SPPNet

■ 가장 큰 장점

1. R-CNN은 이미지 한 장당 수천 번의 CNN을 수행함(Time-Consuming).

→ 이미지 안에서 bounding box → Selective Search 방법을 사용.

→ 반면에 SPPNet은 딱 한 번 CNN을 수행함(one-pass).

→ Image의 Feature Map에서 어떤 위치에 해당하는 정보를 뽑아냄.

2. SPPNet은 이미지의 Scale(확대/축소), Size(크기), Aspect Ratio(종횡비)에 영향을 받지 않음.

■ 기존의 CNN의 입력

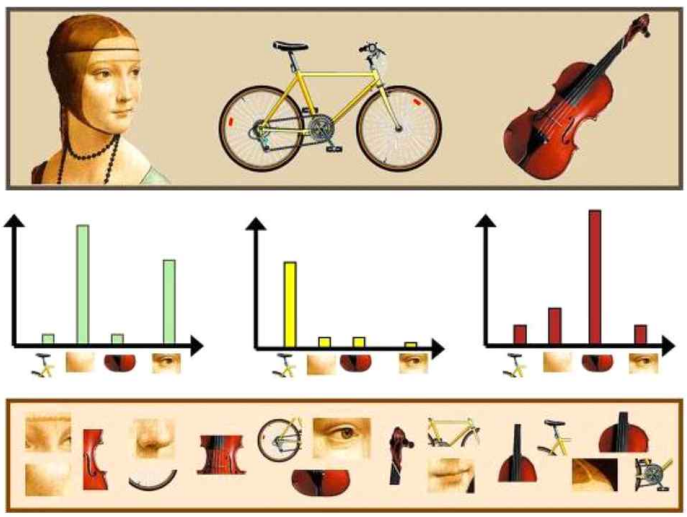
무조건 고정 크기가 입력됨.

: **Fully Connected Layer** 때문이다(fixed-length Vector가 필요함).

1. **Crop**(자르기) : 전체 객체가 들어가지 않는 경우가 발생.
2. **Warp**(줄이거나 늘리기) : 원하지 않는 왜곡이 발생.

사실, Convolution Layer에는 고정 크기의 입력이 필요하지 않음.

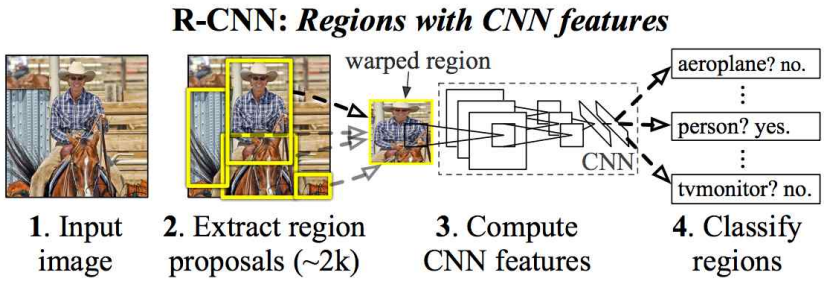
■ BoW(Bag of Words) 기법의 확장



[Box of Words]

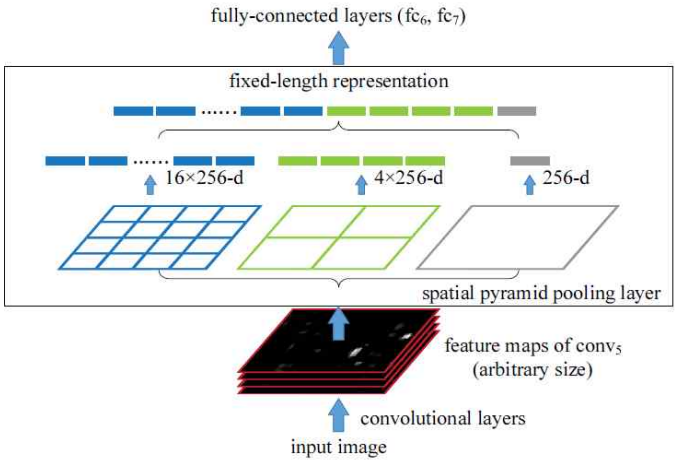
1. Feature Extracting
2. Clustering : K-Means 알고리즘 사용, Codeword 찾아냄
3. Codebook Generation
4. Image Representation : Histogram 시킴(bin 개수 = Codeword 개수)
5. Learning and Recognition : SVM 등의 방법으로 분류

■ **R-CNN 방법과 SPPNet 방법**



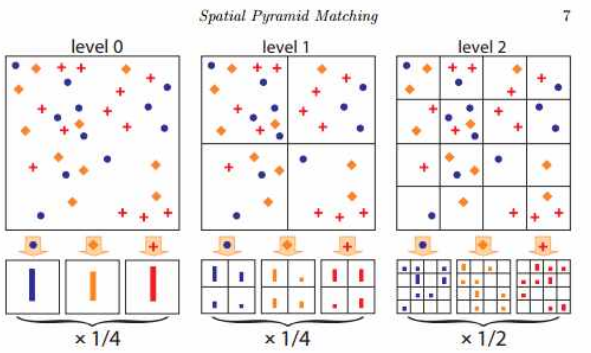
[R-CNN]

R-CNN은 Image 영역(Selective Search 방법으로 수천 개를 만듦.)을 추출하고 **Warp** 시킴.
→ 원하지 않는 왜곡(Unwanted geometric distortion)이 발생함.
→ 이후 추출한 영역을 CNN으로 학습시킴.
→ **매우 느리다(time-consuming).**



[Spatial Pyramid Pooling]

SPPNet 은 Convolution 마지막 층에서 나온 Feature Map을 분할하여 고정 크기로 만들어버림.
→ '06년 발표된 **Spatial Pyramid Matching** 방법에서 나옴.



[Spatial Pyramid Matching]

→ 분할 방법은 “fast” mode of Selective Search

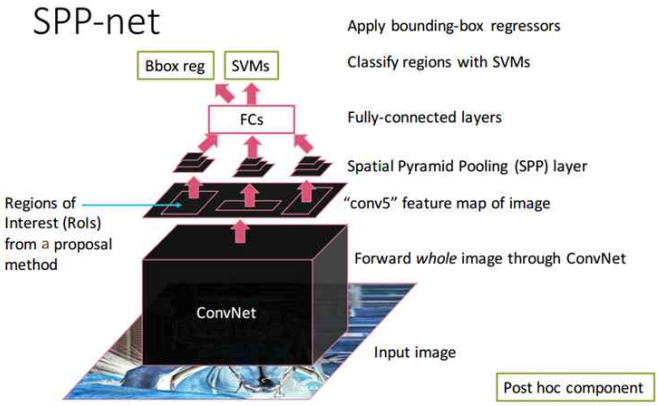
[논문 제목 : **“Segmentation as selective search for object recognition,” in ICCV, 2011.**]

→ 마지막 Pooling Layer를 SPP(Spatial Pyramid Pooling)로 대체함.

+ 내부적으로 Global Max Pooling 사용.

→ 분할하는 크기만 동일하면 어떤 Size, Scale, Aspect Ratio를 가진 이미지가 와도 똑같은 크기의 Vector가 출력됨(이후 FC Layer에 입력으로 들어감).

→ 분류기 : 이진 선형 SVM



[Spatial Pyramid Pooling Network Architecture]

분할 크기(4-level Spatial Pyramid)

1. 50 bins : {6x6, 3x3, 2x2, 1x1} → 12,800-dimension(256x50) representations
2. 30 bins : {4x4, 3x3, 2x2, 1x1}

→ stride를 각각의 크기 별로 따로 적용

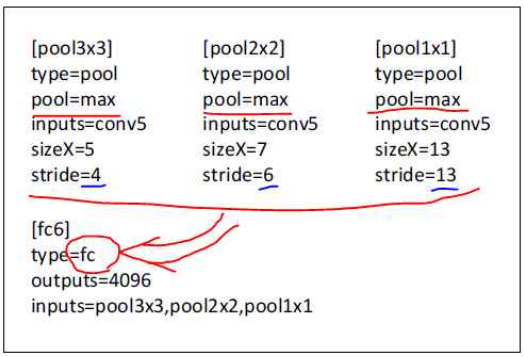


Figure 4: An example 3-level pyramid pooling in the cuda-convnet style [3]. Here sizeX is the size of the pooling window. This configuration is for a network whose feature map size of $conv_5$ is 13×13, so the $pool_{3\times3}$, $pool_{2\times2}$, and $pool_{1\times1}$ layers will have 3×3, 2×2, and 1×1 bins respectively.

SPP Layer의 출력

→ (K * M)-dimensional Vector (K는 마지막 Feature Map의 dimension, M은 bin의 크기)

■ Training

1. Single-size training : 224x224 size로만 학습, {3x3, 2x2, 1x1} SPP 사용.
2. Multi-size training : 180x180 and 224x224 두 종류를 학습.

180-Network와 224-Network의 출력은 동일하다.

→ 왜? 나누고, 평균을 구해서 뽑기 때문.

→ Network Switching 방법? Epoch 1은 180-Network, Epoch 2는 224-Network... 반복!

■ 실험 네트워크

E. 70

E. 90

model	$conv_1$	$conv_2$	$conv_3$	$conv_4$	$conv_5$	$conv_6$	$conv_7$
ZF-5	96×7^2, str 2 LRN, pool 3^2, str 2 map size 55×55	256×5^2, str 2 LRN, pool 3^2, str 2 27×27	384×3^2 13×13	384×3^2 13×13	256×3^2 13×13	-	-
Convnet*-5	96×11^2, str 4 LRN, map size 55×55	256×5^2 LRN, pool 3^2, str 2 27×27	384×3^2 pool 3^2, 2 13×13	384×3^2 13×13	256×3^2 13×13	-	-
Overfeat-5/7	96×7^2, str 2 pool 3^2, str 3, LRN map size 36×36	256×5^2 pool 2^2, str 2 18×18	512×3^2 18×18	512×3^2 18×18	512×3^2 18×18	512×3^2 18×18	512×3^2 18×18

- ZFNet과 AlexNet은 Convolution Layer가 5개
- Overfeat 구조는 FC Layer를 1x1 Convolution으로 이해한 구조.

→ Sliding Window 개념 적용 가능, 연산량 감소 but 정확도는 높지 않음.

■ Improves Accuracy

		top-1 error (%)			
		ZF-5	Convnet*-5	Overfeat-5	Overfeat-7
(a)	no SPP	35.99	34.93	34.13	32.01
(b)	SPP single-size trained	34.98 (1.01)	34.38 (0.55)	32.87 (1.26)	30.36 (1.65)
(c)	SPP multi-size trained	34.60 (1.39)	33.94 (0.99)	32.26 (1.87)	29.68 (2.33)

		top-5 error (%)			
		ZF-5	Convnet*-5	Overfeat-5	Overfeat-7
(a)	no SPP	14.76	13.92	13.52	11.97
(b)	SPP single-size trained	14.14 (0.62)	13.54 (0.38)	12.80 (0.72)	11.12 (0.85)
(c)	SPP multi-size trained	13.64 (1.12)	13.33 (0.59)	12.33 (1.19)	10.95 (1.02)

1. Multi-level Pooling

→ {6x6, 3x3, 2x2, 1x1}과 {4x4, 3x3, 2x2, 1x1}의 차이는 거의 없다.

→ 하지만, SPP를 사용한 것과 사용하지 않은 것은 확실한 차이가 있다.

(50 bin pyramid : 34.98/14.14, no-SPP : 35.99/14.76)

2. Multi-size Training(그림 참조)

Multi-size SPPNet(Overfeat-7) : 29.68%

Single-size SPPNet : 30.36% / no-SPP : 32.01%

3. Full-image Representations

SPP on	test view	top-1 val
ZF-5, single-size trained	1 crop	38.01
ZF-5, single-size trained	1 full	**37.55**
ZF-5, multi-size trained	1 crop	37.57
ZF-5, multi-size trained	1 full	**37.07**
Overfeat-7, single-size trained	1 crop	33.18
Overfeat-7, single-size trained	1 full	**32.72**
Overfeat-7, multi-size trained	1 crop	32.57
Overfeat-7, multi-size trained	1 full	**31.25**

Table 3: Error rates in the validation set of ImageNet 2012 using a single view. The images are resized so $\min(w, h) = 256$. The crop view is the central 224×224 of the image.

→ 다중 조합 뷰가 싱글 Full-Image 뷰보다 더 좋은 경우도 있음.

(We find that the combination of multiple views is substantially better than the single full-image view.)

하지만 여전히 Full-Image View는 장점이 있는데

1. 수십 개의 뷰를 조합한 경우에도 추가적인 두 개의 전체 이미지 뷰에서 약 0.2%의 성능 향상이 있음.
2. 전체 이미지 뷰는 원래의 방식과 동일하다.
3. 응용 프로그램에서 이미지 자체를 표현하는 것을 요구할 수 있음.

■ Detection Result

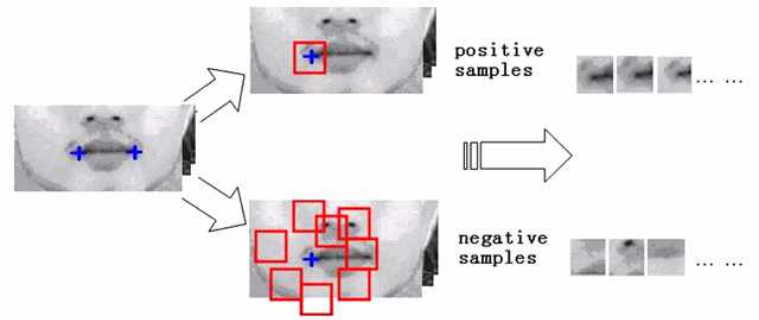
	SPP (1-sc) (ZF-5)	SPP (5-sc) (ZF-5)	R-CNN (Alex-5)
$pool_5$	43.0	44.9	44.2
fc_6	42.5	44.8	46.2
$ftfc_6$	52.3	53.7	53.1
$ftfc_7$	54.5	55.2	54.2
$ftfc_7$ bb	58.0	**59.2**	58.5
conv time (GPU)	0.053s	0.293s	8.96s
fc time (GPU)	0.089s	0.089s	0.07s
total time (GPU)	0.142s	0.382s	9.03s
speedup (*vs.* RCNN)	**64**×	**24**×	-

Table 9: Detection results (mAP) on Pascal VOC 2007. "ft" and "bb" denote fine-tuning and bounding box regression.

	SPP (1-sc) (ZF-5)	SPP (5-sc) (ZF-5)	R-CNN (ZF-5)
$ftfc_7$	54.5	55.2	55.1
$ftfc_7$ bb	58.0	**59.2**	**59.2**
conv time (GPU)	0.053s	0.293s	14.37s
fc time (GPU)	0.089s	0.089s	0.089s
total time (GPU)	0.142s	0.382s	14.46s
speedup (*vs.* RCNN)	**102**×	**38**×	-

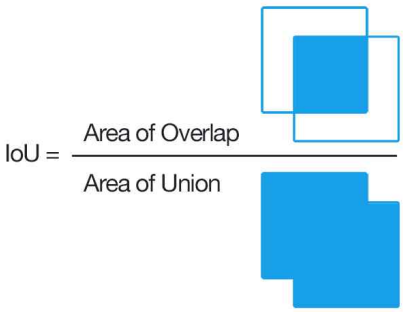
Table 10: Detection results (mAP) on Pascal VOC 2007, **using the same pre-trained model** of SPP (ZF-5).

■ **Positive Sample vs Negative Sample(Data Labeling)**



[Positive Sample and Negative Sample]

positive sample : ground-truth(=GT) Sample을 사용.
negative sample : positive sample과 최대 30% 겹치는 Sample을 사용(IOU 측정법).



[Intersection over Union]

■ **Result**

■ **Conclusion**

- SPPNet은 다른 크기, 다른 사이즈, 다른 종횡비를 다루기 위한 유연한 방법이다.
- Classification과 Detection에서 높은 정확성, 특히 DNN 기반의 Detection을 크게 빠르게 해 줌.