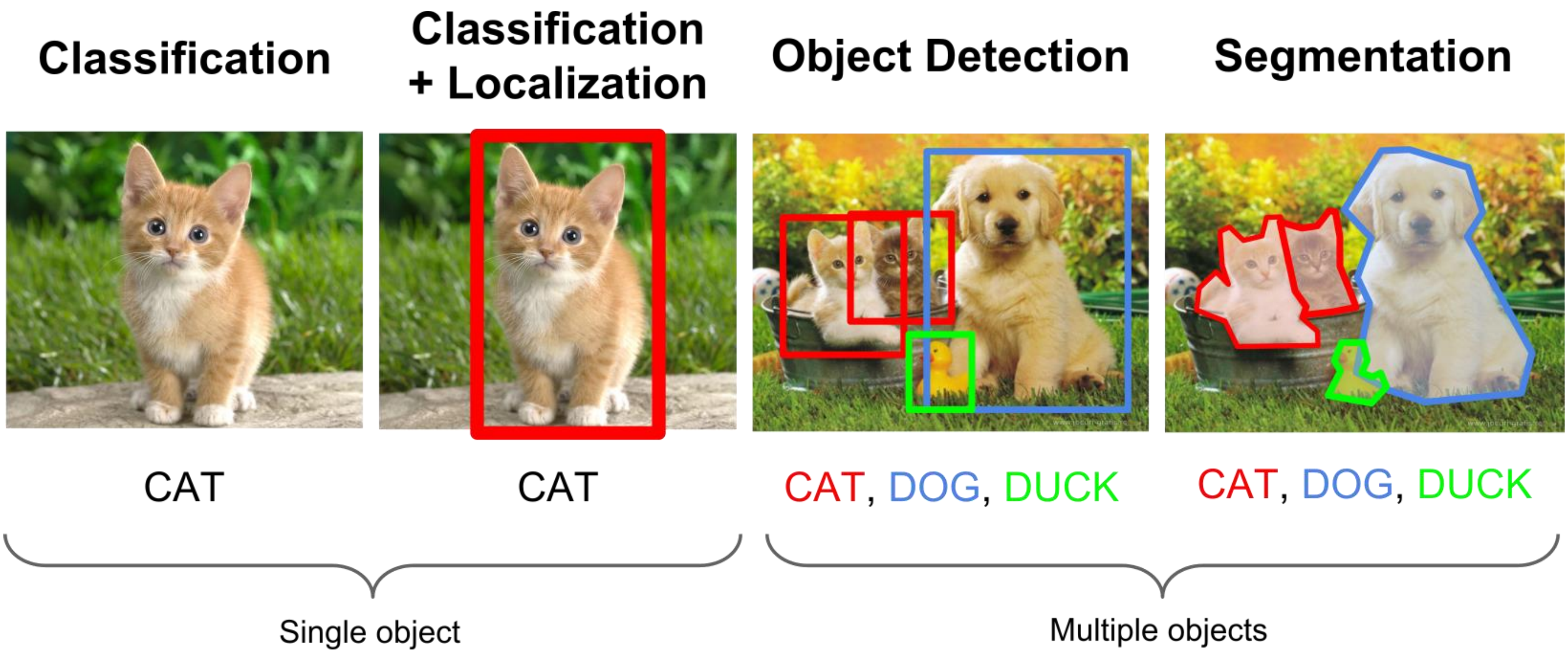
Segmentation and Attention

Jason Park
jason_park@korea.ac.kr
Data Science and Business Analytics lab

Computer Vision



http://cs231n.stanford.edu/slides/winter1516_lecture13.pdf

Problem Definition

Segmentation

Attention

Problem Definition

Segmentation

Attention

Segmentation

Raw image

Semantic Segmentation

Instance Segmentation

https://chaosmail.github.io/deeplearning/2016/10/22/intro-to-deep-learning-for-computer-vision/

- Semantic Segmentation
  - 모든 픽셀을 label을 1개씩 할당하는 것
- Instance Segmentation
  - 각 Instance를 나누어 픽셀에 label을 할당
  - 한 class에 대하여 여러 개체를 구분

Attention

풀꽃

나태주

자세히 보아야
예쁘다.

오래 보아야
사랑스럽다.

너도 그렇다.

Attention

- CNN은 이미지 전체를 이용하여 task를 수행
- 모든 pixel이 같은 중요성을 갖지 않음
- Task에 따라 집중해야 할 영역이 어디인가?

Problem Definition

Segmentation

Attention

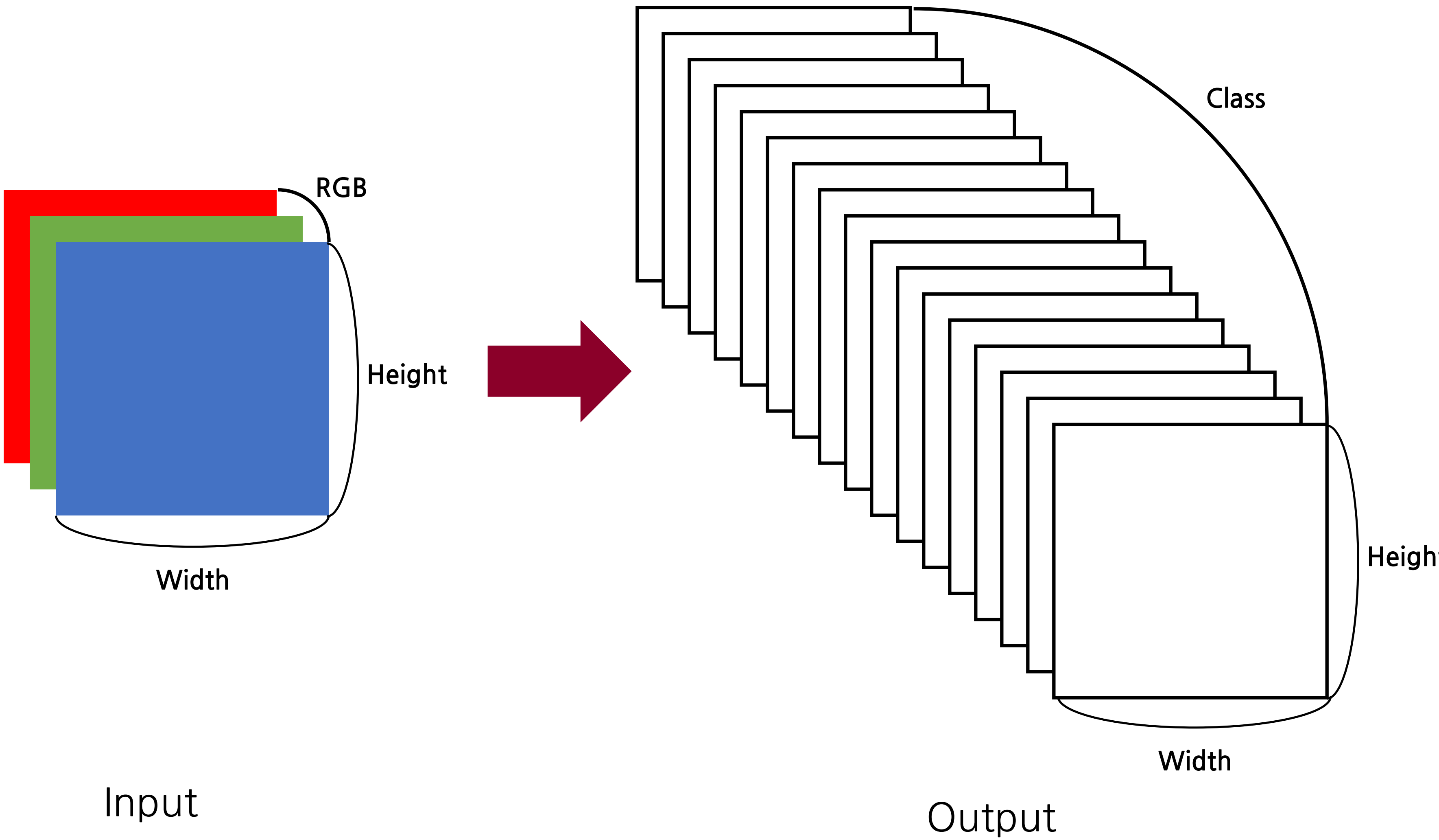
Semantic Segmentation

Semantic Segmentation

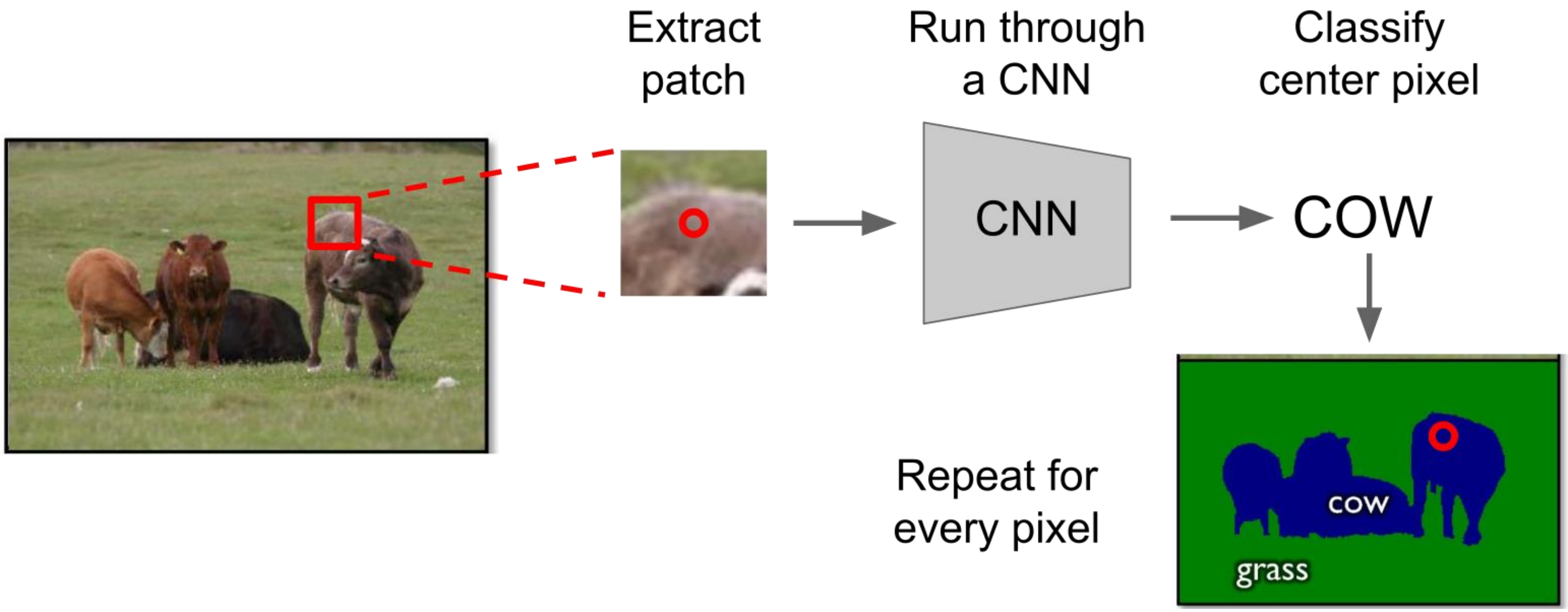
https://chaosmail.github.io/deeplearning/2016/10/22/intro-to-deep-learning-for-computer-vision/

- 이미지의 모든 pixel에 label을 할당
- 몇 번째 고양이의 pixel인지는 구분하지 않음

Semantic Segmentation



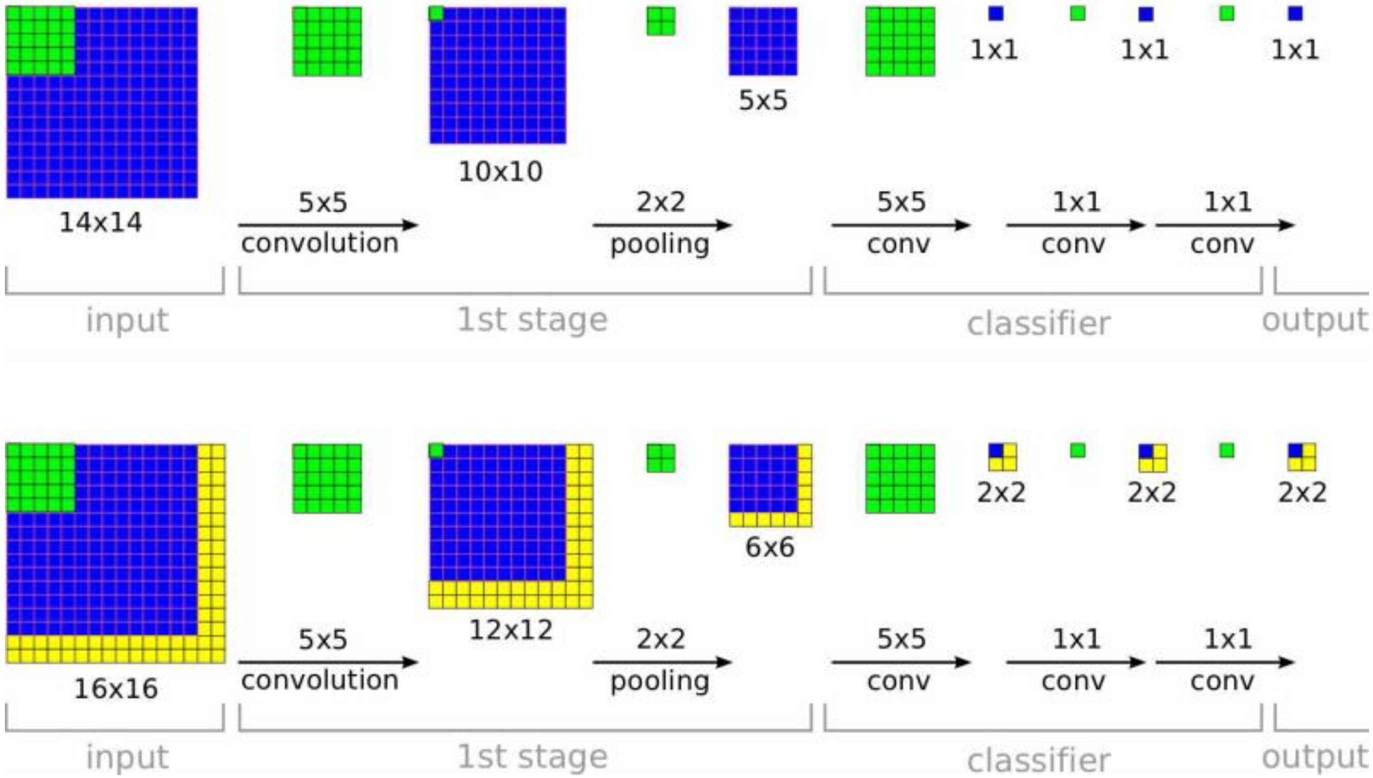
Brute-Force



Expensive!!!

http://cs231n.stanford.edu/slides/winter1516_lecture13.pdf

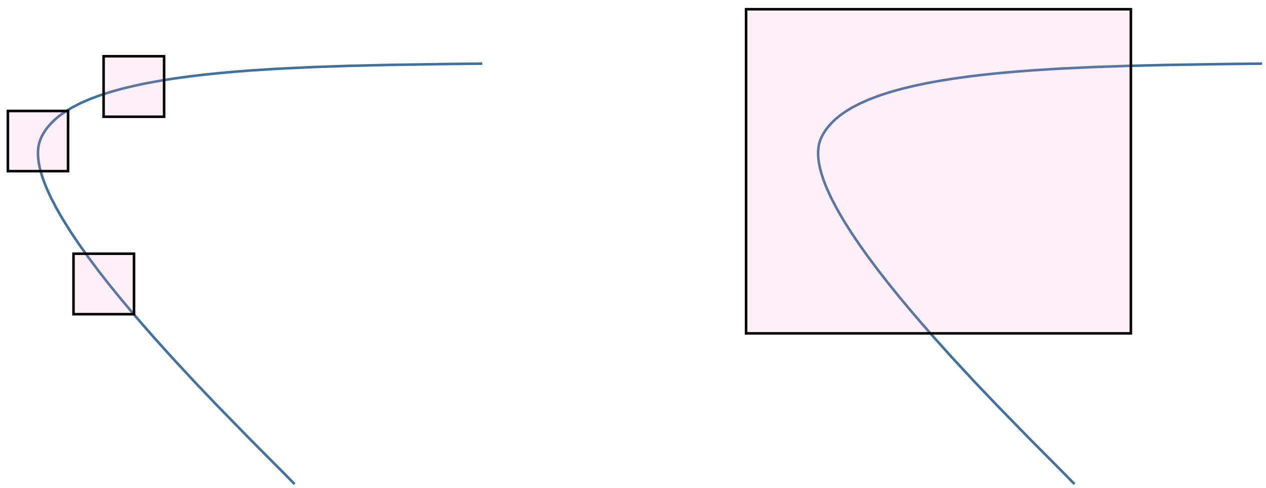
Brute-Force



Trick used in OverFeat

Sermanet et al, “OverFeat:Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Scale 문제



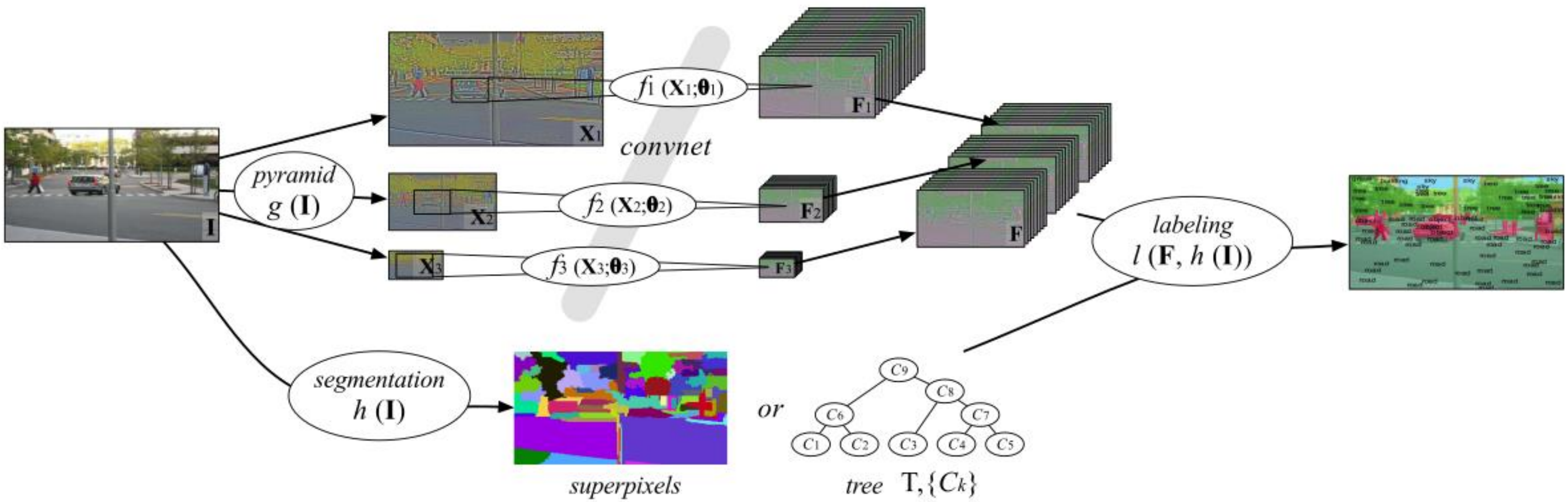
- 같은 선도 Scale에 따라서 corner일 수도, line일 수도 있음
- Segmentation도 scale을 고려하여 판단할 필요가 있음

http://darkpgmr.tistory.com/137 참고

Learning Hierarchical Features for Scene Labeling

Learning Hierarchical Features for Scene Labeling

- Clément Farabet, Camille Couprie, Laurent Najman, Yann Lecun
- IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2013

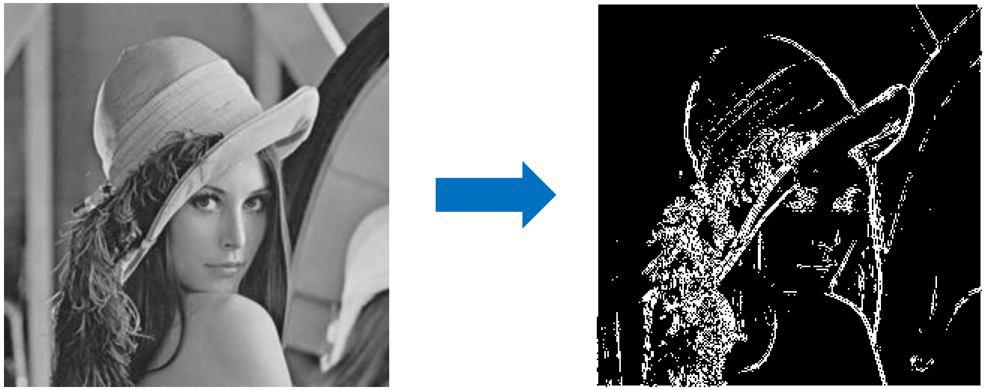


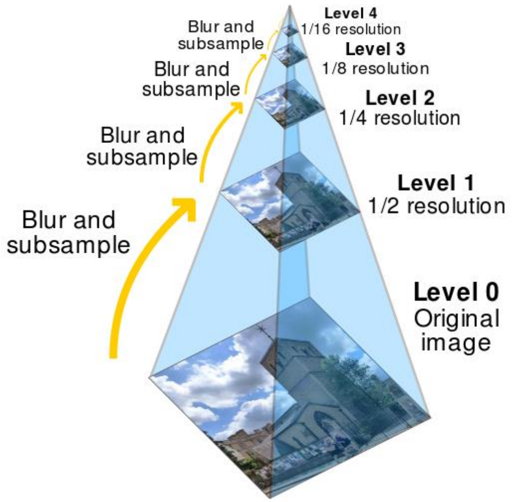
Learning Hierarchical Features for Scene Labeling

- Image Pyramid
  - Filtering → Sampling 반복
  - Gaussian pyramid, Laplacian pyramid, Steerable pyramid…
- Laplacian Pyramid
  - Laplace equation $\Delta f = \nabla \cdot \nabla f = \sum_i \frac{\partial^2 f}{\partial x_i^2} = 0$에서 착안, 다음과 같은 필터 사용

0	1	0
1	-4	1
0	1	0

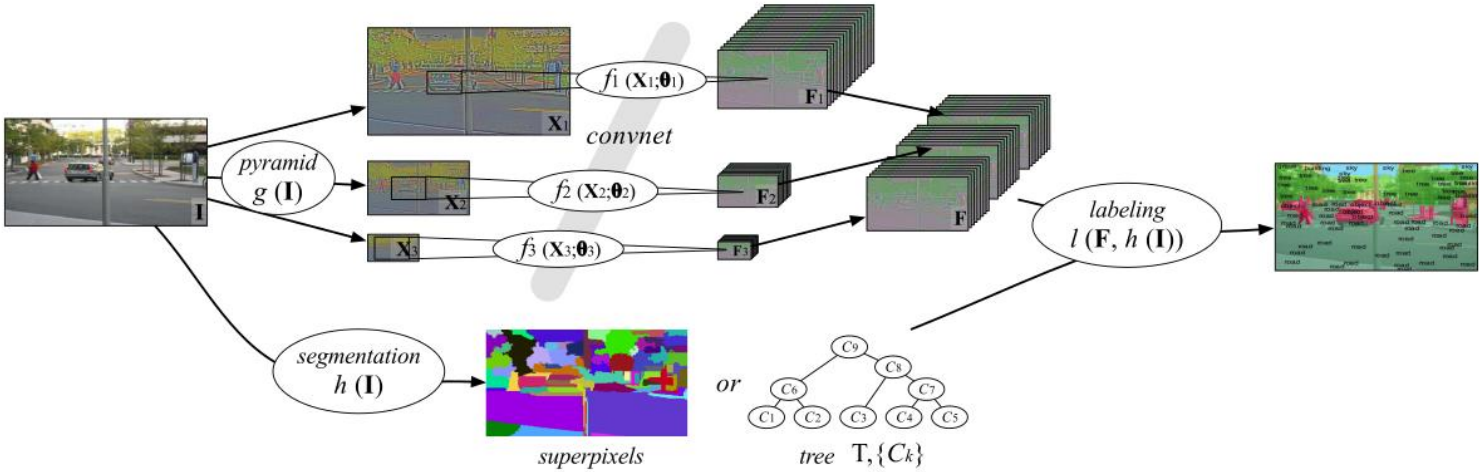
1	1	1
1	-8	1
1	1	1





https://en.wikipedia.org/wiki/Pyramid_(image_processing)

Learning Hierarchical Features for Scene Labeling



- 320×240, 160×120, 80×60 image 생성
- Shared parameter로 CNN 통과
- Feature map 3개 중, 가장 큰 Feature map에 맞추어 upsample, concatenate
  - Upsample 방법이 구체적으로 나와있지 않음
- 한 pixel은 raw image의 46×46, 92×92, 184×184의 정보를 담음
- CNN만을 사용시 모자란 부분을 Superpixel, Conditional Random Field(CRF) 방법론으로 Post-processing

Learning Hierarchical Features for Scene Labeling(2013)

https://www.youtube.com/watch?v=ZJMtDRbqH40

Recurrent Convolutional Neural Networks for Scene Labeling

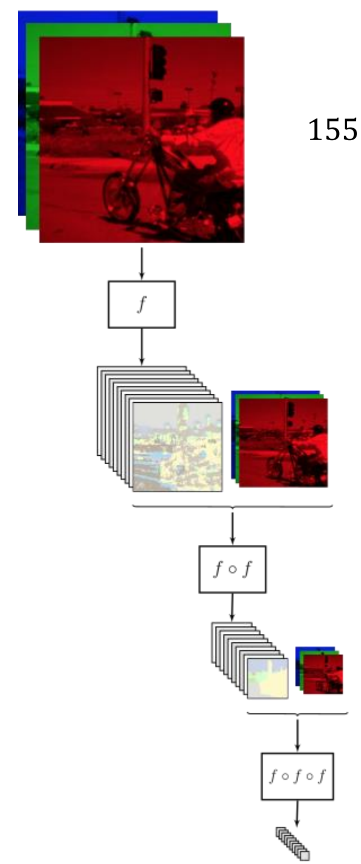
Recurrent Convolutional Neural Networks for Scene Labeling

- Pedro O. Pinheiro, Ronan Collobert
- International Conference on Machine Learning(ICML), 2014

Recurrent Convolutional Neural Networks for Scene Labeling

- Input patch에 해당하는 center pixel에 label
- 이어지는 Image는 원본 이미지를 scale
- 첫 input의 N_{cls} channel은 0으로 pad

$rCNN_3$



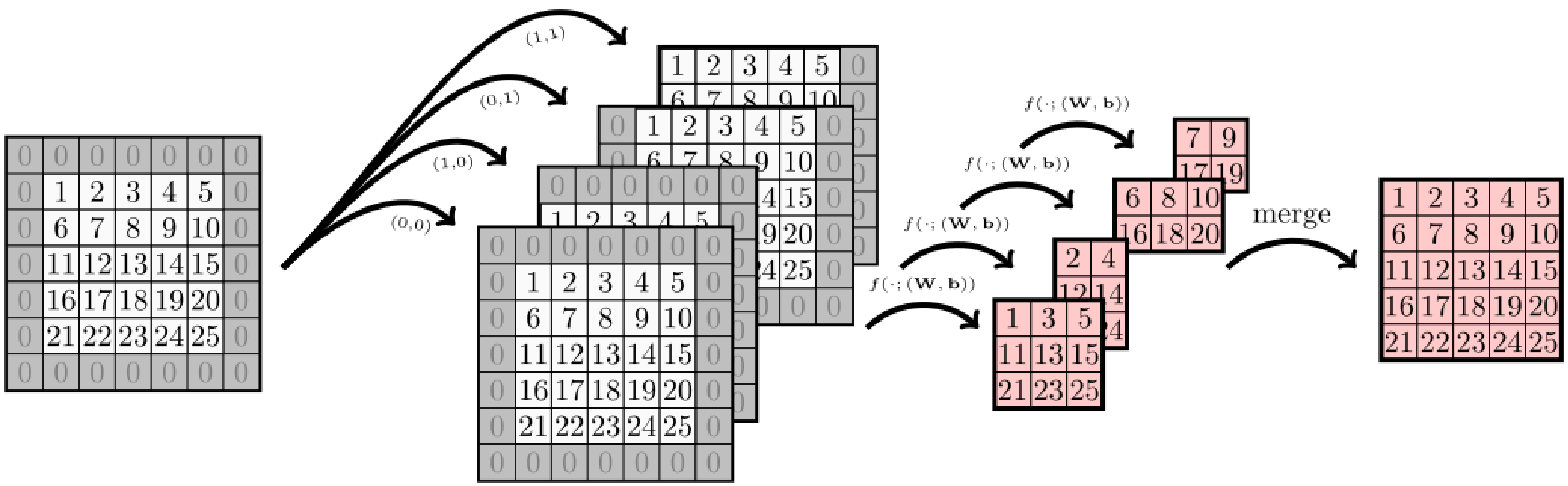
$155 \times 155 \times (3 + N_{cls})$

$67 \times 67 \times (3 + N_{cls})$
Scaled image

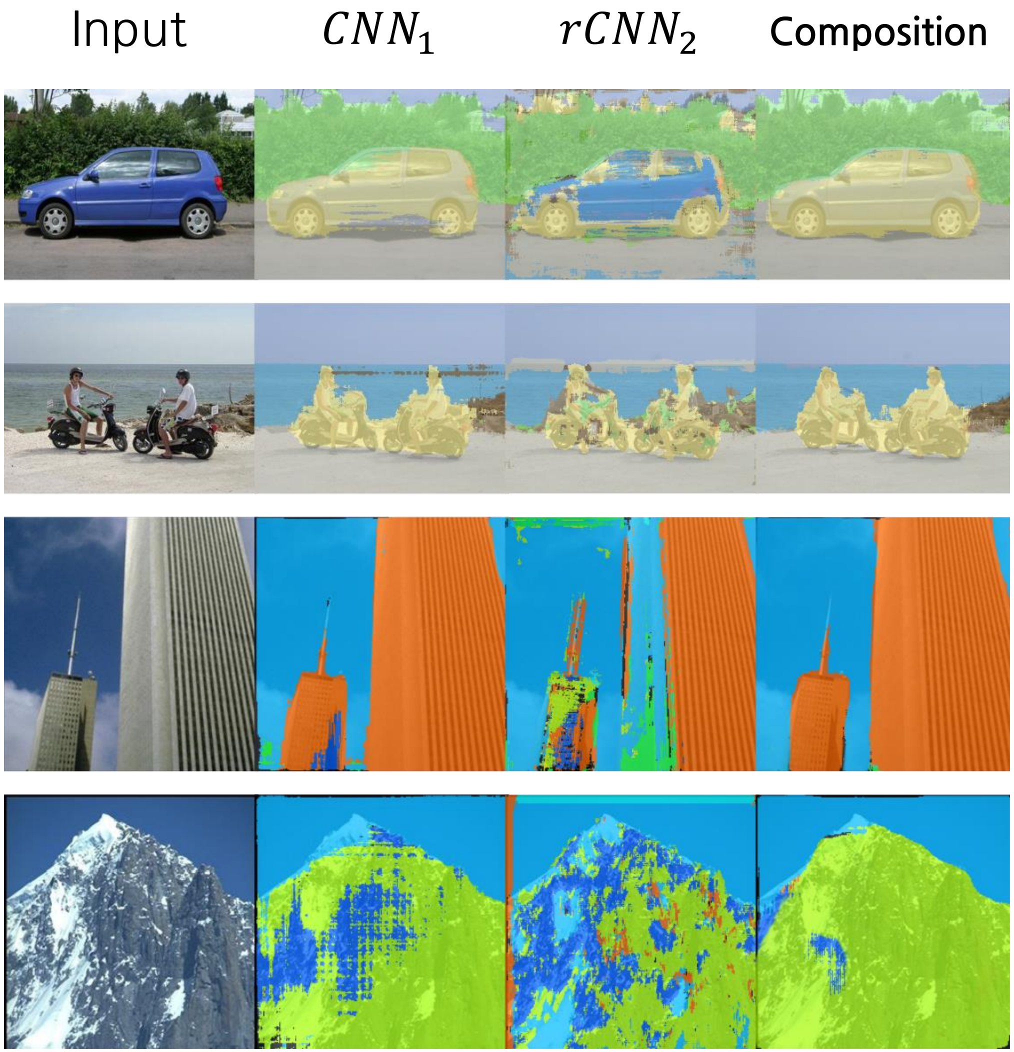
$23 \times 23 \times (3 + N_{cls})$
Scaled image

Recurrent Convolutional Neural Networks for Scene Labeling

- 빠른 계산을 위한 trick
  - OverFeat와 유사하지만 모든 pixel에 대하여 Label을 하기 위해 새로운 trick
  - 2 × 2 pooling layer에 대하여, zero padding을 다르게 적용하여 4번 통과
  - 이후 원 pixel에 대한 label로 복원

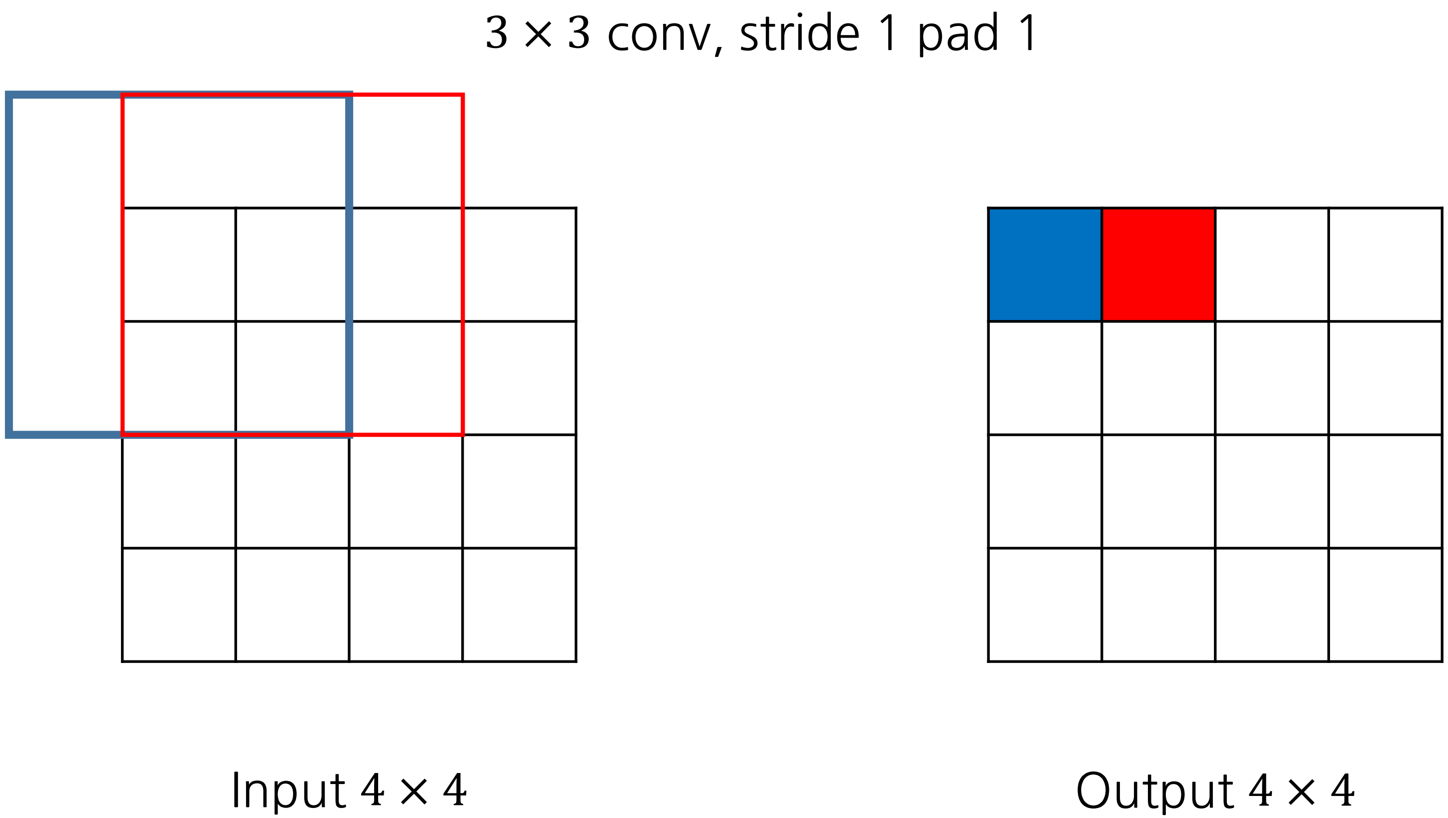


Recurrent Convolutional Neural Networks for Scene Labeling



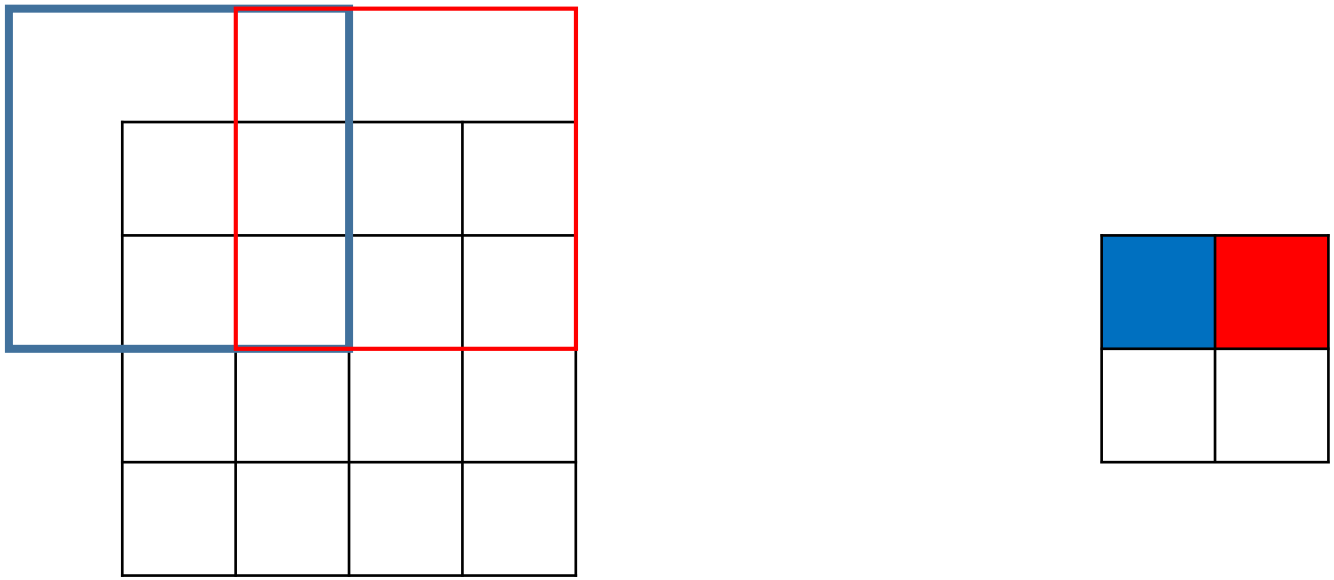
Convolutional Transpose (Deconvolution)

Learnable Upsampling: "Deconvolution"



Learnable Upsampling: "Deconvolution"

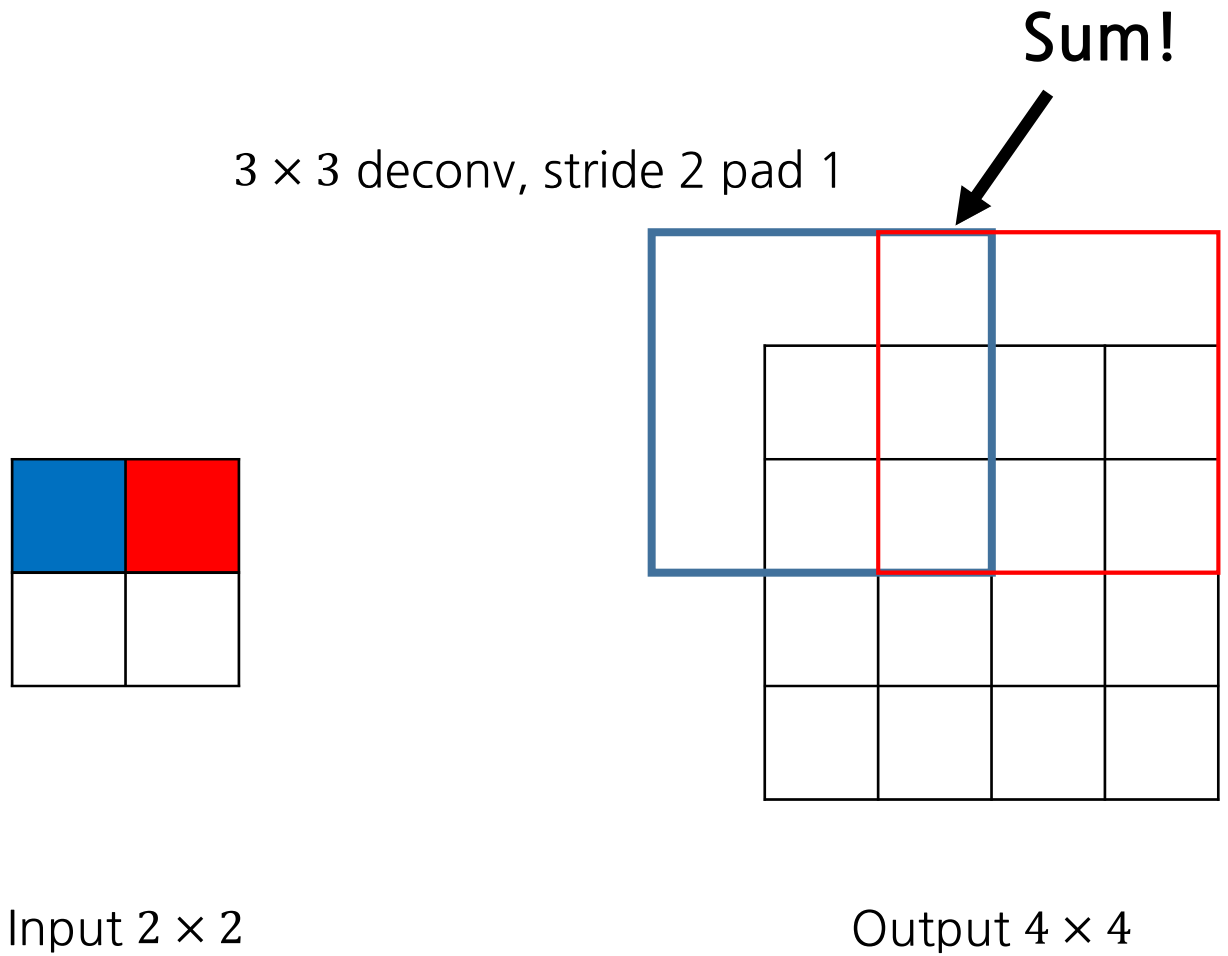
3×3 conv, stride 2 pad 1



Input 4×4

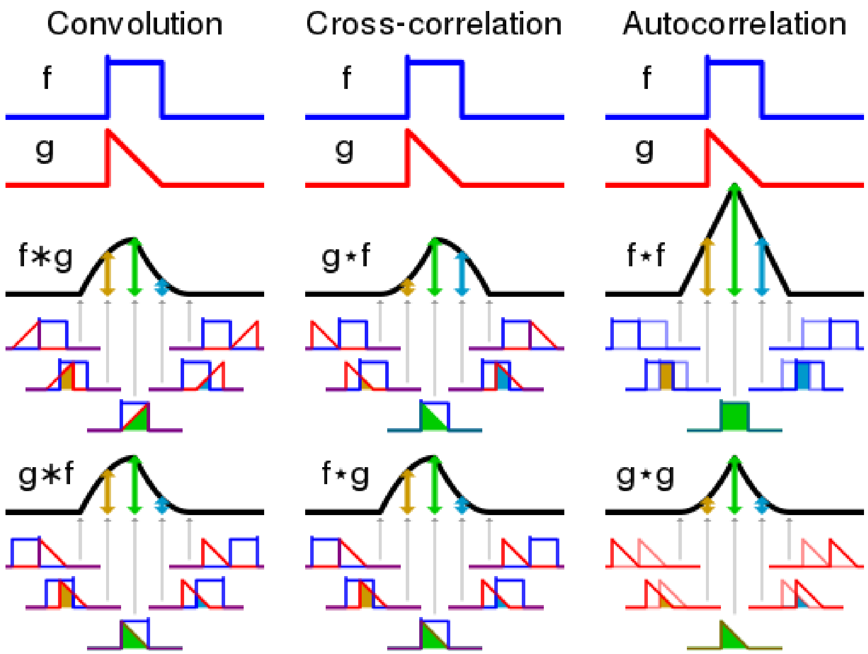
Output 2×2

Learnable Upsampling: “Deconvolution”



Learnable Upsampling: "Deconvolution"

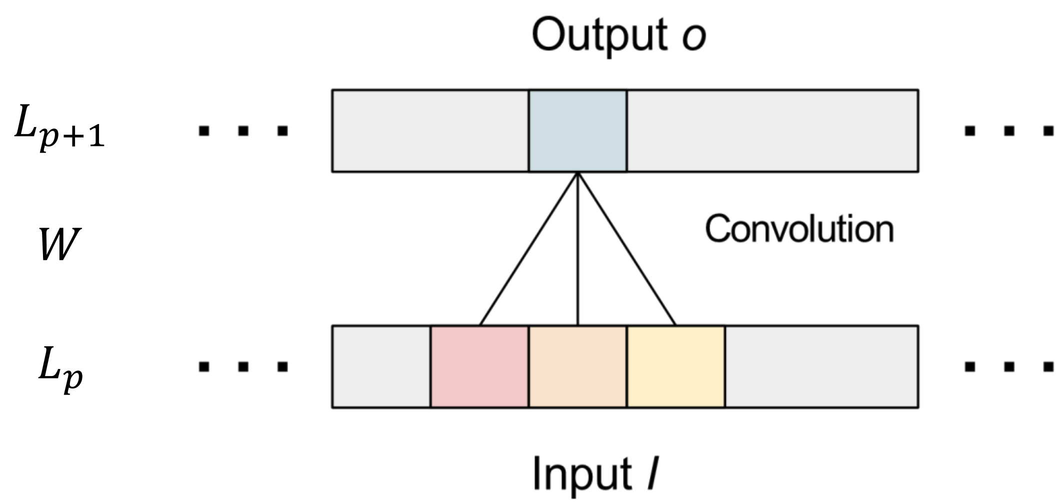
- Convolution and Cross-correlation
  - Convolution
    - $f * g(\tau) \coloneqq \int_{-\infty}^{\infty} f(t)g(\tau - t)dt$
  - Cross-correlation
    - $f \star g(\tau) \coloneqq \int_{-\infty}^{\infty} f(t)g(\tau + t)dt$



https://en.wikipedia.org/wiki/File:Comparison_convolution_correlation.svg

Learnable Upsampling: “Deconvolution”

- 관점에 따라 다르지만, 우리가 지금까지 배운 관점으로는 사실 Convolution이 아니라 Cross-correlation임
- 1-D Convolution



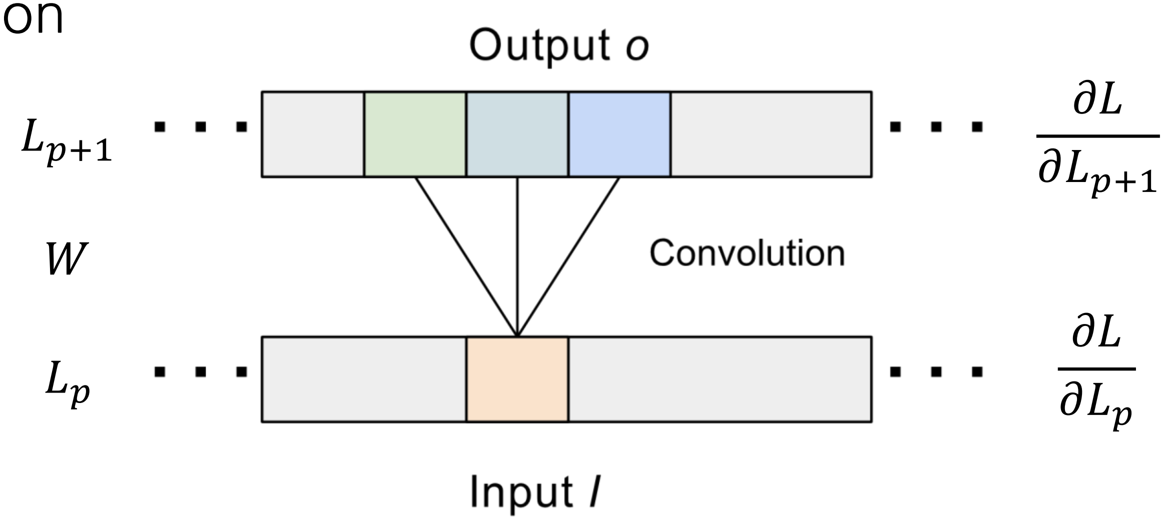
Im et al, “Generating images with recurrent adversarial networks”, arXiv 2016

$$f \star g(\tau) := \int_{-\infty}^{\infty} f(t)g(t+\tau)dt$$

$$L_{p+1}[i] = W \star L_p[i] = \sum_{j=1}^{3} W[j]L_p[j+i-1]$$

Learnable Upsampling: "Deconvolution"

- Gradient of 1-D Convolution



$$L_{p+1}[i] = W \star L_p[i] = \sum_{j=1}^{3} W[j] L_p[j+i-1]$$

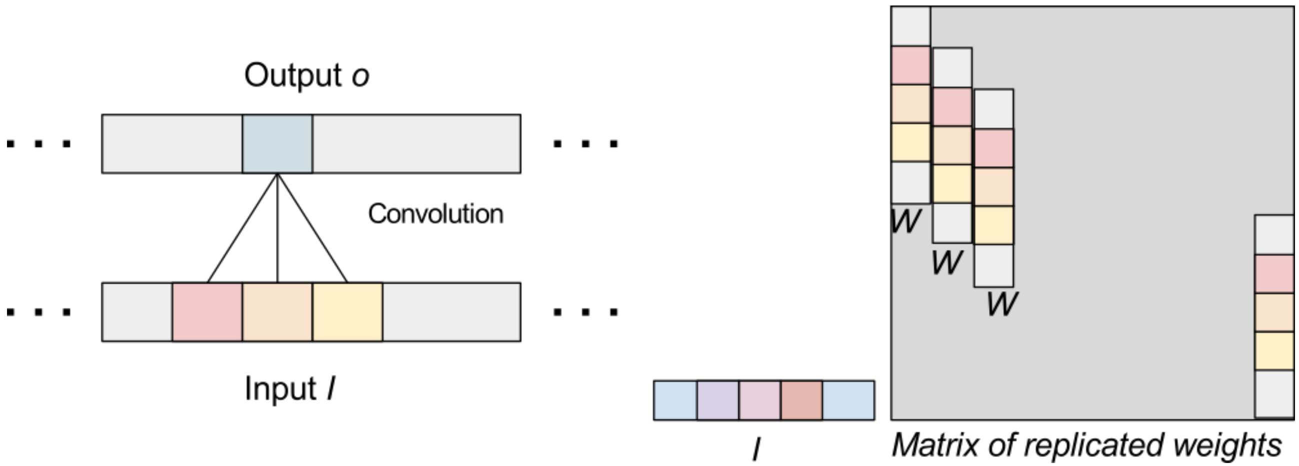
Im et al, "Generating images with recurrent adversarial networks", arXiv 2016

$$\begin{aligned}\frac{\partial L}{\partial W[j]} &= \frac{\partial L}{\partial L_{p+1}} \frac{\partial L_{p+1}}{\partial W[j]} \\ &= \sum_i \frac{\partial L}{\partial L_{p+1}[i]} L_p[j+i-1] \\ &= \left(L_p[i:i+2] \star \frac{\partial L}{\partial L_{p+1}[i]} \right)[j]\end{aligned}$$

$$\begin{aligned}\frac{\partial L}{\partial L_p[i]} &= \frac{\partial L}{\partial L_{p+1}} \frac{\partial L_{p+1}}{\partial L_p[i]} \\ &= \sum_j W[j] \frac{\partial L}{\partial L_{p+1}[i-j+1]} \\ &= \left(W * \frac{\partial L}{\partial L_{p+1}} \right)[i] = \left(W^T \star \frac{\partial L}{\partial L_{p+1}} \right)[i]\end{aligned}$$

Learnable Upsampling: "Deconvolution"

- 1-D Convolution, stride 1
  - $o = i * W$

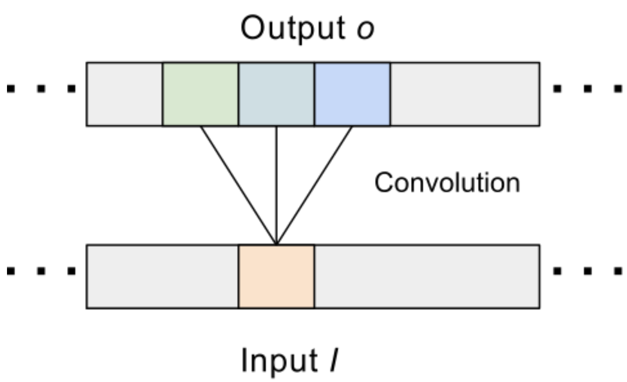


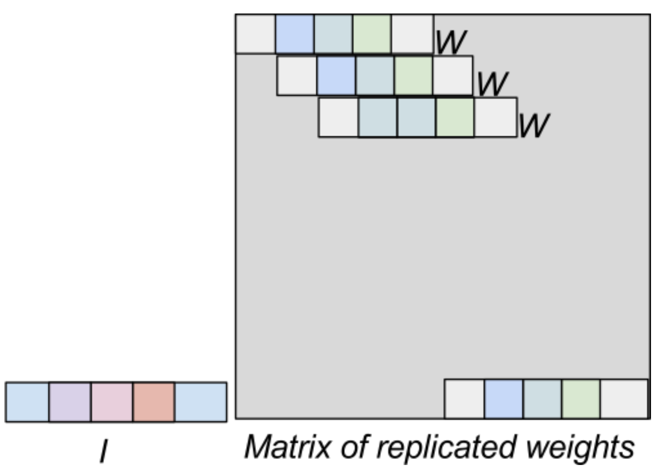
Im et al, "Generating images with recurrent adversarial networks", arXiv 2016

Learnable Upsampling: "Deconvolution"

- 1-D Convolutional transpose, stride 1
  - $\tilde{o} = \tilde{\imath} \star W$

Transposed!

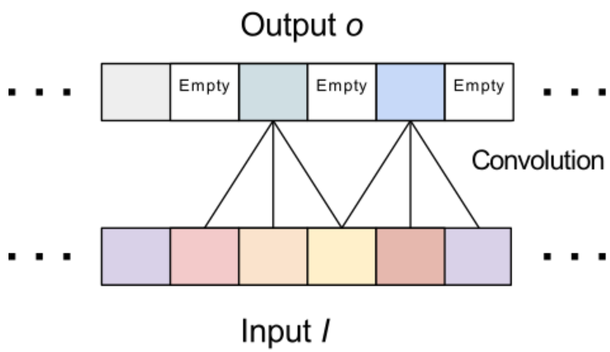


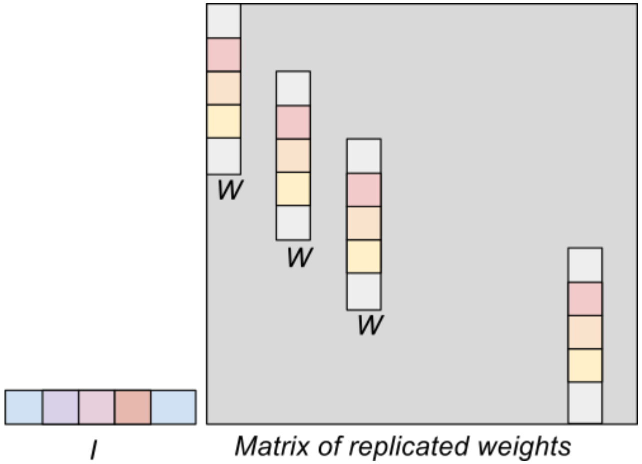


Im et al, "Generating images with recurrent adversarial networks", arXiv 2016

Learnable Upsampling: “Deconvolution”

- 1-D Convolution, stride 2
  - $o = i * W$



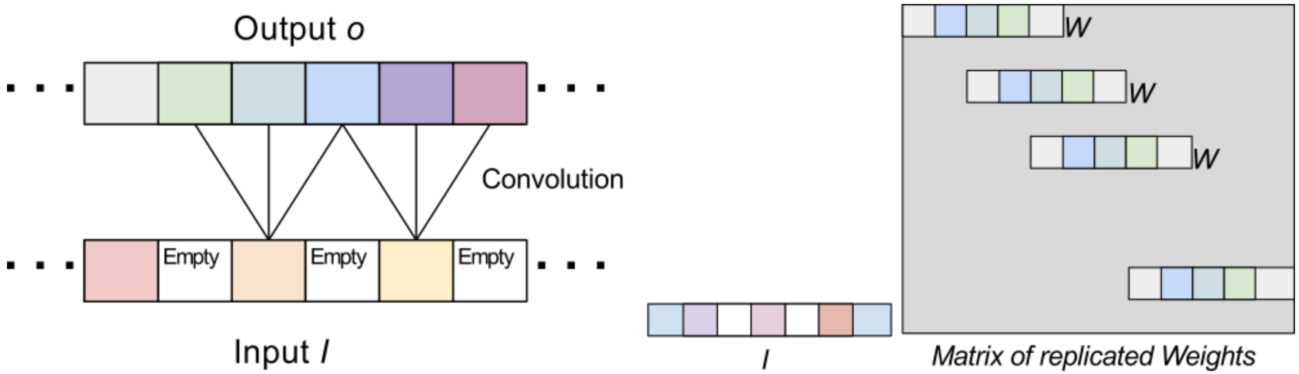


Im et al, “Generating images with recurrent adversarial networks”, arXiv 2016

Learnable Upsampling: "Deconvolution"

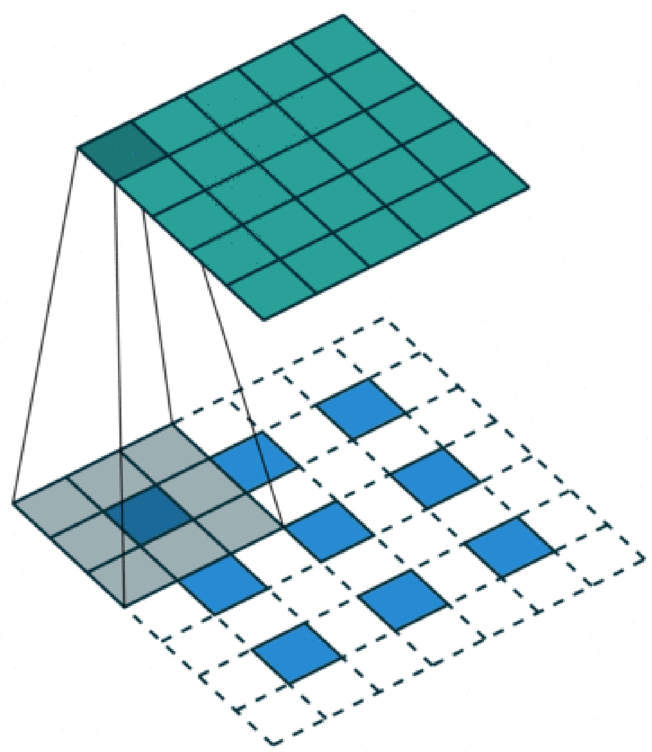
- 1-D Convolutional transpose, stride 1
  - $\tilde{o} = \tilde{\imath} \star W$

Transposed!



Im et al, "Generating images with recurrent adversarial networks", arXiv 2016

Learnable Upsampling: "Deconvolution"



https://github.com/vdumoulin/conv_arithmetic

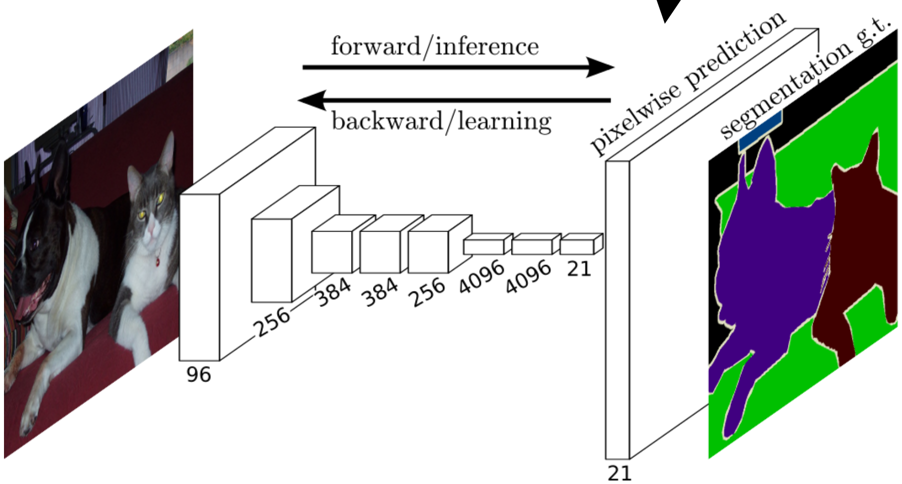
Fully Convolutional Networks for Semantic Segmentation

Fully Convolutional Networks for Semantic Segmentation

- Jonathan Long, Evan Shelhamer, Trevor Darrell
- Conference on Computer Vision and Pattern Recognition(CVPR), 2015

- Patch-wise segmentation은 비효율적
- Fully Convolutional Network(FCN)
  - 모든 layer를 Convolutional layer로 대체
  - Input size에 제한 없음

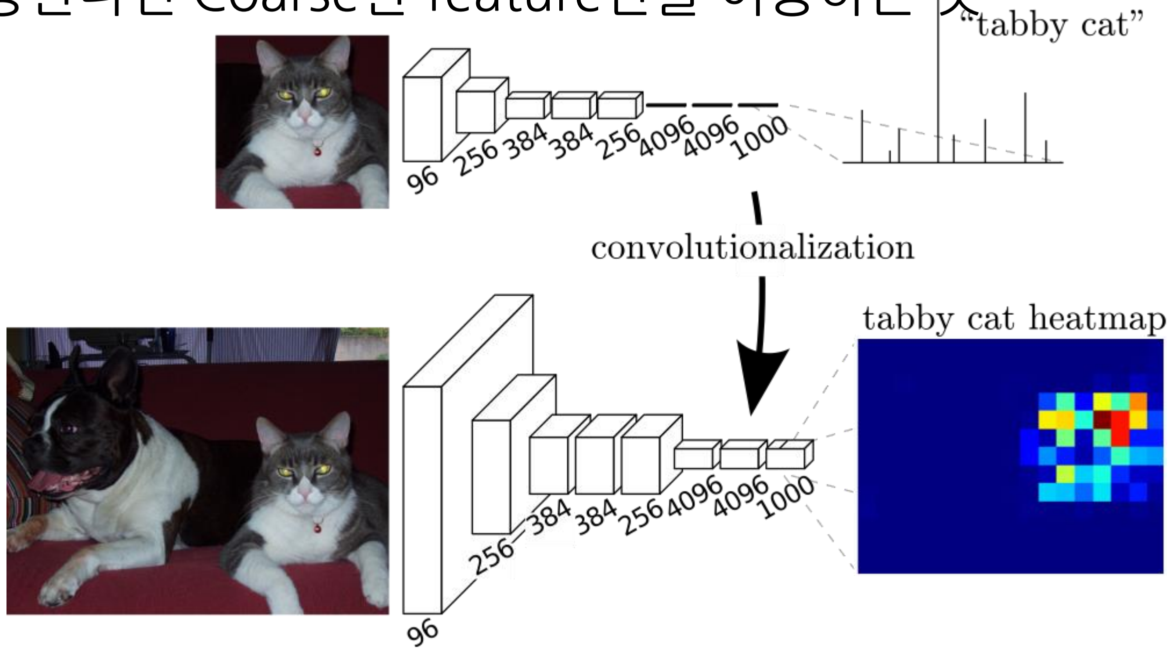
Lecture note와 다르게
이부분은 Learnable Upsample이 아님



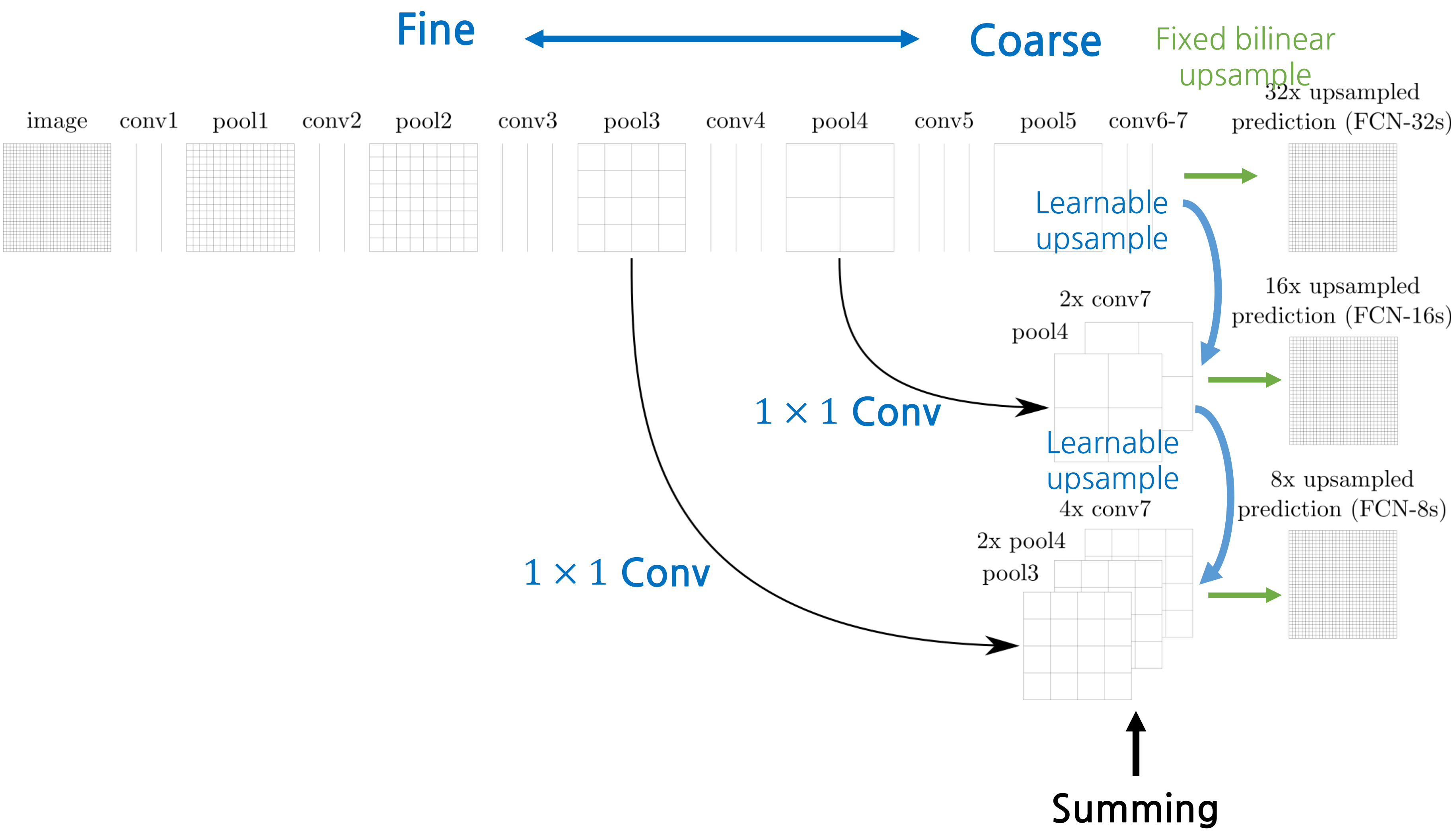
Fully Convolutional Networks for Semantic Segmentation

- AlexNet, VGG-16, GoogLeNet 사용
  - VGG-16으로 설명
- FCN을 이용하면 어느 해당하는 Object가 어디에 있는지 대략적으로 알 수 있음
- 이 결과를 Upsample하여 Segmentation
- 그런데 이 결과만을 이용한다면 Coarse한 feature만을 이용하는 것

  → **Skip structure**

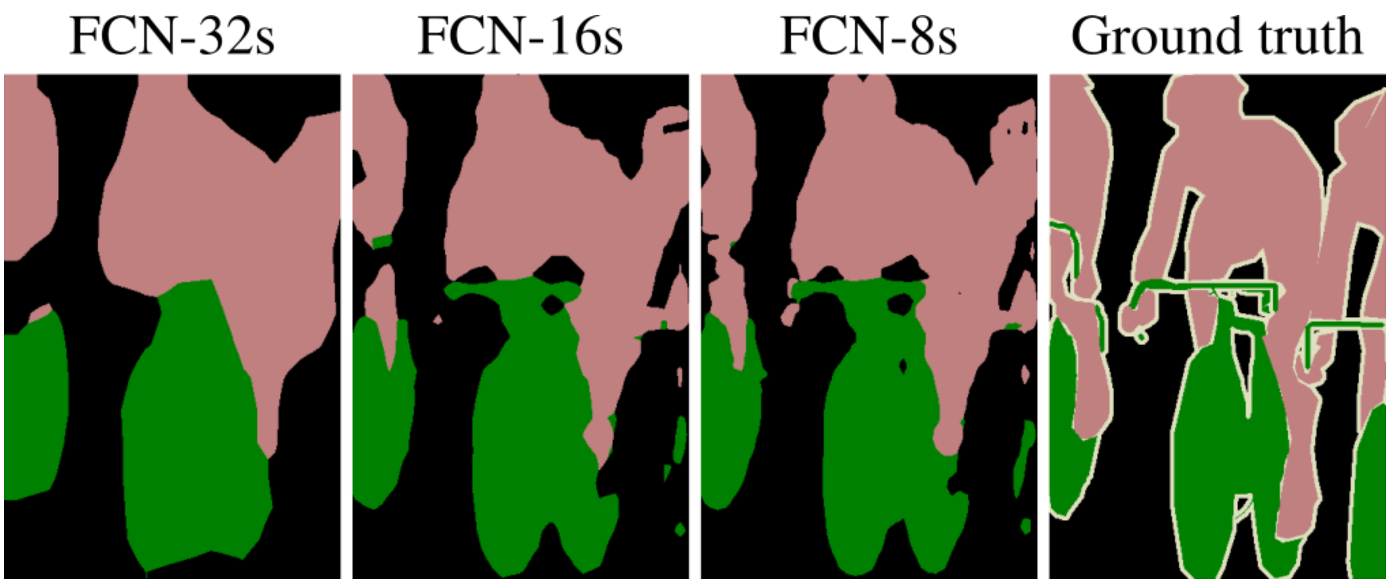


Fully Convolutional Networks for Semantic Segmentation



Fully Convolutional Networks for Semantic Segmentation

- Skip structure로 개략적으로 예측했던 영역이 세밀해짐
- Inference time: 175ms

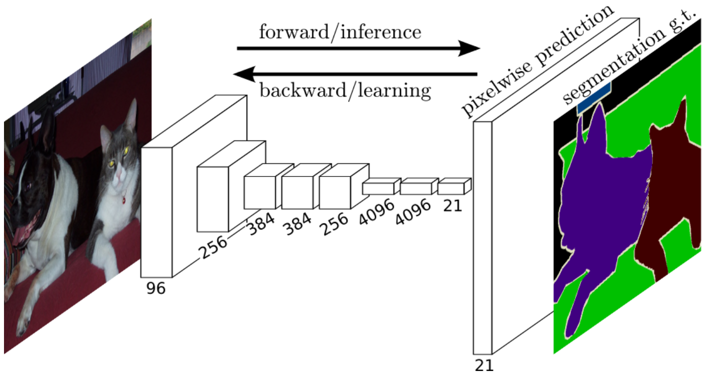


Learning Deconvolution Network for Semantic Segmentation

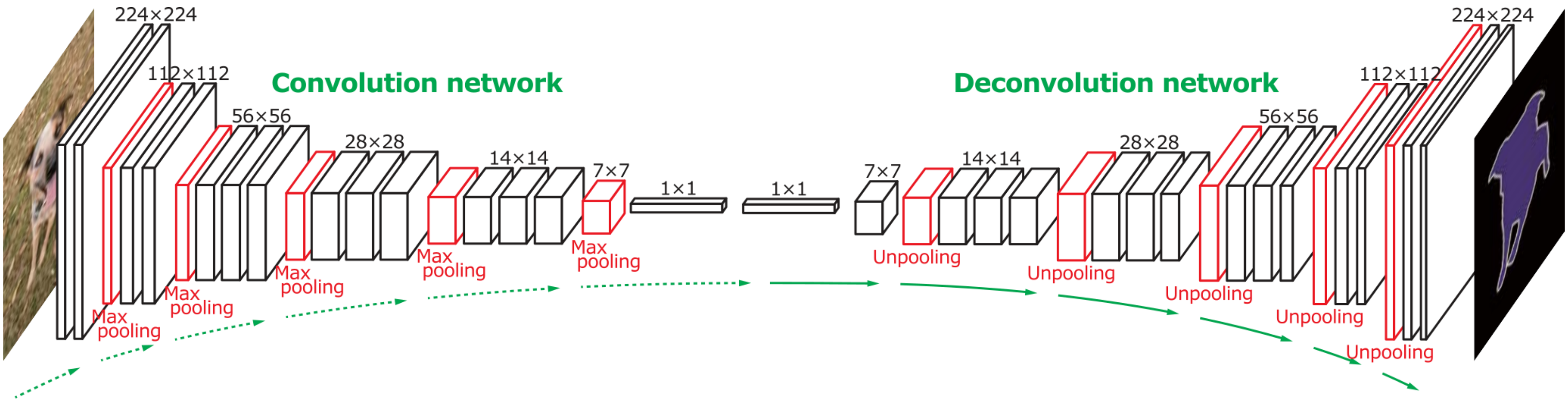
Learning Deconvolution Network for Semantic Segmentation

- Hyeonwoo Noh, Seunghoon Hong, Bohyung Han
- International Conference on Computer Vision(ICCV), 2015

- FCN의 한계 지적
  - 한가지 Scale만을 고려
    - Skip architecture은 근본적인 해결책이 아니며, 성능 향상에 한계가 있음
  - Deconvolution이 병목을 가짐



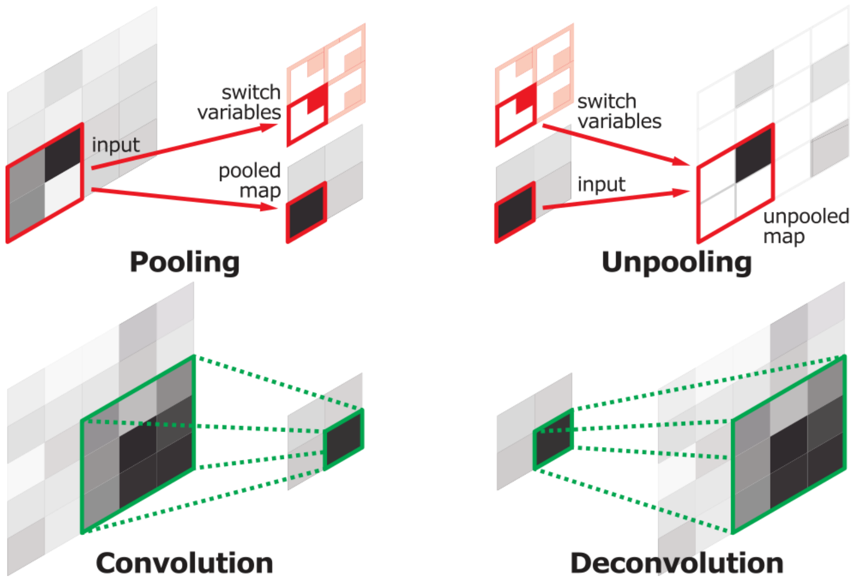
Learning Deconvolution Network for Semantic Segmentation



- Architecture
  - Feature extractor에 거울상에 해당하는 Deconvolutional Network 구성
  - ConvNet은 Feature Extractor, DeconvNet은 Shape Generator의 역할을 함
- 장점
  - Scale에 자유로움
  - Complexity가 낮음

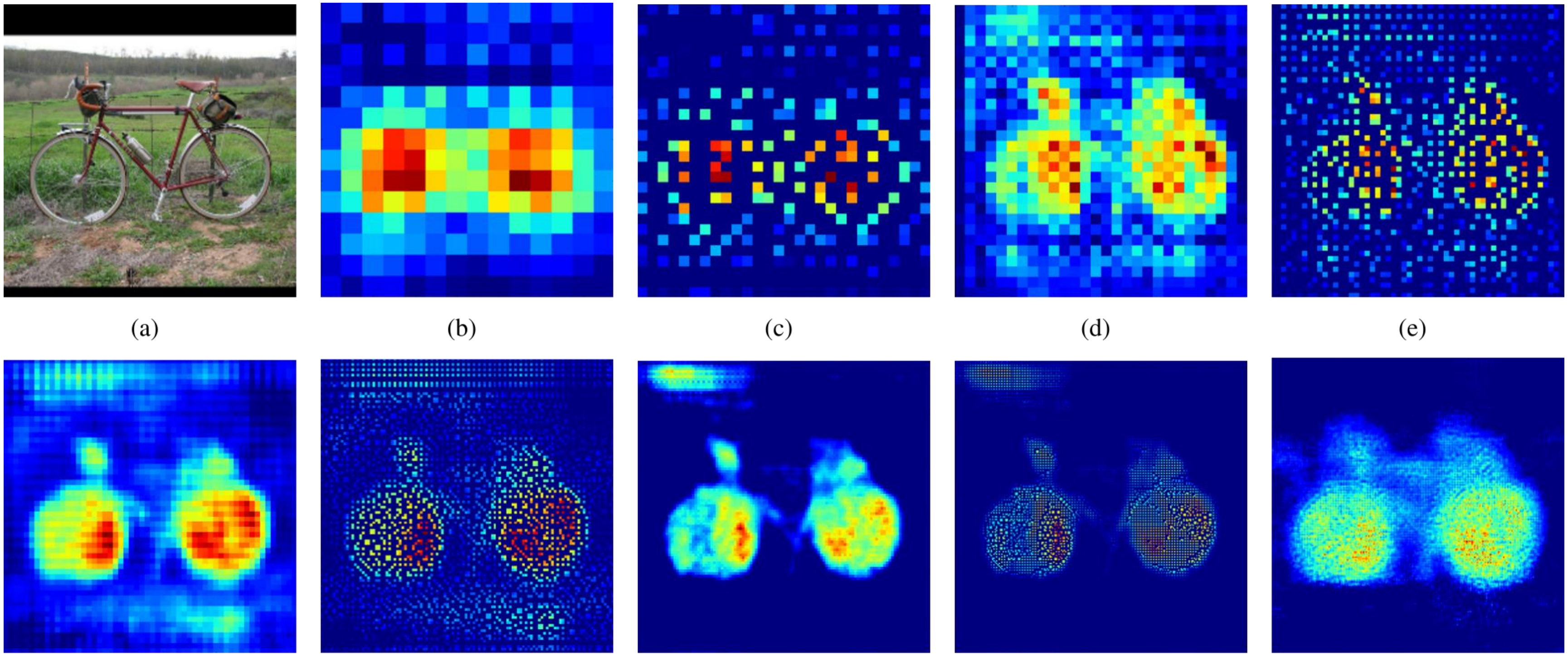
Learning Deconvolution Network for Semantic Segmentation

- Operations
  - Deconvolution: Convolutional transpose
  - Unpooling for max pooling
    - Max pooling의 ArgMax를 기억하여 해당 위치로 복원하며 나머지는 0으로 pad



Learning Deconvolution Network for Semantic Segmentation

- Deconvolutional Network로 Segmentation되는 과정



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Learning Deconvolution Network for Semantic Segmentation

- 사용한 Trick
  - Pre-trained VGG
  - BN
  - 2 stage train
    - Object가 가운데에 위치하여 Crop된 쉬운 데이터를 이용하여 먼저 학습
    - 이후 어려운 데이터 학습
  - Edge-box로 image를 proposal하여 logit을 summation 혹은 maximu으로 병합

- Variation
  - CRF로 Post-processing
  - Ensemble with FCN

Learning Deconvolution Network for Semantic Segmentation

- 의문점
  - FCN과 다르게 왜 224×224 의 고정된 size를 input으로 받을까?
  - 이 논문에서 제안한 모델은 미세한 feature를, FCN은 전체적인 feature를 파악하기에 Ensemble이 잘 작동한다는데, 앞에서 scale에 관계 없이 잘 작동하는 모델이라 서술한 것과 모순되지 않는지?

Instance Segmentation

Instance Segmentation

https://chaosmail.github.io/deeplearning/2016/10/22/intro-to-deep-learning-for-computer-vision/

- 다른 고양이는 다르다고 판단
- Object Detection과 유사
  - Region proposal를 사용

Instance Segmentation

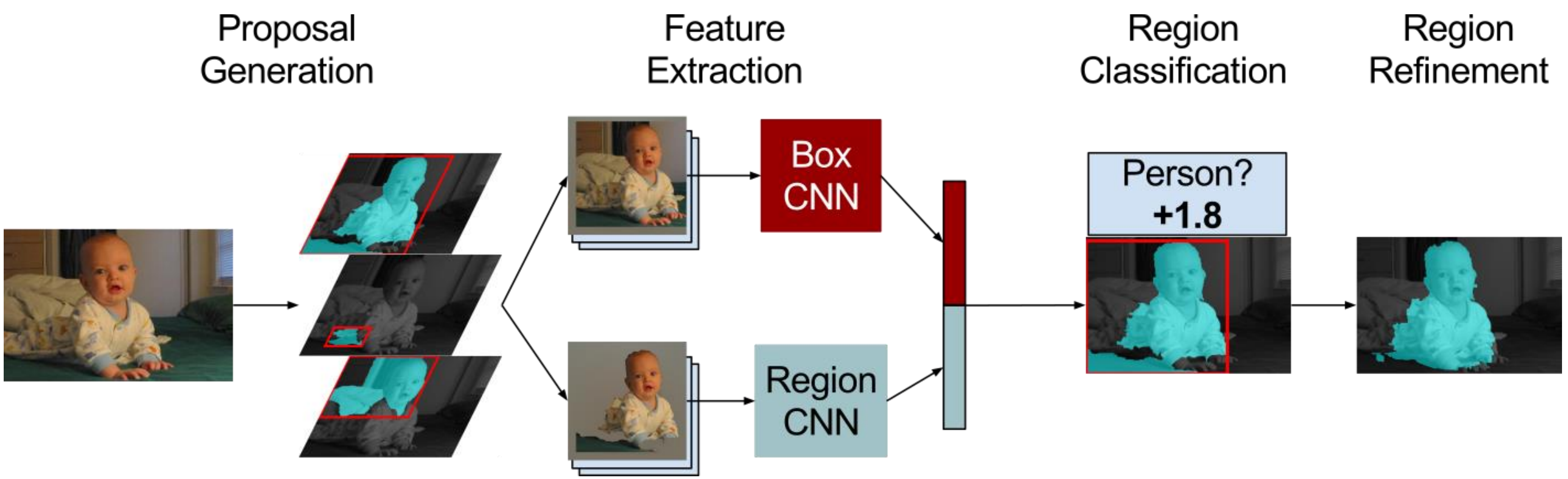
https://chaosmail.github.io/deeplearning/2016/10/22/intro-to-deep-learning-for-computer-vision/

- Sub-task
  - Region proposal
  - Region의 유효성 판단
  - Region 분류

Simultaneous Detection and Segmentation

Simultaneous Detection and Segmentation

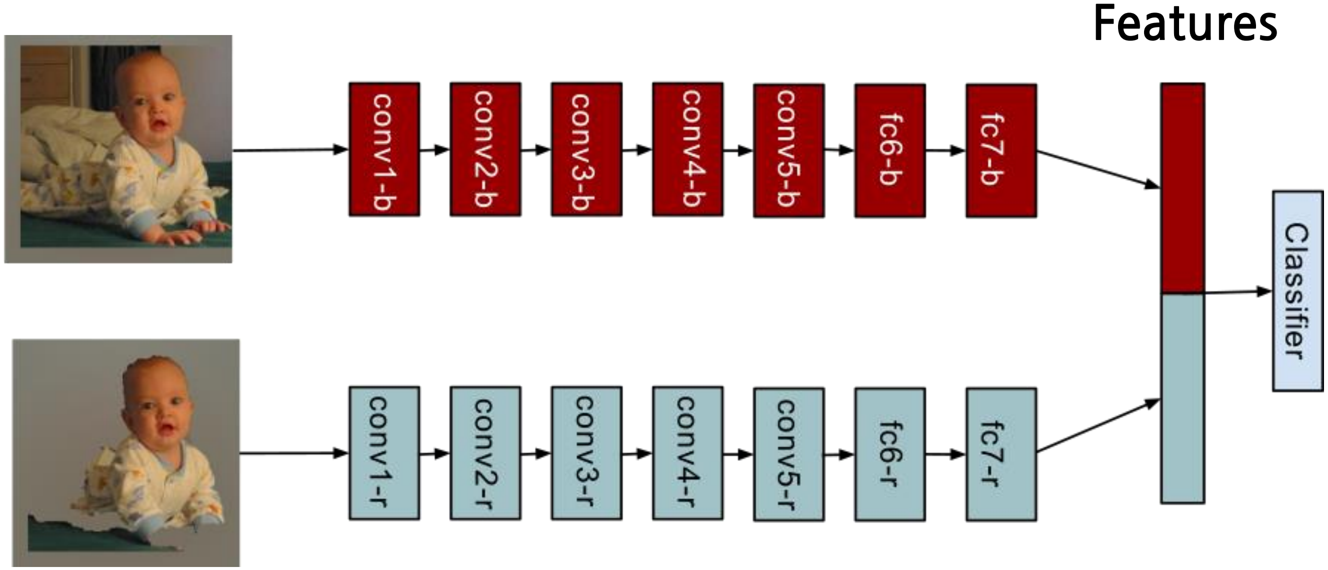
- Bharath Hariharan, Pablo Arbeláez, Ross Girshick, Jitendra Malik
- European Conference on Computer Vision(ECCV), 2014



Simultaneous Detection and Segmentation

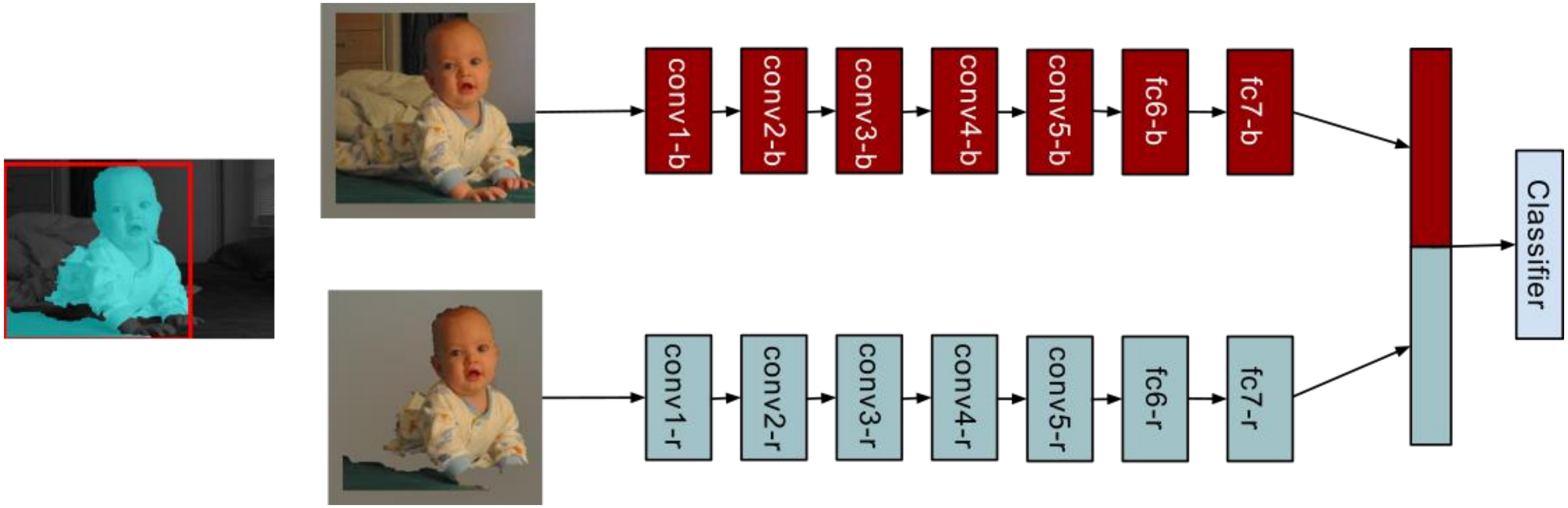
1. **Proposal generation**: Category-independent bottom-up object proposals using MCG
   - 2000 region candidates per image
   - Arbeláez et al, "Multiscale combinatorial grouping", CVPR 2014

2. **Feature extraction**: Extract features from both the bounding box of the region, the region foreground



Simultaneous Detection and Segmentation

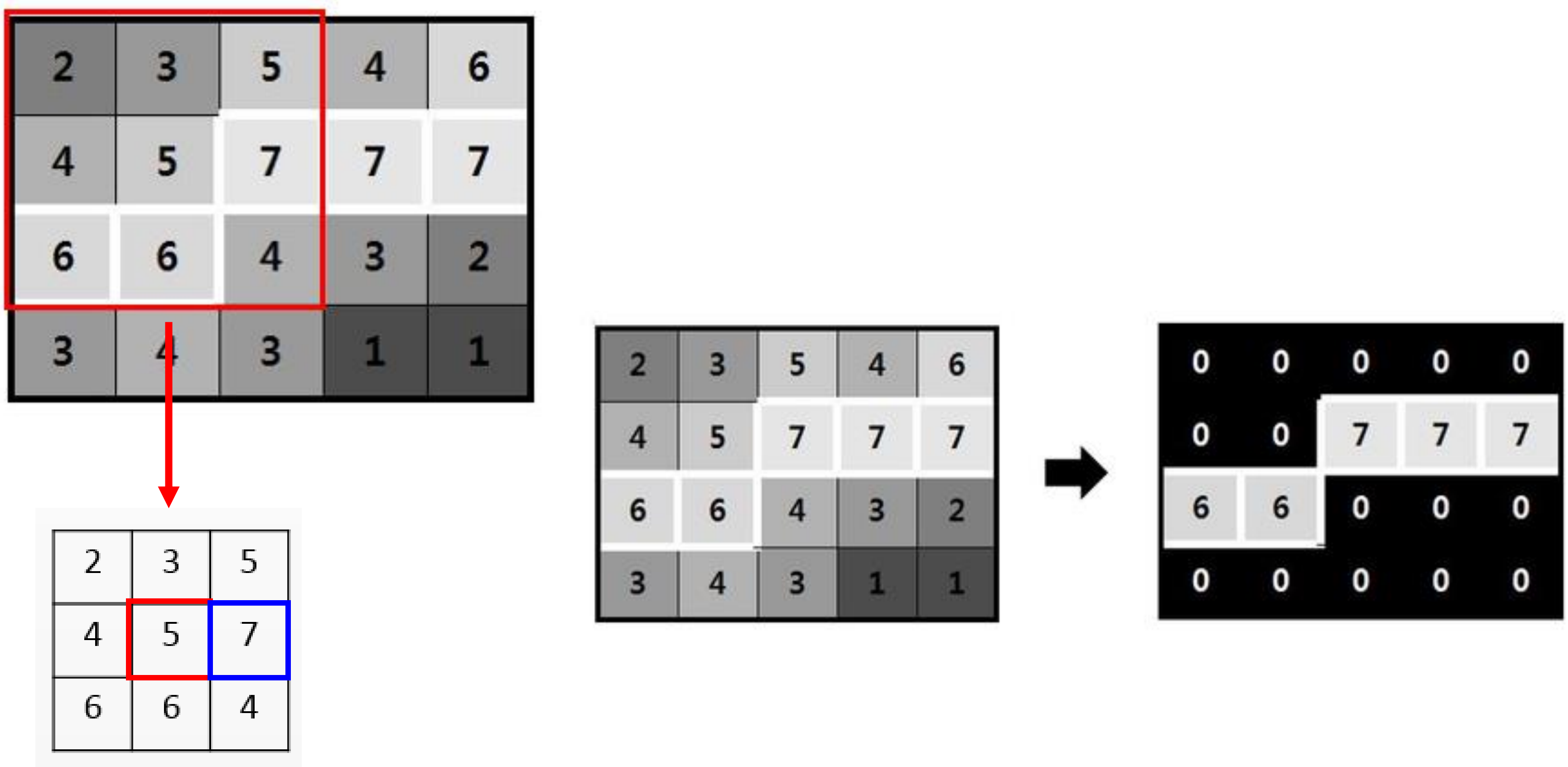
3. **Region classification**: Using the features, train SVM(Top-down)
   - Assign a score for each category(including background)



4. **Region refinement**: Non-maximum suppression(NMS) and refine with superpixels

Simultaneous Detection and Segmentation

- Non-maximum suppression(NMS)



http://blog.naver.com/jinsoo91zz/220511441402

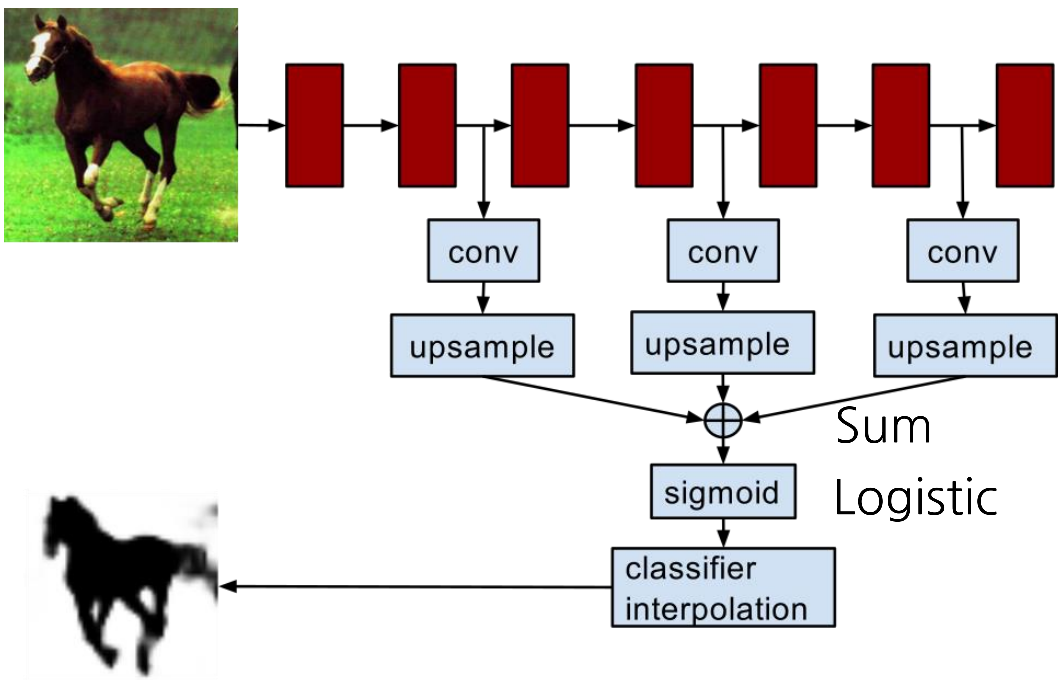
Simultaneous Detection and Segmentation

- Non-maximum suppression(NMS)
  - Canny edge detection에서 사용되는 알고리즘
    - Noise로 검출된 Edge를 제거
  - Object Detection, Instance Segmentation에서는 Window가 아닌 IoU를 이용하여 NMS를 적용할 범위를 결정

Hypercolumns for Object Segmentation and Fine-grained Localization

Hypercolumns

- Bharath Hariharan, Pablo Arbeláez, Ross Girshick
- Computer Vision and Pattern Recognition (CVPR), 2015



1×1 conv

Bilinear upsample

Sum

Logistic

Instance-aware Semantic Segmentation via Multi-task Network Cascades

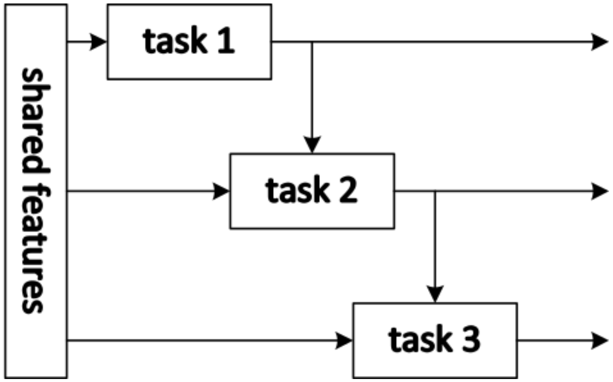
Multi-task Network Cascades

- Jifeng Dai, Kaiming He, Jian Sun
- Computer Vision and Pattern Recognition (CVPR), 2015

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun

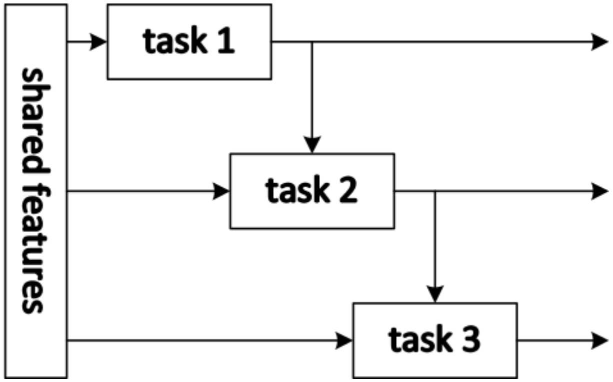
- Region proposal과 Segmentation을 통합
  - Multi-task Cascades(MNCs)
  - Region Proposal Network(RPN) 사용
  - End-to-end learning



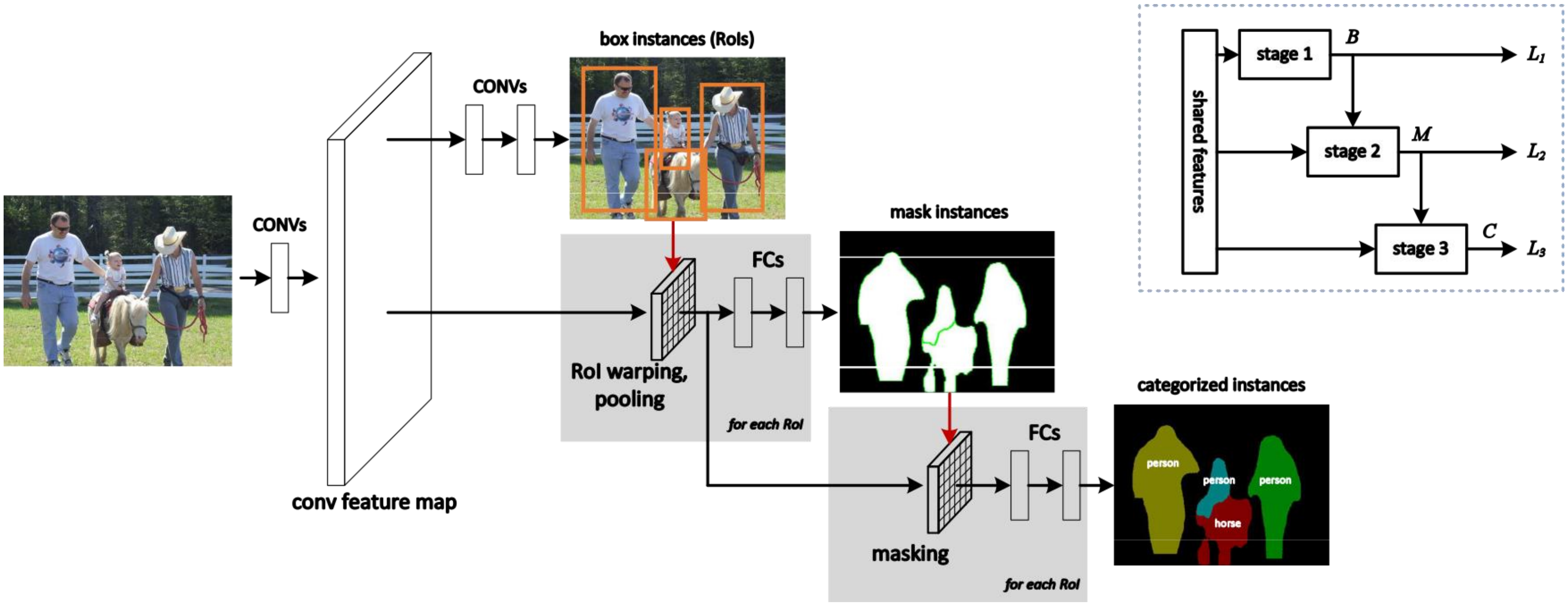
Multi-task Network Cascades

- Decomposition of Instance segmentation
  1. **Differentiating instances**: class agnostic bounding box
  2. **Estimating masks**: pixel-level mask for predicted each instances
  3. **Categorizing objects**: category-wise label is predicted for each mask-level instance

- 위 task들은 순서대로 이루어져야 함
  - Causal cascade(폭포)
  - Multi-task Network Cascades



Multi-task Network Cascades

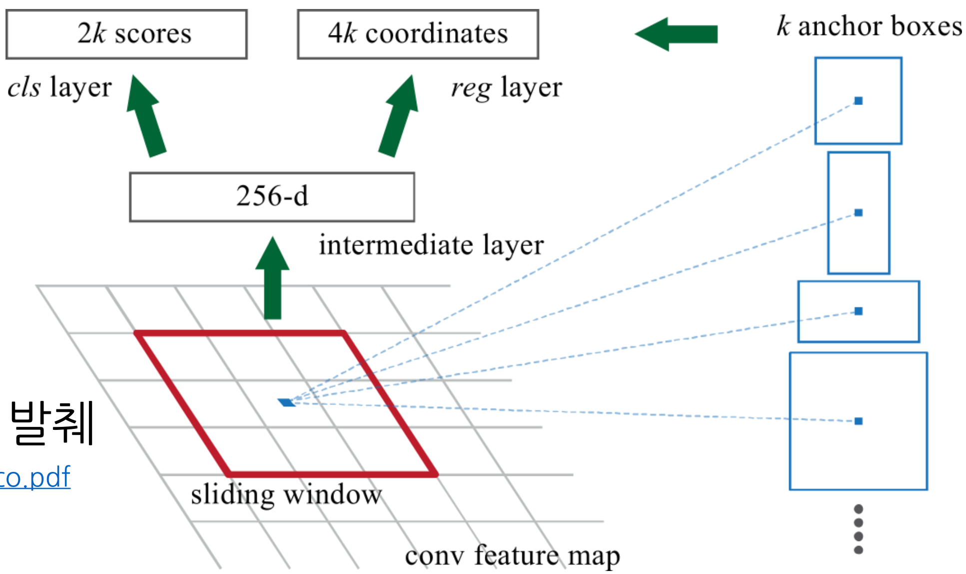


Multi-task Network Cascades

- Anchors in Faster R-CNN
  - k: Maximum possible proposals for each location
  - 한 pixel을 중심으로, scale과 ratio에 따라 anchor box 추출
  - cls layer: 이 box를 proposal의 여부를 판단
  - reg layer: 이 box 내부의 Bounding box를 regress

- Faster R-CNN
  - $k = 9$
- MCNs
  - $k = 12$
  - 논문에는 나오지 않고 발표자료에서 발췌

    http://image-net.org/challenges/talks/2016/ta-fcn_coco.pdf



Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks" *NIPS* 2015

Multi-task Network Cascades

- Regressing Box-level Instances(RPN)
  - Input: Extracted feature
  - $L_1 = L_1\big(B(\Theta)\big)$ where $B = \{B_i\}, B_i = \{x_i, y_i, w_i, h_i, p_i\}$
  - Non-maximum suppression(Threshold of IoU 0.7)

- Regressing Mask-level Instances
  - Input: RoI Pooling feature and RPN box
  - Output: $m \times m$ where $m = 28$
  - $L_2 = L_2(M(\Theta)|B(\Theta))$ where $M = \{M_i\}$, logistic regression
  - 구체적으로 나오지는 않지만, 여기서 유사하다는 DeepMask라는 논문을 보아 Bilinear upsampling으로 원본 image size로 복원하여 Loss를 계산한 것으로 추측

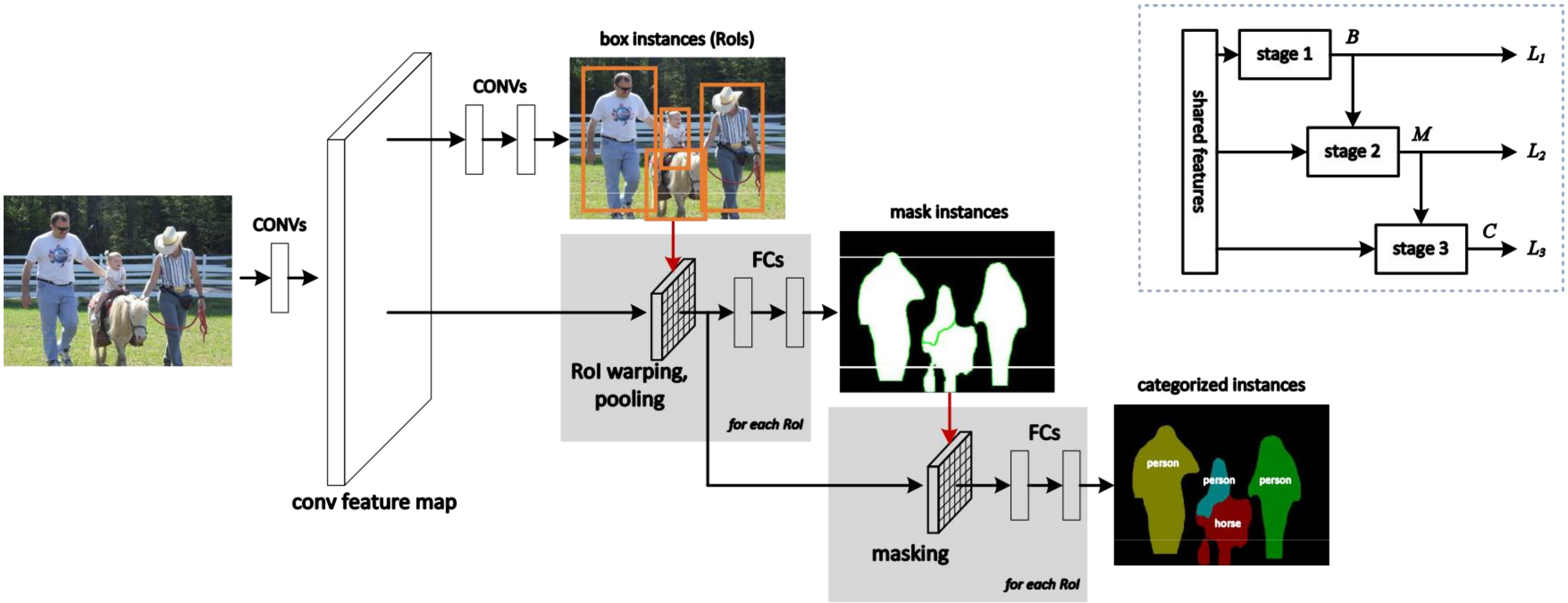
Multi-task Network Cascades

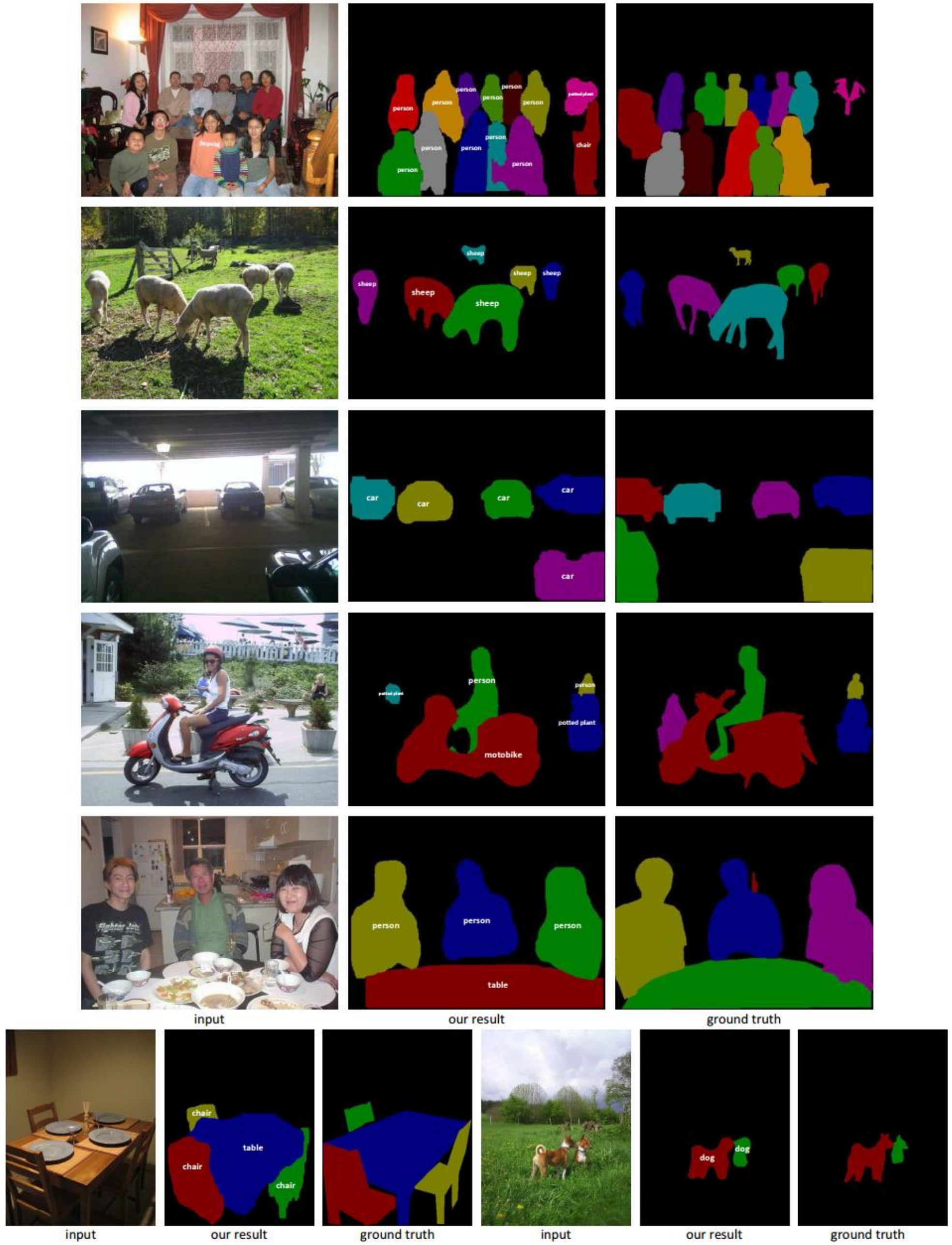
- Categorizing Instances
  - Input: RoI Pooling feature, RPN box and mask prediction
  - $\mathcal{F}_i^{Mask}(\Theta) = \mathcal{F}_i^{RoI}(\Theta) \cdot M_i(\Theta)$
  - For N categories, softmax classifier of $N+1$ classes including background
  - $L_3 = L_3(C(\Theta)|B(\Theta), M(\Theta))$

- End-to-End training
  - $L(\Theta)$=$L_1\big(B(\Theta)\big) + L_2(M(\Theta)|B(\Theta)) + L_3(C(\Theta)|B(\Theta), M(\Theta))$

- 360ms per image on an Nvidia K40

Multi-task Network Cascades




input
our result
ground truth
input
our result
ground truth
input
our result
ground truth

Problem Definition

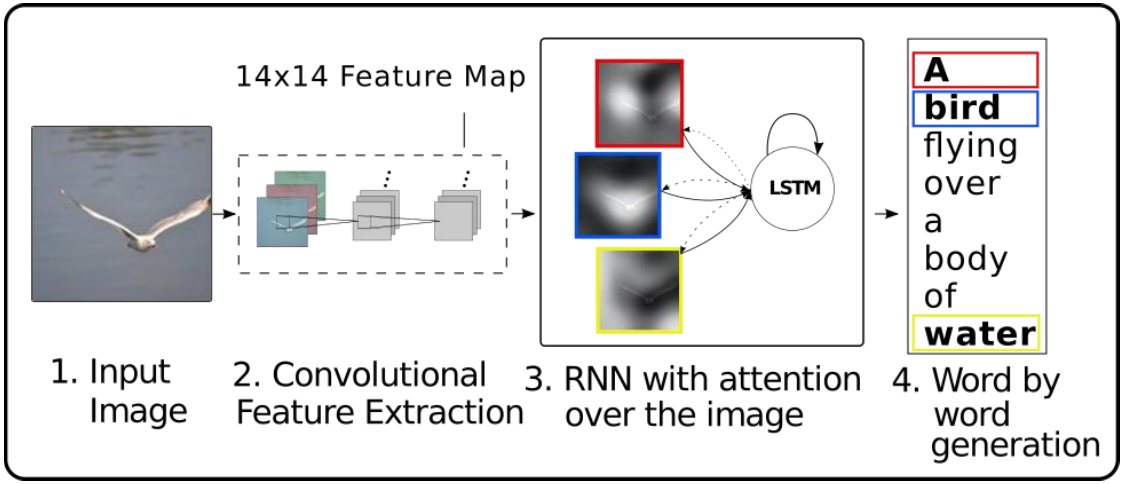
Segmentation

Attention

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

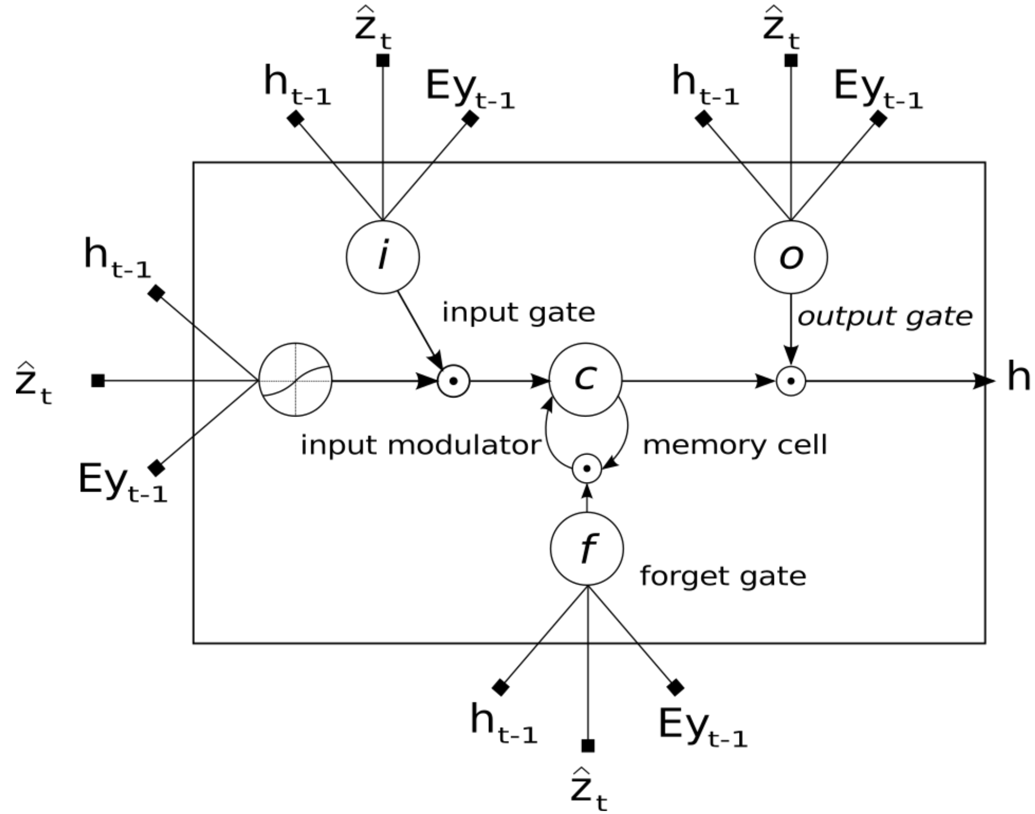
Show, Attend and Tell

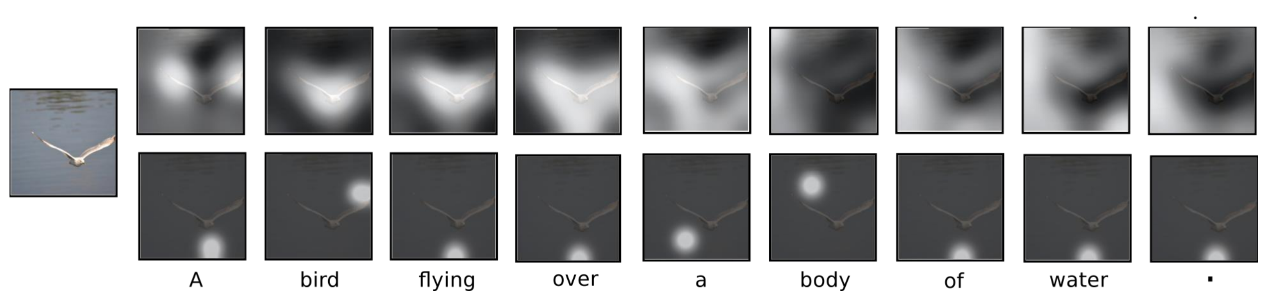
- Kelvin Xu, …, Kyunghyun Cho, …, Yoshua Bengio
- ICML, 2015
- 보여주고, 주시하고, 말하다.



Show, Attend and Tell

- $a = \{\mathbf{a}_1, \dots, \mathbf{a}_L\}, \mathbf{a}_i \in \mathbb{R}^D$
  - L: feature map의 크기
  - D: feature map의 차원
- $y = \{\mathbf{y}_1, \dots, \mathbf{y}_C\}, \mathbf{y}_i \in \mathbb{R}^K$
  - C: caption 단어의 수
  - K: 사전 단어의 수
- Attention model
  - $e_{ti} = f_{att}(\mathbf{a}_i, h_{t-1})$
  - $\alpha_{ti} = \dfrac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$
  - $\hat{\mathbf{z}}_t = \phi(\{\mathbf{a}_i\}, \{\alpha_i\})$





Show, Attend and Tell

- Hard attention
  - $s_{t,i}$: one-hot 변수로, i번째 구역을 주시하면 1, 아니면 0
  - $p\left(s_{t,i} = 1 \middle| s_{j<t}, \mathbf{a}\right) = \alpha_{t,i}$
  - $\hat{\mathbf{z}}_t = \sum_i s_{t,i} \mathbf{a}_i$
  - Loss

$$\begin{aligned} L_s &= \sum_s p(s|\mathbf{a}) \log p(\mathbf{y}|s,\mathbf{a}) \\ &\leq \log \sum_s p(s|\mathbf{a}) p(\mathbf{y}|s,\mathbf{a}) \\ &= \log p(\mathbf{y}|\mathbf{a}) \end{aligned}$$

  - Gradient

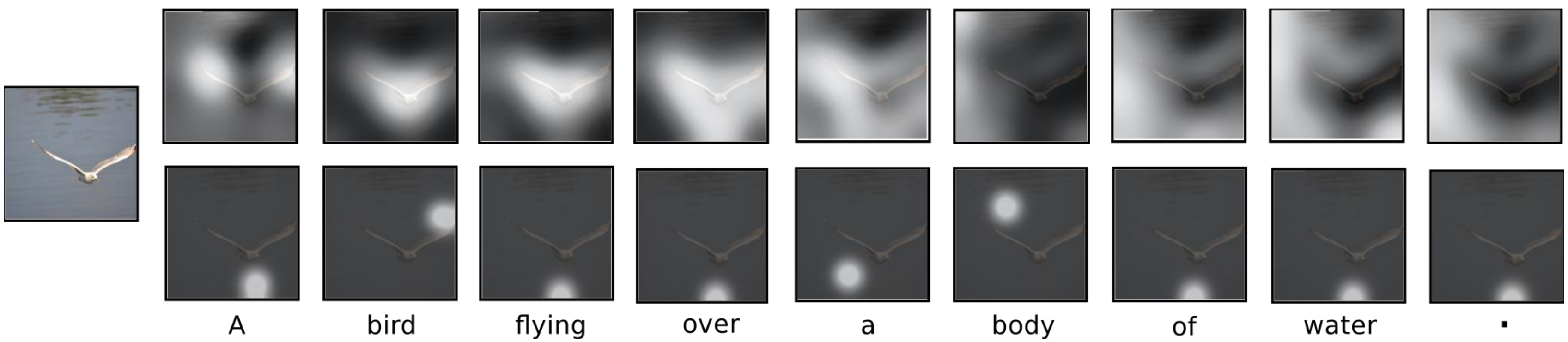
$$\frac{\partial L_s}{\partial W} = \sum_s p(s|\mathbf{a}) \left[\frac{\partial \log p(\mathbf{y}|s,\mathbf{a})}{\partial W} + \log p(\mathbf{y}|s,\mathbf{a}) \frac{\partial \log p(s|\mathbf{a})}{\partial W} \right]$$

    - 이 gradient를 Monte Carlo로 추정

Show, Attend and Tell

- Soft attention

$$\mathbb{E}_{p(s_t|\mathbf{a})}[\hat{\mathbf{z}}_t] = \sum_{i=1}^{L} \alpha_{t,i}\mathbf{a}_i$$



- Soft vs Hard
  - Soft attention은 gradient를 구하는 것이 어렵지 않음
  - Hard attention은 Gradient descent를 사용하기 어려우며, RL이 필요

Show, Attend and Tell

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

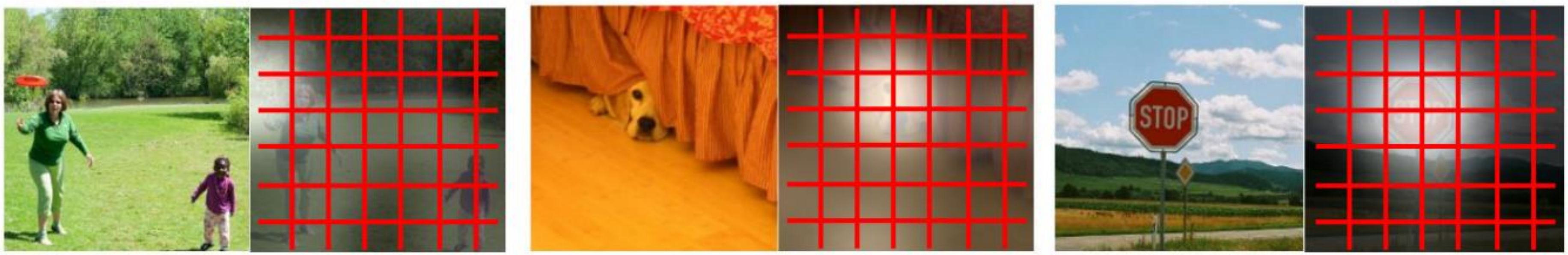
A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Show, Attend and Tell

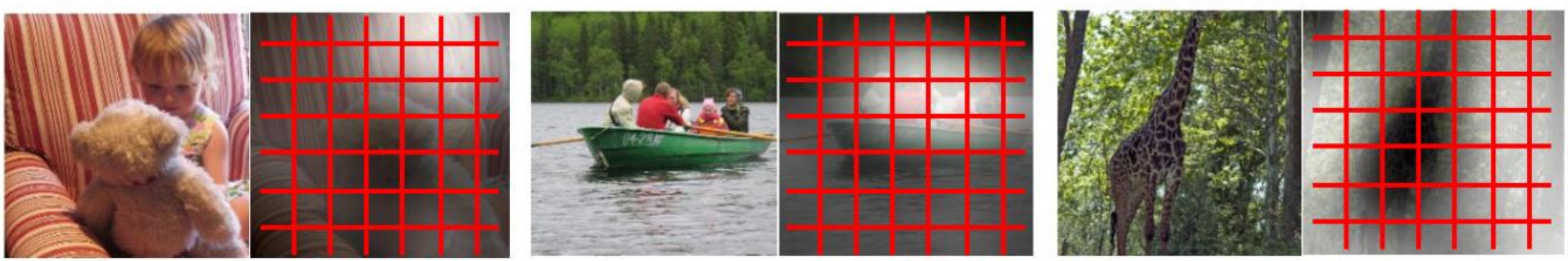
- Attention이 Grid에만 가능



A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.

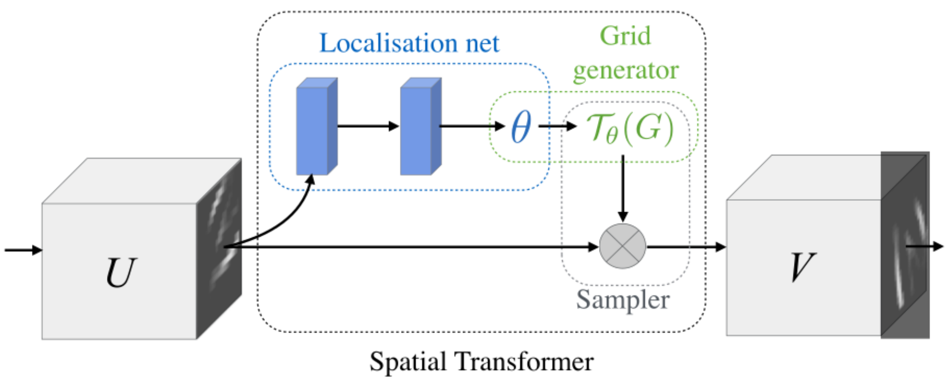
A group of people sitting on a boat in the water.

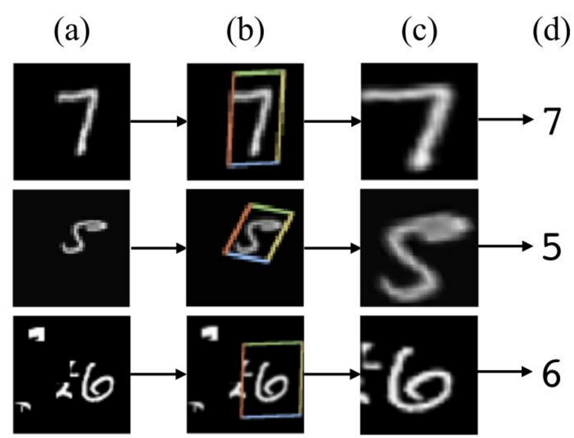
A giraffe standing in a forest with trees in the background.

Spatial Transformer Networks

Spatial Transformer Networks

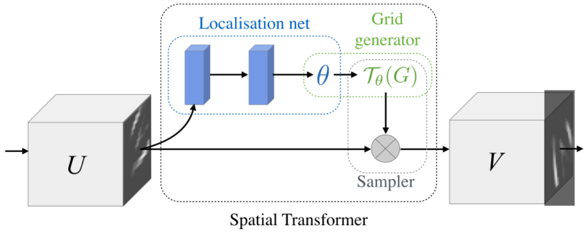
- Max Jaderberg, Karen Simonyan, Andrew Zisserman, Koray Kavukcuoglu
    - DeepMind
- NIPS 2015
- Task에 적절한 Spatial transformation을 학습





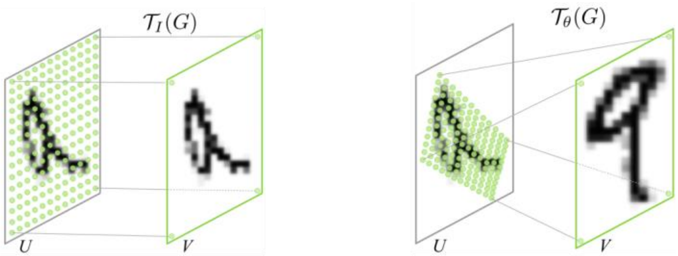
Spatial Transformer Networks

- 구성
  - Localization Network



  - Parmeterized Sampling Grid

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = \mathtt{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$



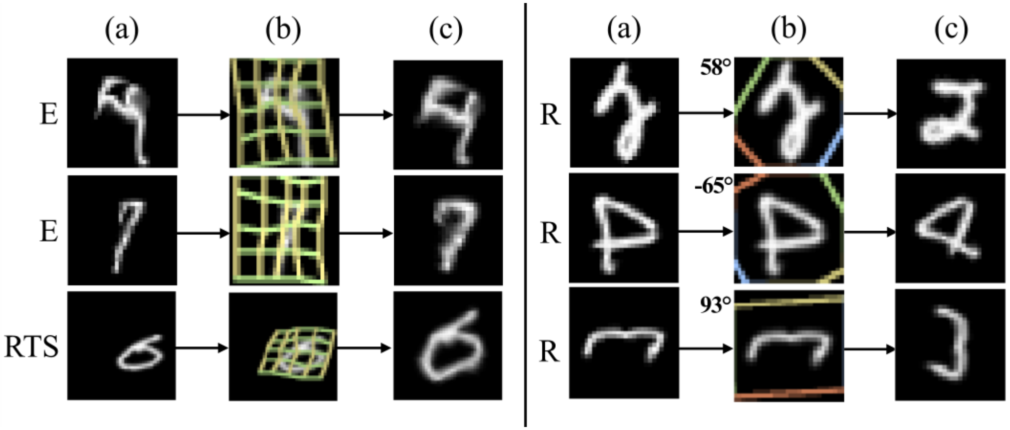
(a) (b)

  - Differentiable Image Sampling

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - n; \Phi_y) \;\; \forall i \in [1 \ldots H'W'] \;\; \forall c \in [1 \ldots C]$$

Spatial Transformer Networks

Model		MNIST Distortion R	RTS	P	E
FCN		2.1	5.2	3.1	3.2
CNN		1.2	0.8	1.5	1.4
ST-FCN	Aff	1.2	0.8	1.5	2.7
	Proj	1.3	0.9	1.4	2.6
	TPS	1.1	0.8	1.4	2.4
ST-CNN	Aff	0.7	0.5	0.8	1.2
	Proj	0.8	0.6	0.8	1.3
	TPS	0.7	0.5	0.8	1.1



Q & A