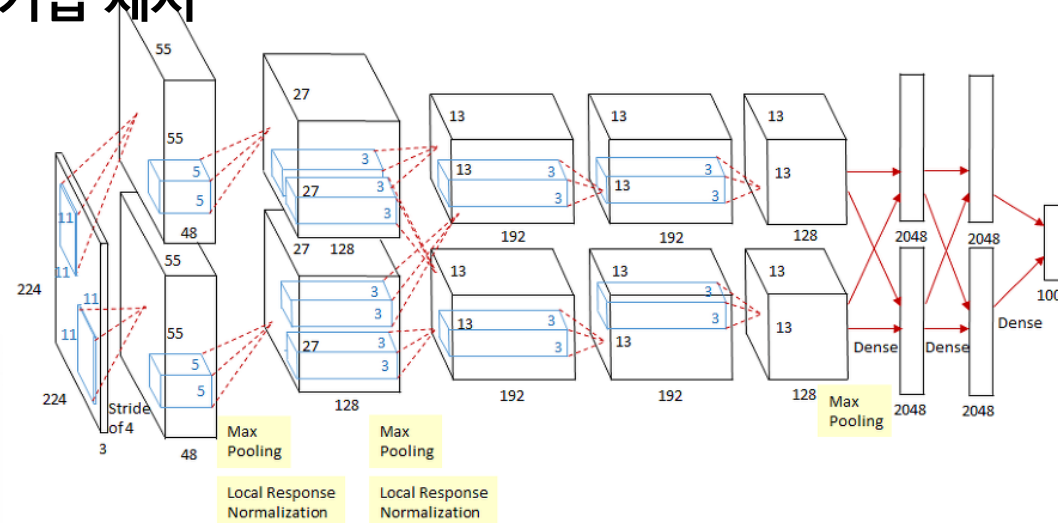
Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.



1.발전과정

1) AlexNet

- ① ImageNet에서 우수한 성과를 나타낸 모델
- ② vram 부족으로 Layer를 2 줄기로 나누어 학습을 수행(Channel을 2배한 것과 유사한 효과)
- ③ **ReLU로 활성화함수를 제시**
 - 기존: tanh 혹은 sigmoid
 - 효과: 학습 속도 향상
- ④ LRN(Local Response Normalization)이라는 정규화 기법 제시
- ⑤ **MaxPooling, Dropout**제시



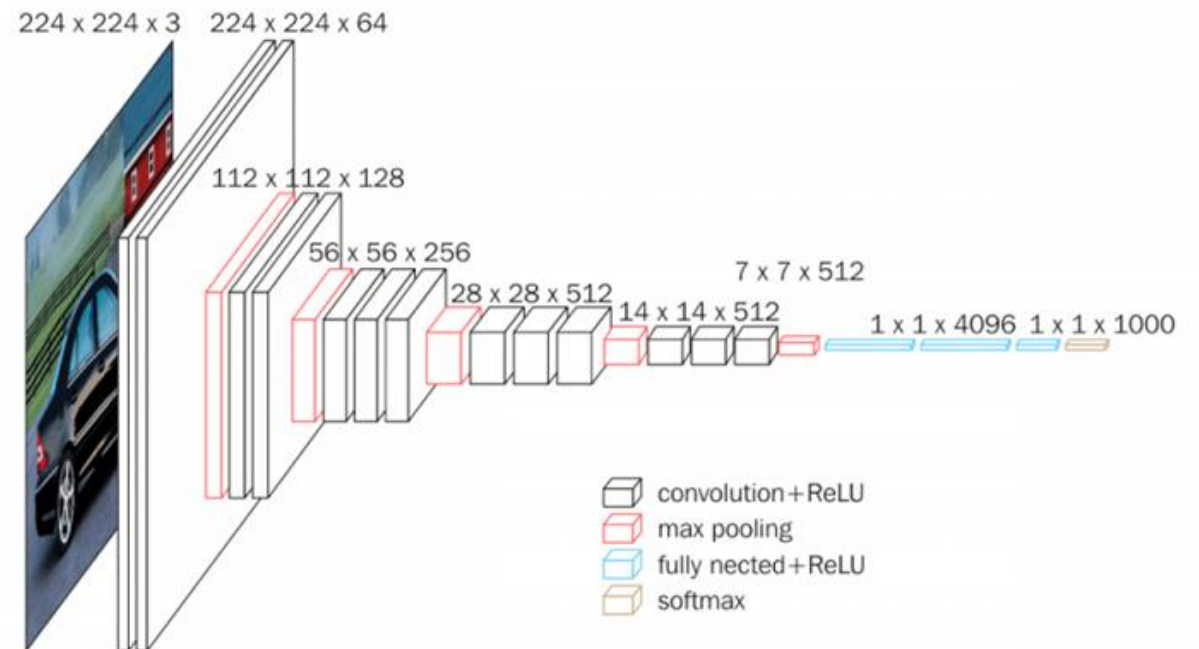


1. 발전과정

1) VGGNet

① kernel size가 크면 이미지 축소가 급격하게 이루어짐

- 깊은 층을 만들기 어려움
- 파라미터 수가 많아져 연산량이 많아짐
- kernel size를 3x3으로 고정

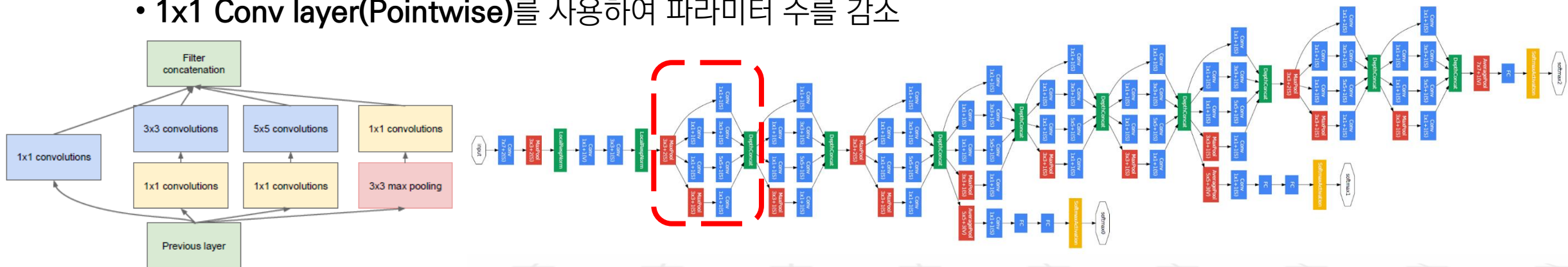




1. 발전과정

1) Inception Net(Google Net)

- ① Deep한 layer 구조로 AlexNet 대비 파라미터가 1/12배 적음에도 우수한 성과를 보임
- ② 레이어의 깊이가 깊어짐에 따라 기울기 소실 및 Overfitting 문제가 발생
 - GAP(Global average pooling) 적용
 - Auxiliary classifier를 사용: 학습 시에 중간 레이어에 output을 만들어 기울기 소실 문제를 방지)
 - Inception module
 - Conv layer에 여러 개의 필터를 적용한 구조
 - 다양한 정보를 추출하면서 학습속도를 개선
 - 1x1 Conv layer(Pointwise)를 사용하여 파라미터 수를 감소

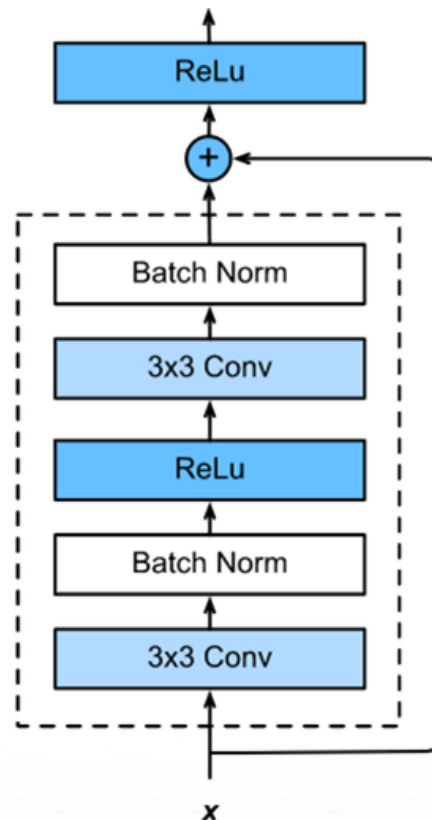




1.발전과정

1) ResNet

- ① 레이어의 깊이에 따른 성능 저하가 발생해서는 안된다고 생각
 - Skip Connection 기법을 고안
 - 기존: 입력값을 이용해서 출력값을 완전히 새로 계산
 - 변경: 입력값과 원래 출력하고자 하는 값의 차이를 학습하는 방식
 - $y = F(x) + x$ 형태로 $F(x)$ 는 잔차를 의미
 - $F(x)$ 가 0이 되어도 x 의 값이 남으므로 기울기 소실 문제가 해결되었음
 - 덧셈 연산만 추가된 것이므로 연산량 또한 부담 없음



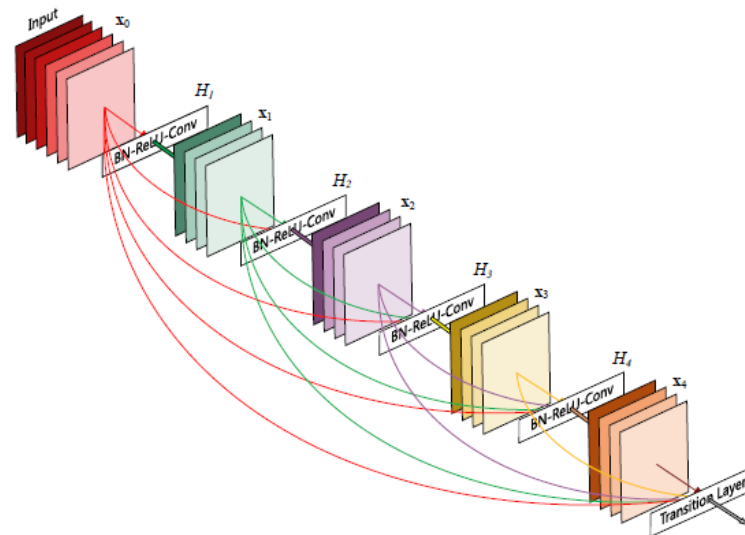


1.발전과정

1) DenseNet

① ResNet과 비슷하게 특성을 연결

- Skip connection 과 유사하지만 **add 연산 대신 concatenate를 활용**
 - concatenate했기때문에 이전 레이어의 출력 모두를 사용 가능
 - 특성 정보가 잘 전달
 - 기울기 소실 문제 방지

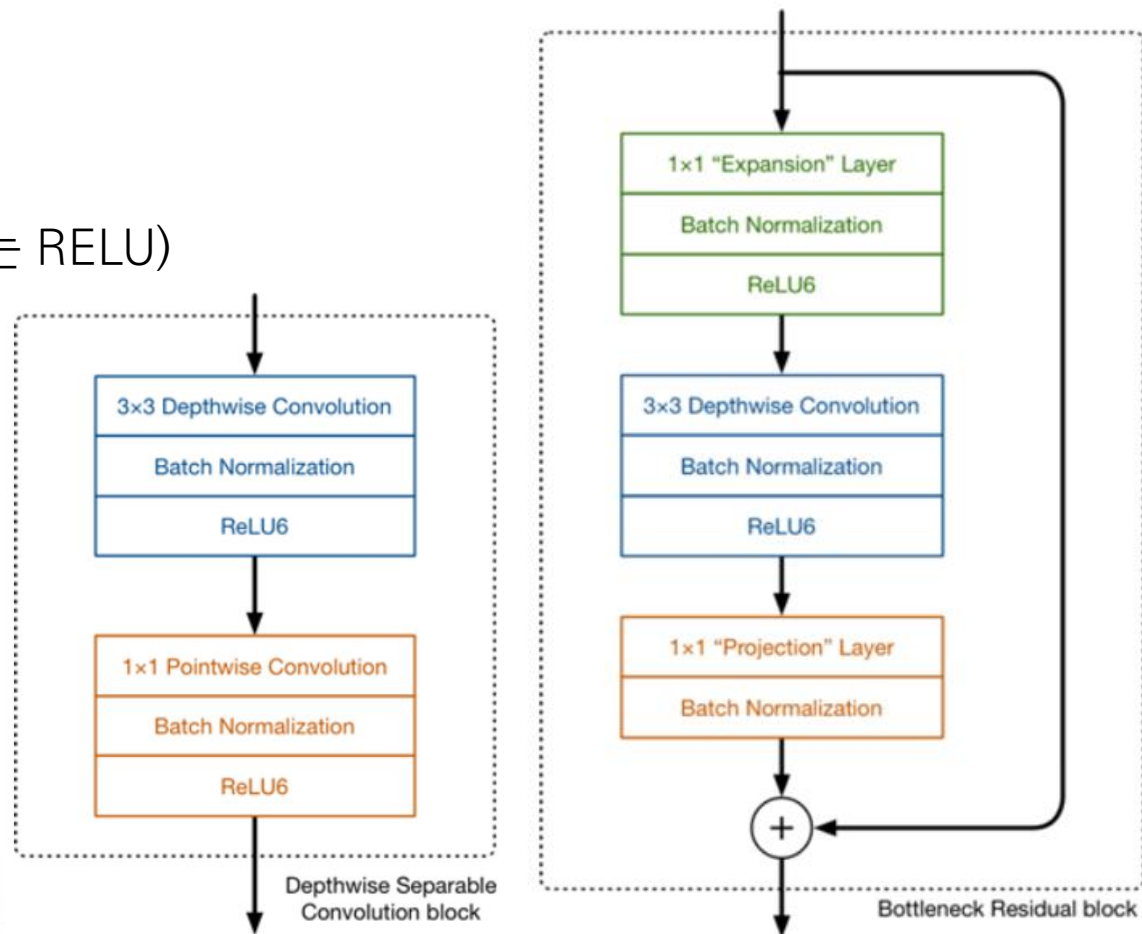
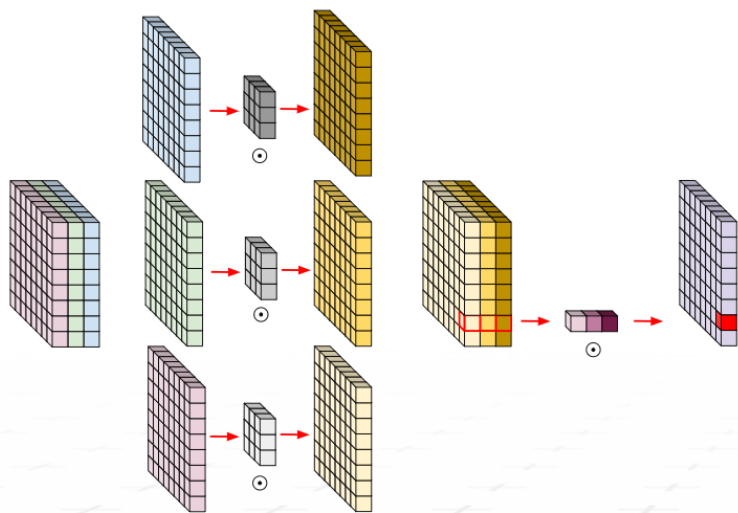


1.발전과정

1) MobileNet

① Depthwise Separable Convolution 제시

- ReLU6 사용으로 연산량 감소(max값을 6으로 사용하는 RELU)
- Inverted Residual Block 적용
 - 1x1 Conv > 3x3 Depthwise Separable Conv
 - 연산량을 효율적으로 감소하며, 성능 저하를 최소화





1.발전과정

1) EfficientNet

① 모델 크기, 연산량, 성능의 균형을 맞추는 것을 목표로 설계

- 기존: 모델 깊이, 너비, 해상도를 개별적으로 조정하여 성능을 향상
- 변경: 깊이, 너비, 해상도를 동시에 균형있게 확장하여 효율적으로 성능을 개선
- 효과: 작은 파라미터와 연산량으로 높은 성능을 달성

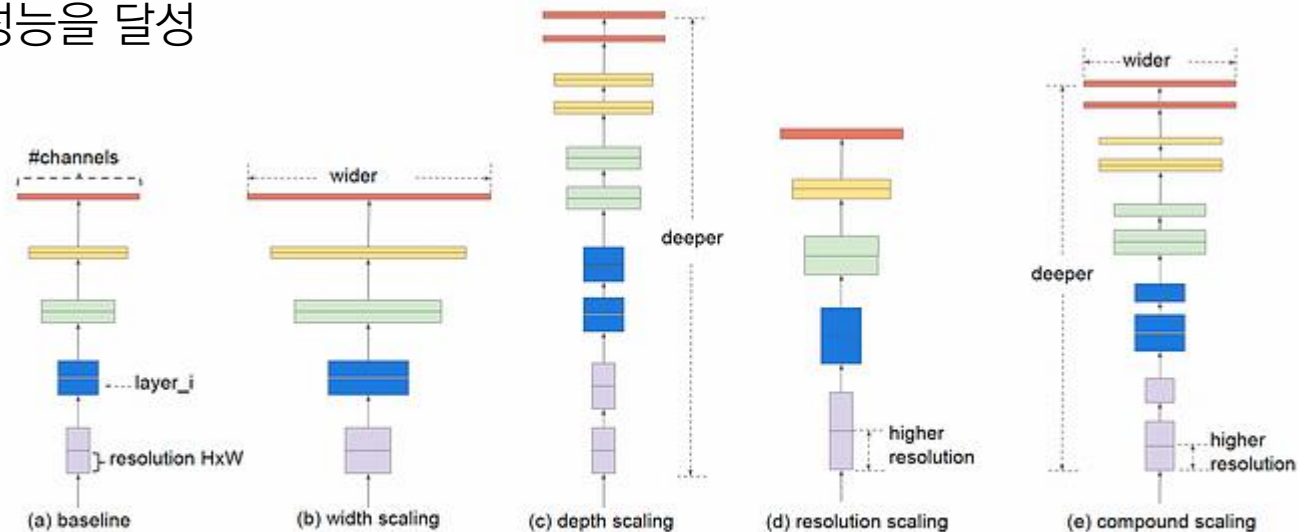


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

1. 분류

1) Semantic segmentation

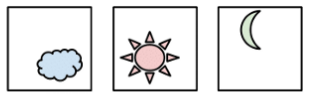
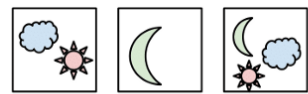
- ① 픽셀에 대한 클래스만을 분류

2) Instance segmentation

- ① 객체 인식 후 객체 분할 수행

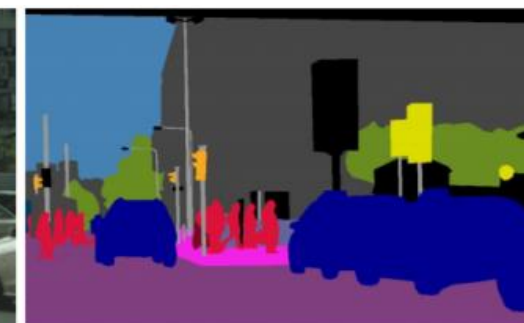
3) panoptic segmentation

- ① 객체 분할 수행을 통해 배경과 객체를 분리
- ② 픽셀단위로 인스턴스를 구분

	Multi-Class	Multi-Label
C = 3		
Samples		
Labels (t)	<p>[0 0 1] [1 0 0] [0 1 0]</p>	<p>[1 0 1] [0 1 0] [1 1 1]</p>



(a) image



(b) semantic segmentation



(c) instance segmentation



(d) panoptic segmentation

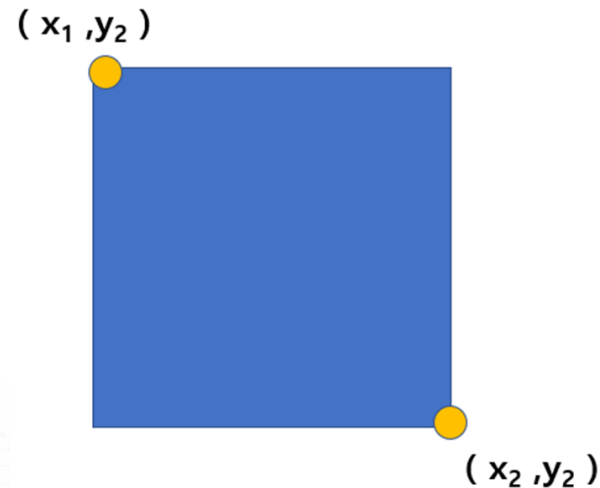
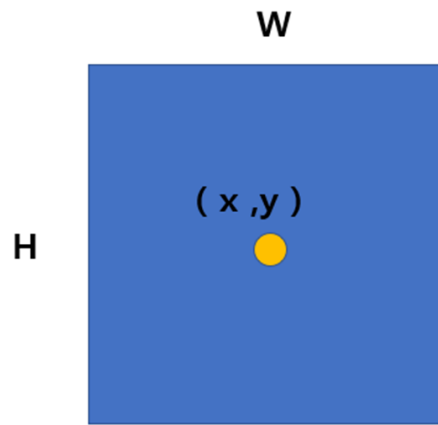
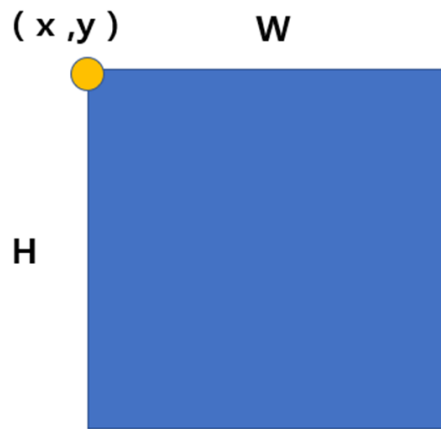


1.레이블 형식

1) 아래 3가지 방식으로 label이 제공될 수 있음

① 중점 방식(Center, Width, Height)

- 객체의 크기나 위치변화에 상대적으로 민감하게 반응할 수 있어 효율적





1. 객체인식(Object detection)

1) 1 stage detector

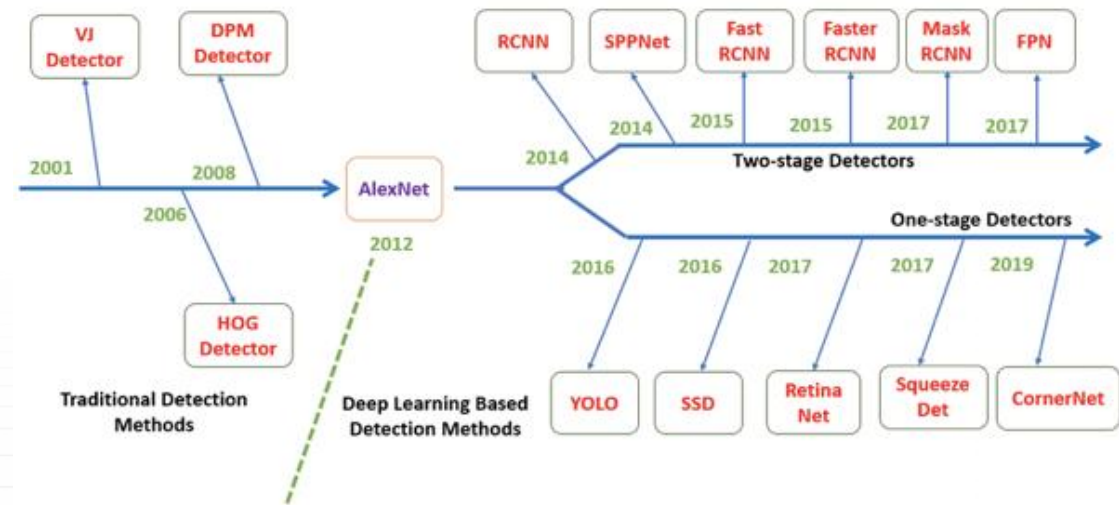
① 물체의 영역과 클래스 한번에 같이 분류

- YOLO
- SSD
- RetinaNet

2) 2 stage detector

① 관심의 영역(ROI)을 찾고, 물체를 분류하는 두가지 단계로 구성

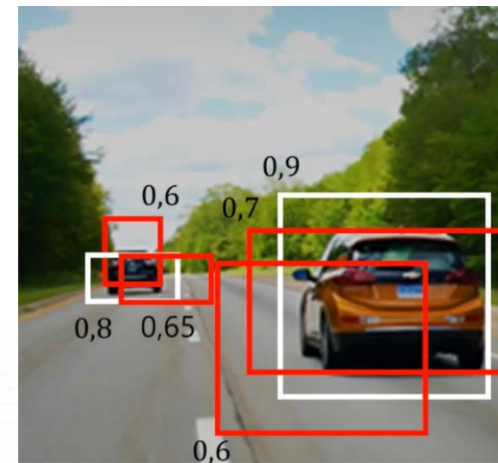
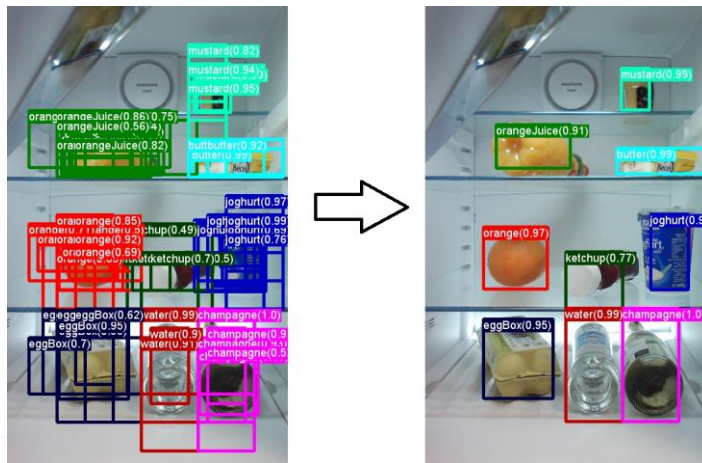
- RCNN
- FastRCNN
- FasterRCNN



1.객체인식(Object detection)

1) NMS(Non Maximum suppression)

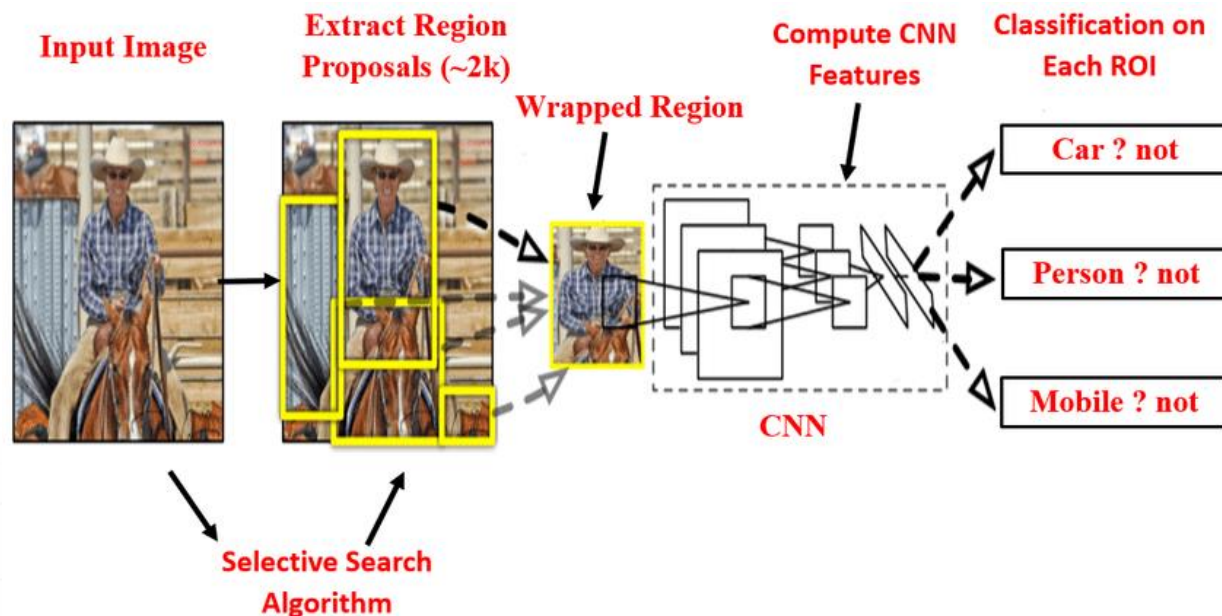
- ① Object Detection 모델 다수가 객체 주변에 아래와 같이 여러 개의 Bounding box가 생김
- ② NMS 알고리즘은 이 중 **하나의 Bounding box만을 선택**하는 알고리즘에 해당
 - Confidence가 Threshold보다 낮은 Bounding box는 제거
 - Confidence 를 기준으로 정렬
 - IOU가 Threshold 보다 큰 박스들은 중복 예측이라 간주하고 Confidence 가 낮은 bounding box를 제거
 - object에 대한 최종 bound box 제시



1.객체인식(Object detection)

1) RCNN

- ① ROI: Selective Search를 통해 Bounding box를 생성하고 계층적으로 군집화하여 제시
 - Selective Search: 색감이나 질감의 차이를 고려하여 유사한 픽셀끼리 묶는 알고리즘
- ② Bounding Box크기 정형화 후 AlexNet을 통해 Feature map 추출
- ③ 추출된 Feature를 통해 SVM모델로 클래스를 분류

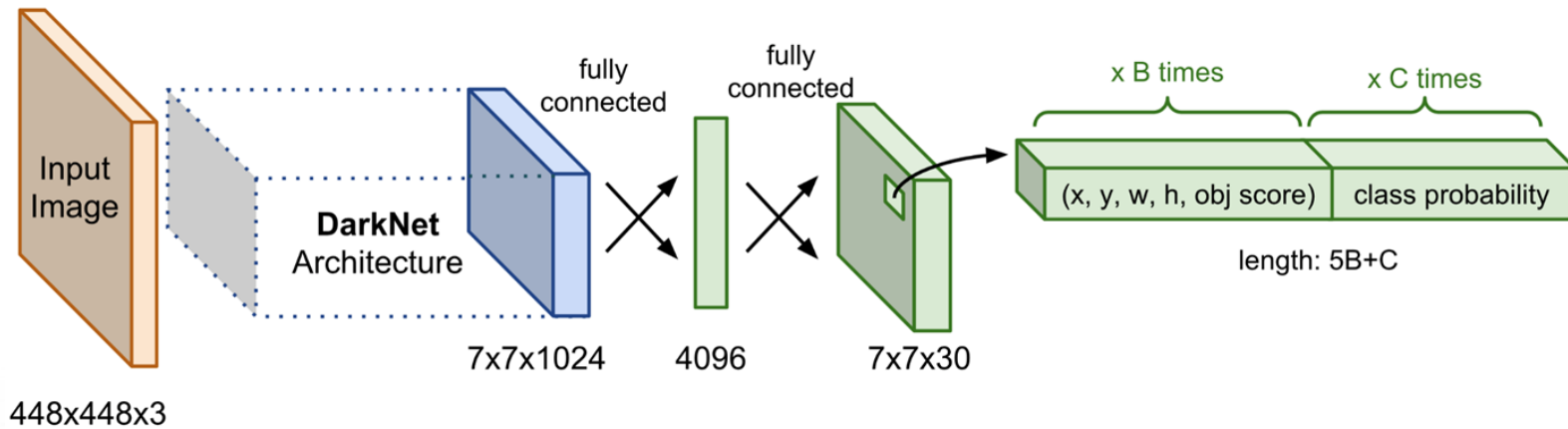




1.객체인식(Object detection)

1) YOLO

- ① DarkNet이라는 독자적인 Backbone을 사용
- ② Backbone의 마지막 레이어의 cell 마다 2개씩 bounding box 제시하게 설계
 - 마지막 레이어의 width, height이 7x7이므로 $49 \times 2 = 98$ 개의 Bounding Box 제시
- ③ Output Layer를 $(x, y, w, h, \text{Confidence}) \times 2 + 20$ 개 클래스에 대한 Probability로 제시
 - $5 \times 2 + 20 = 30$

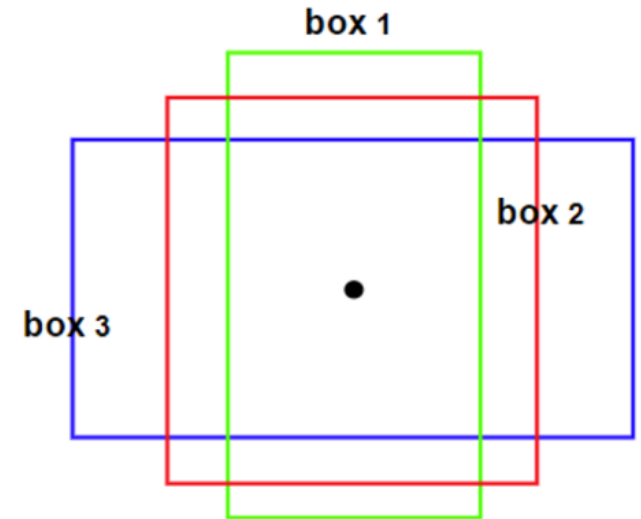
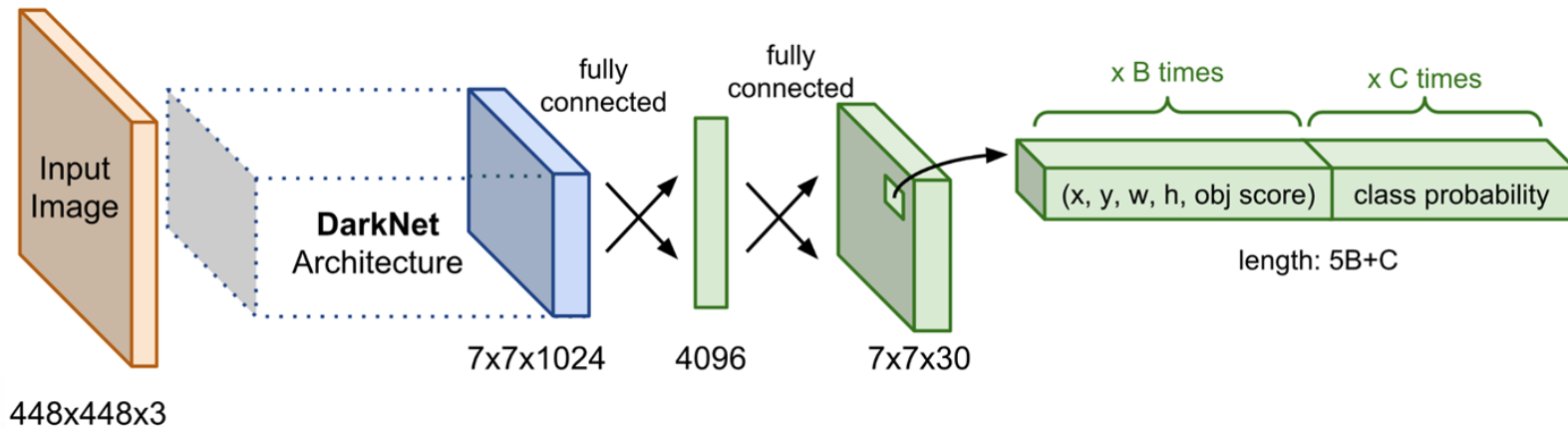




1.객체인식(Object detection)

1) YOLO

- ① Version 2 부터 성능향상을 위해 Anchor를 사용
 - Anchor box의 x, y, w, h 변화량을 추정
- ② Anchor: 다양한 크기와 종횡비의 사각형을 의미



Anchor Box 예시



05.지식 증류(Knowledge distillation)

1.개념

1) 아이디어

- ① 선생이 학생을 가르치는 것을 모방
- ② 잘 학습된 모델(Teacher)을 작은 크기의 모델(Student)로 지식을 전이하는 것을 의미
 - 기존 레이블 방식(Hard label) 이 아닌 Teacher의 Probability를 Label로 활용하는 **Soft label 방식**을 사용
- ③ 경량화 효과를 줄 수 있음

✓ Entropy(2023-10-17)

- 불확실성에 대한 척도
- 확률이 50% 이면 1의 값, 확률이 100%이면 0의 값을 가짐

$$h(p) = - \sum_{i=1}^n p_i \log_2(p_i)$$

✓ Cross Entropy(2023-10-17)

- 모델에서 예측한 확률 값(q)이 실제 값과 비교했을 때 틀릴 수 있는 정보량
- 값이 같으면 0 다르면 inf
- Negative log likelihood라고도 표현 함(엄밀하게 같지는 않음)

$$h(p, q) = - \sum_{i=1}^n p_i \log_2(q_i)$$



✓ Kullbak Leibler Divergence(2023-10-17)

- 경량화를 위한 knowledge distillation 관련 내용
- 두 확률 분포의 차이를 계산하는 데 사용하는 함수
- p와 q의 cross entropy 에서 p의 엔트로피를 뺀 값
- 모델 예측과 실측의 엔트로피와 실측의 엔트로피의 차이(분포 차이로 표현)
- 상대 엔트로피, 정보 획득 량, information divergence라고도 함

$$h(p, q) = - \sum_{i=1}^n p_i \log_2(q_i) - \sum_{i=1}^n p_i \log_2(p_i) - \sum_{i=1}^n p_i \log_2(p_i)$$

$$h(p, q) = h(p) - \sum_{i=1}^n p_i \log_2(q_i) + \sum_{i=1}^n p_i \log_2(p_i)$$

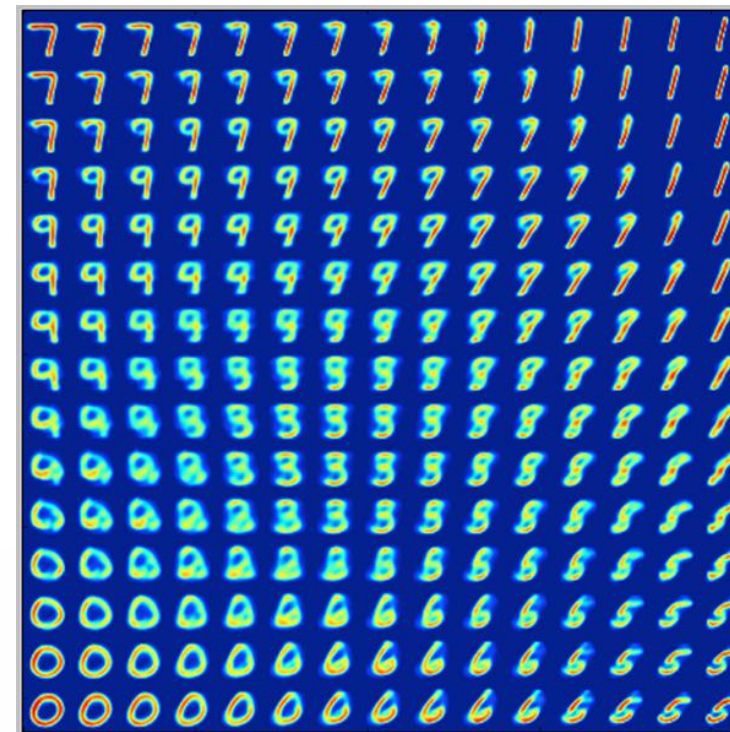
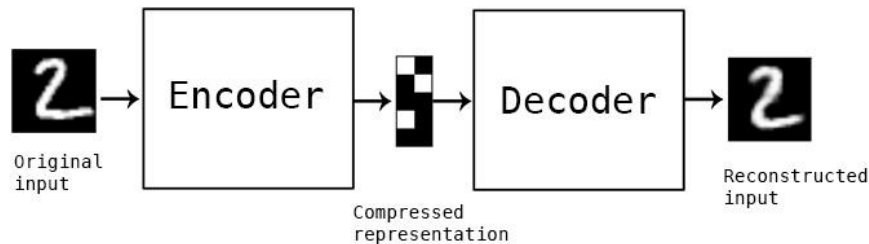
$$h(p, q) = h(p) - \sum_{i=1}^n p_i \log_2\left(\frac{p_i}{q_i}\right)$$

$$h(p, q) - h(p) = - \sum_{i=1}^n p_i \log_2\left(\frac{p_i}{q_i}\right) = D_{KL}(p|q)$$



1.아이디어

- 1) 이미지를 압축하여 잠재공간에 저장하고, 이를 다시 복원하는 알고리즘
- 2) 잠재공간의 변형을 통해 디코더로 새로운 데이터를 생성하는 것을 목표로 함
 - ① Denoising
 - ② Style transfer
 - ③ Super resolution

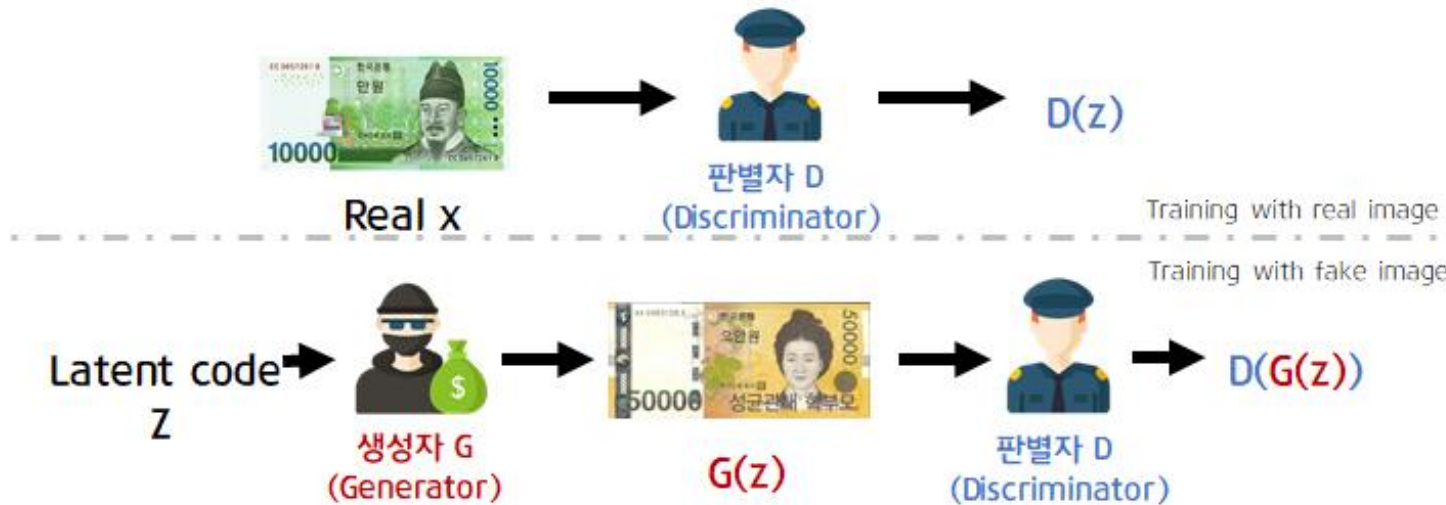


06. Generative adversarial network(GAN)

1. 개념

1) 아이디어

- ① Generator와 Discriminator 두 신경망이 서로 경쟁하며 학습하는 구조
 - Generator: 가짜 데이터를 생성하는 모델
 - Discriminator: 입력된 데이터가 가짜인지 진짜인지 판별하는 모델
- ② Style Transfer, Super resolution, Denoising에 사용 가능

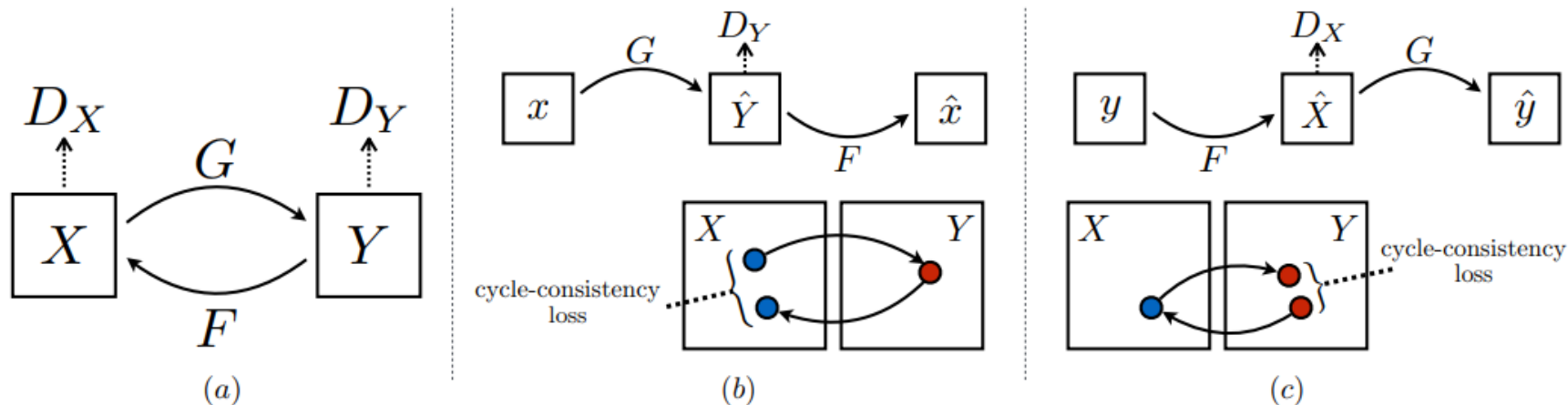


06. Generative adversarial network(GAN)

1. 개념

1) CycleGAN

- ① 생성자와 판별자를 2개씩 사용
- ② 도메인 A to B, B to A 생성
- ③ 손실함수로 Adversarial Loss 외에도 Cycle Consistency Loss와 Identity Loss 사용
 - Cycle Consistency Loss: 도메인 A 이미지를 A to B 생성자로 B 이미지를 생성하고, 이를 다시 B to A 생성자로 복원 시켰을 때 원본 이미지(A)와 유사하도록 학습
 - Identity loss: Target 도메인의 이미지가 입력으로 들어오면, 이미지가 변형되지 않고 유지되도록 학습
 - 도메인 B에 속하는 이미지가 생성자(A to B)를 거쳐 생성된 이미지는 B 이미지와 유사하게 생성되도록 유도





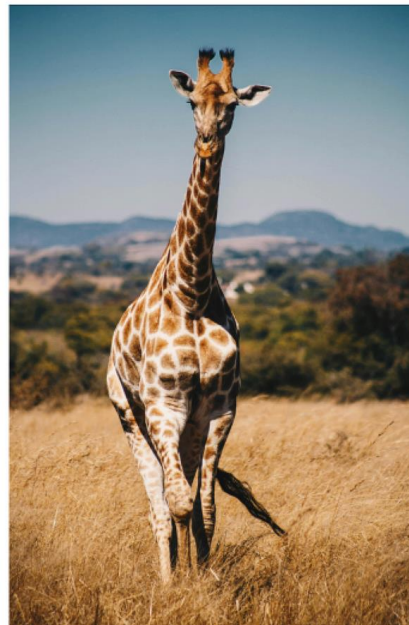
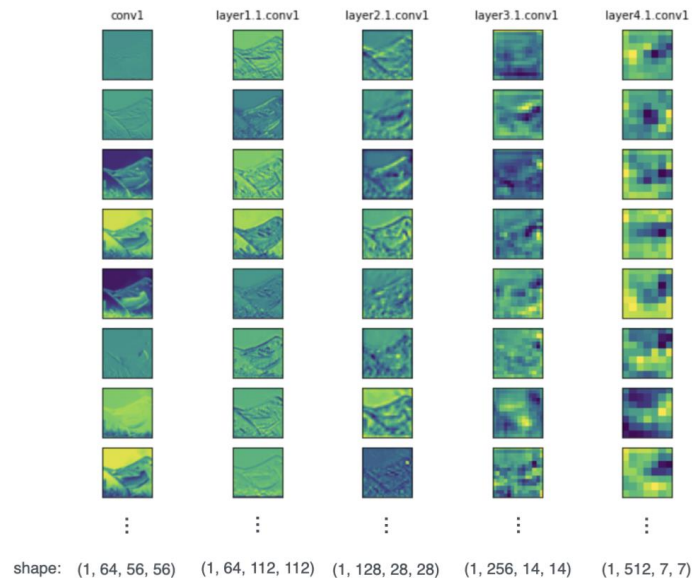
1.CNN 모델의 해석

1) Feature map

- ① 학습된 CNN 모델에서 특정 레이어의 출력을 의미

2) GradCAM

- ① 특정 레이어의 알고자하는 클래스의 출력에서 오차역전파를 수행
- ② 기울기의 GAP값을 시각화



layer2 + layer3 + layer4





Thank You

Email: qkdrk777777@naver.com