Going deeper with convolutions

Christian Szegedy
Google Inc.

Wei Liu
University of North Carolina, Chapel Hill

Yangqing Jia
Google Inc.

Pierre Sermanet
Google Inc.

Scott Reed
University of Michigan

Dragomir Anguelov
Google Inc.

Dumitru Erhan
Google Inc.

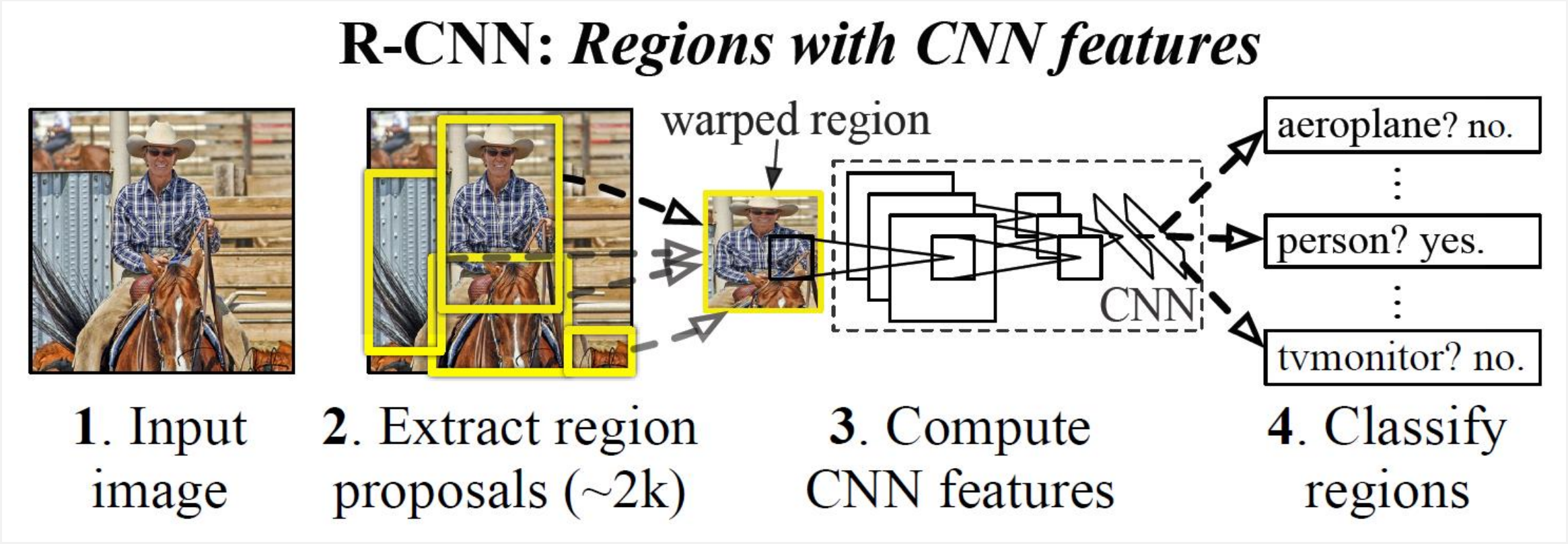
Vincent Vanhoucke
Google Inc.

Andrew Rabinovich
Google Inc.

목차

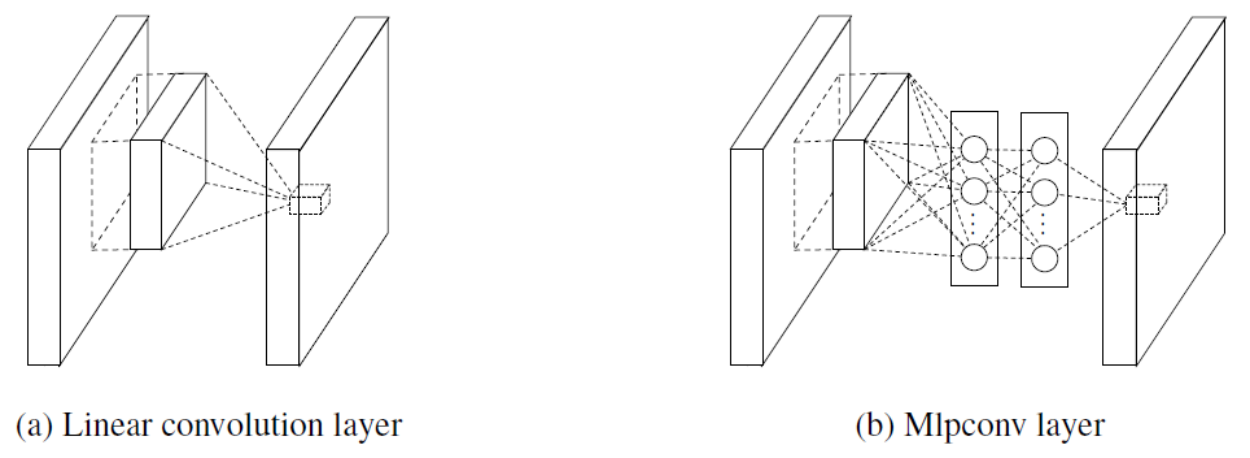
1. R-CNN & Network-in-Network

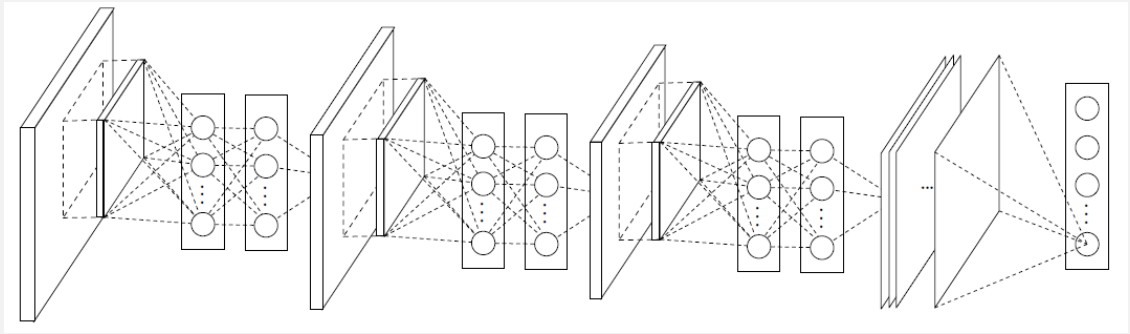
R-CNN

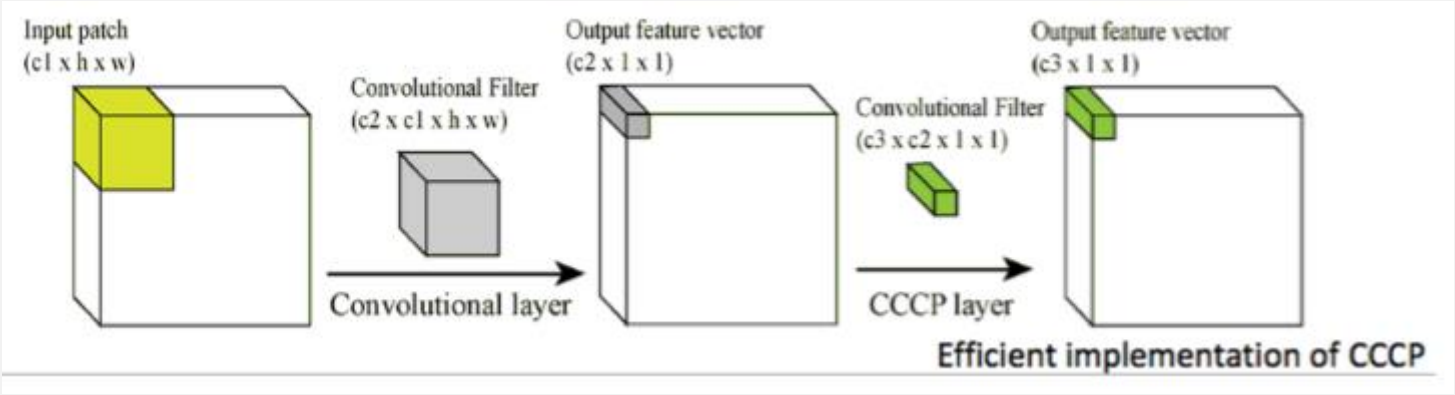


1. R-CNN & Network-in-Network

Network-in-Network







http://nmhkahn.github.io/Casestudy-CNN

2. Introduction & Related Work

INTRODUCTION

- GoogLeNet은 ILSVRC 2014에서 AlexNet보다 12배 적은 parameter 사용
  → 하지만 더 정확하다!

RELATED WORK

- Inception layer → Repeat many times
- Network-in-Network → Dimension Reduction
- R-CNN → Multi-box& Ensemble

3. Motivation and High Level Considerations

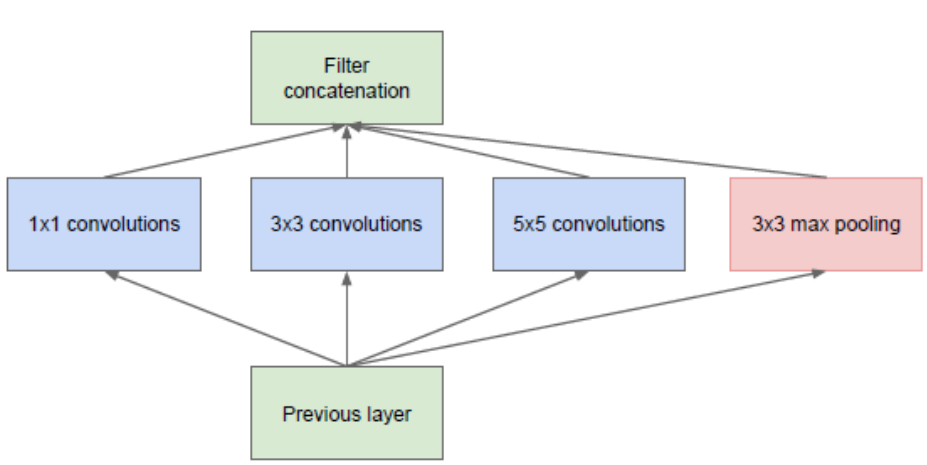
- 성능을 높일 수 있는 방법은 모델의 사이즈를 키우는 것이다.
  → Parameter 수 증가(overfitting ↑) & Computational resources ↑

- Fully connected → Sparsely connected architectures in the convolutions
- Dense submatrices → Clustering sparse matrices

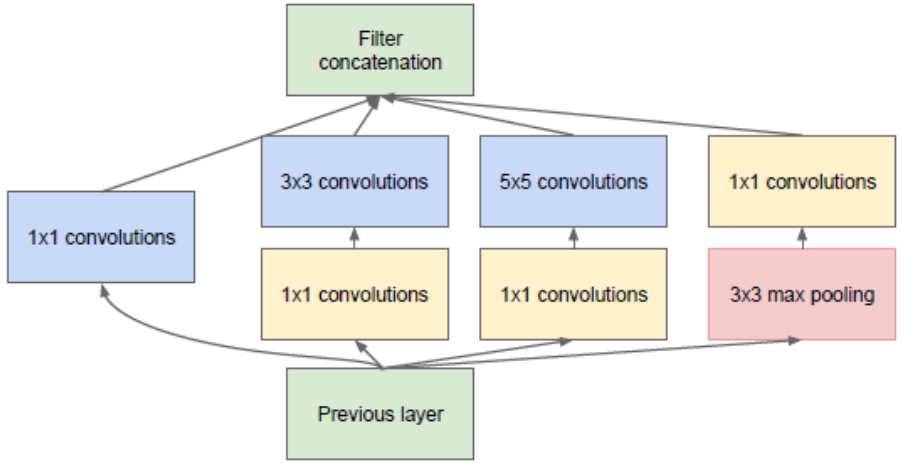
4. Architectural Details

- Inception architecture

  Dense components과 비슷한 성능을 내는 Local sparse structure



(a) Inception module, naïve version



(b) Inception module with dimension reductions

Figure 2: Inception module

5. GoogLeNet

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Table 1: GoogLeNet incarnation of the Inception architecture

https://github.com/rlatjcj/Keras-Model/blob/master/Inception_v1/model.py

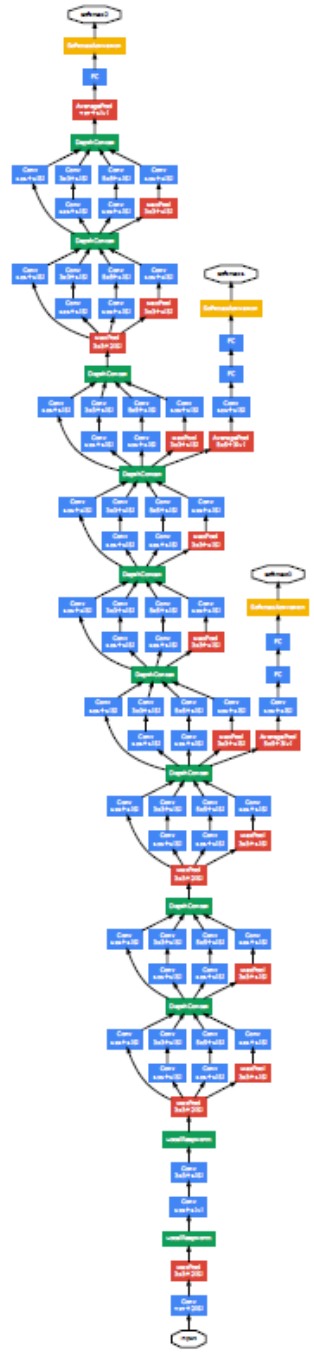


Figure 3: GoogLeNet network with all the bells and whistles

6. Training Methodology

- Stochastic Gradient Descent with 0.9 momentum
- Fixed Learning Rate Schedule (Decreasing the learning rate by 4% every 8 epochs)
- Use Polyak averaging to create the final model used at inference time.
  (http://ttic.uchicago.edu/~shubhendu/Pages/Files/Lecture6_flat.pdf)

- 하지만, transfer learning 할 때는 dropout이나 learning rate 같은 option들을 바꿀 수 있음
- Crop image whose size is distributed evenly between 8% and 100% and aspect ratio is chosen randomly between 3/4 and 4/3
- Random interpolation (bilinear, area, nearest neighbor, cubic with equal probability)

7. ILSVRC 2014 Classification & Detection

Classification

- 7개의 GoogLeNet (with one wider version)을 동일한 초기조건, learning rate policies로 학습
  Sampling methodologies와 입력 이미지의 순서만 다름
  예측할 때 Ensemble 수행

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Table 2: Classification performance

Number of models	Number of Crops	Cost	Top-5 error	compared to base
1	1	1	10.07%	base
1	10	10	9.15%	-0.92%
1	144	144	7.89%	-2.18%
7	1	7	8.09%	-1.98%
7	10	70	7.62%	-2.45%
7	144	1008	6.67%	-3.45%

Table 3: GoogLeNet classification performance break down

7. ILSVRC 2014 Classification & Detection

Detection

- R-CNN과 접근이 유사하지만 Region classifier에서 Inception model로 증강
- Region proposal step에 Selective Search를 추가하여 개선

  → Superpixel size를 2배 증가시켜 Selective Search Algorithm 시행을 절반으로 줄임
- Use Ensemble of 6 ConvNets when classifying each region

→ R-CNN과 달리, 시간 지연의 이유로 bbox regression을 사용하지 않았다.

Team	Year	Place	mAP	external data	ensemble	approach
UvA-Euvision	2013	1st	22.6%	none	?	Fisher vectors
Deep Insight	2014	3rd	40.5%	ImageNet 1k	3	CNN
CUHK DeepID-Net	2014	2nd	40.7%	ImageNet 1k	?	CNN
GoogLeNet	2014	1st	43.9%	ImageNet 1k	6	CNN

Table 4: Detection performance

Team	mAP	Contextual model	Bounding box regression
Trimps-Soushen	31.6%	no	?
Berkeley Vision	34.5%	no	yes
UvA-Euvision	35.4%	?	?
CUHK DeepID-Net2	37.7%	no	?
GoogLeNet	38.02%	no	no
Deep Insight	40.2%	yes	yes

Table 5: Single model performance for detection

Reference

- Inception v1

  C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, *"Going Deeper with Convolutions"*, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. (https://arxiv.org/pdf/1409.4842.pdf)

- R-CNN

  R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", in IEEE Conference on Computer Vision and Pattern Recognition, 2014. (https://arxiv.org/pdf/1311.2524.pdf)

- Network-in-Network

  M. Lin, Q. Chen and S. Yan, "Network in network", in CoRR, 2013. (https://arxiv.org/pdf/1312.4400.pdf)

- AlexNet

  A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", in Neural Information Processing Systems, 2012. (http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf)

감사합니다