DenseCap

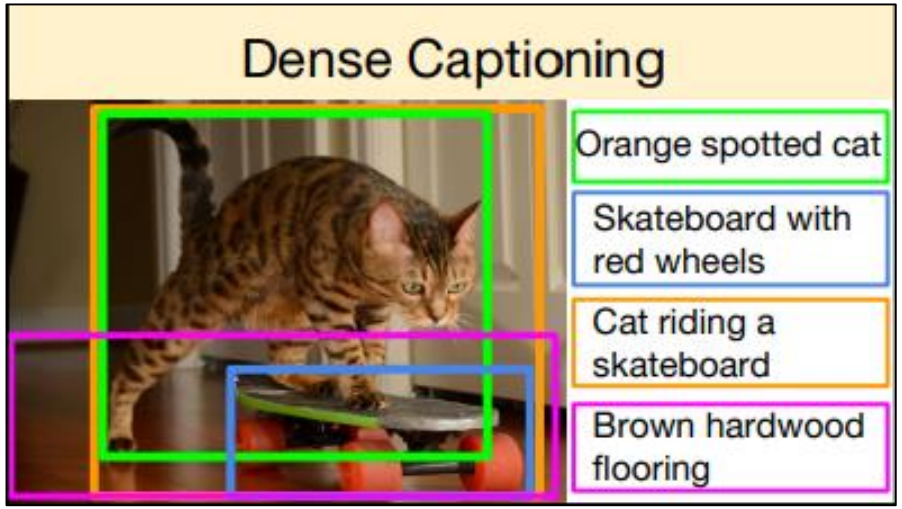
: FCLN (Fully Convolutional Localization Networks) for Dense Captioning

2019-03-16 | 이규희

DenseCap = (1) Localize + (2) Describe salient regions

FCLN (Fully Convolutional Localization Network)

. 별도의 regions proposal 없이 end-to-end로 최적화까지.



Related Work

[Object Detection] RPN (Region Proposal Network)

→ our approach does not require training pipelines.

→ Replace ROI pooling with a **differentiable**, spatial soft attention mechanism

[Image Captioning] Soft attention mechanism

→ Can process arbitrary **affine regions** instead of only discrete grid positions

Model

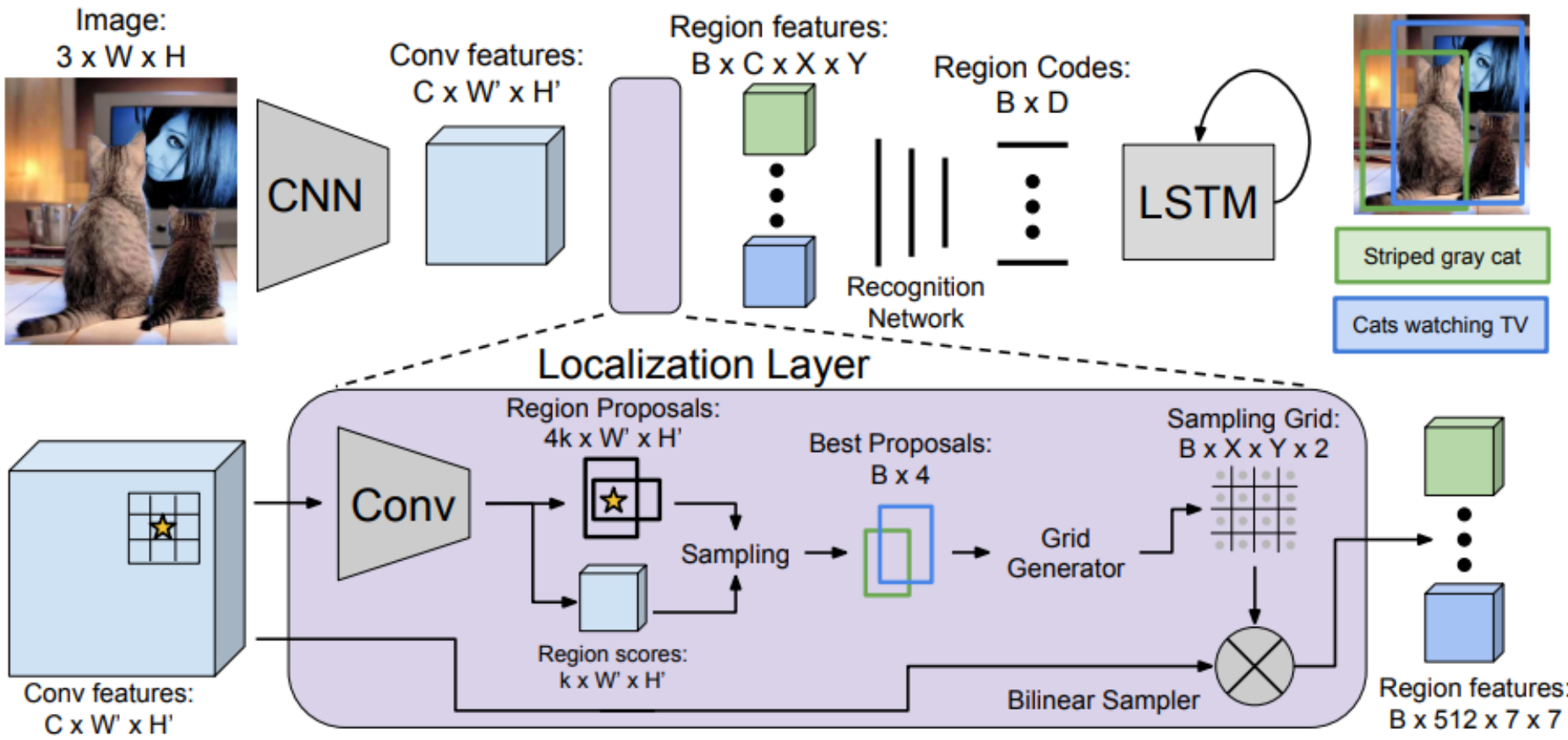
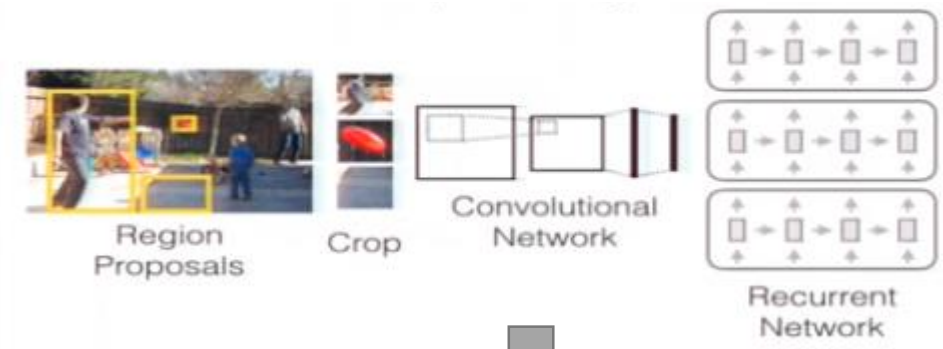


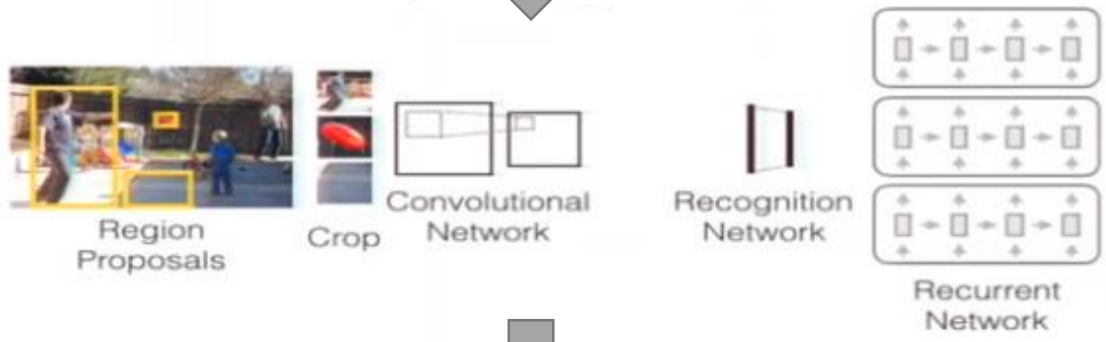
Image:
3 x W x H
CNN
Conv features:
C x W' x H'
Region features:
B x C x X x Y
Region Codes:
B x D
Recognition Network
LSTM
Striped gray cat
Cats watching TV
Localization Layer
Conv features:
C x W' x H'
Conv
Region Proposals:
4k x W' x H'
Region scores:
k x W' x H'
Sampling
Best Proposals:
B x 4
Grid Generator
Sampling Grid:
B x X x Y x 2
Bilinear Sampler
Region features:
B x 512 x 7 x 7

(Dense Captioning Prior Work)

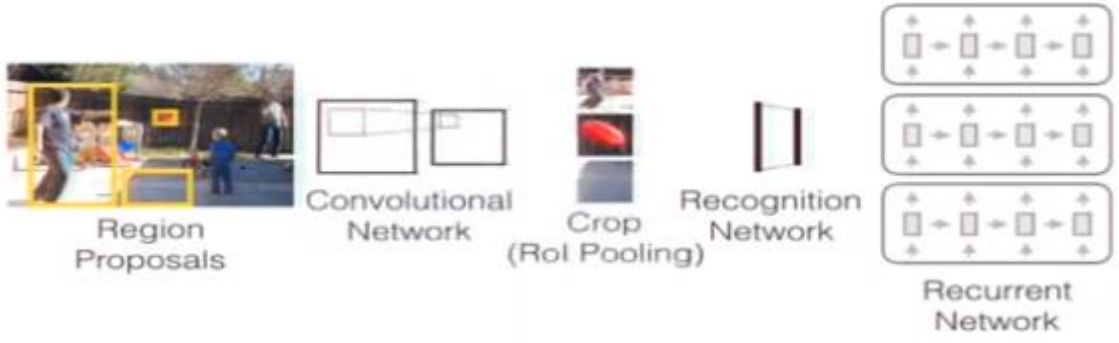
출처: ComputerVisionFoundation Videos (https://youtu.be/2wRnmRSrgCo)



Fast R-CNN처럼 FC layer를 분리함.

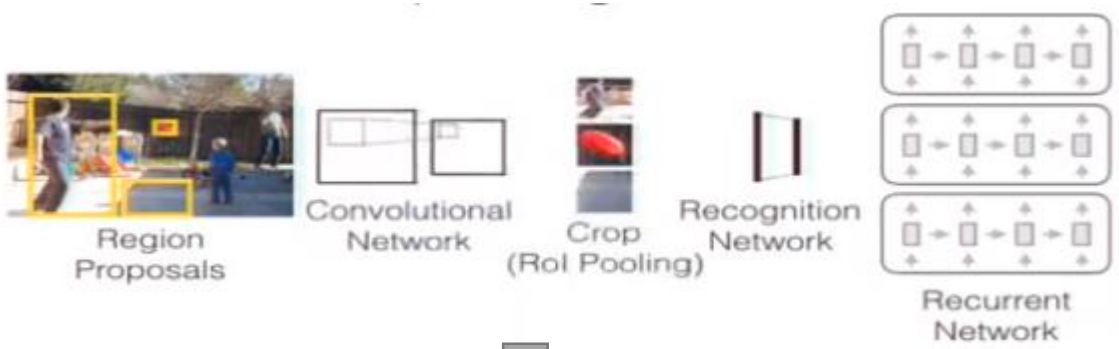


Crop과 Conv의 순서를 바꿈.

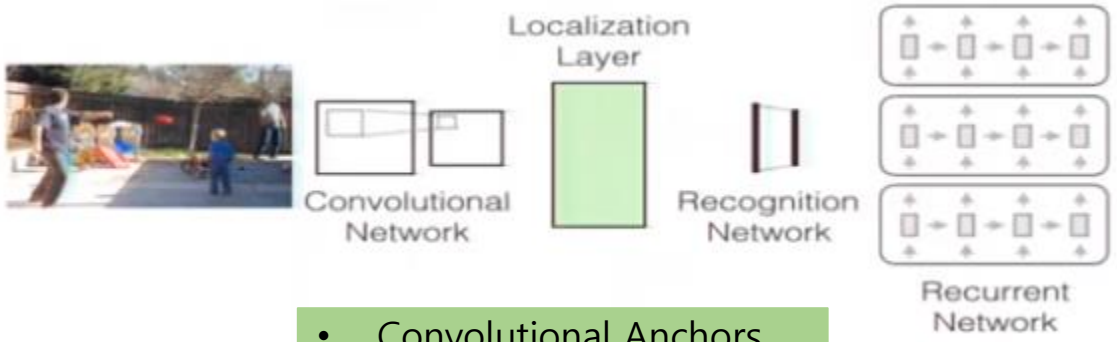


더 고해상도의 이미지를 처리할 수 있고,
전체 이미지에 대한 feature map을 만들 수 있다.

VGG16의 마지막 pooling layer를 제거한,
C(512) X W'(1/16) X H'(1/16).



Localization Layer
(No external region proposal)



- Convolutional Anchors
- Box Regression
- Box Sampling
- Bilinear Interpolation

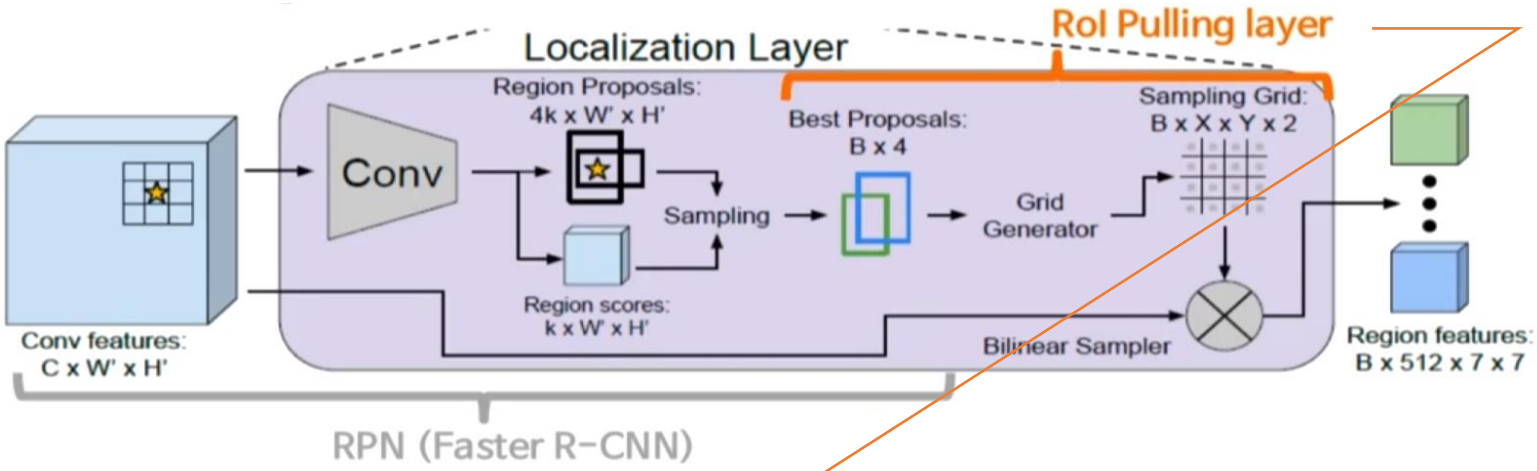
*** Bilinear Interpolation**

- Box sampling의 결과로 크기와 비율이 다른 region proposal을 얻게 되는데, Recognition network에 들어가기 위해서는 fixed-size feature가 필요함.

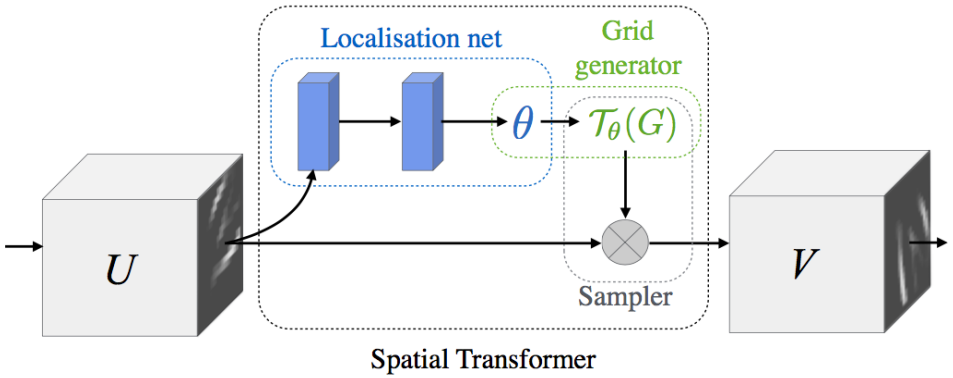
- Fast R-CNN에서는 RoI pooling layer를 썼었음.

- 그런데 RoI pooling은 gradients가 input proposal 좌표까지 backprop이 안 된다는 한계가 있으므로, B.I로 대체함.

Model



[비교] Spatial Transformer Networks



제일 처음, 그 자체로 작은 neural network인 **Localisation Network**은 input feature map U에 적용할 transform의 parameter matrix θ를 추정합니다.

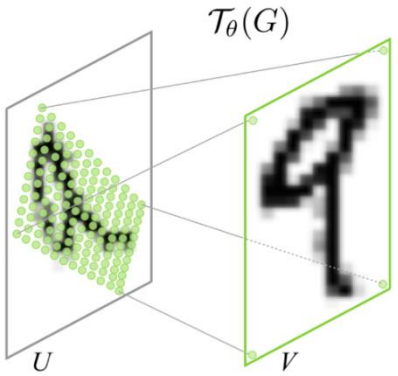
그 다음, **Grid Generator**는 추정한 θ에 따라 input feature map에서 sampling할 지점의 위치를 정해주는 sampling grid $\mathcal{T}_\theta(G)$를 계산합니다.

마지막으로, **Sampler**는 sampling grid $\mathcal{T}_\theta(G)$를 input feature map U에 적용해 변환된 output feature map V를 만듭니다.

https://jamiekang.github.io/2017/05/27/spatial-transformer-networks/

Grid Generator

Grid generator는 추정한 θ에 따라 input feature map에서 sampling할 지점의 위치를 정해주는 sampling grid $\mathcal{T}_\theta(G)$를 계산합니다.



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

Affine transform은 6개의 parameter로 scale, rotation, translation, skew, cropping을 표현할 수 있습니다.

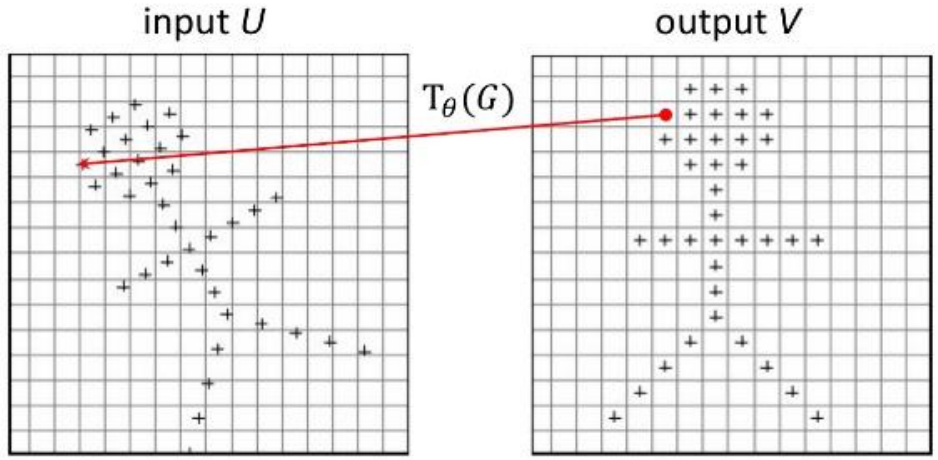
→ Differentiable image sampling!

https://jamiekang.github.io/2017/05/27/spatial-transformer-networks/

Sampler

Sampler는 sampling grid $\mathcal{T}_\theta(G)$를 input feature map U에 적용해 변환된 output feature map V를 만듭니다.

출력 V에서 특정한 pixel 값을 얻기 위해, 입력 U의 어느 위치에서 값을 가져올 지를 sampling grid $\mathcal{T}_\theta(G)$가 가지고 있습니다.



위의 그림에서 보는 것처럼 그 위치가 정확히 정수 좌표 값을 가지지 않을 가능성이 더 높기 때문에, 주변 값들의 interpolation을 통해 값을 계산합니다.

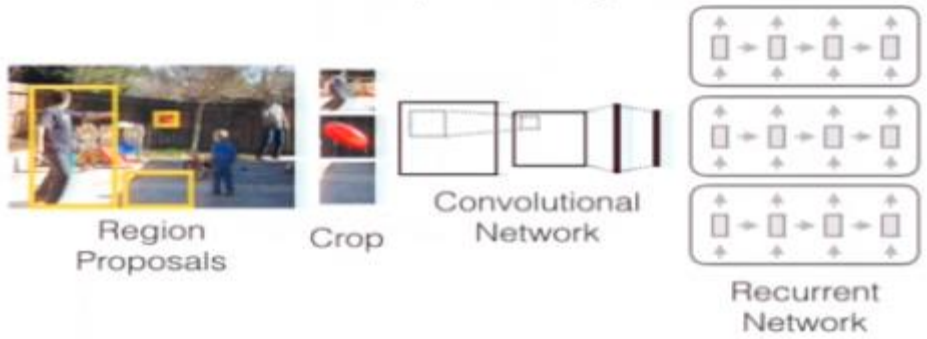
* Interpolation이란 알려진 지점의 값 사이(중간)에 위치한 값을 알려진 값으로부터 추정하는 것을 말한다.

We use bilinear sampling, corresponding to the kernel

$$k(d) = \max(0, 1 - |d|)$$

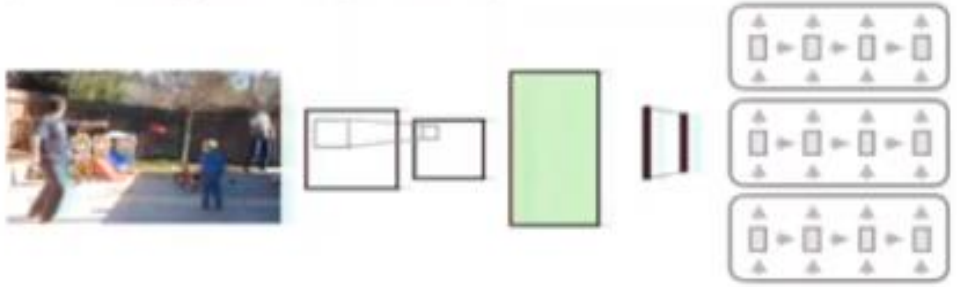
→ Differentiable image sampling!

(Dense Captioning Prior Work)



Advantages over prior work

over [1] (with Region Proposals)



- **No context.** The prediction does not include context outside of each region. **Solved**: features far up the CNN are a function of much larger regions
- **Inefficient.** We must forward every region independently. **Solved**: reuse computation already done inside the ConvNet
- **Not end-to-end.** The use of external region proposals. **Solved**: can be trained end-to-end

[1] Karpathy and Fei-Fei, CVPR 2015

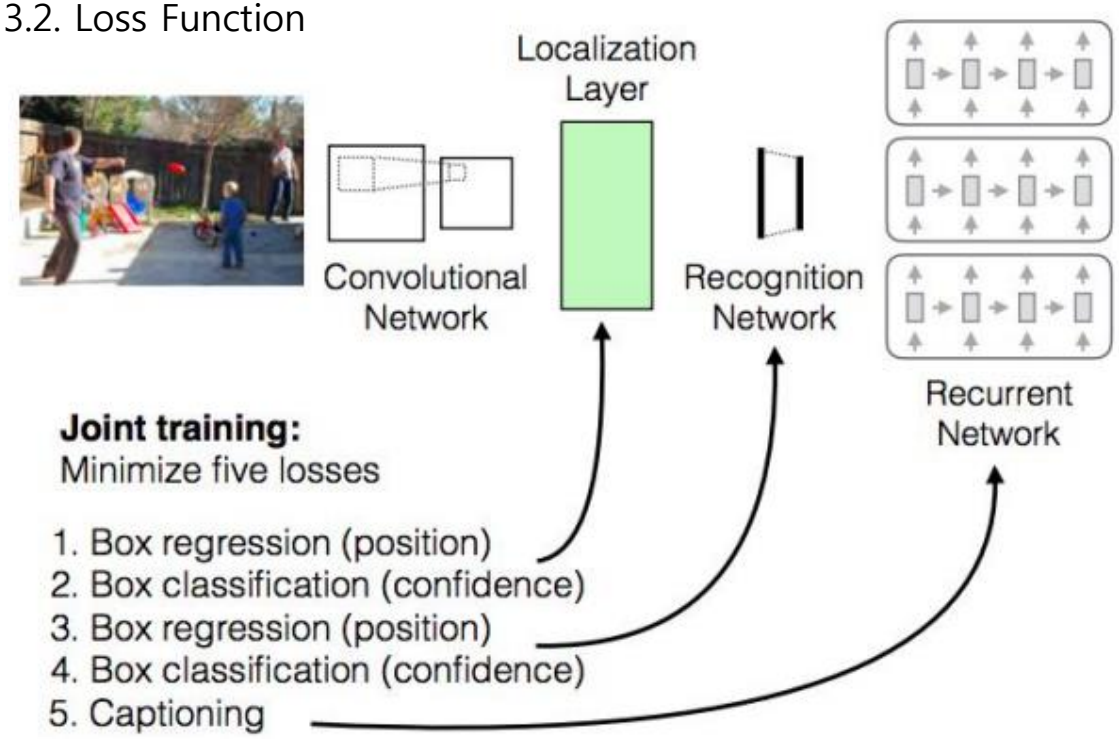
16

전체 이미지에 대해 CNN을 거친 feature map에서 region을 따기 때문에
context에 대한 정보까지 다룰 수 있음

End-to-end Training 가능

3.1.3. Recognition Network
- Localization layer의 output을 받는 two fully-connected layers. Relu랑 dropout씀.
- positive regions만 모아서 BxD(D=4096) RNN으로 들어가.

3.2. Loss Function



3.3. Training and optimization

- training weights : (CNN) pretrained on ImageNet & (Other) a gaussian with standard deviation of 0.01
- optimizer : (CNN) stochastic gradient descent with momentum 0.9 & (Other) Adam
- learning rate = 0.000001
- beta1 = 0.9, beta2 = 0.99
- 효율적인 학습을 위해, CNN 첫 4개 layer는 fine-tuning 안함
- 학습 batch는 하나의 이미지가 들어있고, 가로가 720픽셀로 고정 & resize.
- mini batch 1번이 도는 데 Titan X GPU에서 300ms 소요. 학습 데이터로 모델이 수렴하는데 3일 걸렸다.

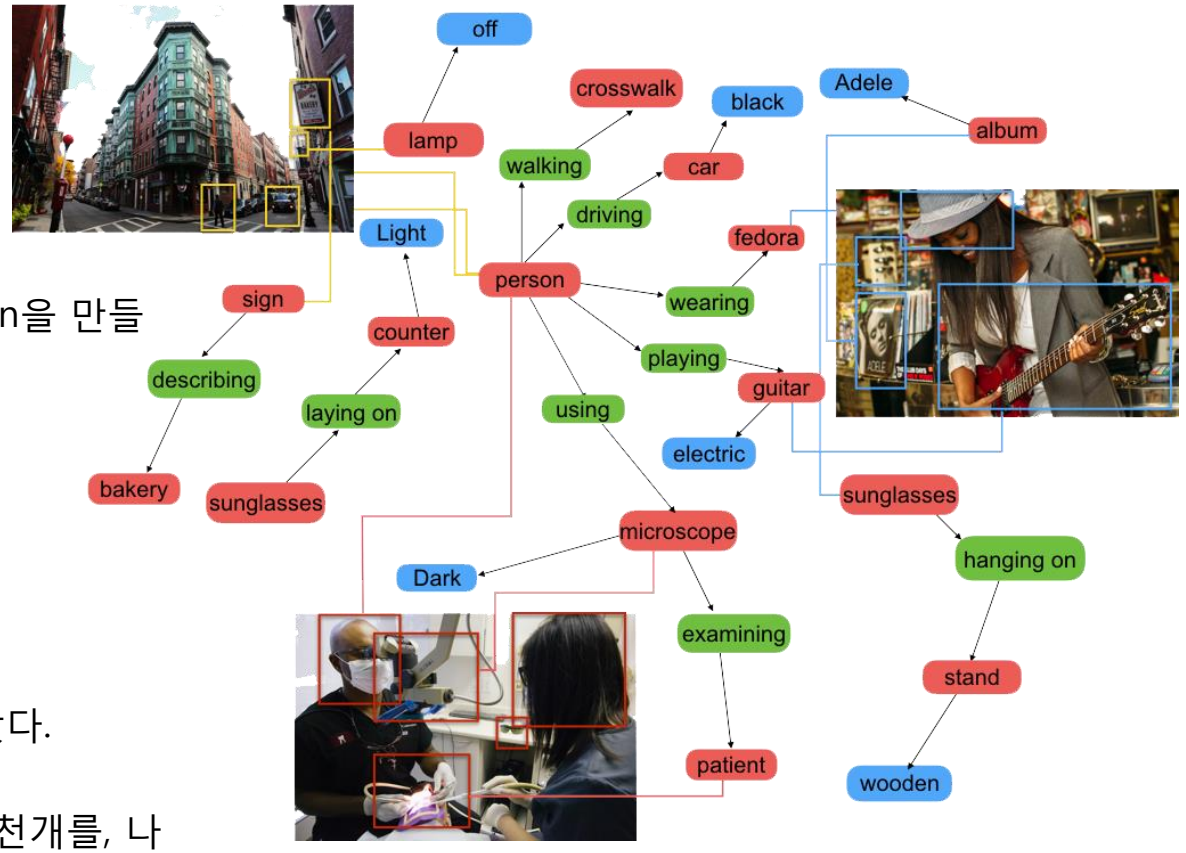
Experiment

*** Dataset**

- individual region captions이 있는 데이터셋이 없었다.
- Visual Genome region captions로 실험했다.
- Amazon Mechanical Turk에서 사람들이 bounding box와 annotation을 만들어줬다.

*** Preprocessing**

- 15번 이하로 언급된 단어는 UNK로 처리. 총 10497개의 단어.
- There Is, This seems to be a 같은 어구는 버림.
- 10단어 이상으로 된 7%의 annotation도 사용하지 않음.
- 20개 미만, 50개 이상의 annotation이 달린 이미지도 사용하지 않았다. (region 수의 분산을 줄일 목적.)
- 그 결과 87,398개의 이미지가 남았고, validation/test data로 각각 5천개를, 나머지를 trainset으로 썼다.



Experiment 1. Dense Captioning

3 sources of Region proposals
GT : Ground Truth boxes
EB : EdgeBoxes (external region proposal method)
RPN : Faster R-CNN Region Proposal Network

mAP across a range of thresholds
for both localization and language accuracy

	Language (METEOR)			**Dense captioning** (AP)			**Test runtime** (ms)			
Region source	EB	RPN	GT	EB	RPN	GT	Proposals	CNN+Recog	RNN	Total
Full image RNN [21]	0.173	0.197	0.209	2.42	4.27	*14.11*	210ms	2950ms	**10ms**	3170ms
Region RNN [21]	0.221	0.244	0.272	1.07	4.26	*21.90*	210ms	2950ms	**10ms**	3170ms
FCLN on EB [13]	**0.264**	**0.296**	0.293	4.88	3.21	*26.84*	210ms	**140ms**	**10ms**	360ms
Our model (FCLN)	**0.264**	0.273	**0.305**	**5.24**	**5.39**	***27.03***	**90ms**	**140ms**	**10ms**	**240ms**

Table 1. Dense captioning evaluation on the test set of 5,000 images. The language metric is METEOR (high is good), our dense captioning metric is Average Precision (AP, high is good), and the test runtime performance for a 720 × 600 image with 300 proposals is given in milliseconds on a Titan X GPU (ms, low is good). EB, RPN, and GT correspond to EdgeBoxes [54], Region Proposal Network [38], and ground truth boxes respectively, used at test time. Numbers in GT columns (italic) serve as upper bounds assuming perfect localization.

Full image RNN : 전체 이미지 vs. 특정 영역에 대한 captioning 비교 목적.
Region RNN : localization layer를 뺀 Image Captioning 모델
FCLN on EB : 모델이 예측한 region이 아닌 고정된 EdgeBox로 region proposal을 학습한 모델.

Experiment 1. Dense Captioning

Region source	Language (METEOR) EB	RPN	GT	Dense captioning (AP) EB	RPN	GT	Test runtime (ms) Proposals	CNN+Recog	RNN	Total
Full image RNN [21]	0.173	0.197	0.209	2.42	4.27	*14.11*	210ms	2950ms	**10ms**	3170ms
Region RNN [21]	0.221	0.244	0.272	1.07	4.26	*21.90*	210ms	2950ms	**10ms**	3170ms
FCLN on EB [13]	**0.264**	**0.296**	0.293	4.88	3.21	*26.84*	210ms	**140ms**	**10ms**	360ms
Our model (FCLN)	**0.264**	0.273	**0.305**	**5.24**	**5.39**	***27.03***	**90ms**	**140ms**	**10ms**	**240ms**

Table 1. Dense captioning evaluation on the test set of 5,000 images. The language metric is METEOR (high is good), our dense captioning metric is Average Precision (AP, high is good), and the test runtime performance for a 720 × 600 image with 300 proposals is given in milliseconds on a Titan X GPU (ms, low is good). EB, RPN, and GT correspond to EdgeBoxes [54], Region Proposal Network [38], and ground truth boxes respectively, used at test time. Numbers in GT columns (italic) serve as upper bounds assuming perfect localization.

*** RPN outperforms external region proposals.**

- 항상 EB region 보다는 RPN의 성능이 좋았다.
- FCLN on EB만 빼고!
  . 사람들이 물체보다 영역을 더 일반화하여 표현하기 때문이라고 추측함.
  . RPN은 raw data의 분포를 학습하지만, EB는 물체에 대한 high recall을 목적으로 설계되었기 때문.
- FCLN > FCLN on EB
  . localization과 독립적인 language score에는 FCLN on EB의 성능이 좋음에도 불구하고,
  . AP기준으로 FCLN의 성능이 좋은 것은, localization을 잘했기 때문이라는 것을 반증한다.

Experiment 1. Dense Captioning

Region source	Language (METEOR) EB	RPN	GT	Dense captioning (AP) EB	RPN	GT	Test runtime (ms) Proposals	CNN+Recog	RNN	Total
Full image RNN [21]	0.173	0.197	0.209	2.42	4.27	*14.11*	210ms	2950ms	**10ms**	3170ms
Region RNN [21]	0.221	0.244	0.272	1.07	4.26	*21.90*	210ms	2950ms	**10ms**	3170ms
FCLN on EB [13]	**0.264**	**0.296**	0.293	4.88	3.21	*26.84*	210ms	**140ms**	**10ms**	360ms
Our model (FCLN)	**0.264**	0.273	**0.305**	**5.24**	**5.39**	***27.03***	**90ms**	**140ms**	**10ms**	**240ms**

Table 1. Dense captioning evaluation on the test set of 5,000 images. The language metric is METEOR (high is good), our dense captioning metric is Average Precision (AP, high is good), and the test runtime performance for a 720 × 600 image with 300 proposals is given in milliseconds on a Titan X GPU (ms, low is good). EB, RPN, and GT correspond to EdgeBoxes [54], Region Proposal Network [38], and ground truth boxes respectively, used at test time. Numbers in GT columns (italic) serve as upper bounds assuming perfect localization.

*** Our model outperforms individual region description.**

- 두 모델은 Localization layer 유무의 차이이다.
- Localization이 있는 Our model이 우수한 이유는, region 바깥의 context 정보까지 사용했기 때문

*** Runtime Evaluation**

- NMS (subsample region)에 80ms가 소요됨.
- Region proposal 개수를 기존 300에서 100으로 줄이면, 총 소요시간이 166ms로 줄어들었다.