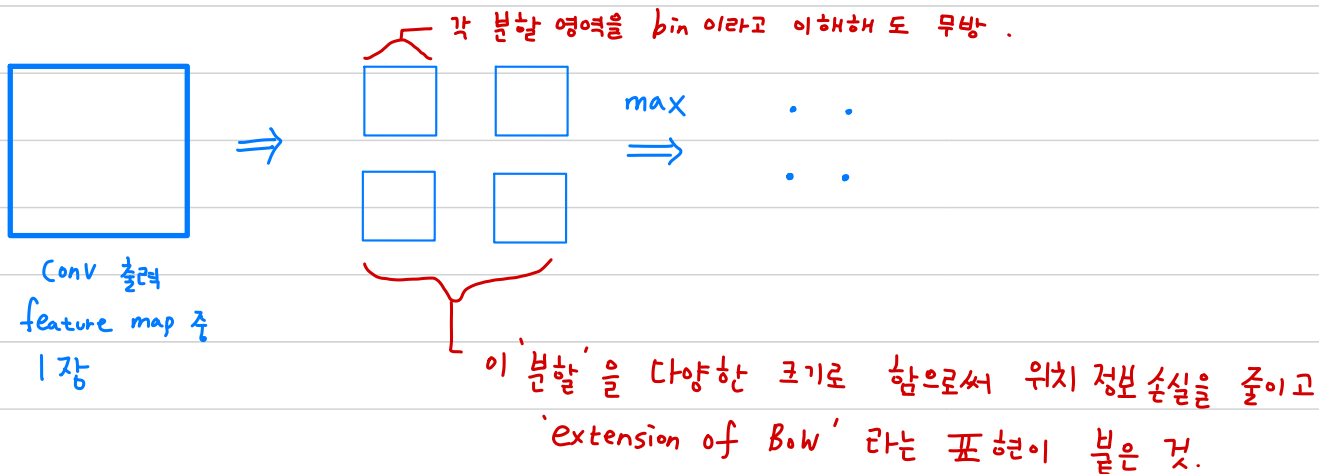
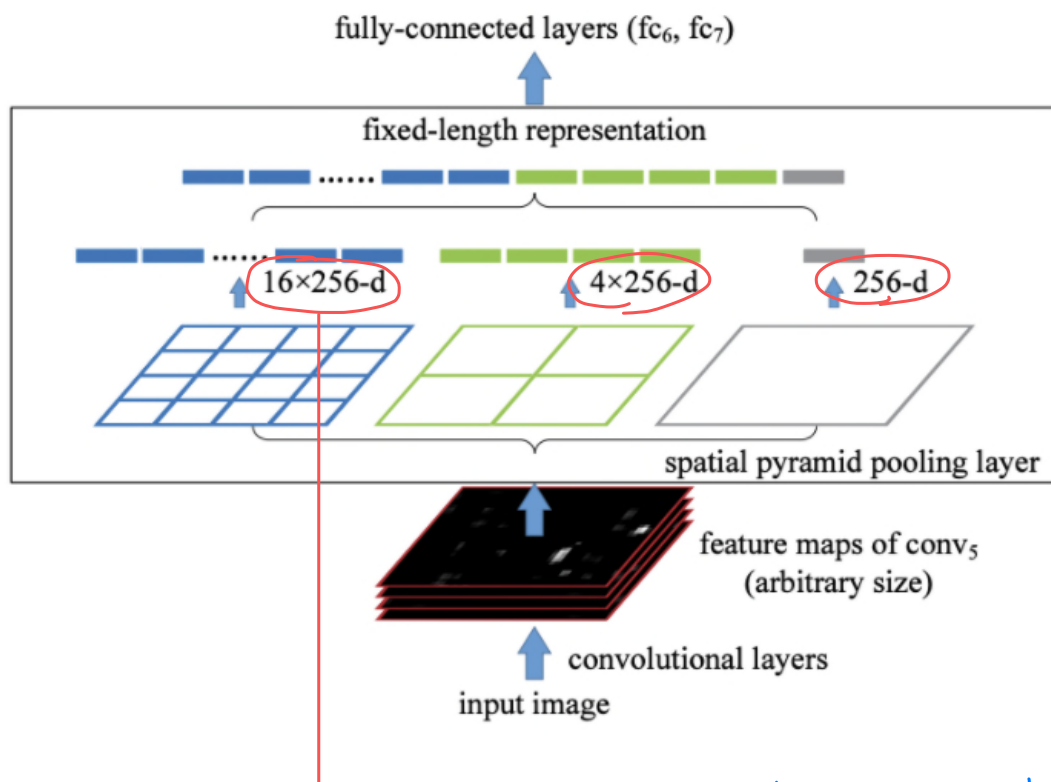


- SPP layer 이론적 아이디어 와 크기 계산 과정



- KM 차원이라는 결과 벡터 계산 이해. ($k = \text{feature map 장 수}$, $M = \text{총 bin 수}$)



		Level 4 (4×4 bin)	Level 2 (2×2 bin)	Level 1 (1×1 bin)
1 장	입력으로 들어 갔을때	16×1	4×1	1×1
2 장	//	16×2	4×2	1×2
⋮				
n 장	//	$16 \times n$	$4 \times n$	$1 \times n$
⋮				
k 장	//	$16 \times k$	$4 \times k$	$1 \times k$

최종: $\text{Concat} (16 \times k, 4 \times k, 1 \times k) = 21 \times k \quad (21 = M)$

- SPP layer 와 헷갈릴 수 있는 개념들

1. GAP

텐서 입력 \rightarrow 벡터 출력이라는 관점에서 비슷할 수 있으나

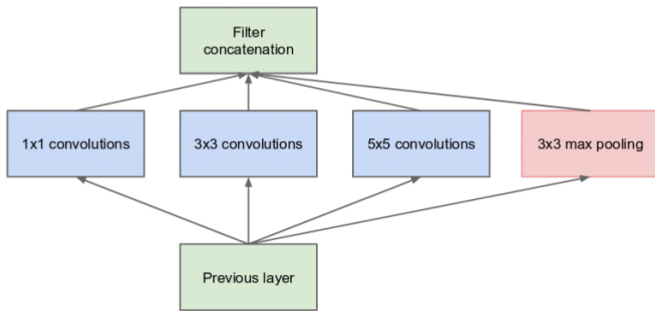
GAP 가 더 작은 개념! SPP layer 에서 level 1 층은 결국 GAP 랑 같다
(Avg 대신 max 지만)

실제로 $a \times a \times k$ 사이즈 feature map 이 들어왔을 때 출력 사이즈 다름

after GAP = k 사이즈 벡터

after SPP layer = $M \cdot k$ 사이즈 벡터 ($M = \text{total \# of bins}$)

2. Inception module



(a) Inception module, naïve version

뒤에 training 내용을 보고보면 inception module 이랑 뭔가 비슷한데 ?? 생각이 들 수 있음.

"한 장의 array 에다 여러 사이즈의 커널로 슬라이딩" 이라는 관점에서 보면 비슷해 보임.

그러나 둘은 전혀 다른 개념

	Inception module	SPP layer
입	feature map	feature map
출	feature map	vector
슬라이딩 목적	feature extraction	Pooling (max)

● SPP layer 정리를 찾아보면 전혀 다른 방식으로

설명을 해놓은 글을 쉽게 찾을 수 있음.

 와 같은 level 2 계산에 대해서

1. 누르는 위 설명처럼 일단 4분할 하고 각 영역에서
max pooling 하는 거라 적어놓음.

2. 누르는 pooling 결과가 2×2 가 되도록 window, stride 를 지정해서
슬라이딩 하면서 pooling 하는 거라 적어놓음



??????

R-CNN 도 그랬지만 개떡같은 정리글 너무 많음 ...

원 논문을 결국 봤더니

결론은 둘다 맞다 ~ 하나는 train, 하나는 test 과정 말하는 거였다 ~

반쪽만 적은 정리글들 길기다 새똥 맞아라 ~

이론적인 아이디어는 이미지 입력 크기에서 상관 없이

3×3 분할, 2×2 분할 등 겁치지 않게 영역을 극분하여

max pooling 을 하는 거였지만 (실제 test 에서는 이렇게 함)

학습 과정에서는 GPU 를 통한 모델링이 고정된 입력으로 학습해야

의미가 있어서 아이디어에서 조금 변형을 하여서 실무에 적용 시킨 것

어떤 대체 방안이었는지 알아보기 전에

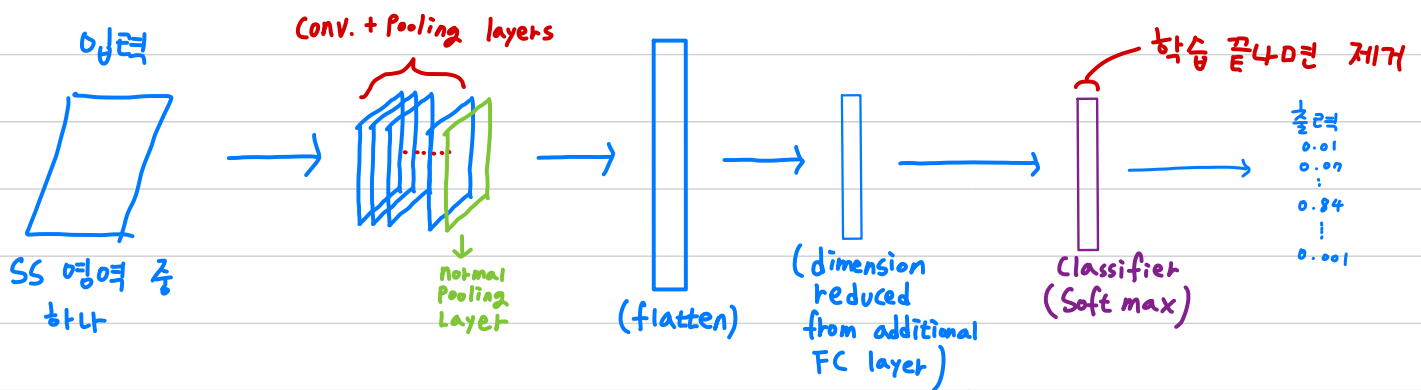
무엇을? 학습한다는 건지 먼저 보자면

R-CNN, SPPNet 모두 3-module 로 이루어진 구조이며 학습또한 각 모듈 별로 진행.

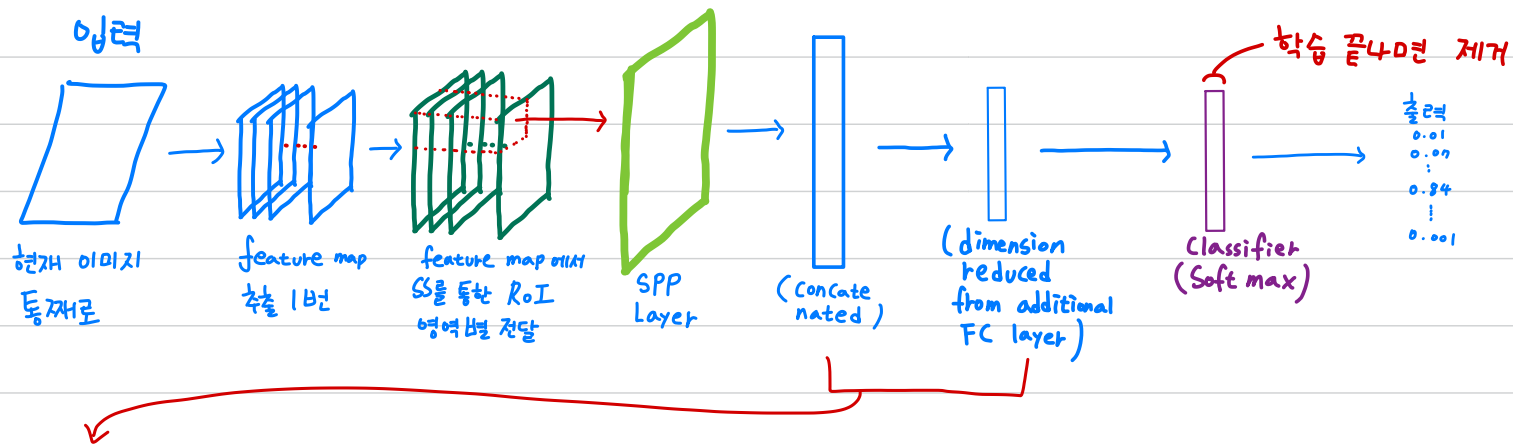
그중 나머지 둘 (SVM, Bbox regressor) 의 학습 방법은 거의 차이 없음.

차이는 CNN 모듈에 존재하는데

• R-CNN 의 CNN 모듈 학습



• SPPNet 의 CNN 모듈 학습

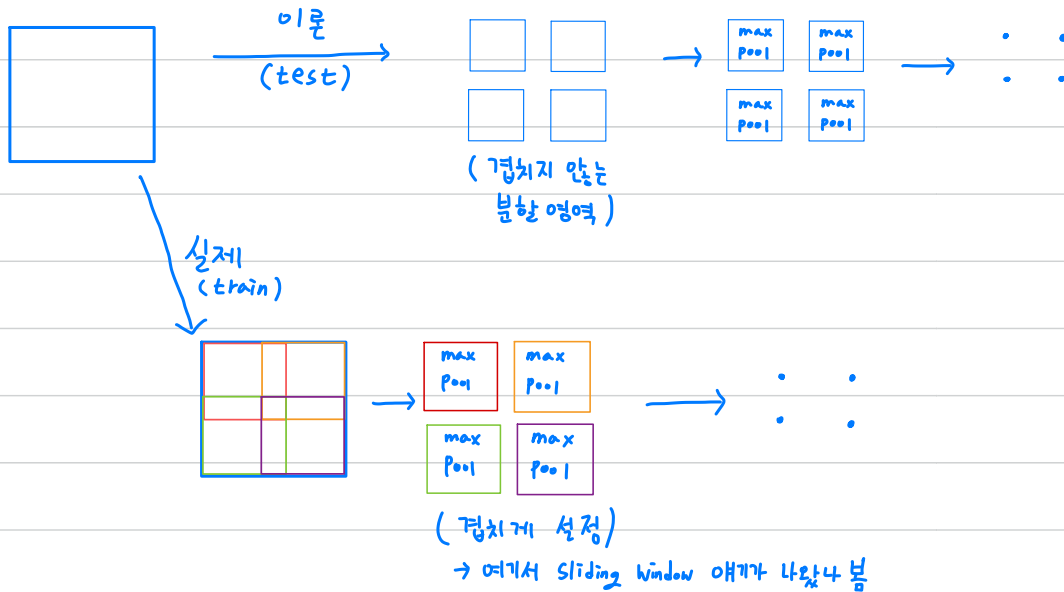


이 부분이 바로 SPPNet CNN 모듈에서 학습이 가능한 곳

* 사실상 conv. layer 들은 학습 불가 (한계점에서 다른 내용)

4 위의 저 부분을 학습하기 위해 Single-size, Multi-size 학습 방식을 고안하였다고 함.

입력적으로 들어가는 고정된 사이즈 종류만 1개 VS 2+개 차이
SPP layer 계산이 이론과 다르다는게 핵심.



Single-size

↳ 224x224 를 원본 이미지에서 Crop 해서 증식 데이터 만들어서 학습

* 후보영역을 224x224로 warping 하는거 X, 이미지동재 들어가는거 주의

Multi-size

↳ Single-size 처럼 우선 224x224 Crop 으로 증식데이터 생성

↳ 각 증식데이터를 180x180 로 resizing (← 난 이것도 정보손실 아닌가 했는데 crop, warp와는 다르게 이전 이미지와 ratio는 같고 scale만 달라서 해상도 차이만 있다고 함.)

↳ 1 epoch 씩 번갈아가며 각 사이즈 이미지로부터 FC 가 학습할 수 있게 함.

(연구과정에서는 중간중간 180 ~ 224 사이 랜덤 SxS 사이즈 에폭도 추가해서 학습하는 jittering 기법도 시도했다고 함. 성능은 잘 모르겠음)