SPPNet

Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition

2021210088 허지혜

Abstract

- 기존 CNN에서는 고정된 크기를 갖는 이미지를 입력으로 삼는다.

- Spatial pyramid pooling 방법을 적용한 Network를 이용하여 고정된 크기의 입력 이미지가 아닌 크기/비율과 상관없이 일정한 크기의 출력을 반환한다. 이를 통해 이미지 분류와 객체 탐지에서 모두 좋은 성능을 나타낼 수 있다.

- 특히 장점은 feature map을 한 번만 수행하기 때문에 속도가 빠르다. Pascal voc 2007 기준 r-cnn보다 좋은 정확도를, 속도도 24배 ~ 102배 빠르다. ILSVRC 2014에서는 객체 탐지 분야 2등, 이미지 분류 분야 3등을 차지하였다.

1. Introduction

- CV 분야는 CNN과 빅데이터에 의해 빠르게 발전하고 있다. 특히, 딥러닝 기반의 접근 방식이 이미지 분류, 객체 탐지 등에 상당한 기여를 하고 있다.
- 하지만, CV 분야에서 CNN을 활용하는 데 기술적인 문제가 있다. CNN에서는 고정된 입력 이미지를 필요로 한다. 다양한 크기로 변환을 하기 위해서는 Crop, Warp 등의 작업이 필요하다.

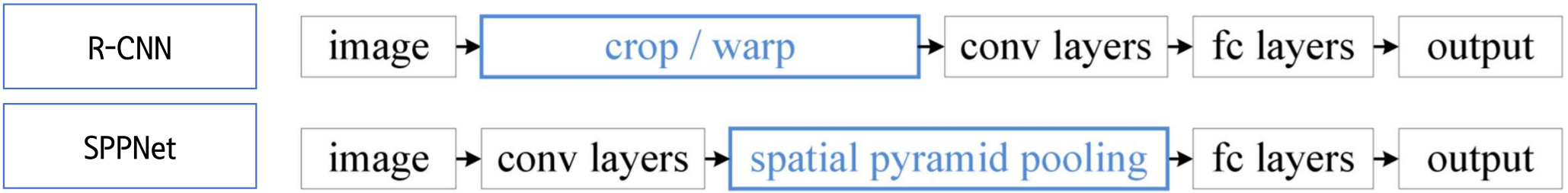
crop

warp

- 하지만 위 작업은 객체 전체를 포함하지 못하거나 가로세로 비율이 달라져 찌그러지는 등 단점을 포함한다. 고정된 입력 이미지를 쓰기 위해선 성능을 떨어트려야 한다.
- CNN은 크게 Convolution layer와 FC layer로 나뉘는데, 고정된 입력 이미지는 FC layer 때문이다.

1. Introduction

- 본 논문에서는 SPP(Spatial Pyramid Pooling) 방식을 소개한다. 이 방식은 CNN이 고정된 크기의 입력 이미지를 받는다' 라는 제약조건을 해결해준다.
- Convolution layer와 FC layer 사이에 SPP 방식을 적용한다. SPP 방식을 이용하여 고정된 크기의 결과값을 내주기 때문에 따로 데이터를 전처리할 필요가 없다.

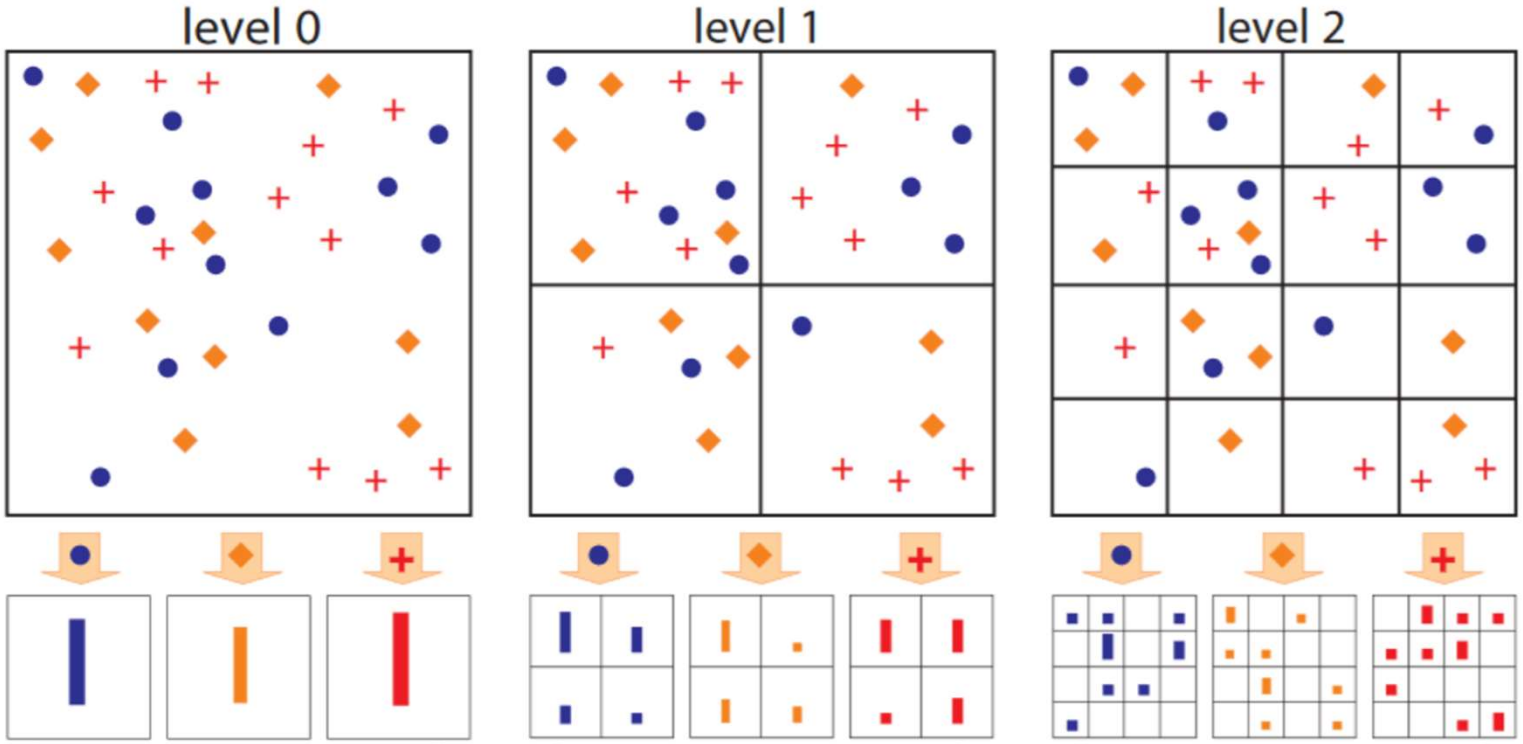


- 두 모형의 차이를 나타내보면 위와 같은 순서로 나타낼 수 있다.

1. Introduction

Spatial pyramid pooling 방식이란?

- 이미지를 여러 영역으로 나눈 후 각 영역별 특성을 파악하는 방식이다.



1. Introduction

- Spatial pyramid pooling 기법은 CNN이 유행하기 전까지 우수한 성능을 보인 기법이다. 하지만 CNN과 Spatial pyramid pooling을 결합하려는 시도는 지금까지의 논문에서는 없었다.
- 연구진은 SPP 방식을 CNN에 적용할 때 특징이 있음을 발견한다.

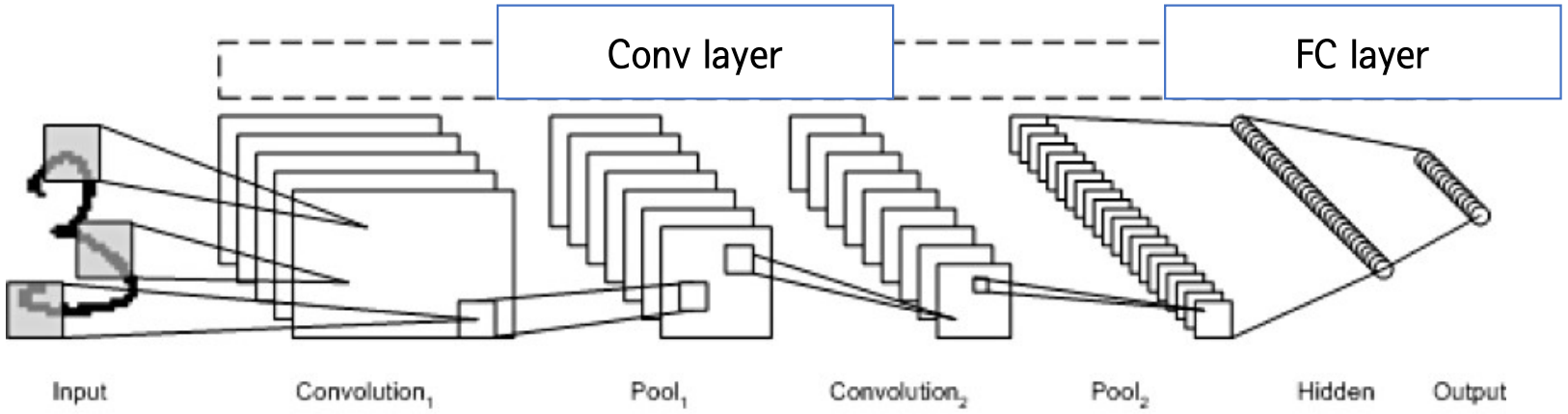
1) 입력 이미지 size와 상관없이 고정된 크기의 결과를 출력한다.
2) Multi-level spatial bins를 사용한다.
3) SPP는 다양한 scale의 feature를 pooling 할 수 있다.

- 이러한 특징 덕분에 객체 탐지 정확도가 높아질 수 있었다. 또한 별도의 처리 과정을 하지 않아도 되기 때문에 이미지 크기에 강건해지고 과대적합도 방지할 수 있게 되었다.
- 본 논문에서는 CNN과 SPP 방식을 결합하는 방식을 새롭게 소개하였다.

2. Deep Networks With Spatial Pyramid Pooling

2.1 Convolutional Layers and Feature Maps

- 7 Layer를 가지는 CNN 구조를 생각해보자.

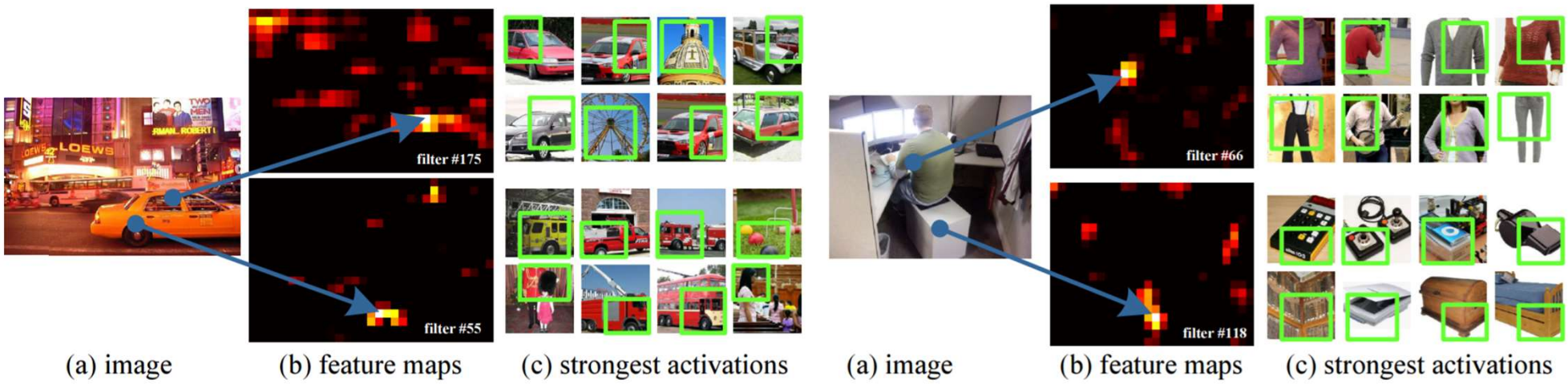


http://taewan.kim/post/cnn/

- Conv layer에서는 어떤 크기의 입력 이미지를 받아도 괜찮다. 즉, 고정되지 않아도 된다. 반면 FC layer는 고정된 크기의 feature map이 필요하다 .

2. Deep Networks With Spatial Pyramid Pooling

2.1 Convolutional Layers and Feature Maps



- Con5의 filter가 도출한 feature map을 시각화하면 다음과 같다. Feature map은 **response 강도와 위치 정보**를 포함한다는 것을 위 시각화를 통해 보여준다.

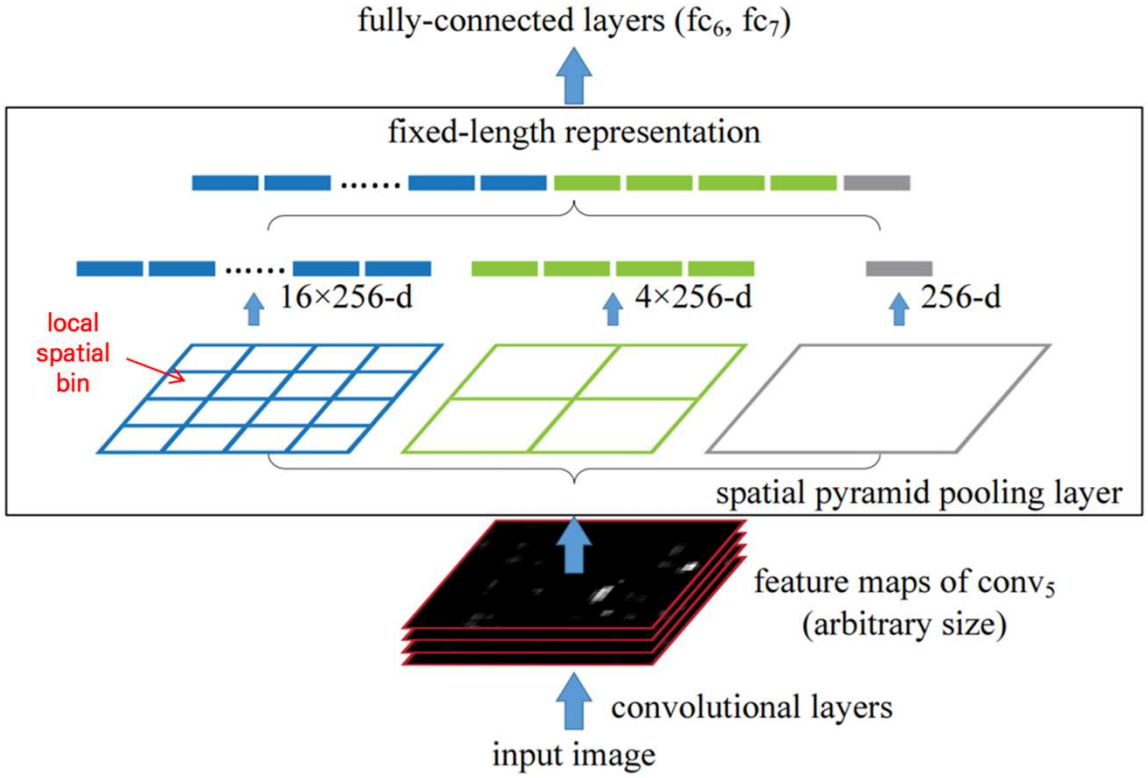
2. Deep Networks With Spatial Pyramid Pooling

2.2 The Spatial Pyramid Pooling Layer

- Spatial pyramid pooling 방식은 local spatial bins(부분 공간 격자)를 통해 spatial information(공간 정보)를 추출한다. 이때 local spatial bins의 크기는 입력 이미지의 크기에 **비례**한다. 따라서 큰 이미지는 local spatial bins를 늘리면 되고, 작으면 줄이면 되기 때문에 이미지 크기에 구애받지 않는다.
- 다양한 크기의 입력 이미지에 대응하기 위해 마지막 pool5 layer를 spatial pyramid pooling layer로 변경해야 한다.

2. Deep Networks With Spatial Pyramid Pooling

2.2 The Spatial Pyramid Pooling Layer



- 첫번째, Spatial pyramid pooling layer는 bins를 한 개만 갖는다. 1개(1x1)의 bin이 이미지 전체를 커버할 수 있다.
- 중간은 4개(2x2)의 bins가, 오른쪽은 16개(4x4)의 bins가 있는 것을 확인할 수 있다.
- 각 spatial bins마다 response를 max pooling한다.
- 위 구조로 설명했을 때의 결과값은 (16+4+1)*256 = 5376이다.
- 입력 이미지가 어떻게 들어오든 bins 값에 따라 fc layer에 동일한 크기로 들어올 수 있다는 것을 알 수 있다.

2. Deep Networks With Spatial Pyramid Pooling

2.3 Training the Network

Single size training

- 한 가지 크기의 이미지로만 훈련하는 방식
- 입력 이미지의 크기가 일정하면 SPP에서 사용할 bin의 크기를 미리 계산할 수 있다.
- Single size training의 목적은 multi-level pooling을 하기 위해서다.
- Conv5를 거친 feature map의 크기가 13x13이라고 하고, spatial bin 크기를 2x2로 만들고 싶으면 window = 7, stride =6으로 설정하여 만들면 된다.
- 이를 공식으로 나타내면 다음과 같다.
- **Window 크기 = a/n 올림 값**
- **Stride 크기 = a/n 내림 값**

```
[pool3x3]
type=pool
pool=max
inputs=conv5
sizeX=5
stride=4

[pool2x2]
type=pool
pool=max
inputs=conv5
sizeX=7
stride=6

[pool1x1]
type=pool
pool=max
inputs=conv5
sizeX=13
stride=13

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

2. Deep Networks With Spatial Pyramid Pooling

2.3 Training the Network

Multi size training

- 여러 크기의 이미지로 훈련하는 방식
- 이미지 크기가 다양해지는 문제를 다루기 위해 (180x180), (224x224) 크기로 생각을 해 보았을 때, 먼저 (180x180) 를 전체 epochs만큼 훈련한 후 (224x224) 를 전체 epochs만큼 훈련한다. 이렇게 반복하면 Multi-size training은 Single-size training과 비슷해진다.
- 이를 하는 이유는 다양한 크기의 입력 이미지로 실험을 해보기 위함이다. 이렇게 훈련하면 다양한 크기의 이미지로 test에 SPPNet을 적용할 수 있다고 한다.

3.1 Experiments on ImageNet 2012 Classification

- 1000개의 class를 가지는 ImageNet 2012 dataset을 이용하여 SPPNet을 훈련하였다.
- h,w는 256이 되도록 이미지의 size를 조절하고 이미지의 center와 네 모서리 중 하나를 224x224 size로 crop하여 뽑는다. Horizontal flipping, color altering 등의 data augmentation도 한다.
- 마지막 FC layer에서는 dropout을 적용하고 learning rate는 기본으로는 0.01로 설정한다. 오류값이 정체되면 10으로 나누어서 다시 진행한다.
- 전체 네트워크를 GeForce GTX Titan GPU(6GB)로 훈련하면 2주 ~ 4주가 걸린다.

3.1 Experiments on ImageNet 2012 Classification

3.1.1 Baseline Network Architectures

Filter 개수 x Filter 크기

Local Response Normalization

mode		$conv_2$	$conv_3$	$conv_4$	$conv_5$	$conv_6$	$conv_7$
ZF-5	96×7^2, str 2 LRN, po map siz	256×5^2, str 2 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

- **ZF-5**: Zeiler and Fergus(ZF) 'fast' 모델. Conv layer가 5개이다.
- **Convnet*-5**: Krizhevsky의 네트워크를 조금 변형한 모델. conv1과 conv2 대신 conv2와 conv3 다음에 pooling layer를 두었다. 변형 결과, 각 계층별 피처 맵 크기는 ZF-5와 같다.
- **Overfeat-5/7**: Overfeat을 조금 변형한 모델. ZF-5/Convnet*-5와 다르게, 이 네트워크는 마지막 pooling layer 전에 더 큰 feature map을 도출한다(13 x 13 대신 18 x 18). conv3과 그 이후 Conv layer에서 더 큰 필터 크기(512)를 사용한다. Conv layer 7개를 갖는 Overfeat-7은 conv3부터 conv7까지 같은 구조를 가진다.

3.1 Experiments on ImageNet 2012 Classification

3.1.1 Baseline Network Architectures

		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)

4 level pyramid 사용
{(6x6), (3x3), (2x2), (1x1)}

		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)

- ImageNet 2012의 검증 데이터셋 오류율을 표시하였다. 여기서 괄호 안 숫자들은 no SPPNet과의 차이를 나타내었다.
- ZF-5 모형만 70epochs로, 나머지는 90epchs로 훈련하였다.
- Single size train한 것 보다 multi-size로 train한 게 더 좋다

3.1 Experiments on ImageNet 2012 Classification

3.1.2 Multi-level Pooling Improves Accuracy

- 1000개의 class를 가지는 ImageNet 2012 dataset을 이용하여 SPPNet을 훈련하였다.
- h,w는 256이 되도록 이미지의 size를 조절하고 이미지의 center와 네 모서리 중 하나를 224x224 size로 crop하여 뽑는다. Horizontal flipping, color altering 등의 data augmentation도 한다.
- 마지막 FC layer에서는 dropout을 적용하고 learning rate는 기본으로는 0.01로 설정한다. 오류값이 정체되면 10으로 나누어서 다시 진행한다.
- 전체 네트워크를 GeForce GTX Titan GPU(6GB)로 훈련하면 2주 ~ 4주가 걸린다.

3.1 Experiments on ImageNet 2012 Classification

3.1.4 Full-image Representations Improve Accuracy

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**

- 전체 이미지를 이용해 성능을 측정해보면 다음과 같다.
- 가로세로 비율을 유지하면서 h,w가 256이 되도록 이미지 size를 조정 한 후 SPP 방식을 적용한다.
- 공정함을 위하여 test는 전체 이미지와 crop 이미지를 사용하여 성능을 측정하였다. 모든 경우에서 전체 이미지를 사용해 test를 한 경우 오류율이 낮아졌다. 이는, **전체 이미지를 살리는 것도 중요하다는 것**을 보여준다.

3.1 Experiments on ImageNet 2012 Classification

3.1.6 Summary and Results for ILSVRC 2014

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**

- ILSVRC 2012에서 우승한 Krizhevsky et al 모형과 ILSVRC 2013에서 우수했던 Overfeat, Howard, Zeiler & Fergus 모형과 비교를 해보았다.

3.1 Experiments on ImageNet 2012 Classification

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 3등을 했다.
- 1등, 2등은 지금도 유명한 GoogLeNet와 VGG이다.

3.2 Experiments on VOC 2007 Classification

3.1.6 Summary and Results for ILSVRC 2014

Baseline model

객체의 scale이 모형의 성능에 영향을 준다.

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	-	-	-
$conv_5$	66.24	65.43	-	-	-
$pool_{5/7}$ (6×6)	69	76	70.82	71.67	76.09
$fc_{6/8}$	74.66	75.55	77.32	78.78	81.58
$fc_{7/9}$	75.90	76.45	78.39	80.10	**82.44**

깊어질수록 성능이 좋아진다.

Backbone network가 좋으면 성능이 좋아진다.

- VOC 2007 dataset에도 적용시켜보았다.
- Train : 5011개, Test : 4952개의 이미지가 있고 20개의 class가 있는 데이터셋이다.
- 판단 척도는 mAP(mean Average Precision)이다.

4. SPPNet For Object Detection

	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**×	-

- Ft : fine tuning
- Ftfc7 : fc7에 fine tuning 적용 모형
- Ftfc7 bb : ftfc7에 bounding box regression 적용
- 판단 척도 Map 사용하였다.

5. Conclusion

- SPP는 이미지 scale, size, 가로세로 비율을 유연하게 처리하는 알고리즘이다.
- 이는 객체 탐지 분야에서 중요한 점이 될 수 있는데 딥러닝 객체 탐지 모형은 이미지 scale이나 비율을 유연하게 다룰 수 없었다.
- 본 논문은 spatial pyramid pooling 기법을 딥러닝에 적용하는 방법을 소개하였다.
- 이를 적용한 SPPNet은 이미지 분류나 객체 인식 분야에서 성능도 뛰어나고 속도도 빠르다.

감사합니다