

6.1.5 - 6.3

6.1.5 ResNet

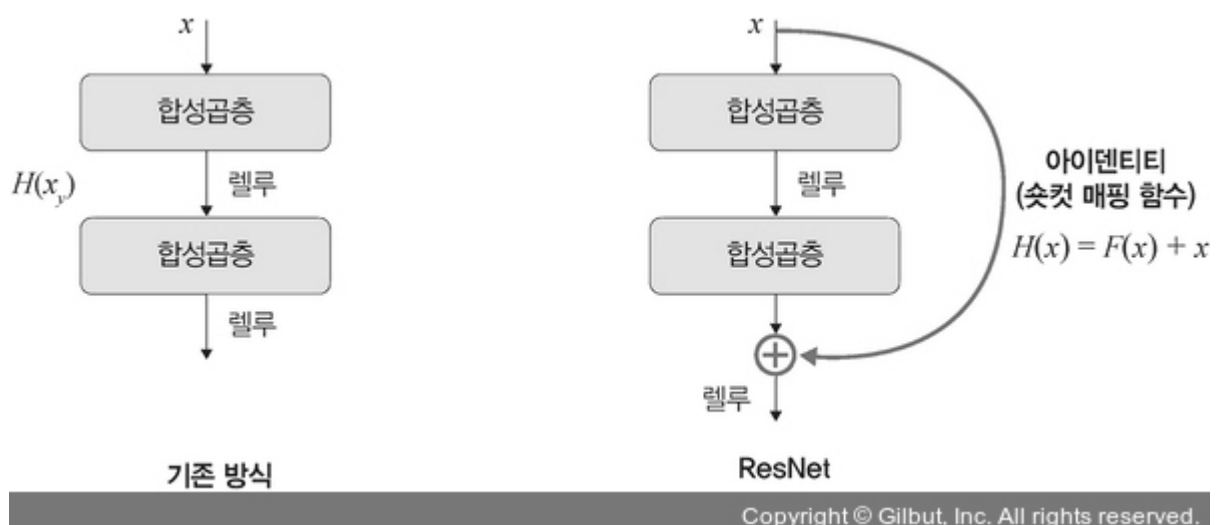
ResNet은 마이크로소프트에서 개발한 알고리즘으로 “Deep Residual Learning for Image Recognition”이라는 논문에서 발표됨.

ResNet 핵심은 깊어진 신경망을 효과적으로 학습하기 위한 방법으로 레지듀얼(residual) 개념을 고안한 것

신경망은 깊이가 깊어질수록 성능이 좋아지다가 일정한 단계에 다다르면 오히려 성능이 나빠짐

→ ResNet은 바로 이러한 문제를 해결하기 위해 레지듀얼 블록(residual block)을 도입.

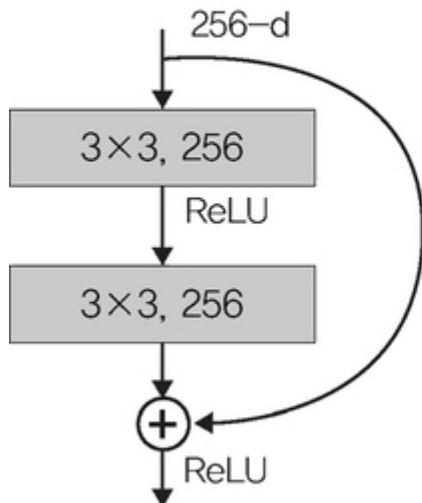
레지듀얼 블록은 기울기가 잘 전파될 수 있도록 일종의 숏컷(shortcut, skip connection)을 만들어줌.



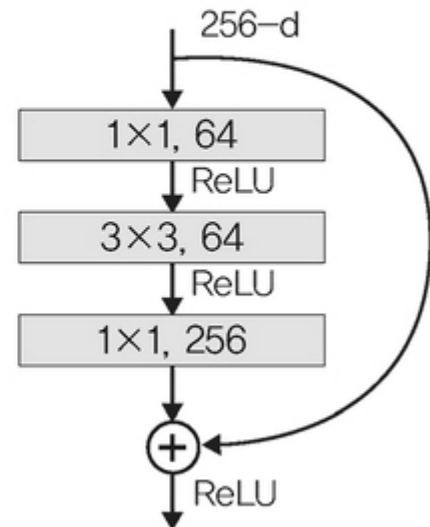


- 합성곱층을 하나의 블록으로 묶음
- 블록을 구분했는데 이렇게 묶인 계층들을 하나의 레지듀얼 블록이라 함
- 레지듀얼 블록을 여러 개 쌓은 것을 ResNet

**ResNet34
레지듀얼 블록**



**ResNet50
레지듀얼 블록**

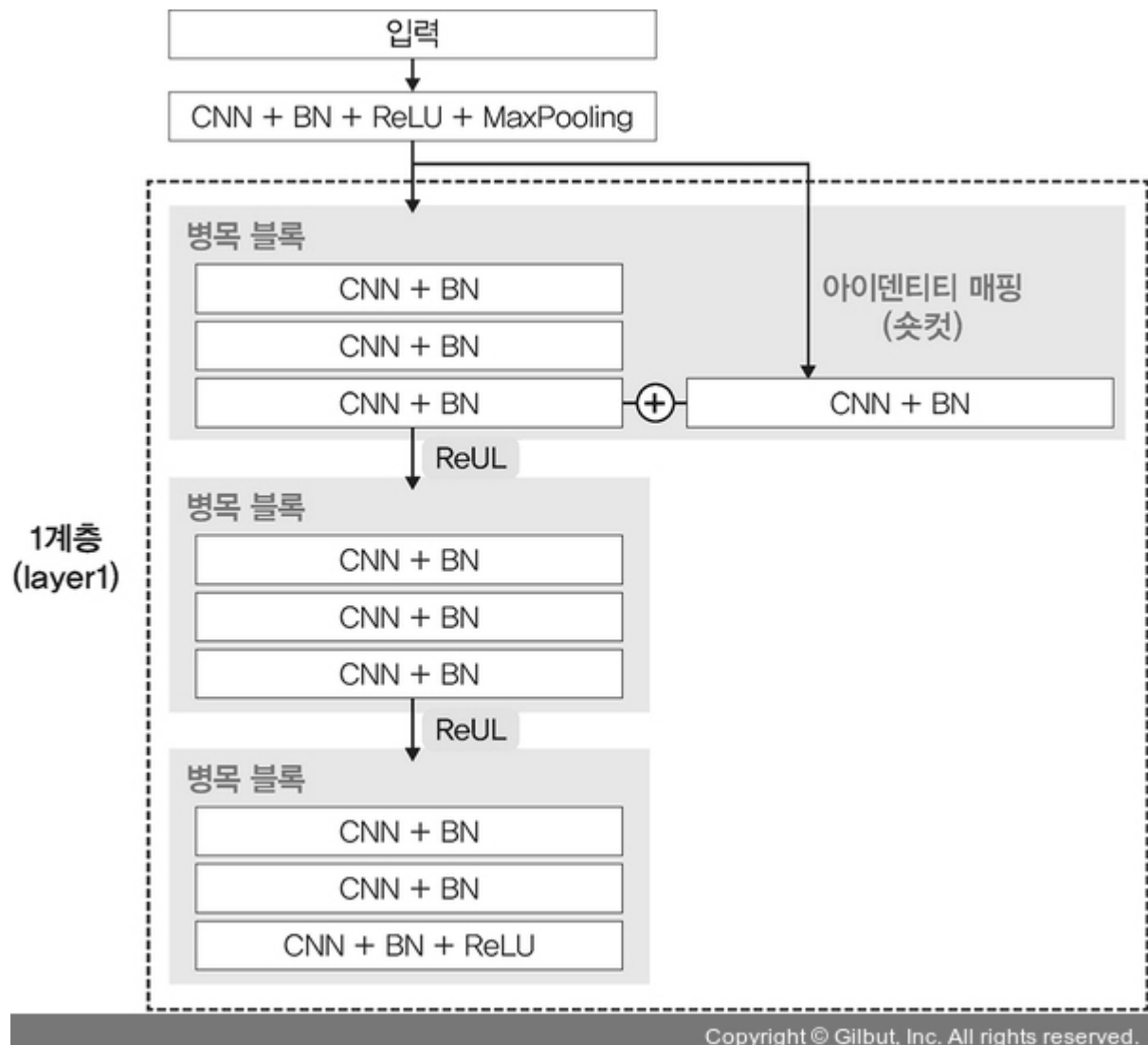


계층의 깊이가 깊어질수록 파라미터가 무제한으로 커지는 문제를 해결하기 위해 병목 블록 (bottleneck block)을 둬.

- ResNet 34는 기본 블록, 50은 병목 블록 사용 → 깊이가 깊어져도 파라미터 수는 감소
 - ResNet34와는 다르게 ResNet50에서는 3x3 합성곱층 앞뒤로 1x1 합성곱층이 붙어 있는데, 1x1 합성곱층의 채널 수를 조절하면서 차원을 줄였다 늘리는 것이 가능하기 때문에 파라미터 수를 줄일 수 있었던 것

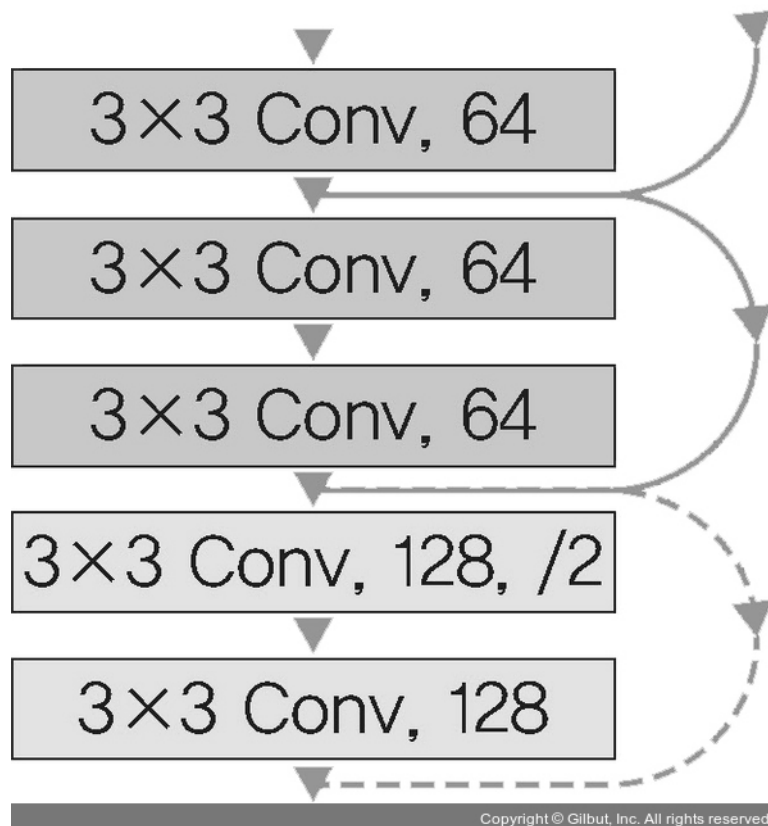
아이덴티티 매핑(identity mapping)(혹은 숏컷(shortcut), 스킵 연결(skip connection))

- 입력 x 가 어떤 함수를 통과하더라도 다시 x 라는 형태로 출력되도록 하는 것



다운샘플(downsample)

- 특성 맵(feature map) 크기를 줄이기 위한 것으로 풀링과 같은 역할을 함.

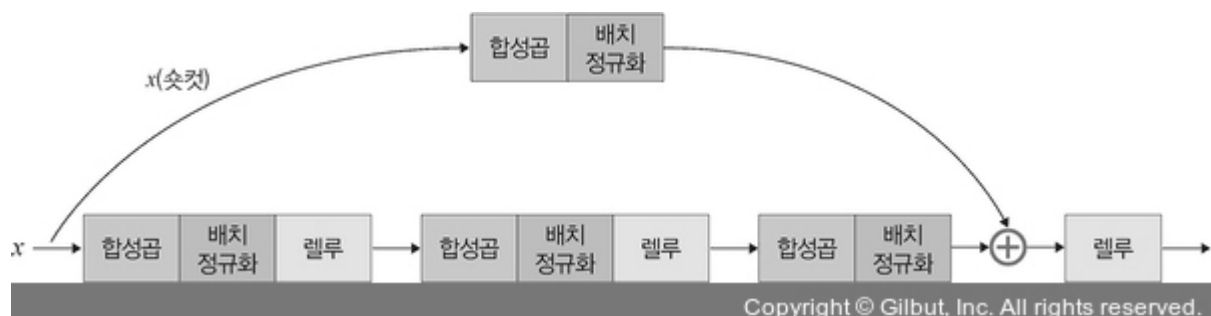


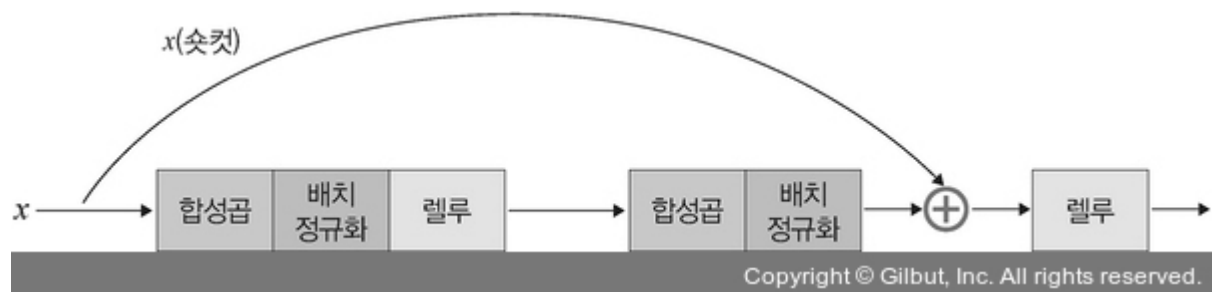
프로젝션 숏컷(projection-shortcut) 혹은 합성곱 블록

- 입력 및 출력 차원이 동일하지 않고 입력의 차원을 출력에 맞추어 변경해야 하는 것

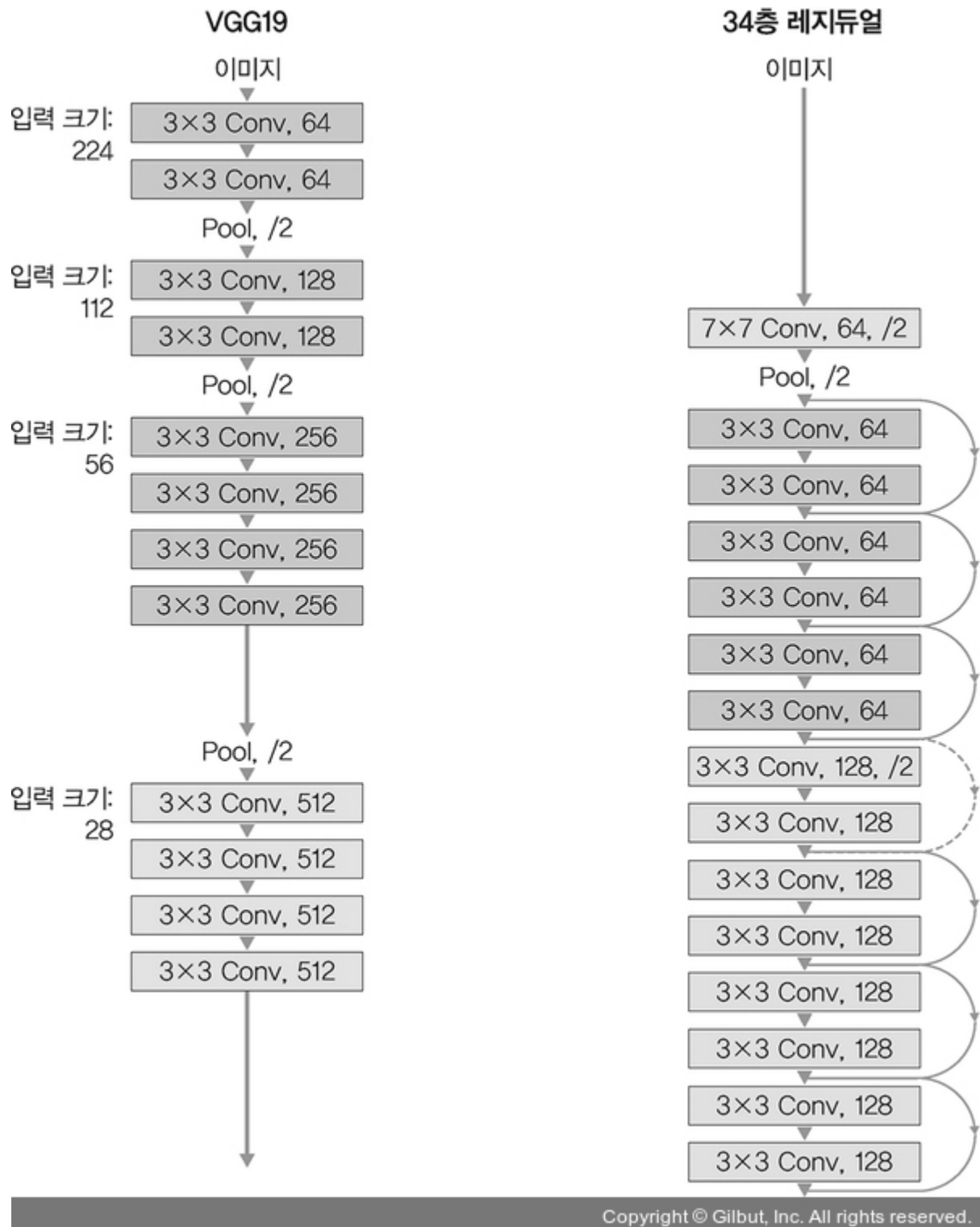


입력과 출력의 차원이 같은 것을 아이덴티티 블록이라고 하며, 입력 및 출력 차원이 동일하지 않고 입력의 차원을 출력에 맞추어 변경해야 하는 것을 프로젝트 숏컷(projection-shortcut) 혹은 합성곱 블록이라고 함.





ResNet은 기본적으로 VGG19 구조를 뼈대로 하며, 거기에 합성곱층들을 추가해서 깊게 만든 후 숏컷들을 추가하는 것!!



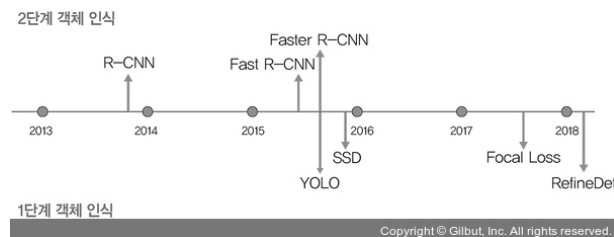
6.2 객체 인식을 위한 신경망

- 객체 인식

- 이미지나 영상 내에 있는 객체를 식별하는 컴퓨터 비전 기술
- 이미지나 영상 내에 있는 여러 객체에 대해 각 객체가 무엇인지 분류하는 문제와 그 객체 위치가 어디인지 박스(bounding box)로 나타내는 위치 검출(localization) 문제를 다루는 분야

객체 인식 = 여러 가지 객체에 대한 분류 + 객체의 위치 정보를 파악하는 위치 검출

딥러닝을 이용한 객체 인식 알고리즘은 크게 1단계 객체 인식(1-stage detector)과 2단계 객체 인식(2-stage detector)으로 나눌 수 있음



1단계 객체 인식

- 분류와 위치 검출을 동시에 행하는 방법
- YOLO(You Only Look Once) 계열과 SSD 계열 등이 포함

2단계 객체 인식

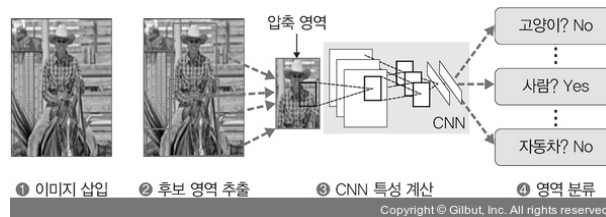
- 분류와 위치 검출을 순차적으로 행하는 방법
- CNN을 처음으로 적용시킨 R-CNN 계열이 대표적

→ 1단계 객체 인식은 비교적 빠르지만 정확도가 낮고, 2단계 객체 인식은 비교적 느리지만 정확도가 높음.

6.2.1 R-CNN

이미지 분류를 수행하는 CNN과 이미지에서 객체가 있을 만한 영역을 제안해 주는 후보 영역 알고리즘을 결합한 알고리즘

R-CNN 학습 절차



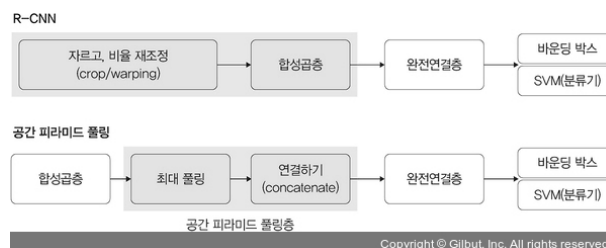
1. 이미지를 입력으로 받습니다.
2. 2000개의 바운딩 박스(bounding box)를 선택적 탐색 알고리즘으로 추출한 후 잘라 내고(cropping), CNN 모델에 넣기 위해 같은 크기(227×227 픽셀)로 통일합니다(warping).
3. 크기가 동일한 이미지 2000개에 각각 CNN 모델을 적용합니다.
4. 각각 분류를 진행하여 결과를 도출합니다.

6.2.2 공간 피라미드 풀링

- 입력 이미지의 크기에 관계없이 합성곱층을 통과시키고, 완전연결층에 전달되기 전에 특성 맵들을 동일한 크기로 조절해 주는 풀링층을 적용하는 기법

입력 이미지의 크기를 조절하지 않고 합성곱층을 통과시키기 때문에 원본 이미지의 특징이 훼손되지 않는 특성 맵을 얻을 수 있음.

또한, 이미지 분류나 객체 인식 같은 여러 작업에 적용할 수 있다는 장점이 있음.

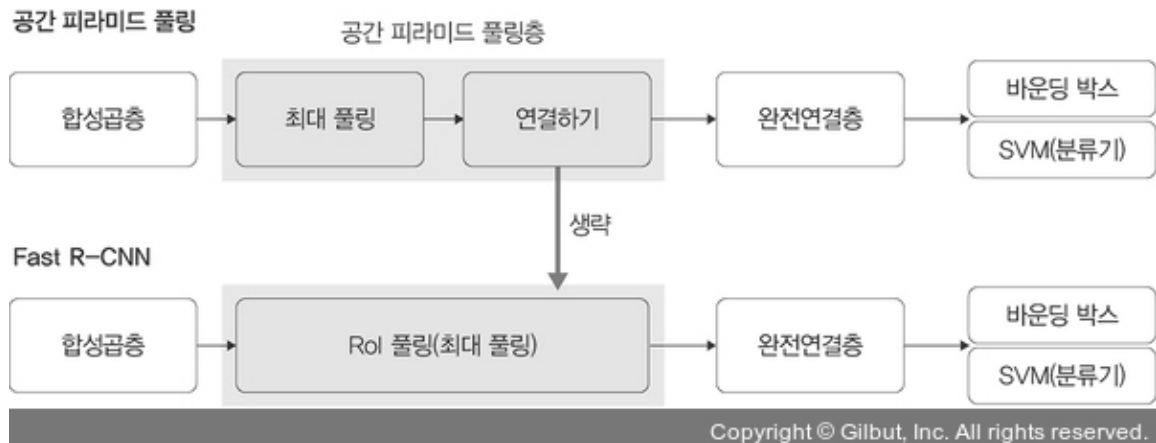


6.2.3 Fast R-CNN

속도 문제를 개선하기 위해 RoI 풀링 도입

선택적 탐색에서 찾은 바운딩 박스 정보가 CNN을 통과하면서 유지되도록 하고 최종 CNN 특성 맵은 풀링을 적용하여 완전연결층을 통과하도록 크기 조정

→ 바운딩 박스마다 CNN 돌리는 시간 단축

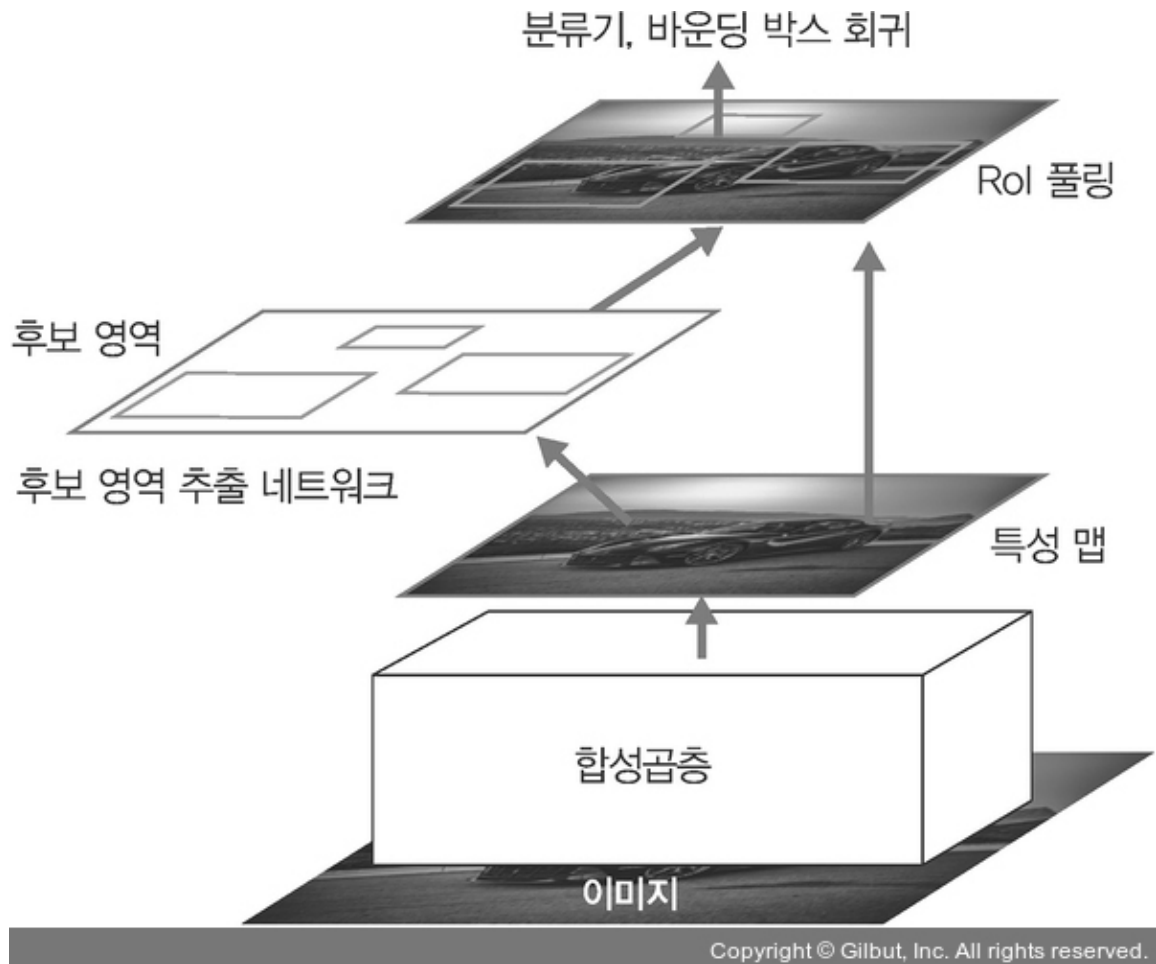


RoI 풀링

RoI 풀링(RoI pooling)은 크기가 다른 특성 맵의 영역마다 스트라이드를 다르게 최대 풀링을 적용하여 결괏값 크기를 동일하게 맞추는 방법.

6.2.4 Faster R-CNN

- '더욱 빠른' 객체 인식을 수행하기 위한 네트워크
- 후보 영역 생성을 CNN 내부 네트워크에서 진행할 수 있도록 설계
- 외부의 느린 선택적 탐색(CPU로 계산) 대신 내부의 빠른 RPN(GPU로 계산)을 사용
- RPN은 마지막 합성곱층 다음에 위치, 그 뒤에 RoI풀링, 분류기, 바운딩 박스 회귀 위치



6.3 이미지 분할을 위한 신경망

- 이미지 분할
 - 신경망을 훈련시켜 이미지를 픽셀 단위로 분할하는 것
 - 이미지를 픽셀 단위로 분할하여 이미지에 포함된 객체 추출