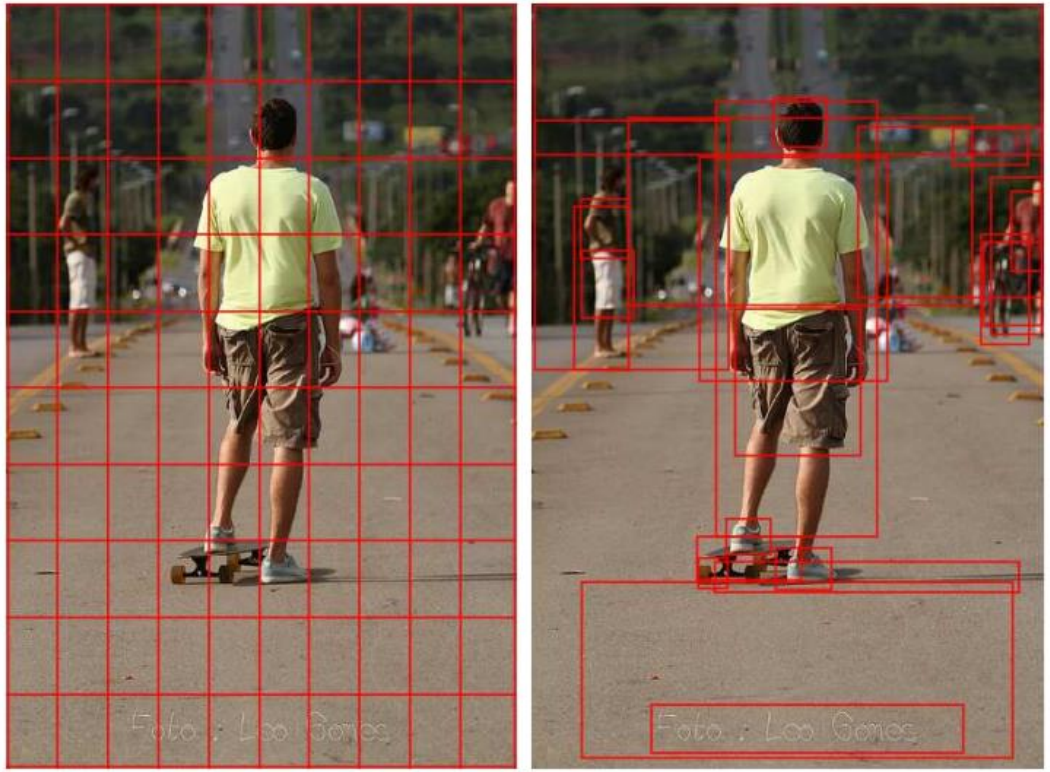
Bottom-up and Top-Down Attention
For Image Captioning and Visual Question Answering

Choi. w. h.

Bottom-up and Top-down

Top-Down 방식

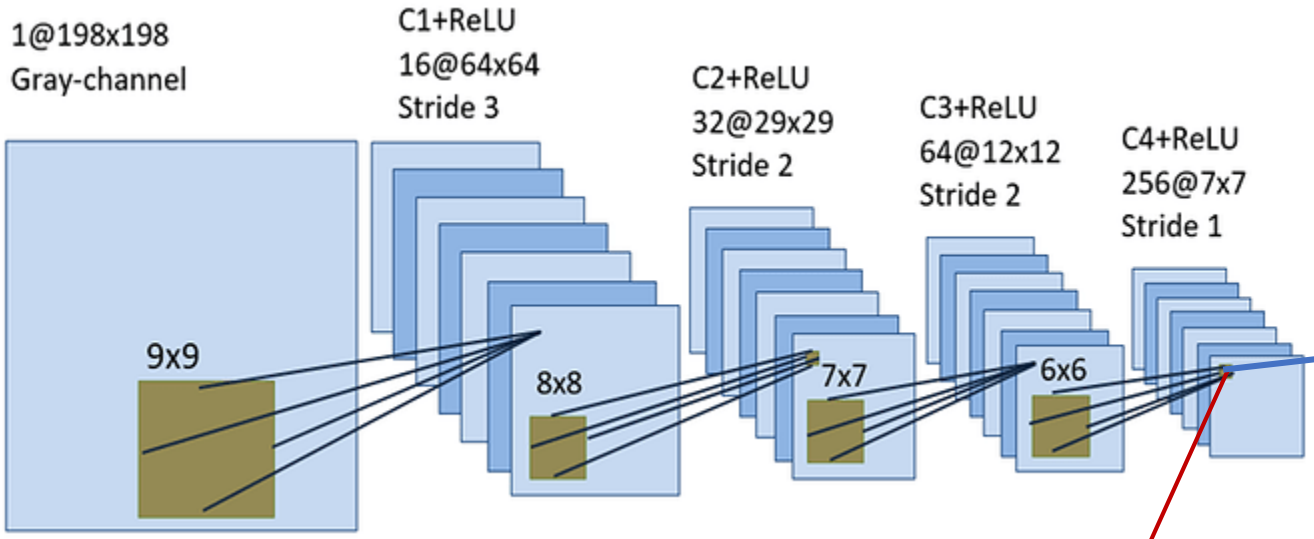


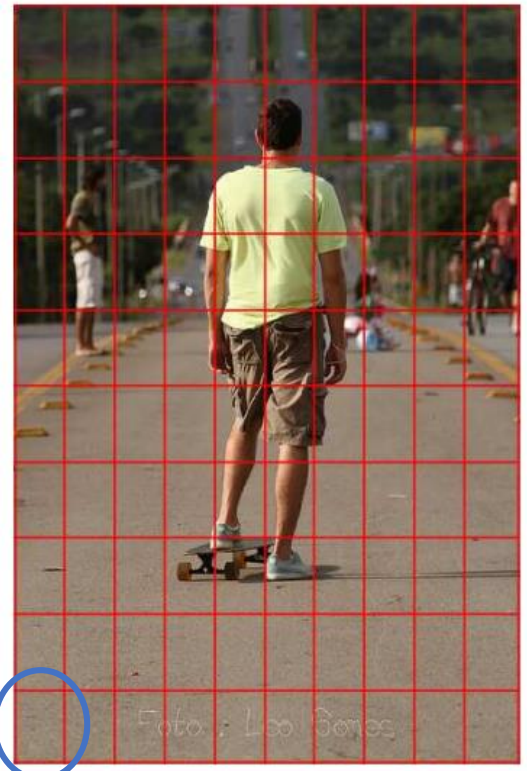
Bottom-Up 방식

Figure 1. Typically, attention models operate on CNN features corresponding to a uniform grid of equally-sized image regions (left). Our approach enables attention to be calculated at the level of objects and other salient image regions (right).

Top-Down

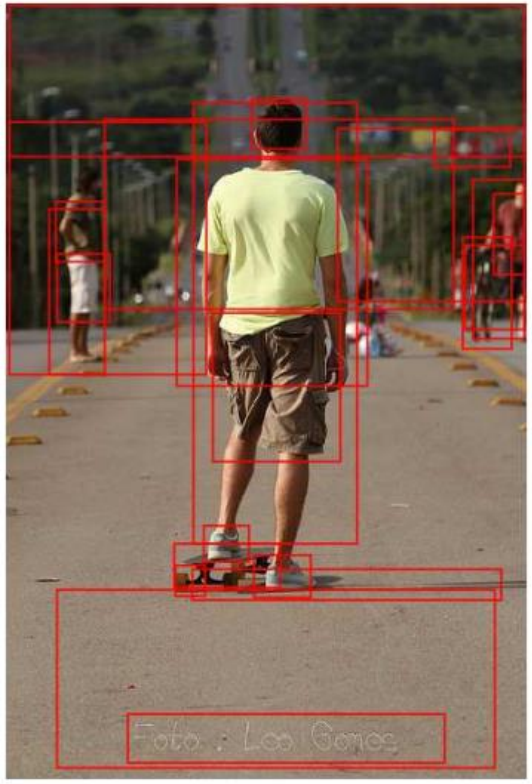
Top-Down Attention은 기존 방법





Deep Layer의 Feature Map 1개의 값은
이미지의 어느 부분의 공간적 정보를 담고 있다

Bottom-Up



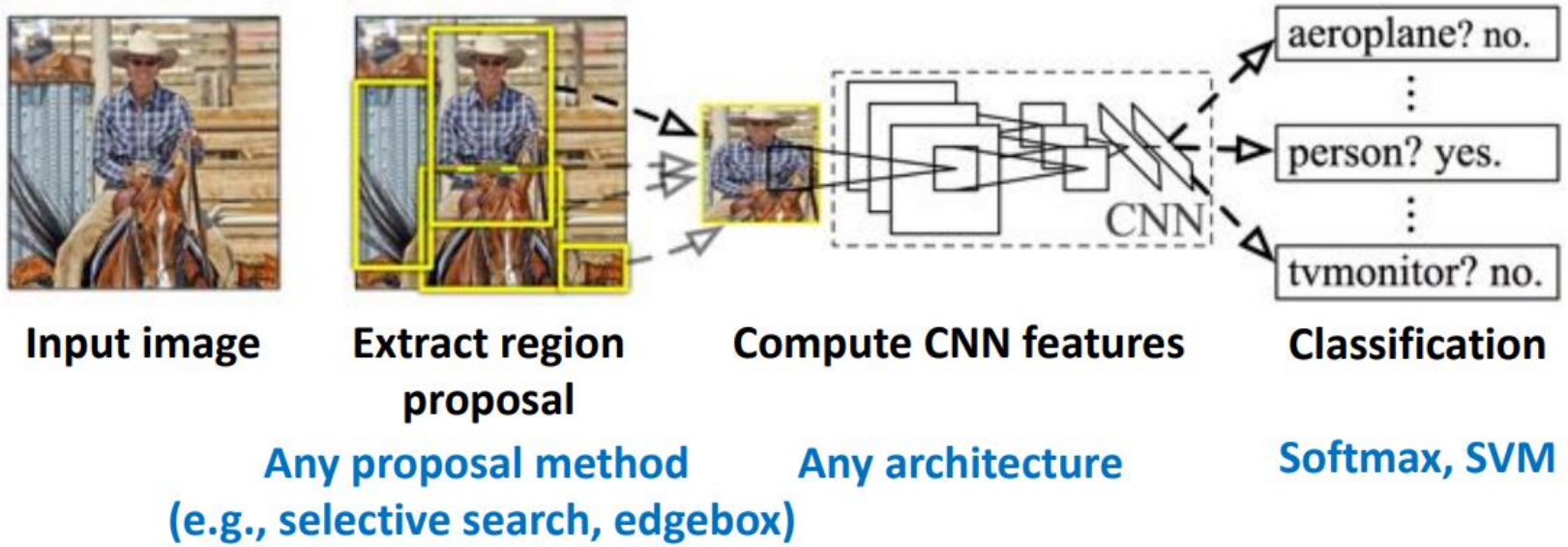
- Bottom-Up Mechanism은 인간과 유사한 Visual Attention Mechanism
- Faster R-CNN을 사용하여 정보 획득

Proposal

Top-Down 방식과 Bottom-Up 방식을 같이 활용하여

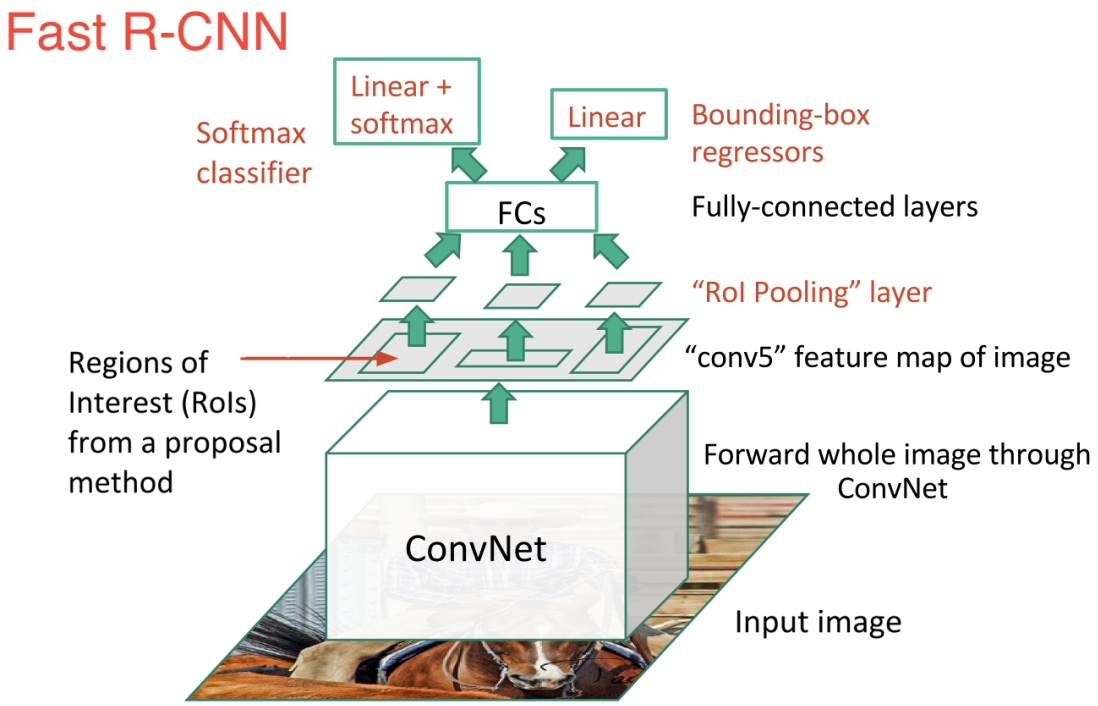
Image Captioning과 Visual Question Answering Task를 잘 할 수 있다.

R-CNN



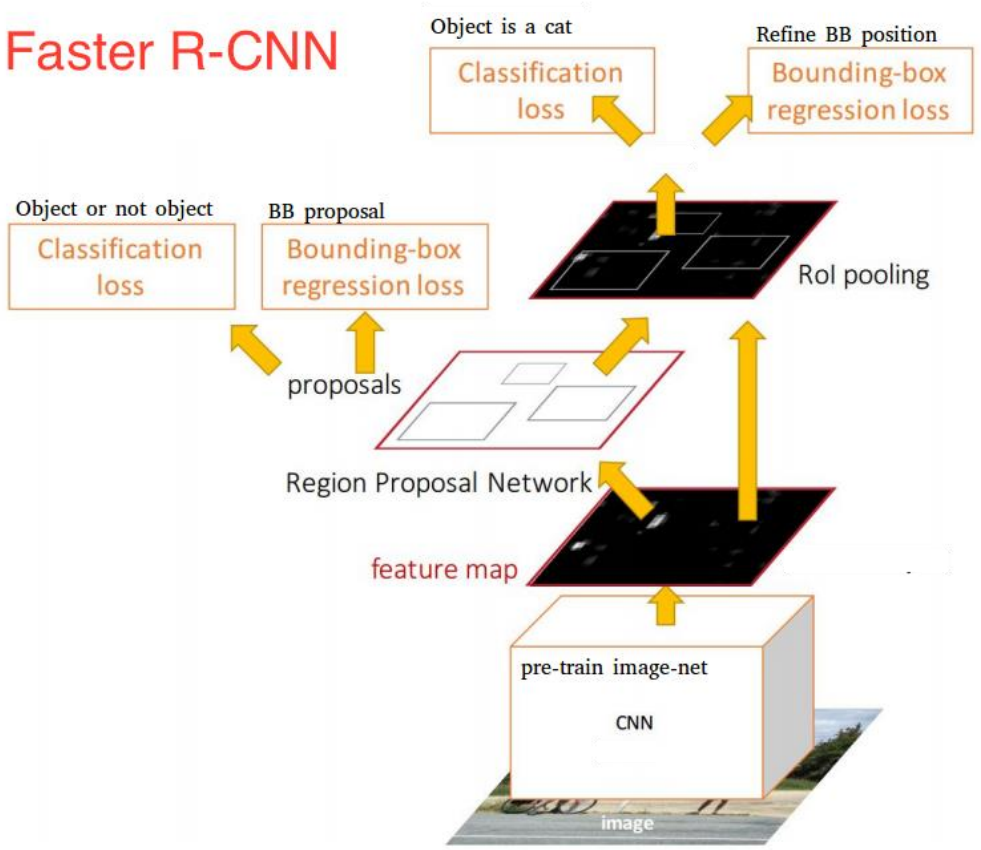
- Object가 있을 만한 Region을 찾아낸다.
- 해당 영역들을 동일한 size로 조정하고 CNN Network에 넣어 Classification을 한다.

Fast R-CNN



- Image를 ConvNet을 통과시켜 Feature를 뽑아낸다.
- 해당 Feature에서 Region of Interest(ROI)를 찾아낸다. (proposal method를 통해서)
- Feature을 FC Layer를 통과하여 Bounding-box regression과 Classification 한다.

Faster R-CNN



- Image를 ConvNet을 통과시켜 Feature를 뽑아낸다.
- 해당 Feature에서 Region of Interest(ROI)를 찾아낸다. (proposal network를 통해서)
- Feature을 FC Layer를 통과하여 Bounding-box regression과 Classification 한다.

concat

Top-Down Attention LSTM Input : $\boldsymbol{x}_t^1 = [\boldsymbol{h}_{t-1}^2, \bar{\boldsymbol{v}}, W_e \Pi_t]$

$$\bar{v} = \frac{1}{k}\sum_i v_i$$

v_i : Bottom-Up output Feature

$W_e \in \mathbb{R}^{E \times |\Sigma|}$ Word-embedding matrix

Σ : Vocabulary

Π_t One-hot encoding of the input word at timestep t

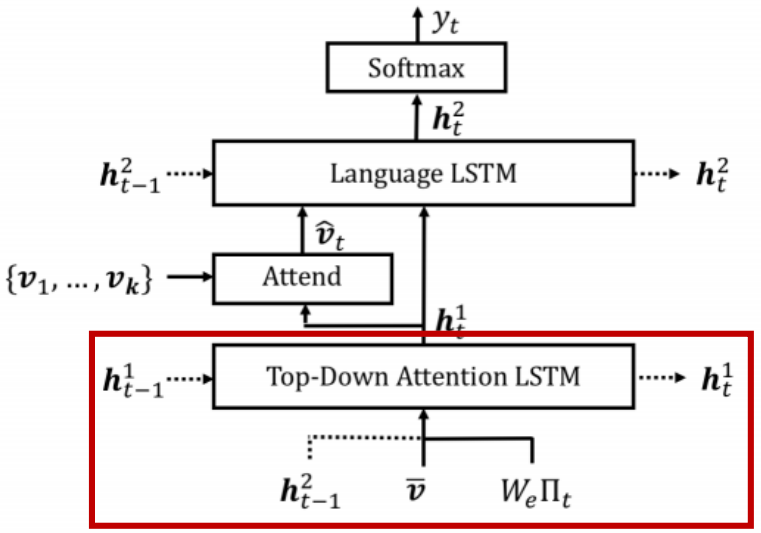


Image Captioning

$$a_{i,t} = \boldsymbol{w}_a^T \tanh\left(W_{va}\boldsymbol{v}_i + W_{ha}\boldsymbol{h}_t^1\right)$$
$$\boldsymbol{\alpha}_t = \text{softmax}\left(\boldsymbol{a}_t\right)$$

where $W_{va} \in \mathbb{R}^{H\times V}$, $W_{ha} \in \mathbb{R}^{H\times M}$ and $\boldsymbol{w}_a \in \mathbb{R}^H$

$$\hat{\boldsymbol{v}}_t = \sum_{i=1}^{K} \alpha_{i,t}\boldsymbol{v}_i$$

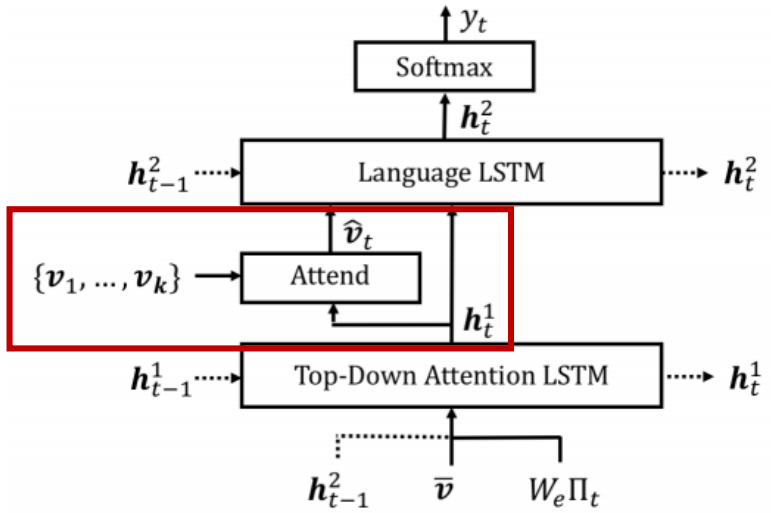


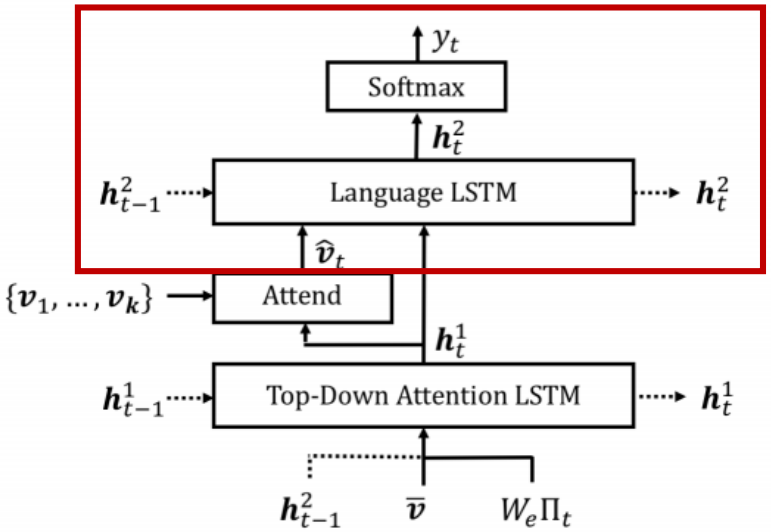
Image Captioning

concat

Language LSTM Input : $\boldsymbol{x}_t^2 = [\hat{\boldsymbol{v}}_t, \boldsymbol{h}_t^1]$

$$p(y_t \mid y_{1:t-1}) = \text{softmax}\left(W_p \boldsymbol{h}_t^2 + \boldsymbol{b}_p\right)$$

where $W_p \in \mathbb{R}^{|\Sigma| \times M}$ and $\boldsymbol{b}_p \in \mathbb{R}^{|\Sigma|}$



Complete output sequences $p(y_{1:T}) = \prod_{t=1}^{T} p(y_t \mid y_{1:t-1})$

Objective

Given a target ground truth sequence $y^*_{1:T}$ and a captioning model with parameters θ, we minimize the following cross entropy loss:

$$L_{XE}(\theta) = -\sum_{t=1}^{T} \log(p_\theta(y^*_t \mid y^*_{1:t-1}))$$

For fair comparison with recent work we also report results optimized for CIDEr . Initializing from the cross-entropy trained model, we seek to minimize the negative expected score:

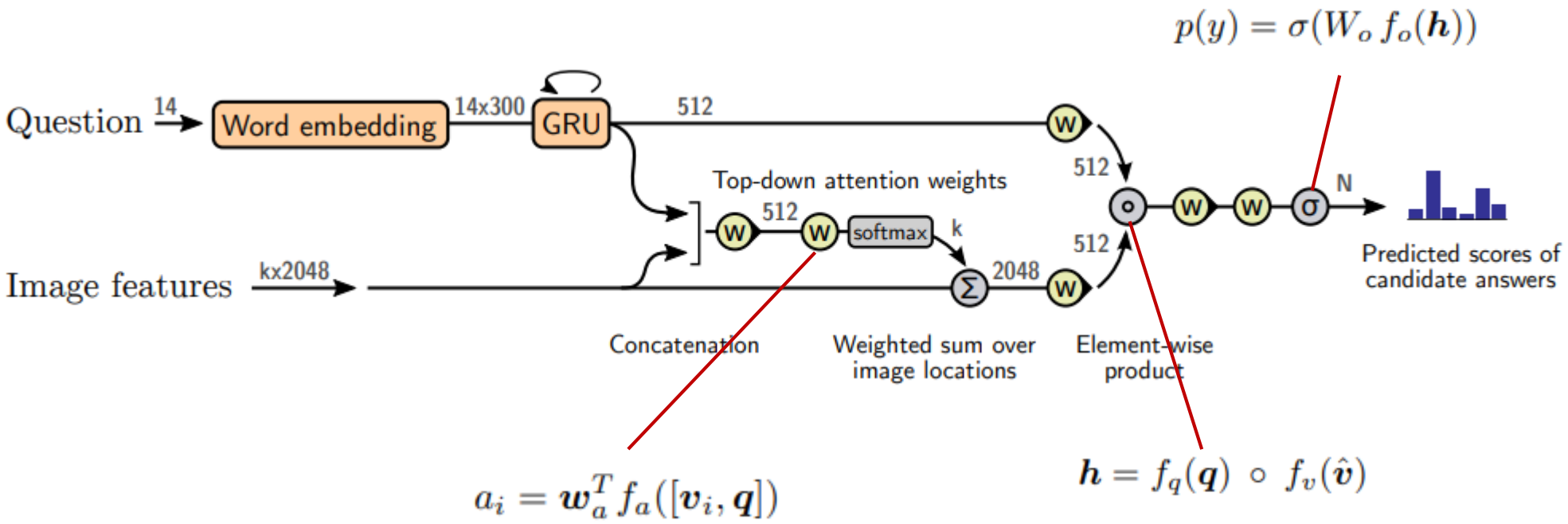
$$L_R(\theta) = -\mathbf{E}_{y_{1:T} \sim p_\theta}[r(y_{1:T})]$$

where r is the score function (e.g., CIDEr). Following the approach described as Self-Critical Sequence Training (SCST), the gradient of this loss can be approximated:

$$\nabla_\theta L_R(\theta) \approx -(r(y^s_{1:T}) - r(\hat{y}_{1:T}))\nabla_\theta \log p_\theta(y^s_{1:T})$$

where $y^s_{1:T}$ is a sampled caption and $r(\hat{y}_{1:T})$ defines the baseline score obtained by greedily decoding the current model.

VQA


$p(y) = \sigma(W_o\, f_o(\boldsymbol{h}))$
Question
14
Word embedding
14x300
GRU
512
Top-down attention weights
512
softmax
k
Image features
kx2048
2048
512
512
N
Predicted scores of candidate answers
Concatenation
Weighted sum over image locations
Element-wise product
$a_i = \boldsymbol{w}_a^T f_a([\boldsymbol{v}_i, \boldsymbol{q}])$
$\boldsymbol{h} = f_q(\boldsymbol{q}) \circ f_v(\hat{\boldsymbol{v}})$

Result

Image Captioning

	Cross-Entropy Loss						CIDEr Optimization					
	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
SCST:Att2in [33]	-	31.3	26.0	54.3	101.3	-	-	33.3	26.3	55.3	111.4	-
SCST:Att2all [33]	-	30.0	25.9	53.4	99.4	-	-	34.2	26.7	55.7	114.0	-
Ours: ResNet	74.5	33.4	26.1	54.4	105.4	19.2	76.6	34.0	26.5	54.9	111.1	20.2
Ours: Up-Down	**77.2**	**36.2**	**27.0**	**56.4**	**113.5**	**20.3**	**79.8**	**36.3**	**27.7**	**56.9**	**120.1**	**21.4**
Relative Improvement	4%	8%	3%	4%	8%	6%	4%	7%	5%	4%	8%	6%

Table 1. Single-model image captioning performance on the MSCOCO Karpathy test split. Our baseline ResNet model obtains similar results to SCST [33], the existing state-of-the-art on this test set. Illustrating the contribution of bottom-up attention, our Up-Down model achieves significant (3–8%) relative gains across all metrics regardless of whether cross-entropy loss or CIDEr optimization is used.

	Cross-Entropy Loss							CIDEr Optimization						
	SPICE	Objects	Attributes	Relations	Color	Count	Size	SPICE	Objects	Attributes	Relations	Color	Count	Size
Ours: ResNet	19.2	35.4	8.6	5.3	12.2	4.1	3.9	20.2	37.0	9.2	6.1	10.6	12.0	**4.3**
Ours: Up-Down	**20.3**	**37.1**	**9.2**	**5.8**	**12.7**	**6.5**	**4.5**	**21.4**	**39.1**	**10.0**	**6.5**	**11.4**	**18.4**	3.2

Table 2. Breakdown of SPICE F-scores over various subcategories on the MSCOCO Karpathy test split. Our Up-Down model outperforms the ResNet baseline at identifying objects, as well as detecting object attributes and the relations between objects.

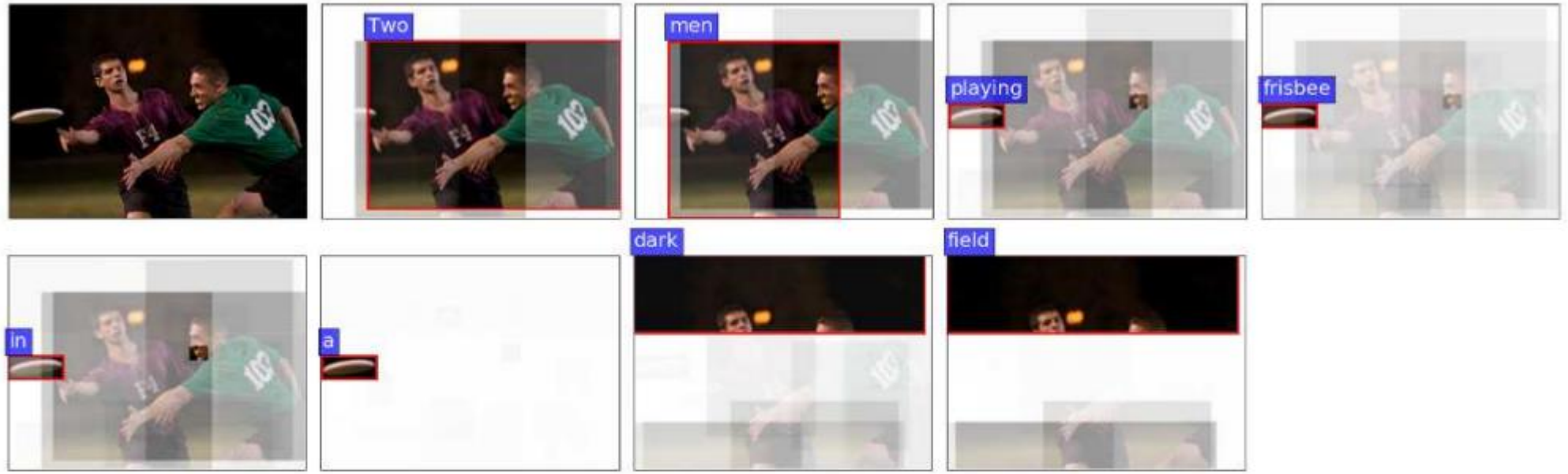
Result

Image Captioning

	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr		SPICE	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Review Net [47]	72.0	90.0	55.0	81.2	41.4	70.5	31.3	59.7	25.6	34.7	53.3	68.6	96.5	96.9	18.5	64.9
Adaptive [27]	74.8	92.0	58.4	84.5	44.4	74.4	33.6	63.7	26.4	35.9	55.0	70.5	104.2	105.9	19.7	67.3
PG-BCMR [24]	75.4	-	59.1	-	44.5	-	33.2	-	25.7	-	55	-	101.3	-	-	-
SCST:Att2all [33]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.7	20.7	68.9
LSTM-A_3 [48]	78.7	93.7	62.7	86.7	47.6	76.5	35.6	65.2	27	35.4	56.4	70.5	116	118	-	-
Ours: Up-Down	**80.2**	**95.2**	**64.1**	**88.8**	**49.1**	**79.4**	**36.9**	**68.5**	**27.6**	**36.7**	**57.1**	**72.4**	**117.9**	**120.5**	**21.5**	**71.5**

Table 3. Highest ranking published image captioning results on the online MSCOCO test server. Our submission, an ensemble of 4 models optimized for CIDEr with different initializations, outperforms previously published work on all reported metrics. At the time of submission (18 July 2017), we also outperformed all unpublished test server submissions.

Image Captioning

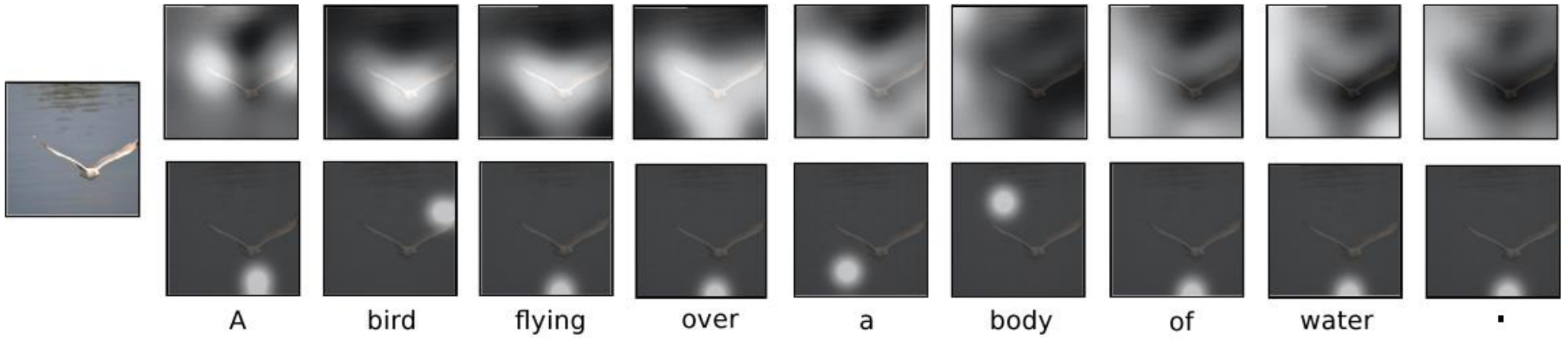


Two men playing frisbee in a dark field.

Figure 5. Example of a generated caption showing attended image regions. For each generated word, we visualize the attention weights on individual pixels, outlining the region with the maximum attention weight in red. Avoiding the conventional trade-off between coarse and fine levels of detail, our model focuses on both closely-cropped details, such as the frisbee and the green player's mouthguard when generating the word 'playing', as well as large regions, such as the night sky when generating the word 'dark'.

Result

기존 Top-Down 방식의 Attention



VQA

	Yes/No	Number	Other	Overall
d-LSTM+n-I [26, 12]	73.46	35.18	41.83	54.22
MCB [11, 12]	78.82	38.28	53.36	62.27
UPMC-LIP6	82.07	41.06	57.12	65.71
Athena	82.50	44.19	59.97	67.59
HDU-USYD-UNCC	84.50	45.39	59.01	68.09
Ours: Up-Down	**86.60**	**48.64**	**61.15**	**70.34**

Table 5. VQA v2.0 test-standard server accuracy as at 8 August 2017, ranking our submission against published and unpublished work for each question type. Our approach, an ensemble of 30 models, outperforms all other leaderboard entries.

Question: What room are they in? Answer: kitchen

Figure 6. VQA example illustrating attention output. Given the question 'What room are they in?', the model focuses on the stove-top, generating the answer 'kitchen'.

APPENDIX

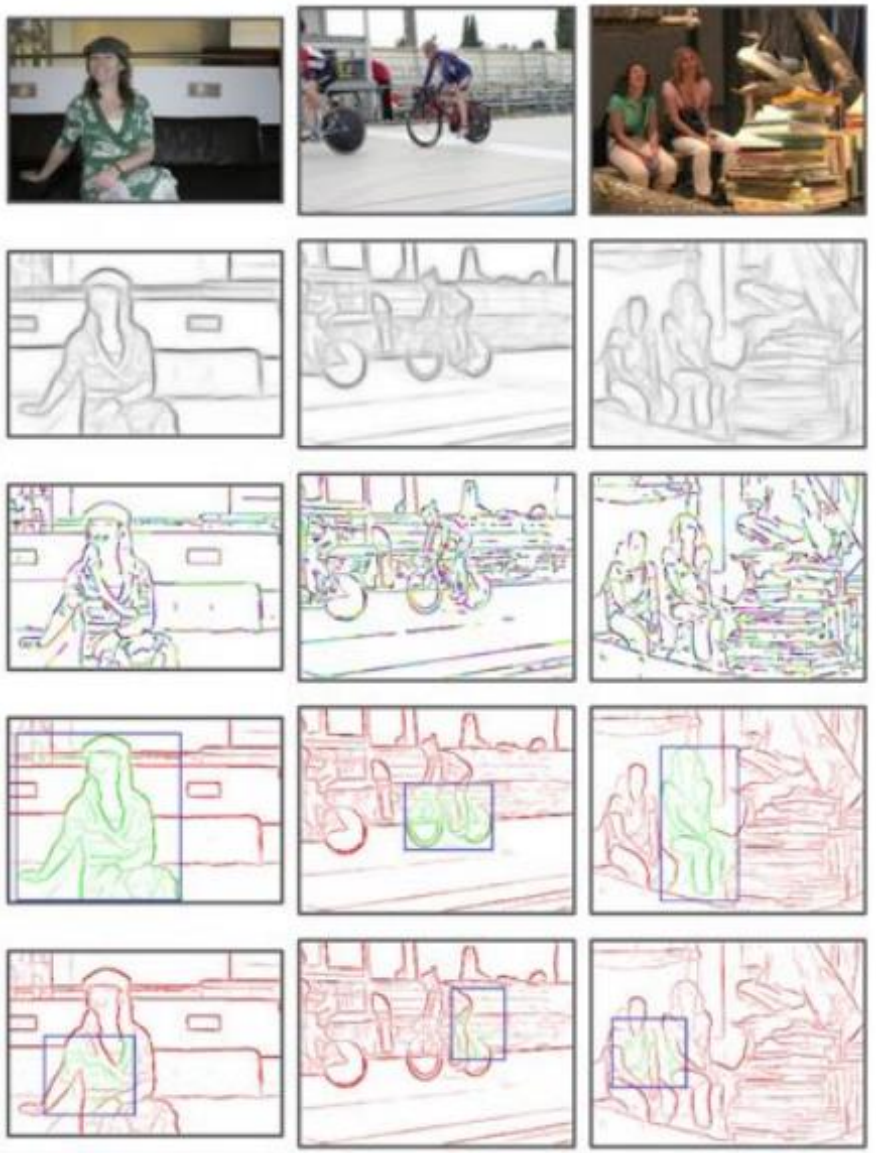
Selective Search

Greedy hierarchical superpixel segmentation

EdgeBox

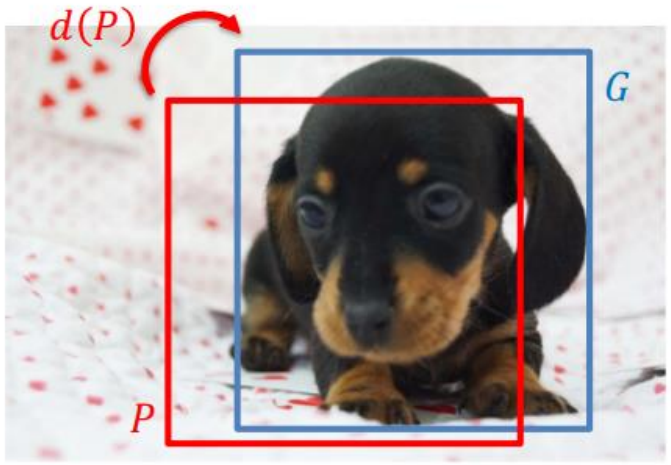
<Box score>

(number of edges in the box)
-
(number of edges that overlab the box boundary)



Bounding Box Regression

- Learning a transformation of bounding box
  - Region proposal: $P = (P_w, P_h, P_w, P_h)$
  - Ground-truth: $G = (G_x, G_y, G_w, G_h)$
  - Transformation: $d(P) = (t_x, t_y, t_w, t_h)$



$$\hat{G}_x = P_w d_x(P) + P_x$$
$$\hat{G}_y = P_h d_y(P) + P_y$$
$$\hat{G}_w = P_w \exp\big(d_w(P)\big)$$
$$\hat{G}_h = P_h \exp\big(d_h(P)\big)$$

$$d_i(P) = \boldsymbol{w}_i^{\mathrm{T}} \phi_5(P)$$

CNN pool5 feature

$$\boldsymbol{w}_i^* = \underset{\boldsymbol{w}_i}{\operatorname{argmin}} \sum_{k=1}^{N} \left(t_i^k - \boldsymbol{w}_i^{\mathrm{T}} \phi_5\big(P^k\big)\right)^2 + \lambda \|\boldsymbol{w}_i\|^2$$