Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition

Paper

> Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition
> Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

Abstract

기존의 CNN 아키텍쳐들은 모두 입력 이미지가 고정되어야 했습니다. (ex. 224 x 224) 그렇기 때문에 신경망을 통과시키기 위해서는 이미지를 고정된 크기로 크롭하거나 비율을 조정(warp)해야 했습니다.

하지만 이렇게 되면 물체의 일부분이 잘리거나, 본래의 생김새와 달라지는 문제점이 있습니다.

1. Introduction

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

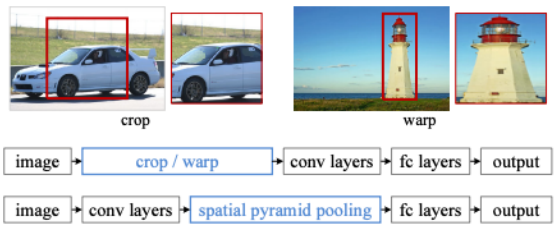


그림1 SPPNet 핵심 아이디어

Convolution 필터들은 사실 입력 이미지가 고정될 필요가 없습니다. sliding window 방식으로 작동하기 때문에, 입력 이미지의 크기나 비율에 관계 없이 작동합니다. 입력 이미지 크기의 고정이 필요한 이유는 바로 컨볼루션 레이어들 다음에 이어지는 fully connected layer가 고정된 크기의 입력을 받기 때문입니다. 여기서 Spatial Pyramid Pooling(이하 SPP)이 제안됩니다.

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

입력 이미지의 크기를 조절하지 않은 채로 컨볼루션을 진행하면 원본 이미지의 특징을 고스란히 간직한 피쳐 맵을 얻을 수 있습니다. 또한 사물의 크기 변화에 더 견고한 모델을 얻을 수 있다는 것이 저자들의 주장입니다. 또한 이는 Image Classification이나 Object Detection과 같은 여러 테스크들에 일반적으로 적용할 수 있다는 장점이 있습니다.

전체 알고리즘은 다음과 같습니다.

1. 먼저 전체 이미지를 미리 학습된 CNN을 통과시켜 피쳐맵을 추출합니다.

2. Selective Search를 통해서 찾은 각각의 RoI들은 제 각기 크기와 비율이 다릅니다. 이에 SPP를 적용하여 고정된 크기의 feature vector를 추출합니다.

3. 그 다음 fully connected layer들을 통과 시킵니다.

4. 앞서 추출한 벡터로 각 이미지 클래스 별로 binary SVM Classifier를 학습시킵니다.

(SVM : Support Vector Machine)

5. 마찬가지로 앞서 추출한 벡터로 bounding box regressor를 학습시킵니다.

본 논문의 가장 핵심은 Spatial Pyramid Pooling을 통해서 각기 크기가 다른 CNN 피쳐맵 인풋으로부터 고정된 크기의 feature vector를 뽑아내는 것에 있습니다. 그 이후의 접근 방식은 R-CNN과 거의 동일합니다. 그렇다면 SPP에 대해서 좀 더 자세히 알아보고, Object Detection에서는 어떻게 적용되는 지 알아보겠습니다.

2. Deep Networks with Spatial Pyramid Pooling

2.1 Convolutional Layers and Feature Maps

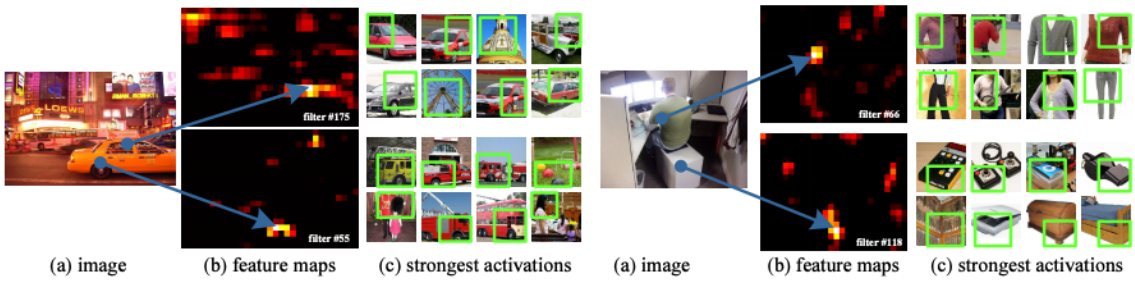


그림2

conv5 층의 어떤 필터들에 의해 생성된 것임.

그림2-(c) 는 ImageNet dataset에서 이런 필터들에 의해 가장 강하게 activate된 이미지를 보여줌

- Crop/warp를 하지 않았으므로 feature map의 크기가 다름
- 화살표는 feature의 강한 응답을 가리킴
- 필터는 의미가 있는 내용들에 대해 activate됨

2.2 The Spatial Pyramid Pooling Layer

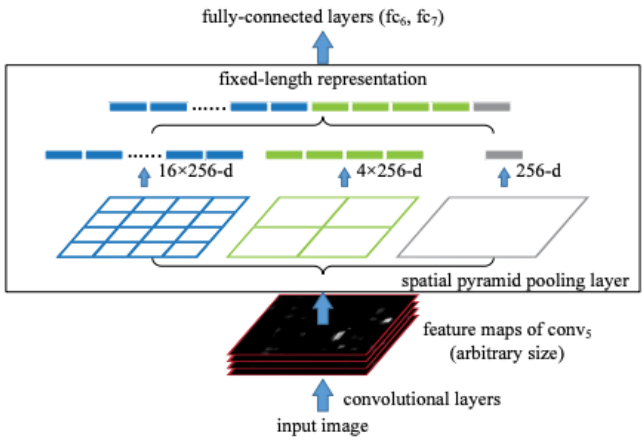


그림3 Spatial Pyramid Pooling Structure

먼저 Conv Layer들을 거쳐거 추출된 피쳐맵을 인풋으로 받습니다.

그리고 이를 미리 정해져 있는 영역으로 나누어 줍니다. 위의 예시에서는 미리 4×4, 2×2, 1×1 세 가지 영역을 제공하며, 각각을 하나의 피라미드라고 부릅니다.

즉, 해당 예시에서는 3개의 피라미드를 설정한 것입니다. 피라미드의 한 칸을 bin 이라고 합니다.

예를 들어 입력이 64 x 64 x 256 크기의 피쳐 맵이 들어온다고 했을 때, 4×4의 피라미드의 bin의 크기는 16×16이 됩니다.

> 분할 크기 (4-level Spatial Pyramid)
> 50 bins : {6*6, 3*3, 2*2, 1*1} → 12,800(256*50)-dimension representations
> 30 bins : {4*4, 3*3, 2*2, 1*1} → 7,680(256*30)-dimension repersentations
> → stride를 각각의 크기 별로 따로 적용하였다.

이제 각 bin에서 가장 큰 값만 추출하는 max pooling을 수행하고, 그 결과를 쭉 이어붙여 줍니다.
입력 피쳐맵의 채널 크기(conv5 layer에서 출력한 feature map의 filter 수)를 k, bin의 개수를 M(사전의 설정한)이라고 했을 때 SPP의 최종 아웃풋은 kM 차원의 벡터입니다.
위의 예시에서 k = 256(채널 크기), M = (16(4*4) + 4(2*2) + 1(1*1)) = 21 이 됩니다.

SPP layer는 256 * 21차원의 벡터를 출력합니다.

정리해보면 입력 이미지의 크기와는 상관없이 미리 설정한 bin의 개수와 CNN 채널 값으로 SPP의 출력이 결정되므로, 항상 동일한 크기의 결과를 리턴한다고 볼 수 있습니다.

이를 통해 다양한 입력 이미지 크기를 입력받아 다양한 feature map size가 생성되고 SPP layer를 거쳐서 고정된 크기의 벡터가 생성됩니다.

실제 실험에서 저자들은 1×1, 2×2, 3×3, 6×6 총 4개의 피라미드로 SPP를 적용합니다.

> 1. Convolution 을 거친 feature map들을 input으로 받음
> 2. 피라미드 라고 부르는 각각의 영역으로 나누어줌 (위 그림에선 4×4, 2×2, 1×1)
> 3. 각각의 영역에서 max pooling을 수행 후 이를 이어붙힌다.
> 4. 최종 output은 kM차원의 벡터 / k = 256, M = (16 + 4 + 1) = 21

입력 사이즈가 다양하므로 conv5에서 출력하는 feature map의 크기도 다양하게 됩니다.

다양한 feature에서 pooling의 window size와 stride 만을 조절하여 출력 크기를 결정합니다.

window size = ceiling(feature map size / pooling size)

stride = floor(feature map size / pooling size) 로 계산하면 어떠한 feature map 크기가 오더라도 고정된 pyramid size를 얻을 수 있습니다.

2.3 Training the Network

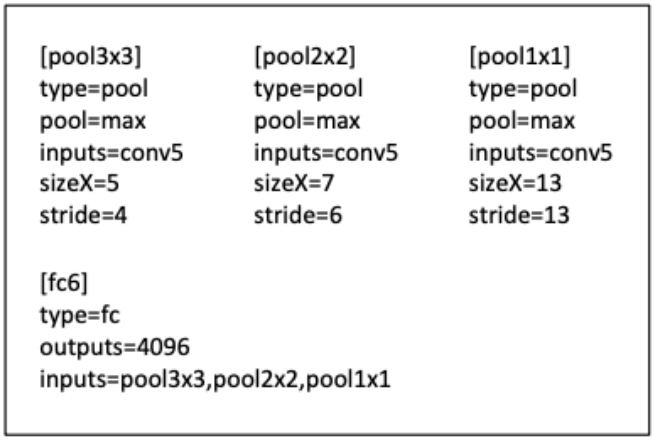


그림4 13×13 feature map에서 각각의 pooling의 window size, stride를 계산한 표

[pool3×3]의 window size = ceiling(13 / 3) = 5, stride = floor(13 / 3) = 4 로 설정되었습니다.

Single-size training = 224×224 size {3×3, 2×2, 1×1} SPP 사용

Multi-size training = 180×180, 224×224 size 학습

180-Network와 224-Network의 출력은 동일하다.
→ 나누고, 평균을 구해서 뽑기 때문이다.
입력 이미지의 크기와는 상관없이 미리 설정한 bin의 개수와 CNN 채널 값으로 SPP의 출력이 결정되므로, 항상 동일한 크기의 결과를 리턴한다고 볼 수 있습니다.

→ Network Switching 방법 : Epoch 1은 180-Network, Epoch 2는 224-Network의 과정을 반복

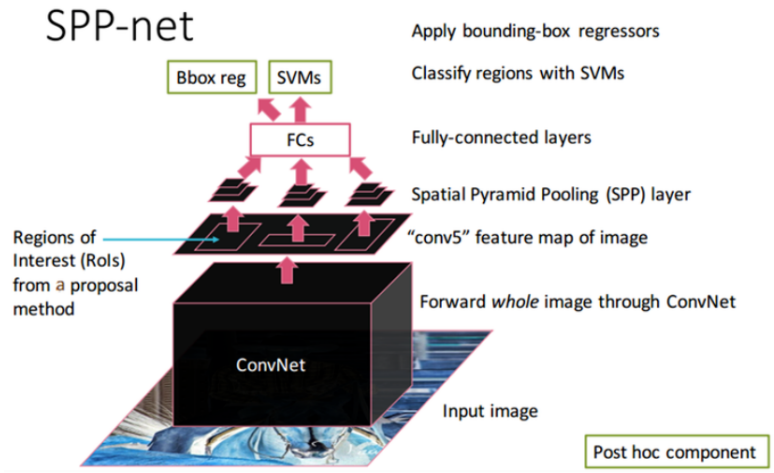
SPPnet을 거친 feature에 softmax를 수행하여 분류에 사용

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

테이블1

ZFNet과 AlexNet은 Convolution Layer가 5개.

추가 자료. SPPNet 작동 방식



1. Selective Search를 사용하여 약 2000개의 region proposals를 생성합니다.
2. 이미지를 CNN에 통과시켜 feature map을 얻습니다.
3. 각 region proposal로 경계가 제한된 feature map을 SPP layer에 전달합니다.
4. SPP layer를 적용하여 얻은 고정된 벡터 크기(representation)를 FC layer에 전달합니다.
5. SVM으로 카테고리를 분류합니다.
6. Bounding box regression으로 bounding box 크기를 조정하고 non-maximum suppression을 사용하여 최종 bounding box를 선별합니다.

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)

테이블2

SPP를 적용한 모델과 그렇지 않은 모델의 비교이다. CNN의 구조와 관계 없이 보장된 성능 향상을 보여준다. Multi-size로 훈련한 모델이 성능이 더 좋다.

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**

테이블3

Input size에 제한이 없기 때문에 cropping이나 warpping 을 거치지 않은 full-image를 넣는 것이 가능하다. 위의 표를 보면 full-image를 넣은 경우가 성능이 더 좋은 것을 확인할 수 있다.

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**

테이블4

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

테이블5

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**

테이블6

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

테이블7

3.3 Experiments on Caltech101

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**

테이블8

4. SPP-NET FOR OBJECT DETECTION

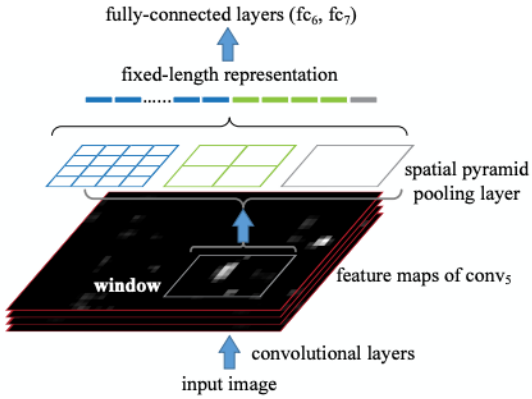
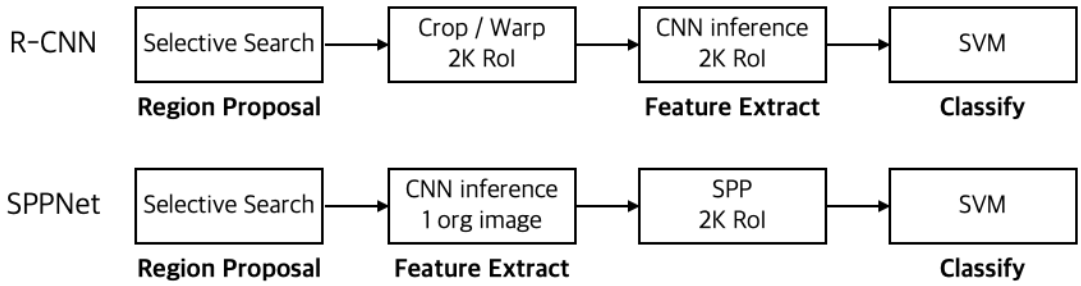
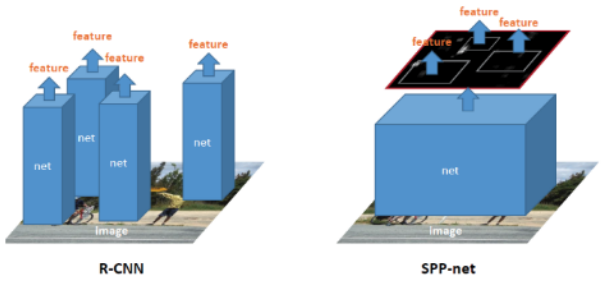


그림5 SPP in Object Detection

Object Detection에 SPP를 적용할 수 있습니다. 먼저 저자들은 R-CNN의 문제점을 지적하며 SPP를 이용한 더 효율적인 object detection을 제안합니다. R-CNN은 Selective Search로 찾은 2천개의 물체 영역을 모두 고정 크기로 조절한 다음, 미리 학습된 CNN 모델을 통과시켜 feature를 추출합니다. 때문에 속도가 느릴 수 밖에 없습니다.

반면 SPPNet은 입력 이미지를 그대로 CNN에 통과시켜 피쳐 맵을 추출한 다음, 그 feature map에서 2천개의 물체 영역을 찾아 SPP를 적용하여 고정된 크기의 feature를 얻어냅니다. 그리고 이를 FC와 SVM Classifier에 통과시킵니다.





💡 R-CNN은 Selective Search로 찾은 2000개의 영역을 모두 고정 크기로 조절한 후, CNN 모델을 통과시켜 feature를 추출함
→ 때문에 시간이 오래 걸린다는 단점이 발생한다.

SPPNet은 입력 이미지를 그대로 CNN에 통과시켜 feature map을 추출한 후 feature map에서 2000개의 영역을 찾아 SPP를 적용하여 고정된 크기의 feature를 얻어냄
→ SPPNet은 CNN을 이미지에 한번만 적용하여 속도가 빠르다.
→ CNN으로 생성된 feature map에서 selective search를 적용하여 생성된 window에서 특징을 추출합니다.
→ feature map에서 임의의 크기 window로 특징 추출이 가능합니다.
→ 입력 이미지의 scale, size, aspect ratio에 영향을 받지 않습니다.

4.1 Detection Algorithm

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**×	-

테이블9

	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**×	-

테이블10

비슷한 성능을 보여주지만 SPP를 적용하는 편이 속도가 훨씬 빠르다.

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**

테이블11

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**

테이블12

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

테이블13

5. CONCLUSION

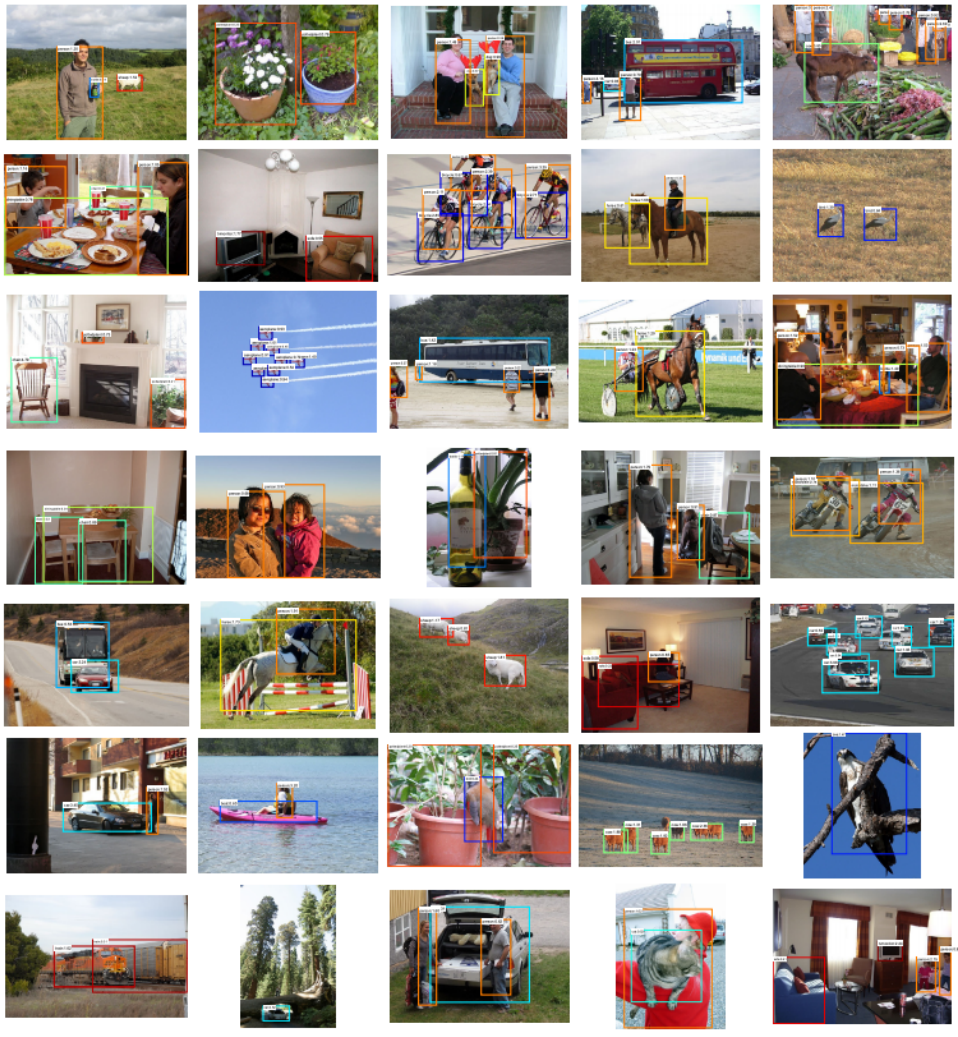


그림6

SPPNet은 다른 크기, 다른 사이즈, 다른 종횡비를 다루기 위한 유연한 방법이다.

Classification과 Detection에서 높은 정확성, 특히 DNN 기반의 Detection을 크게 빠르게 해주었다.

한계점

SPPNet은 기존 R-CNN이 모든 RoI에 대해서 CNN inference를 한다는 문제점을 획기적으로 개선하였지만 여전히 한계점이 있습니다.

1. end-to-end 방식이 아니라 학습에 여러 단계가 필요하다. (fine-tuning, SVM training, Bounding Box Regression)

2. 여전히 최종 클래시피케이션은 binary SVM, Region Proposal은 Selective Search를 이용한다.

3. fine tuning 시에 SPP를 거치기 이전의 Conv 레이어들을 학습 시키지 못한다. 단지 그 뒤에 Fully Connnected Layer만 학습시킨다.

여기서 세 번째 한계점에 대해서 저자들은 **"for simplicity"** 라고만 설명이 되어 있습니다. 이 뒤에 이어지는 Fast R-CNN 논문에서는 앞서 언급된 SPPNet의 한계점을 대폭 개선한 접근 방법이 소개됩니다.

Ref

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

SPPNet : Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition SPPNet ■ 가장 큰 장점 1. R-CNN은 이미지 한 장당 수천 번의 CNN을 수행함(Time-Consuming). → 이미지 안에서 bo..

https://n1094.tistory.com/30

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

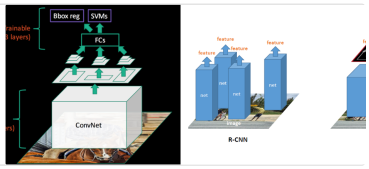
이번에 리뷰할 논문은 SPPnet 'Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition' 입니다. SPPnet은 CNN 구조가 고정된 입력 이미지 크기를 입력으로 취하는 데에서 발생한 문제점을 개선하기 위해 고안되었습니다. 기존 CNN은 고정된 입력 크기를 맞춰주기 위해서 crop, wrap을 적용합니다. 참고로, crop과

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

[논문리뷰] SPPNet - Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition

논문리뷰] Pyramid_pooling - Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition FC layer에서 고정된 크기의 입력을 받으므로 기존의 CNN은 입력이미지의 크기가 고정되어야 함 (crop, warp 사용) 입력 크기에 관계없이 conv를 진행 후 고정된 크기의 feature map을 출력하는 pooling을 적용함 Image Classification나 Object

https://blackchopin.github.io/imagerecognition/Pyramid_pooling/



SPPnet(Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition)

Fast R CNN이 영감을 받았다는 SPPnet이다. Kaiming He가 1저자로 있다. 1. Spatial pyramid pooling을 이용해서 다양한 size/scale의 input에 대해 fixed-length representation을 만들 수 있고, 그러므로 학습 시킬 수 있다. 2. CNN architecture를 가리지 않고 성능 향상이 가능하다. 3. Object detection에서 기존 R CNN과 비슷한 성능을 내면서 최대

https://yun905.tistory.com/19

Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition

arxiv.org/abs/1406.4729 [Submitted on 18 Jun 2014 (v1), last revised 23 Apr 2015 (this version, v4)] 1. 요약 1.1 현황 - 현존하는 심층 합성곱 신경망들은 고정 크기의 (e.g 224 x 224) 입력 이미지를 받고 있습니다. 이러한 요구사항은 이미지 혹은 고정 크기/스케일 하부이미지의 인식 정확도를 떨어트릴수도 있습니다.

https://throwexception.tistory.com/1237

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 중요한 개념이므로 리뷰하고 넘

https://yeomko.tistory.com/14