SPPNet: Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun

SPPNet: Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition
Abstract
1. Introduction
2. Deep Networks with Spatial Pyramid Pooling
    2.1. Convolutional Layers and Feature Maps
    2.2. The Spatial Pyramid Pooling Layer
    2.3. Training the Network
3. SPP-Net for Image Classification
    3.1. Experiments on ImageNet 2012 Classification
    3.1.1. Baseline Network Architectures
    3.1.2. Multi-level Pooling Improves Accuracy
    3.1.3. Multi-size Training Improves Accuracy
    3.1.4. Full-image Representations Improve Accuracy
    3.1.5. Multi-view Testing on Feature Maps
    3.1.6. Summary and Results for ILSVRC 2014
    3.2. Experiments on VOC 2007 Classification
    3.3. Experiments on Caltech 101
4. SPP-Net for Object Detection
    4.1. Detection Algorithm
    4.2. Detection Results
    4.3. Complexity and Running Time
    4.4. Model Combination for Detection
    4.5. ILSVRC 2014 Detection
5. Conclusion
Appendix A
Reference

Abstract

현존하는 CNN은 **고정 크기의 입력 이미지를 요구**

이는 '인위적'이며 그 이미지나 임의의 크기로 변환된 부분 이미지에 대한 인식 정확도를 해침

→ 'spatial pyramid pooling'이라는 다른 방법을 제시

이미지의 크기에 상관없이 고정 길이의 대표값(representation)을 생성

pyramid pooling은 **object 변형에도 강건**함

Contribution

- CNN 기반의 이미지 classification을 개선하여 **ImageNet 2012**에 결과를 보여줌
- PASCAL VOC 2007 및 Caltech101 dataset에 대해서도 하나의 이미지 대표값으로 fine-tuning 없이도 최고의 결과를 보여줌
- object detection에서도 강력한데, 전체 이미지에서 feature map을 한번만 계산하고 detector를 학습하기 위해 임의의 region에 대해 고정 크기의 대표값을 생성함
- 테스트 이미지 처리시에 이 방법은 **R-CNN보다 20~102배 빠르고** 정확도는 비슷함
- ILSVRC 2014에서 **detection 부문 2위**, **classification 부문 3위**

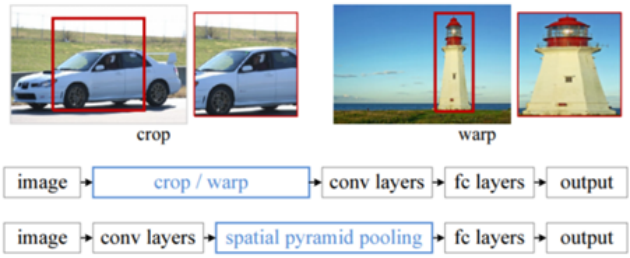
1. Introduction

기존의 CNN 아키텍쳐들은 모두 **입력 이미지가 고정**되어야 했음 (ex. 224 x 224)

→ 신경망을 통과시키기 위해서는 이미지를 **고정된 크기로 크롭**하거나 **비율을 조정**(warp)해야 함

하지만 이렇게 되면 물체의 일부분이 잘리거나, 본래의 생김새와 달라지는 문제점

"입력 이미지의 크기나 비율에 관계 없이 CNN을 학습 시킬 수는 없을까?"



Convolution 필터들은 사실 입력 이미지가 고정될 필요가 없음

sliding window 방식으로 작동하기 때문에, 입력 이미지의 크기나 비율에 관계 없이 작동함

입력 이미지 크기의 고정이 필요한 이유는 바로 컨볼루션 레이어들 다음에 이어지는 **fully connected layer**가 고정된 크기의 입력을 받기 때문

여기서 **Spatial Pyramid Pooling(SPP)**이 제안됨

"입력 이미지의 크기에 관계 없이 Conv layer들을 통과시키고, FC layer 통과전에 피쳐 맵들을 동일한 크기로 조절해주는 pooling을 적용하자!"

입력 이미지의 크기를 조절하지 않은 채로 컨볼루션을 진행하면

1. **원본 이미지의 특징**을 고스란히 간직한 피쳐 맵을 얻을 수 있음
2. 사물의 **크기 변화에 더 견고**한 모델을 얻을 수 있음
3. Image Classification이나 Object Detection과 같은 여러 테스크들에 **일반적으로 적용**할 수 있음

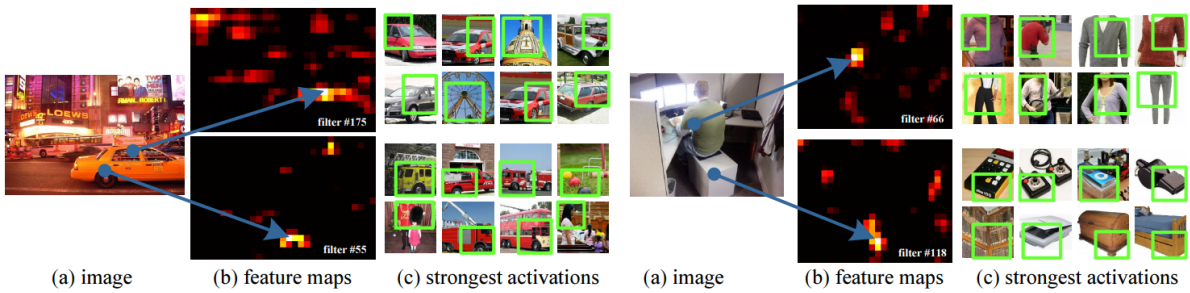
💡 **전체 알고리즘**

1. 먼저 전체 이미지를 미리 학습된 **CNN을 통과시켜 피쳐맵**을 추출함
2. Selective Search를 통해서 찾은 각각의 RoI들은 제 각기 크기와 비율이 다름
   이에 SPP를 적용하여 **고정된 크기의 feature vector**를 추출함
3. 그 다음 **fully connected layer**들을 통과 시킴
4. 앞서 추출한 벡터로 각 이미지 클래스 별로 **binary SVM Classifier**를 학습시킴
5. 마찬가지로 앞서 추출한 벡터로 **bounding box regressor**를 학습시킴

본 논문의 가장 핵심은 Spatial Pyramid Pooling을 통해서 **각기 크기가 다른 CNN 피쳐맵 인풋으로부터 고정된 크기의 feature vector를 뽑아내는 것**에 있음

2. Deep Networks with Spatial Pyramid Pooling

2.1. Convolutional Layers and Feature Maps



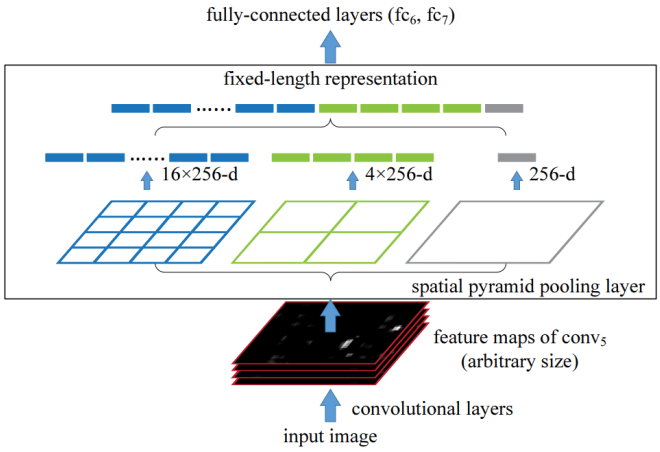
(a) Pascal VOC 2007의 2개의 이미지
(b) conv5 의 특정 피쳐 맵
(c) 대응하는 필터의 응답이 가장 강한 receptive field

convolution 층은 sliding filters를 사용하며, 그 출력은 입력과 동일한 aspect ratio를 가짐

필터는 의미 있는 content 에 의해서 active 하게 됨

ex. 55번째 필터(왼쪽 하단)는 원 모양으로, 66번째 필터(오른쪽 상단)는 ∧ 모양, 118번째 필터(오른쪽 하단)는 ∨ 모양으로 가장 활성화 됨

2.2. The Spatial Pyramid Pooling Layer



SPP Layer가 있는 네트워크 구조

1. 먼저, Conv Layer들을 거쳐서 추출된 **feature map을 인풋**으로 받음
2. 그리고 이를 미리 정해져 있는 영역으로 나누어 줌

   (위의 경우, 미리 4x4, 2x2, 1x1 세 가지 영역을 제공하며, 각각을 하나의 **피라미드**라고 부름. 즉, 해당 예시에서는 **3개의 피라미드**를 설정한 것, **피라미드의 한 칸을 bin** 이라고 함)

   ex. 64 x 64 x 256 크기의 피쳐 맵이 들어온다고 했을 때, 4x4의 피라미드의 bin의 크기는 16x16

3. 이제 각 bin에서 가장 큰 값만 추출하는 **max pooling**을 수행하고, 그 결과를 쭉 이어붙여 줌
4. 입력 피쳐맵 **채널 크기를 k**, **bin의 개수를 M**이라고 했을 때 SPP의 최종 아웃풋은 k*M 차원 벡터
   (위의 예시에서 k = 256, M = (16 + 4 + 1) = 21)

→ 입력 이미지의 크기와 상관없이 **미리 설정한 bin 개수와 CNN 채널 값으로 SPP의 출력이 결정됨**
→ 항상 **동일한 크기**의 결과를 리턴한다고 볼 수 있음
실제 실험에서 저자들은 1x1, 2x2, 3x3, 6x6 총 4개의 피라미드, 50개의 bin으로 SPP를 적용

2.3. Training the Network

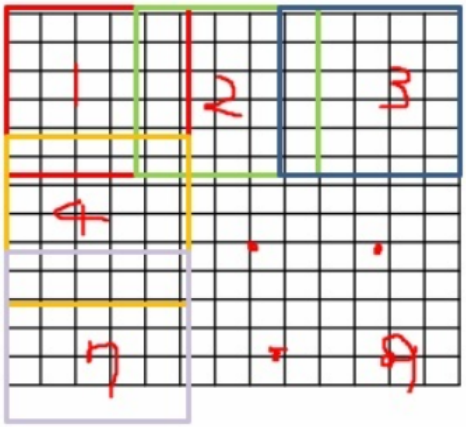
이론적으로 위의 네트워크는 **입력 이미지 크기에 관계없이** 표준 back-propagation을 사용해 훈련
but, 실제로는 cuda-convnet, Caffe와 같은 GPU 구현 시 **고정 이미지 크기를 선호**함
다음은 SPP 동작을 유지하면서 이러한 GPU 구현의 이점을 살리는 학습 방법을 기술함

Single-size training

이전 작업처럼 224x224의 고정 크기로 crop된 이미지를 먼저 고려하면 crop은 data 증강을 위한 것
주어진 이미지 크기에 대해 먼저 SPP에 필요한 bin 사이즈를 미리 계산함

SPP 예시

- ROI feature - 13x13
- Spatial bin - 3x3 (pooling 연산을 통해 3x3 feature map을 얻겠다는 의미)
- 아래의 ROI feature에서 3x3 feature map을 얻기 위해서는 window size =5, stride = 4 로 설정
- 아래와 같이 설정된 window (작은 네모 박스)는 이동하면서 max pooling 연산을 적용



SPP 적용 예시

```
[pool3x3]          [pool2x2]          [pool1x1]
type=pool          type=pool          type=pool
pool=max           pool=max           pool=max
inputs=conv5       inputs=conv5       inputs=conv5
sizeX=5            sizeX=7            sizeX=13
stride=4           stride=6           stride=13

[fc6]
type=fc
outputs=4096
inputs=pool3x3,pool2x2,pool1x1
```

cuda-convnet 환경에서 구현된 3-level pyramid (3x3, 2x2, 1x1)

but, 보통은 1x1, 2x2, 4x4 spatial bin 사용

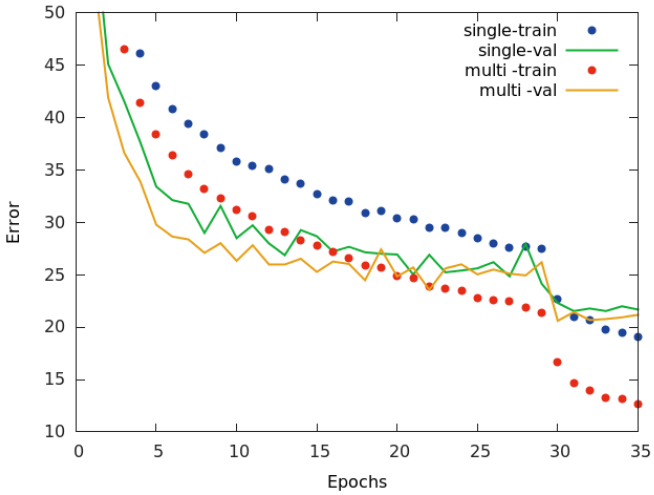
다양한 spatial bin을 가지고 있다는 의미에서 Spatial Pyramid 라고 함

위의 SPP를 통해 1x1, 2x2, 4x4 spatial bin 을 얻었다면 spatial bin을 모두 flatten하게 됨

그럼 총 16+4+1=21개의 feature가 만들어 지는 것!

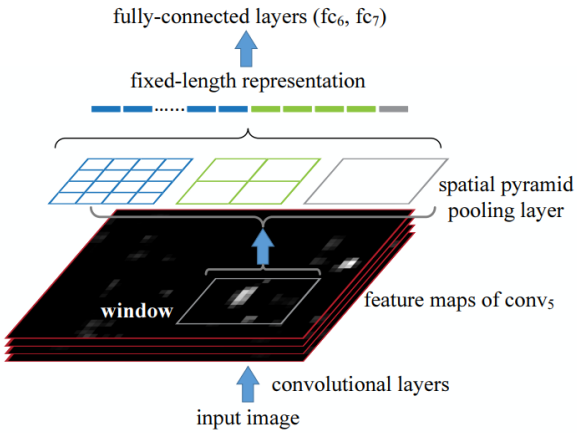
21개의 고정된 feature들은 fc layer로 넘어가게 됨

Multi-size training



single-size training vs. multi-size training

다양한 크기의 서로 다른 input image size를 넣어 네트워크의 **수렴을 빠르게 하고 성능을 개선**함



이미지를 6개의 스케일 s=(224, 256, 300, 360, 448, 560) 로 조절

각 스케일에 대해 전체 이미지에 대한 feature map을 계산

어떤 스케일에서든 224x224를 뷰 크기로 사용하여 이 뷰는 원래 다른 스케일의 이미지에 대해 상대적으로 다른 크기를 가지게 됨

각 스케일에 18개의 뷰를 사용 (4 코너와 중앙, 각 변 중심에서 4개, 각각을 플립)

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

4가지 네트워크 구조를 활용하여 SPP-Net을 적용한 결과 개선되는 점을 실험

3. SPP-Net for Image Classification

3.1. Experiments on ImageNet 2012 Classification

3.1.1. Baseline Network Architectures

3.1.2. Multi-level Pooling 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)

성능 : multi-size trained > single-size trained > no SPP

4개의 level pyramid, 50개의 bin

multi-size trained를 사용하면 parameter를 더 사용하면서 object의 변형에 robust 에러율 향상

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**

성능 : 전체 이미지 > 잘린 이미지

3.1.3. Multi-size Training Improves Accuracy

3.1.4. Full-image Representations Improve Accuracy

3.1.5. Multi-view Testing on Feature Maps

method	test scales	test views	top-1 val	top-5 val	**top-5 test**
Krizhevsky *et al.* [3]	1	10	40.7	18.2	
Overfeat (fast) [5]	1	-	39.01	16.97	
Overfeat (fast) [5]	6	-	38.12	16.27	
Overfeat (big) [5]	4	-	35.74	14.18	
Howard (base) [36]	3	162	37.0	15.8	
Howard (high-res) [36]	3	162	36.8	16.2	
Zeiler & Fergus (ZF) (fast) [4]	1	10	38.4	16.5	
Zeiler & Fergus (ZF) (big) [4]	1	10	37.5	16.0	
Chatfield *et al.* [6]	1	10	-	13.1	
ours (SPP O-7)	1	10	29.68	10.95	
ours (SPP O-7)	6	96+2full	**27.86**	**9.14**	**9.08**

ImageNet 2012 에서 최고 수준이었던 모델보다 에러율 향상

3.1.6. Summary and Results for ILSVRC 2014

rank	team	top-5 test
1	GoogLeNet [32]	**6.66**
2	VGG [33]	7.32
3	ours	8.06
4	Howard	8.11
5	DeeperVision	9.50
6	NUS-BST	9.79
7	TTIC_ECP	10.22

ILSVRC 2014에서 3위에 해당하는 좋은 결과를 보여줌

3.2. Experiments on VOC 2007 Classification

model	(a) no SPP (ZF-5)	(b) SPP (ZF-5)	(c) SPP (ZF-5)	(d) SPP (ZF-5)	(e) SPP (Overfeat-7)
size	crop 224×224	crop 224×224	full 224×-	full 392×-	full 364×-
$conv_4$	59.96	57.28	-	-	-
$conv_5$	66.34	65.43	-	-	-
$pool_{5/7}$ (6×6)	69.14	68.76	70.82	71.67	76.09
$fc_{6/8}$	74.86	75.55	77.32	78.78	81.58
$fc_{7/9}$	75.90	76.45	78.39	80.10	**82.44**

model	(a) no SPP (ZF-5)	(b) SPP (ZF-5)	(c) SPP (ZF-5)	(d) SPP (Overfeat-7)
size	crop 224×224	crop 224×224	full 224×-	full 224×-
$conv_4$	80.12	81.03	-	-
$conv_5$	84.40	83.76	-	-
$pool_{5/7}$ (6×6)	87.98	87.60	89.46	91.46
SPP $pool_{5/7}$	-	89.47	91.44	**93.42**
$fc_{6/8}$	87.86	88.54	89.50	91.83
$fc_{7/9}$	85.30	86.10	87.08	90.00

3.3. Experiments on Caltech 101

method	VOC 2007	Caltech101
VQ [15]†	56.07	74.41±1.0
LLC [18]†	57.66	76.95±0.4
FK [19]†	61.69	77.78±0.6
DeCAF [13]	-	86.91±0.7
Zeiler & Fergus [4]	75.90‡	86.5±0.5
Oquab *et al.* [34]	77.7	-
Chatfield *et al.* [6]	**82.42**	88.54±0.3
ours	**82.44**	**93.42±0.5**

4. SPP-Net for Object Detection

4.1. Detection Algorithm

Object Detection에 SPP를 적용할 수 있음

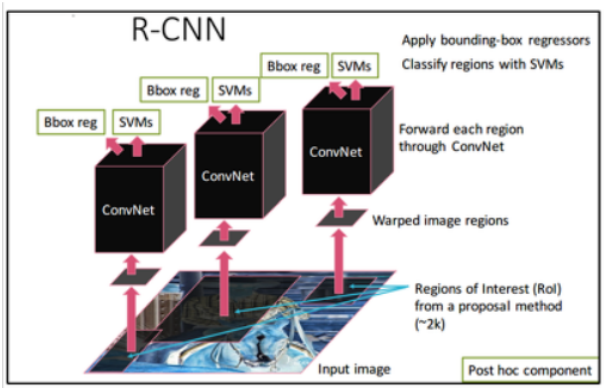
저자들은 R-CNN의 문제점을 지적하며 SPP를 이용한 더 효율적인 object detection을 제안

R-CNN은 Selective Search로 찾은 2천개의 물체 영역을 모두 고정 크기로 조절한 다음, 미리 학습된 CNN 모델을 통과시켜 feature를 추출 → 속도가 엄청 느려짐

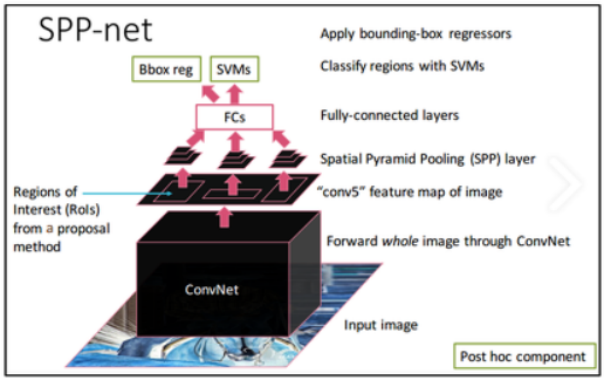
cf) SPPNet은 입력 이미지를 그대로 CNN에 통과시켜 피쳐 맵을 추출한 다음, 그 feature map에서 2천개의 물체 영역을 찾아 SPP를 적용하여 고정된 크기의 feature를 얻어냄

그리고 이를 FC와 SVM Classifier에 통과시킴

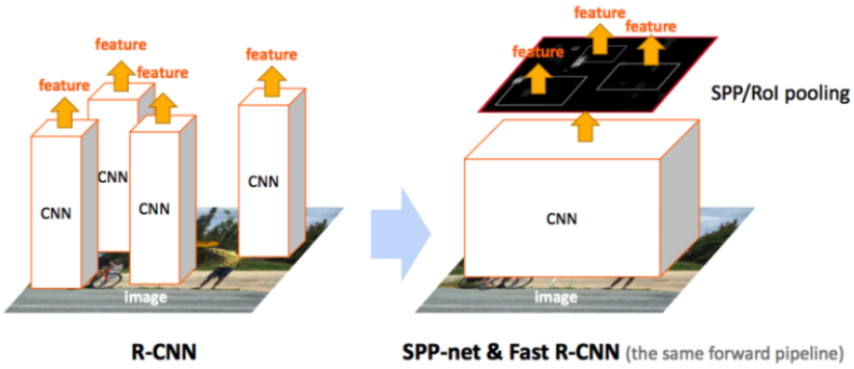
▼ **R-CNN vs. SPP-Net**



R-CNN 네트워크 구조



SPP-Net 네트워크 구조



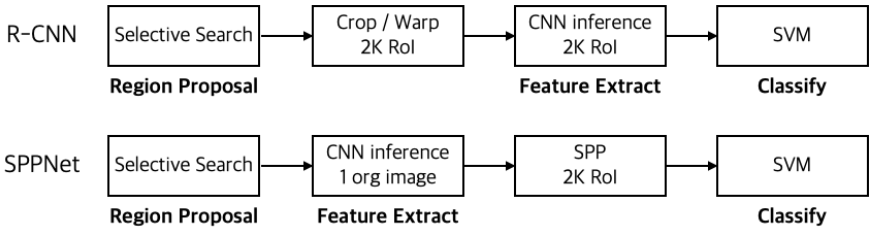
R-CNN vs. SPP-net&Fast R-CNN

R-CNN에서는 입력 이미지에서부터 region proposal 방식을 이용해 candidate bounding box를 선별하고 모든 candidate bounding box에 대해서 CNN 작업을 함

→ 2000개의 candidate bounding box가 나오게 되면 2000번의 CNN 과정을 수행

SPP-Net은 입력 이미지를 먼저 CNN 작업을 진행하고 다섯번째 conv layer에 도달한 feature map을 기반으로 region proposal 방식을 적용해 candidate bounding box를 선별

→ CNN 연산은 1번이 됨



∴ R-CNN 2000번 → SPP-Net 1번의 CNN Operation 절감효과, 시간을 빠르게 단축

Training & Test Time

4.2. Detection Results

	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**×	-

	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**×	-

Pascal VOC 2007 mAP 결과값

scale을 변화시키며 SPP-Net를 적용한 결과

vs. R-CNN 을 이용해 fine-tuning과 bounding box regression을 이용한 결과 더 좋은 성능

4.3. Complexity and Running Time

method	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
DPM [23]	33.7	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5
SS [20]	33.8	43.5	46.5	10.4	12.0	9.3	49.4	53.7	39.4	12.5	36.9	42.2	26.4	47.0	52.4	23.5	12.1	29.9	36.3	42.2	48.8
Regionlet [39]	41.7	54.2	52.0	20.3	24.0	20.1	55.5	68.7	42.6	19.2	44.2	49.1	26.6	57.0	54.5	43.4	16.4	36.6	37.7	59.4	52.3
DetNet [40]	30.5	29.2	35.2	19.4	16.7	3.7	53.2	50.2	27.2	10.2	34.8	30.2	28.2	46.6	41.7	26.2	10.3	32.8	26.8	39.8	47.0
RCNN $ftfc_7$ (A5)	54.2	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7
RCNN $ftfc_7$ (ZF5)	55.1	64.8	68.4	47.0	39.5	30.9	59.8	70.5	65.3	33.5	62.5	50.3	59.5	61.6	67.9	54.1	33.4	57.3	52.9	60.2	62.9
SPP $ftfc_7$ (ZF5)	55.2	65.5	65.9	51.7	38.4	32.7	62.6	68.6	69.7	33.1	66.6	53.1	58.2	63.6	68.8	50.4	27.4	53.7	48.2	61.7	64.7
RCNN bb (A5)	58.5	68.1	72.8	56.8	**43.0**	36.8	**66.3**	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8
RCNN bb (ZF5)	**59.2**	68.4	**74.0**	54.0	40.9	35.2	64.1	**74.4**	69.8	**35.5**	66.9	53.8	**64.2**	69.9	69.6	**58.9**	**36.8**	**63.4**	**56.0**	62.8	64.9
SPP bb (ZF5)	**59.2**	**68.6**	69.7	**57.1**	41.2	**40.5**	**66.3**	71.3	**72.5**	34.4	**67.3**	**61.7**	63.1	**71.0**	**69.8**	57.6	29.7	59.0	50.2	**65.2**	**68.0**

method	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SPP-net (1)	59.2	**68.6**	69.7	57.1	41.2	40.5	66.3	71.3	72.5	34.4	**67.3**	61.7	63.1	71.0	69.8	57.6	29.7	59.0	50.2	65.2	68.0
SPP-net (2)	59.1	65.7	71.4	57.4	**42.4**	39.9	67.0	71.4	70.6	32.4	66.7	61.7	64.8	71.7	70.4	56.5	30.8	59.9	53.2	63.9	64.6
combination	**60.9**	68.5	**71.7**	**58.7**	41.9	**42.5**	**67.7**	**72.1**	**73.8**	**34.7**	67.0	**63.4**	**66.0**	**72.5**	**71.3**	**58.9**	**32.8**	**60.9**	**56.1**	**67.9**	**68.8**

4.4. Model Combination for Detection

4.5. ILSVRC 2014 Detection

rank	team	mAP
1	NUS	**37.21**
2	ours	35.11
3	UvA	32.02
-	(our single-model)	(31.84)
4	Southeast-CASIA	30.47
5	1-HKUST	28.86
6	CASIA_CRIPAC_2	28.61

ILSVRC 2014에서 2위에 해당하는 좋은 결과를 보여줌

5. Conclusion

SPPNet은 기존 R-CNN이 모든 RoI에 대해서 CNN inference를 한다는 문제점을 획기적으로 개선

하지만 여전히 한계점이 있는데,

1. end-to-end 방식이 아니라 **학습에 여러 단계**가 필요함

   (fine-tuning, SVM training, Bounding Box Regression)

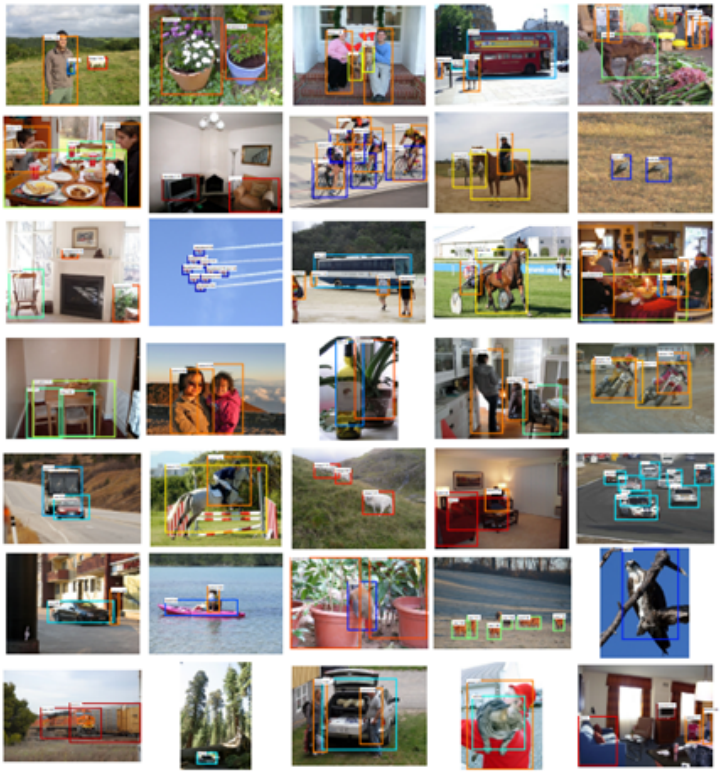
2. 여전히 최종 클래시피케이션은 binary SVM, Region Proposal은 **Selective Search**를 이용

3. **fine tuning 시**에 SPP를 거치기 이전의 Conv 레이어들을 학습 시키지 못함

   단지 그 뒤에 **Fully Connnected Layer만 학습**시킴

   → 저자들은 ***"for simplicity"*** 라고만 설명함

Appendix A



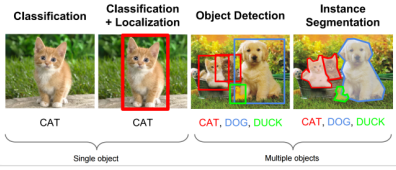
Reference

- **Object Detection 논문 흐름 및 리뷰**

[Object Detection] 1. Object Detection 논문 흐름 및 리뷰

Object Detection 분야에 대해서 살펴보고, 구조가 어떤 방식으로 되어있으며 어떤 방식으로 발전되어 왔는지 살펴보고자 합니다. Deep Learning을 이용하는 Computer Vision Task 중에서 세 번째 그림에 해당이 됩니다.

https://nuggy875.tistory.com/20



- **GitHub 링크** (Papers with Code 기준)

GitHub - yueruchen/sppnet-pytorch: A simple Spatial Pyramid Pooling layer which could be added in CNN

A simple Spatial Pyramid Pooling layer which could be added in CNN - GitHub - yueruchen/sppnet-pytorch: A simple Spatial Pyramid Pooling layer which could be added in CNN

https://github.com/yueruchen/sppnet-pytorch

- **논문 리뷰 - 유튜브**

[Paper Review] Introduction to Object Detection Task : Overfeat, RCNN, SPPNet, FastRCNN

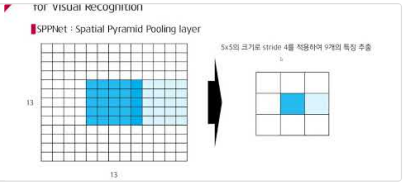
1] 발표자: DSBA 연구실 석사과정 정의석[2] 발표 논문: 본 발표는 Computer Vision Task 중 Object Detection Task의 기반이 되는 논문 4개를 소개합니다.- OverFeat: Integrated Recognition, Localization...

https://www.youtube.com/watch?v=SMEtbrqJ2YI

천우진 - Spatial pyramid pooling in deep convolutional networks for visual recognition

Paper Review : Spatial pyramid pooling in deep convolutional networks for visual recognition

https://www.youtube.com/watch?v=i0lkmULXwe0



- **논문 리뷰 - 블로그**

6. SPP Net

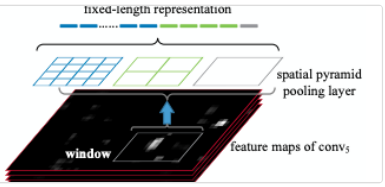
안녕하세요~ 이번글에서는 RCNN의 단점을 극복하고자 나온 SPP-Net object detection 모델에 대해서 알아보도록 할게요~ 1) Too CNN operation RCNN은 selective search를 통해 대략 2000개의 candidate bounding box를 만들어내요. 2000개의 candidate bounding box를 CNN에 입력하게되면 하나의 이미지에 대해서 학습

https://89douner.tistory.com/89

갈아먹는 Object Detection [2] Spatial Pyramid Pooling Network

갈아먹는 Object Detection [1] R-CNN 지난 시간 R-CNN에 이어서 오늘은 SPP-Net[1]을 리뷰해보도록 하겠습니다. 저 역시 그랬고, 많은 분들이 R-CNN 다음으로 Fast R-CNN 논문을 보시는데요, 해당 논문을 보다 보면 SPPNet에서 많은 부분들을 참고한 것을 확인할 수 있습니다. 특히나 핵심인 Spatial Pyramid Pooling은 중요

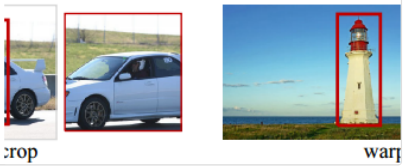
https://yeomko.tistory.com/14



[논문 리뷰] SPPnet (2014) 리뷰, Spatial Pyramid Pooling Network

이번에 리뷰할 논문은 SPPnet 'Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition' 입니다. SPPnet은 CNN 구조가 고정된 입력 이미지 크기를 입력으로 취하는 데에서 발생한 문제점을 개선하기 위해 고안되었습니다. 기존 CNN은 고정된 입력 크기를 맞춰주기 위해서 crop, wrap을 적용합니

https://deep-learning-study.tistory.com/445



SPPnet

SPPNet (https://arxiv.org/pdf/1406.4729.pdf) Abstract 현존하는 CNN은 고정 크기의 입력 이미지를 요구하는데 이는 '인위적'이며 그 이미지나 임의의 크기로 변환된 부분 이미지에 대한 인식 정확도를 해침...

https://blog.daum.net/sotongman/7