Methods for Dense Captioning

Alex Yue & Ryan McCaffrey

Roadmap

1. Dense captioning
2. Paper 1: Visual-Semantic Alignments for Generating Image Descriptions
    - Overview
    - Methods:
        i. Representing images and sentences
        ii. Data alignment
    - Evaluation and Metrics
3. Paper 2: DenseCap
    - Captioning
    - Architecture
    - Evaluation and Metrics

Recognition Tasks

Deep Visual-Semantic Alignments for Generating Image Descriptions

Andrej Karpathy, Li Fei-Fei

[Paper]

Goals + Motivation

- Design model that reasons about content of images and their representation in the domain of natural language
- Make model free of assumptions about hard-coded templates, rules, or categories
    - Previous work in captioning uses fixed vocabulary or non-generative methods

Datasets

- Flickr8K (validation, testing, training)
- Flickr30K (validation, testing, training)
- MSCOCO (validation, testing)

- A man is snowboarding over a structure on a snowy hill.
- A snowboarder jumps through the air on a snowy hill.
- a snowboarder wearing green pants doing a trick on a high bench
- Someone in yellow pants is on a ramp over the snow.
- The man is performing a trick on a snowboard high in the air.

Image credit: Flickr8K

Input Data

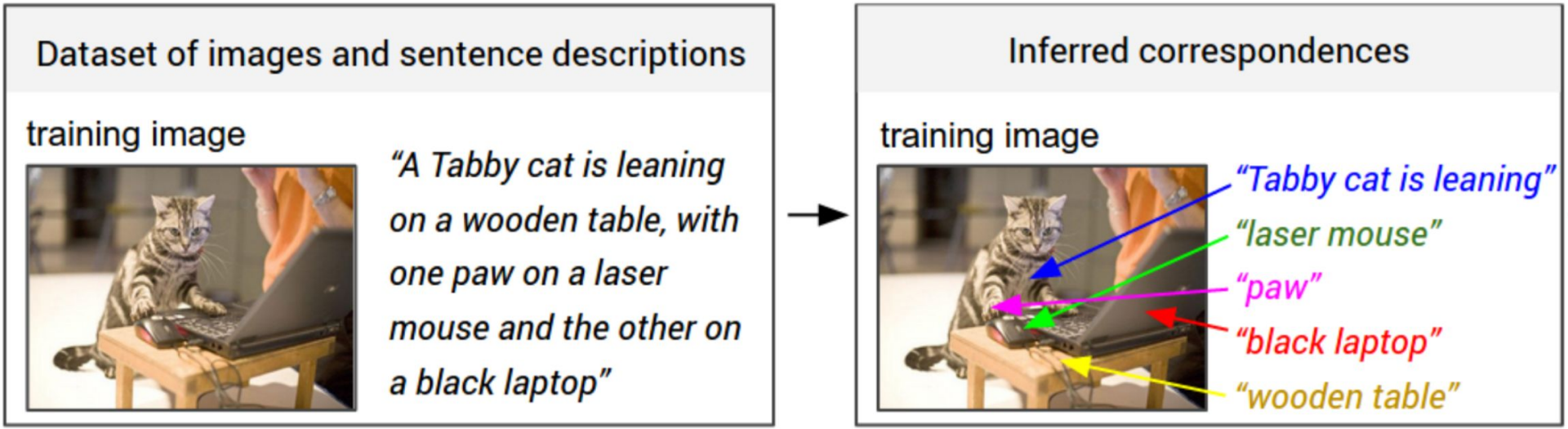


Image credit: A. Karpathy & L. Fei-Fei

Making Training Data Usable

1. Vectorizing images
2. Vectorizing descriptions
3. Grounding image with descriptions

Making Training Data Usable

1. **Vectorizing images**
2. Vectorizing descriptions
3. Grounding image with descriptions

1.1: Representing Images

- Use Region Convolutional Neural Network (R-CNN) pre-trained on ImageNet
- Take top 19 detected locations + whole image, compute representations

$$v = W_m[CNN_{\theta_c}(I_b)] + b_m$$

Making Training Data Usable

1. Vectorizing images
2. **Vectorizing descriptions**
3. Grounding image with descriptions

Recurrent Neural Network (RNN) Review

We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

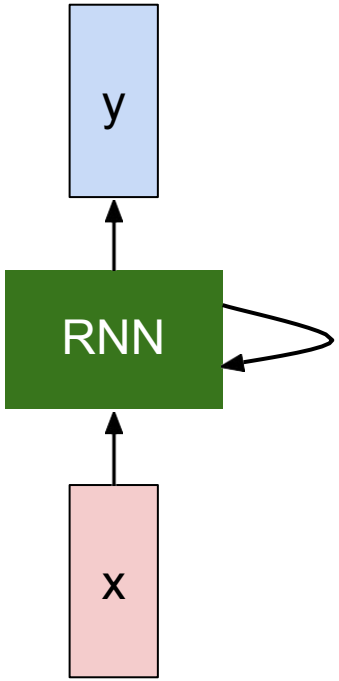
$$h_t = f_W(h_{t-1}, x_t)$$

new state

old state

some function with parameters W

input vector at some time step



Slide Credit: CS231n. Fei-Fei Li & Justin Johnson & Serena Yeung

Bidirectional RNN Overview

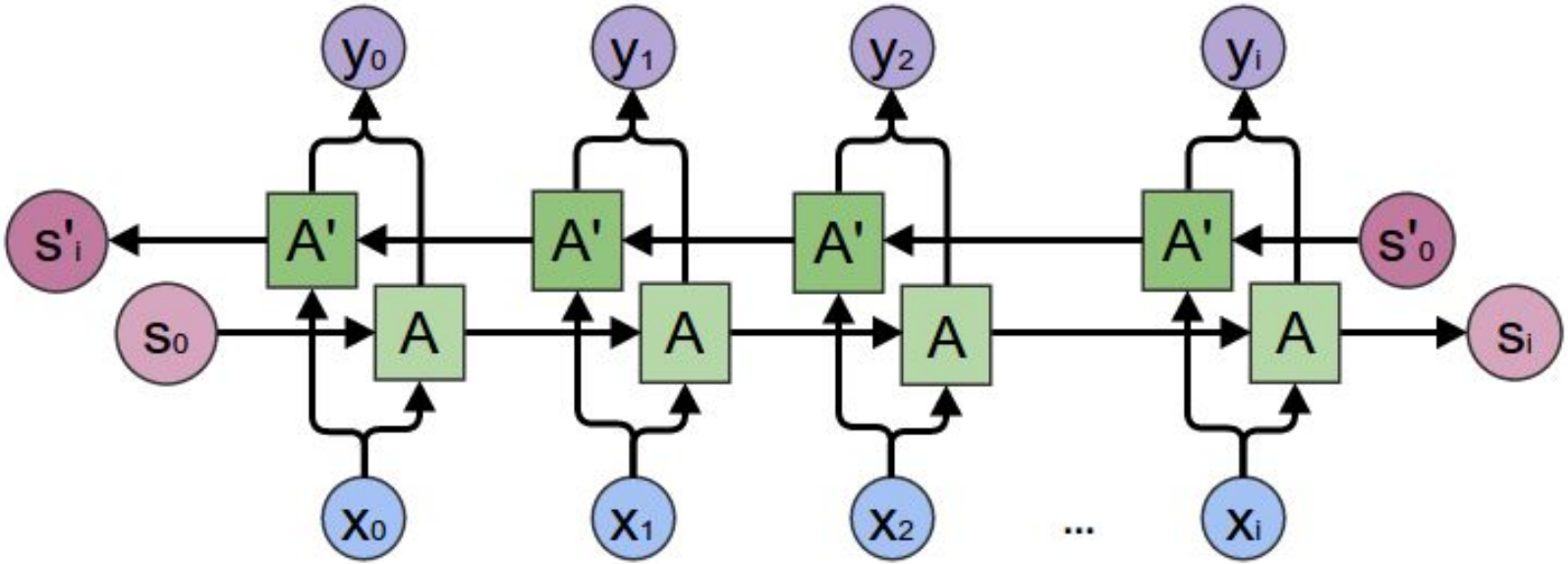


Image credit: [colah's blog]

Word Embedding Matrix

- 300 x n matrix, where n is the number of words in the word vocabulary
- Weights initialized with word2vec weights
  - Word2vec a project led by Tomas Mikolov at Google. Project is trained to reconstruct linguistic context of words. Word vectors positioned in space such that words that share common contexts are close in proximity

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97
...						
size						
cost						
alive						

Image credit: Andrew Ng

1.2: Representing Sentences

- BRNN takes N words and transforms each into h-dimensional vector

$$x_t = W_w \mathbb{I}_t$$

$$e_t = f(W_e x_t + b_e)$$

$$h_t^f = f(e_t + W_f h_{t-1}^f + b_f) \longleftarrow \text{Forward output}$$

$$h_t^b = f(e_t + W_b h_{t+1}^b + b_b) \longleftarrow \text{Backward output}$$

$$s_t = f(W_d(h_t^f + h_t^b) + b_d)$$

Making Training Data Usable

1. Vectorizing images
2. Vectorizing descriptions
3. **Grounding image with descriptions**

Aligning Image + Sentences

$$S_{kl} = \sum_{t \in g_l} \max_{i \in g_k} v_i^\top s_t$$

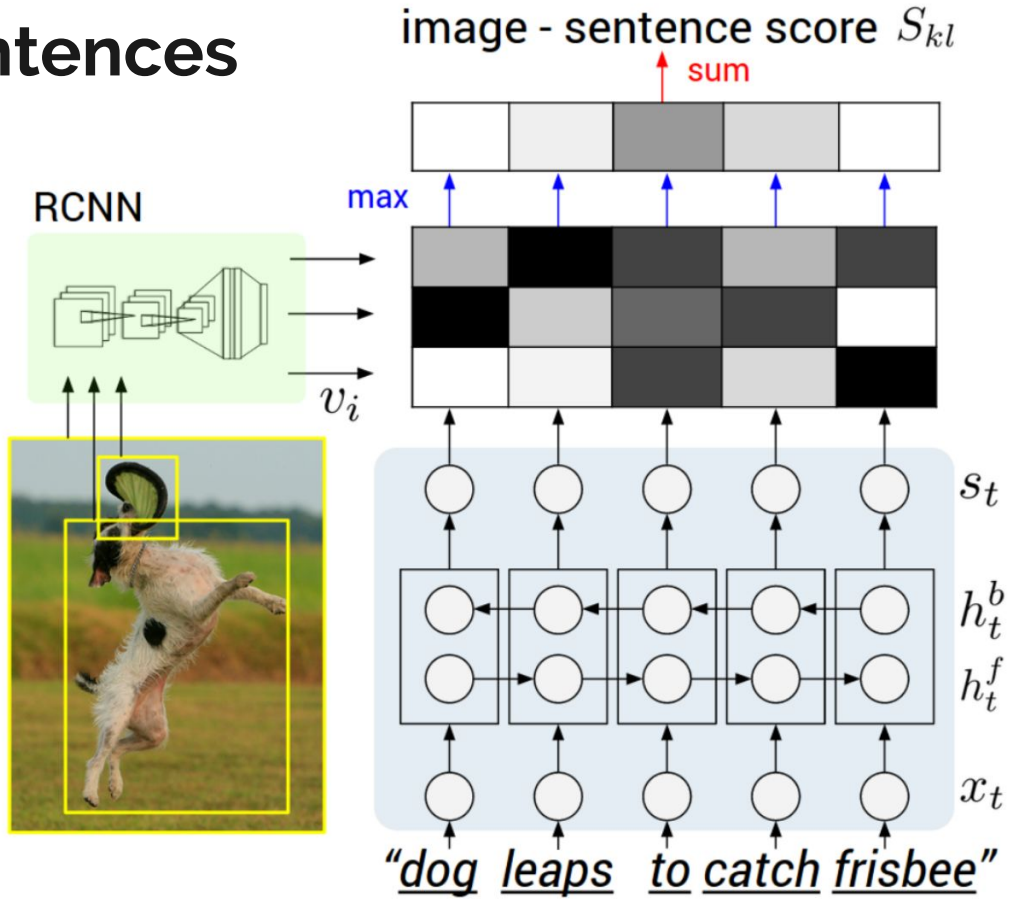
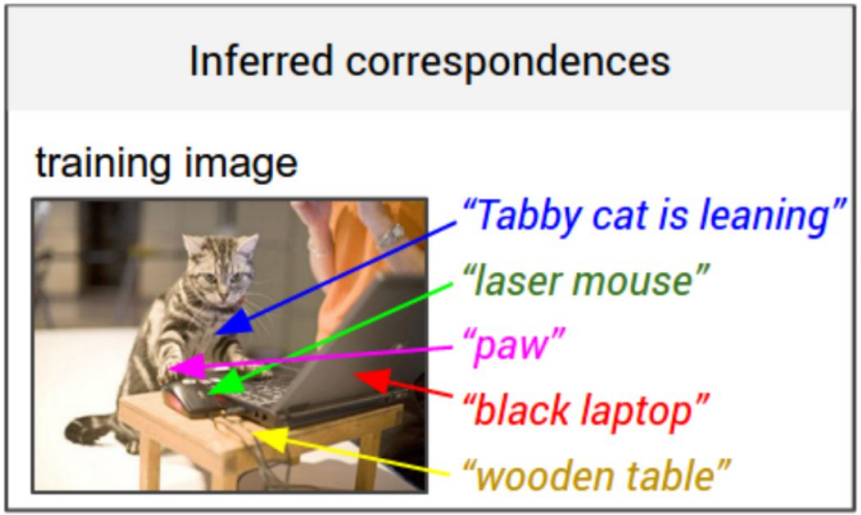


Image credit: A. Karpathy & L. Fei-Fei

Aligning Image + Sentences

$$E(a) = \sum_{j=1...N} \psi_j^U(a_j) + \sum_{j=1...N-1} \psi_j^B(a_j, a_{j+1})$$

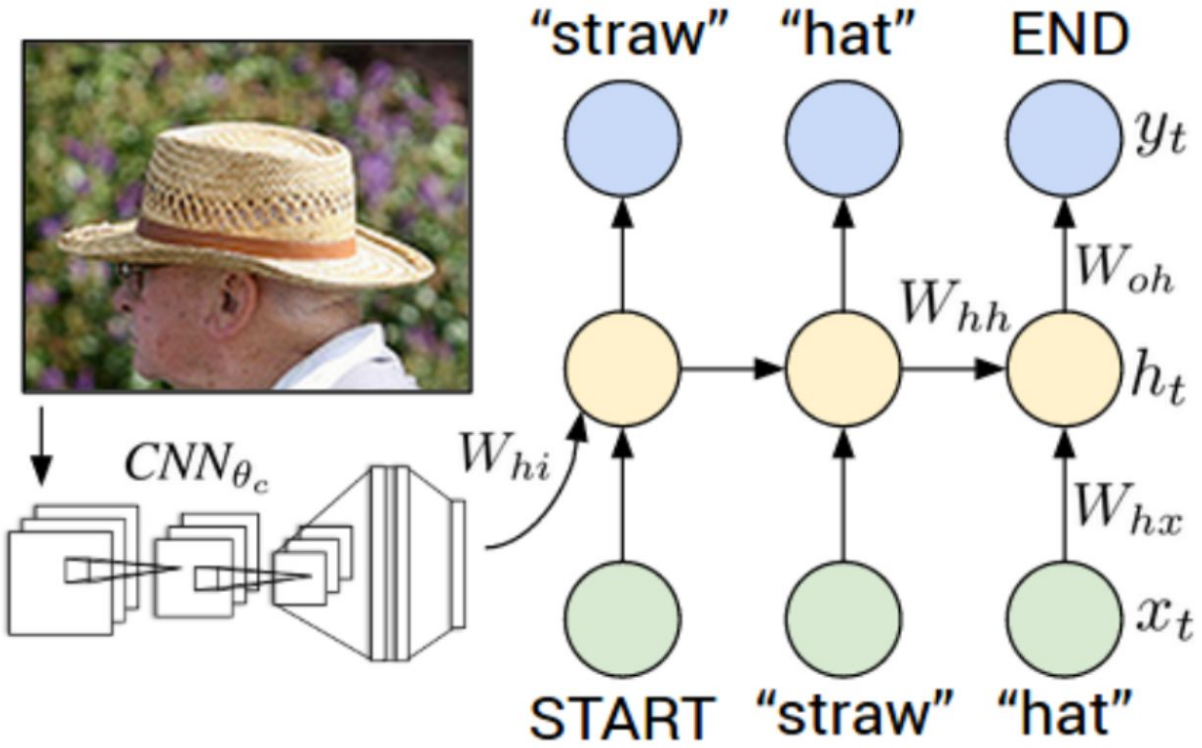


$$\psi_j^U(a_j = t) = v_i^\top s_t$$

$$\psi_j^B(a_j, a_{j+1}) = \beta \mathbb{I}[a_j = a_{j+1}]$$

Image credit: A. Karpathy & L. Fei-Fei

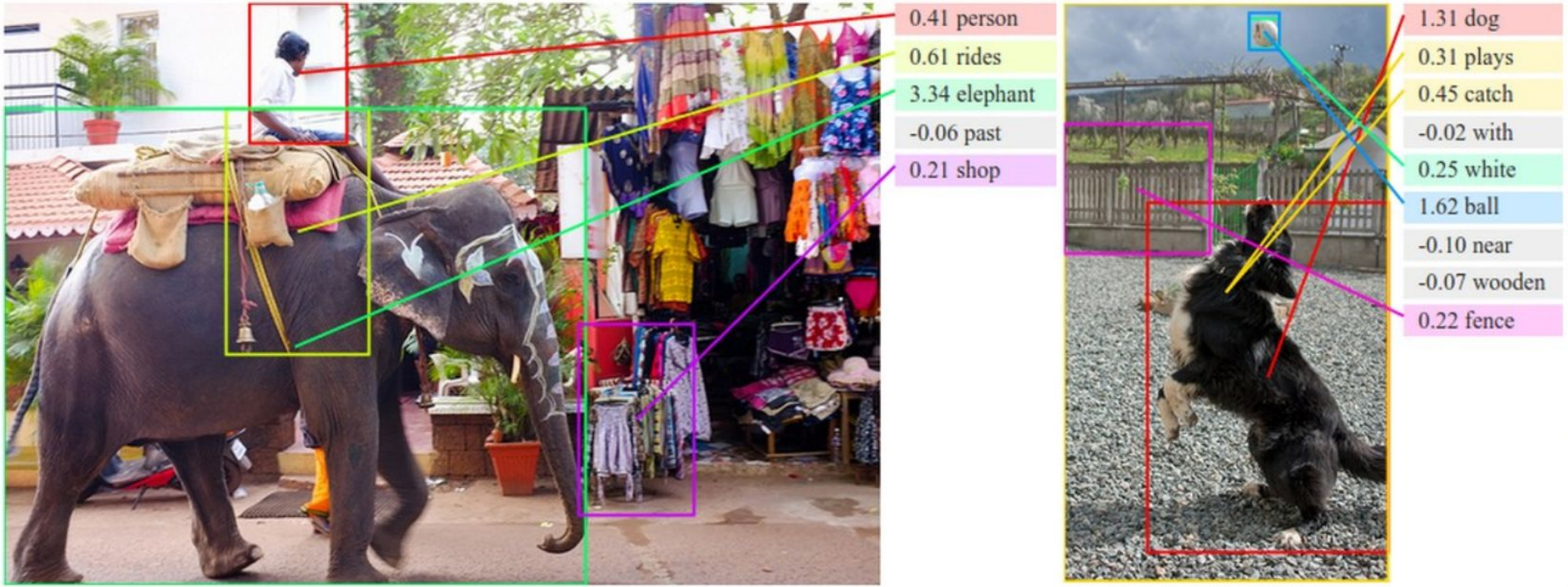
Generating Descriptions



Results (alignment)

Model	Image Annotation				Image Search			
	R@1	**R@5**	**R@10**	**Med *r***	**R@1**	**R@5**	**R@10**	**Med *r***
Flickr30K								
SDT-RNN (Socher et al. [49])	9.6	29.8	41.1	16	8.9	29.8	41.1	16
Kiros et al. [25]	14.8	39.2	50.9	10	11.8	34.0	46.3	13
Mao et al. [38]	18.4	40.2	50.9	10	12.6	31.2	41.5	16
Donahue et al. [8]	17.5	40.3	50.8	9	-	-	-	-
DeFrag (Karpathy et al. [24])	14.2	37.7	51.3	10	10.2	30.8	44.2	14
Our implementation of DeFrag [24]	19.2	44.5	58.0	6.0	12.9	35.4	47.5	10.8
Our model: DepTree edges	20.0	46.6	59.4	5.4	15.0	36.5	48.2	10.4
Our model: BRNN	**22.2**	**48.2**	**61.4**	**4.8**	**15.2**	**37.7**	**50.5**	**9.2**
Vinyals et al. [54] (more powerful CNN)	23	-	63	5	17	-	57	8
MSCOCO								
Our model: 1K test images	38.4	69.9	80.5	1.0	27.4	60.2	74.8	3.0
Our model: 5K test images	16.5	39.2	52.0	9.0	10.7	29.6	42.2	14.0

Results (alignment)



Results (full image captioning)

man in black shirt is playing guitar.

construction worker in orange safety vest is working on road.

two young girls are playing with lego toy.

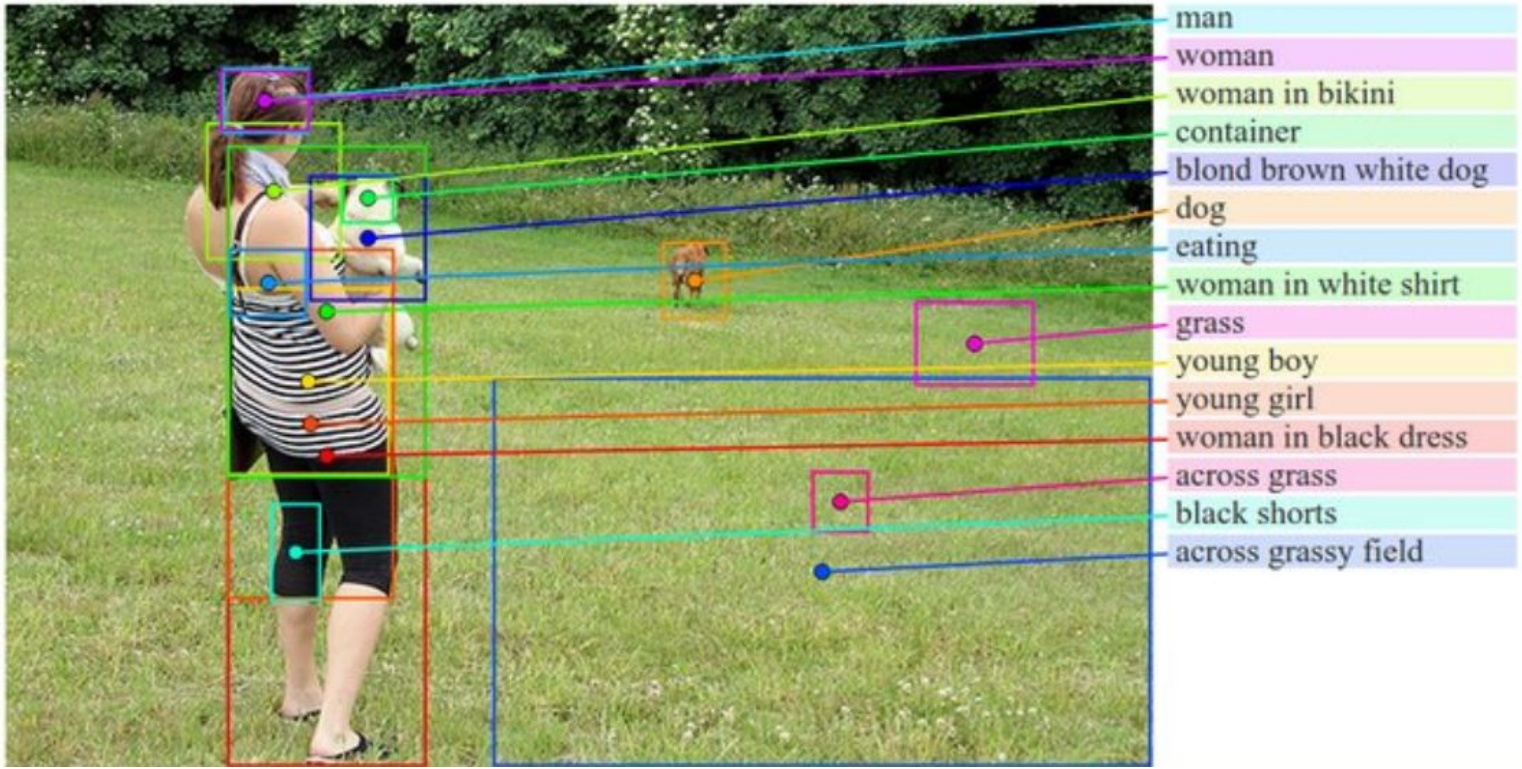
boy is doing backflip on wakeboard.

Results (dense captioning)

Model	B-1	B-2	B-3	B-4
Human agreement	61.5	45.2	30.1	22.0
Nearest Neighbor	22.9	10.5	0.0	0.0
RNN: Fullframe model	14.2	6.0	2.2	0.0
RNN: Region level model	**35.2**	**23.0**	**16.1**	**14.8**

BLEU scores for fullframe vs. region level models

Results (dense captioning)



Key Points + Thoughts

- Limitation: uses two separate models, one for image-sentence alignment and one for prediction
- Important for dense captioning and non-restrictive description generation

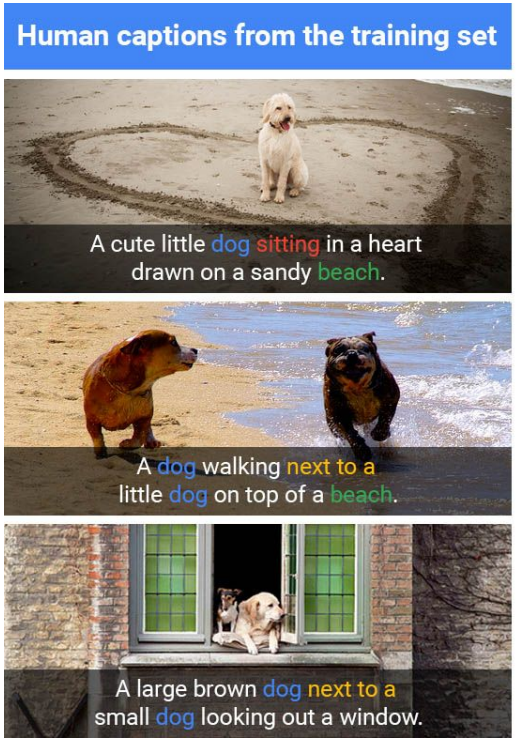
DenseCap: Fully Convolutional Localization Networks for Dense Computing

Justin Johnson*, Andrej Karpathy*, Li Fei-Fei
*Indicates Equal Contribution

[Paper]

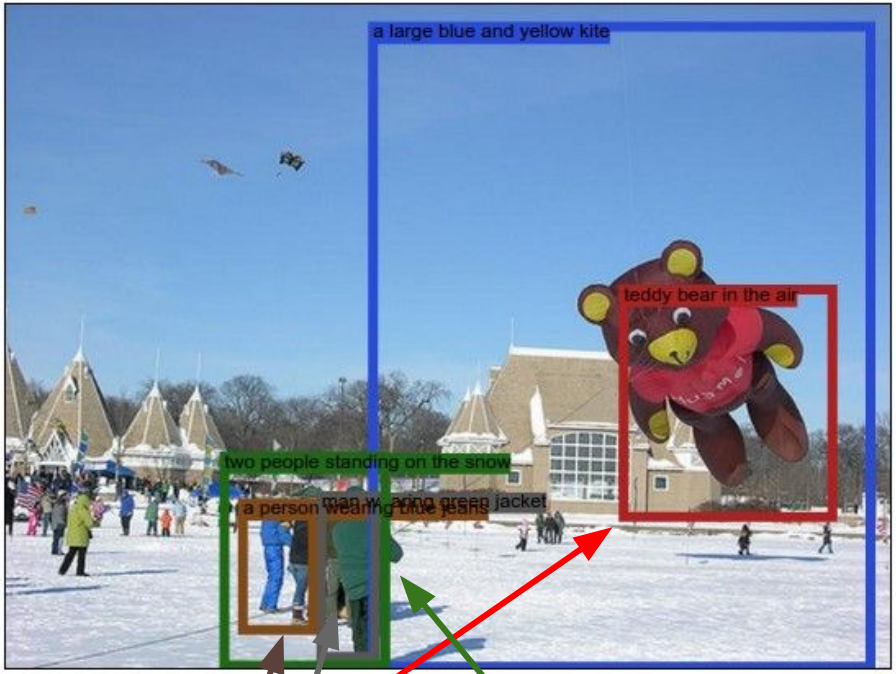
Recognition Tasks

Captioning

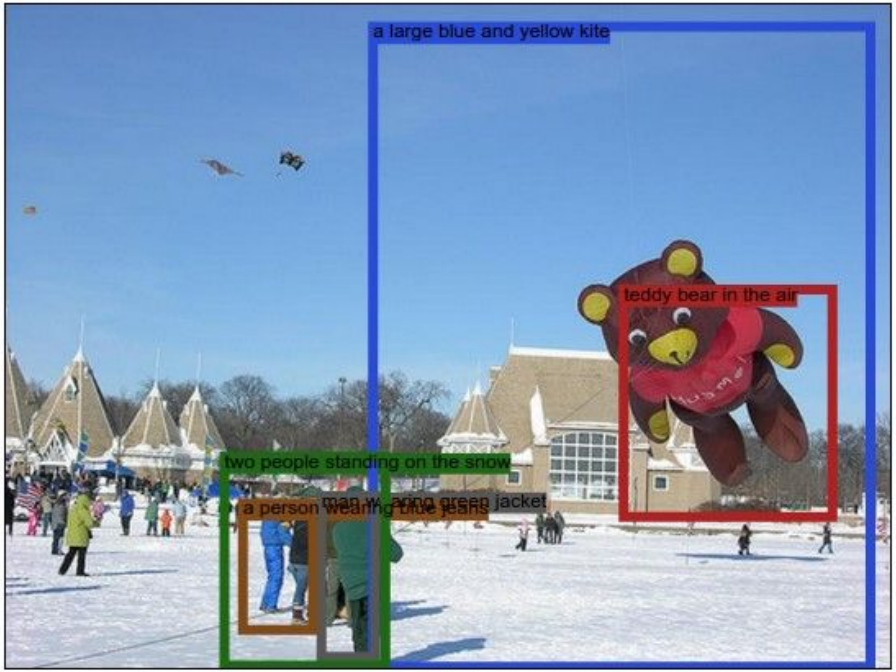


Google Research
https://research.googleblog.com/2016/09/show-and-tell-image-captioning-open.html

Dense Captioning



Dense Captioning



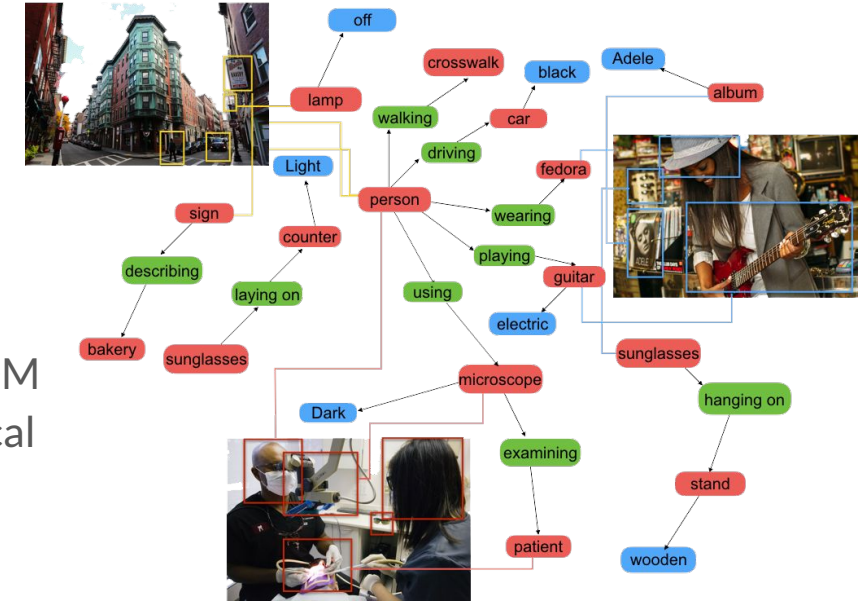
teddy bear in the air. a large blue and yellow kite. man wearing green jacket. two people standing on the snow. a person wearing blue jeans.

Key Developments

- Develop Fully Convolutional Localization Network (FCLN) to aid in dense captioning task.
- Introduce a new dense localization layer to predict a set of regions of interest in a given image.
    - Uses bilinear interpolation to crop activations into requested size.
- Can be trained end-to-end, and saves numerous recomputation when compared to other methods.

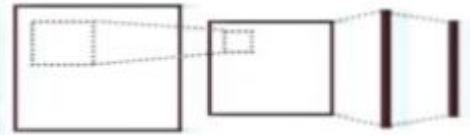
Training Data

- Visual Genome
  - 94,000 images and 4,100,000 region captions
  - Roughly 43.5 region captions per image
- Images derived from MS COCO and YFCC100M
- Annotations collected from Amazon Mechanical Turk
- 10,497 words in vocabulary
- Each image contains between 20 and 50 annotations

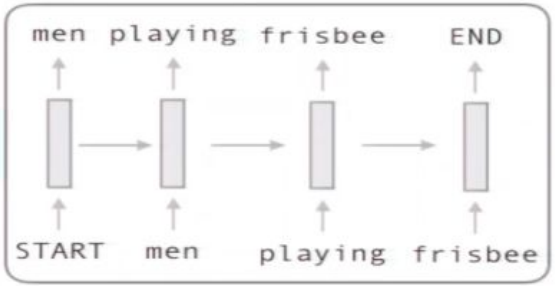


http://visualgenome.org/

Related Work



Convolutional Network

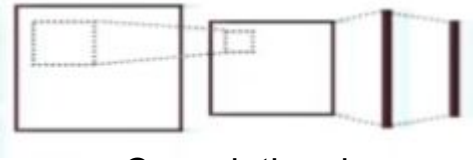


RNN

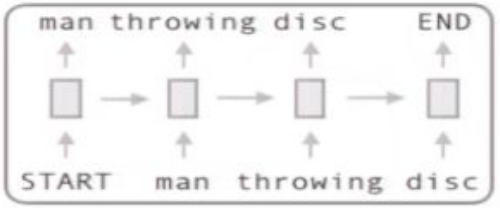
Related Work

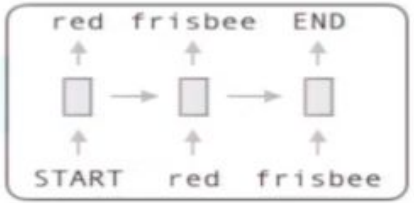
Region Proposals

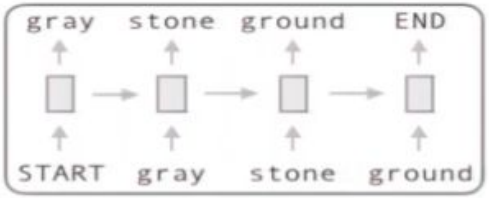
Crop



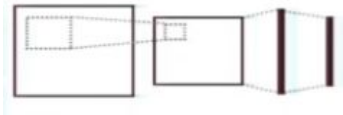
Convolutional Network

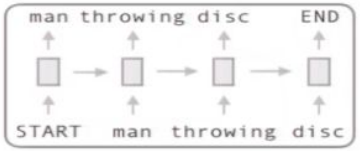


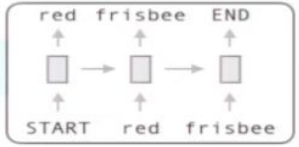


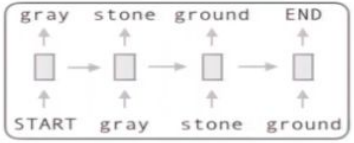


RNN









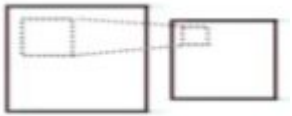
Problems

- No context -- Prediction has no context outside of region.
- Inefficient -- Redundant computations, each region is forwarded separately.
- Not End-To-End -- Use of external region proposals reduces efficiency significantly.

Approach

- End-To-End Learning -- Formulate a single differentiable function from inputs to outputs
- Localize regions of interest first after convolution, then use model to describe each region using natural language.

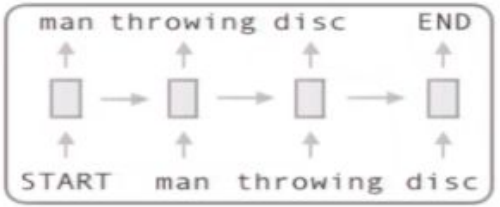
Region Proposals

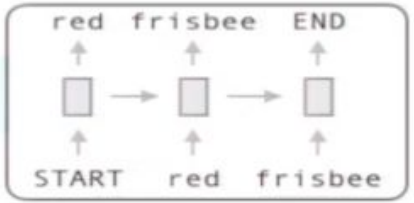


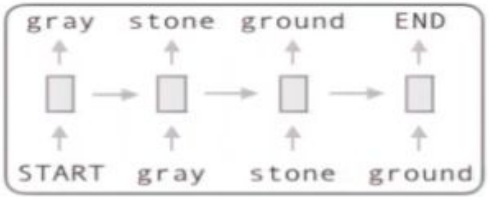
Convolutional Network

Crop (Bilinear Interpolation)

Recognition Network

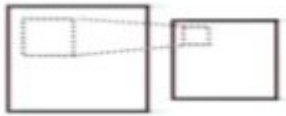






RNN

Region Proposals

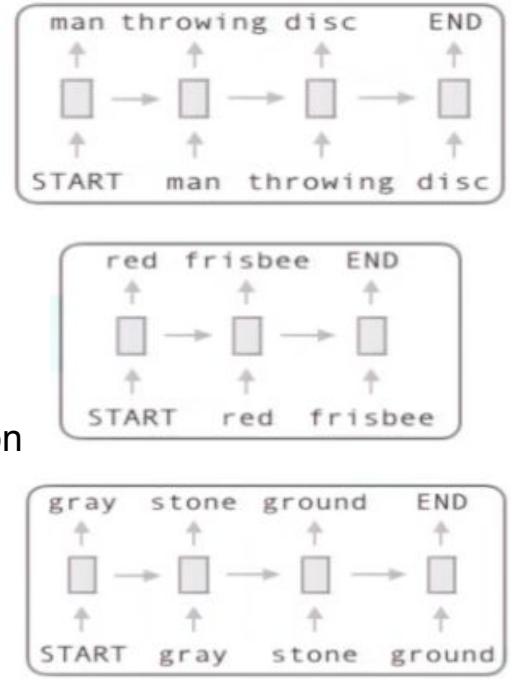


Convolutional Network

Localization Layer

Crop (Bilinear Interpolation)

Recognition Network



RNN

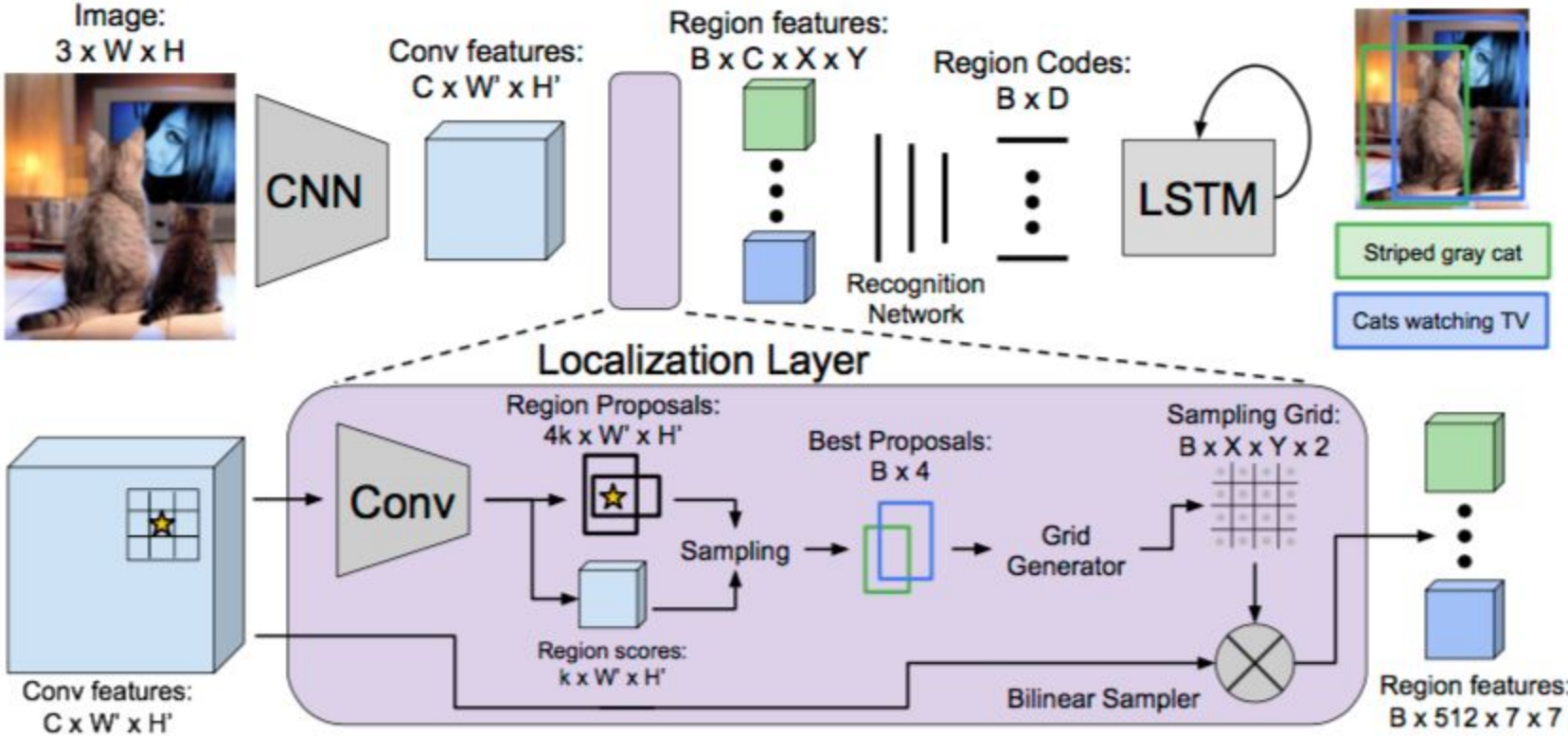


Image:
3 x W x H
CNN
Conv features:
C x W' x H'
Region features:
B x C x X x Y
Region Codes:
B x D
Recognition Network
LSTM
Striped gray cat
Cats watching TV
Localization Layer
Conv features:
C x W' x H'
Conv
Region Proposals:
4k x W' x H'
Region scores:
k x W' x H'
Sampling
Best Proposals:
B x 4
Grid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

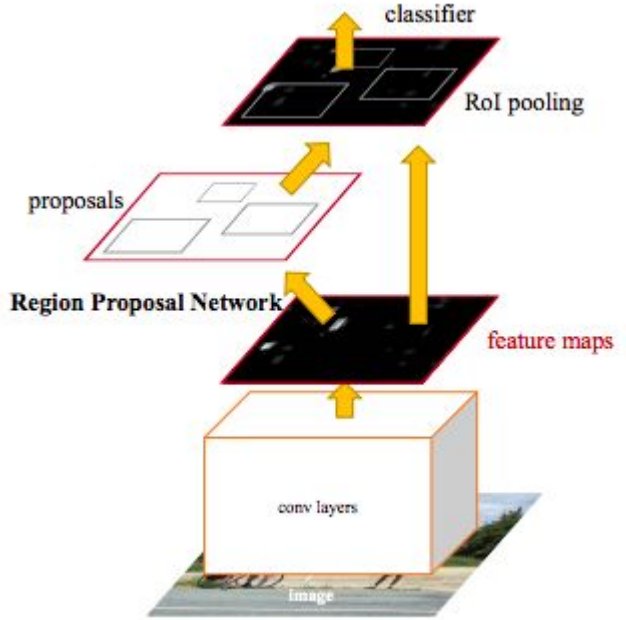
Convolution Network

- VGG-16 network for state-of-the-art performance in converting input images to tensor feature matrix.
- 13 layers of 3 × 3 convolutions interspersed with 5 layers of 2 × 2 max pooling.
- Input of 3 × W × H with output tensor features of 512 × W/16 × H/16.

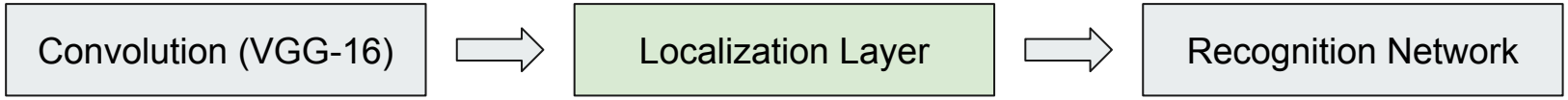
Convolution (VGG-16) ⇨ Localization Layer ⇨ Recognition Network

Faster R-CNN: Region Proposal Networks

- N × N sliding window across generated feature map.
- Anchor Boxes
  - (x_a, y_a, w_a, h_a)
- Predict for *k*-proposals (translation invariant)
  - $k \times (t_x, t_y, t_w, t_h)$
    - $x = x_a + t_x w_a$
    - $w = w_a \exp(t_w)$
- Proposals passed to box-regression and box-classification layer



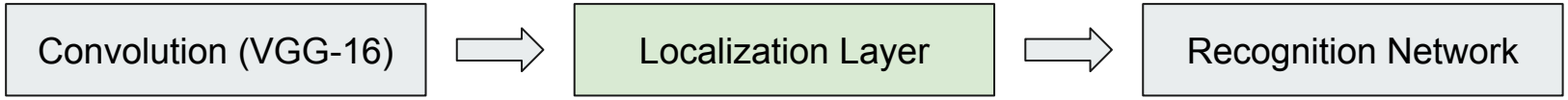
http://web.eng.tau.ac.il/deep_learn/wp-content/uploads/2017/01/YOLO_pre.pdf



Anchor Generation

- Similar to Faster R-CNN Region Proposal Networks
- Project each point in the W/16 × H/16 grid of input features back into the W × H image plane
- Generate *k* anchor boxes and predict a confidence score for each and four scalars regressing from the anchor to the predicted box coordinates.
- For a given anchor box (x_a, y_a, w_a, h_a), predict (t_x, t_y, t_w, t_h) to generate

$$x = x_a + t_x w_a \qquad y = y_a + t_y h_a$$
$$w = w_a \exp(t_w) \qquad h = h_a \exp(h_w)$$



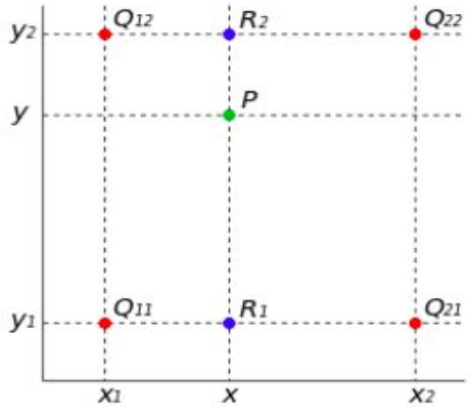
Fully Convolutional Localization Layer

- Accepts tensor of activations of size C × W/16 × H/16.
- Selects *B* regions of interest, returning output tensors:
  - Region Coordinates: A matrix of shape B × 4 giving bounding box coordinates for each output region.
  - Region Scores: A vector of length B giving a confidence score for each output region. Regions with high confidence scores are more likely to correspond to ground-truth regions of interest.
  - Region Features: A tensor of shape B × C × X × Y giving features for output regions; is represented by an X × Y grid of C-dimensional features.

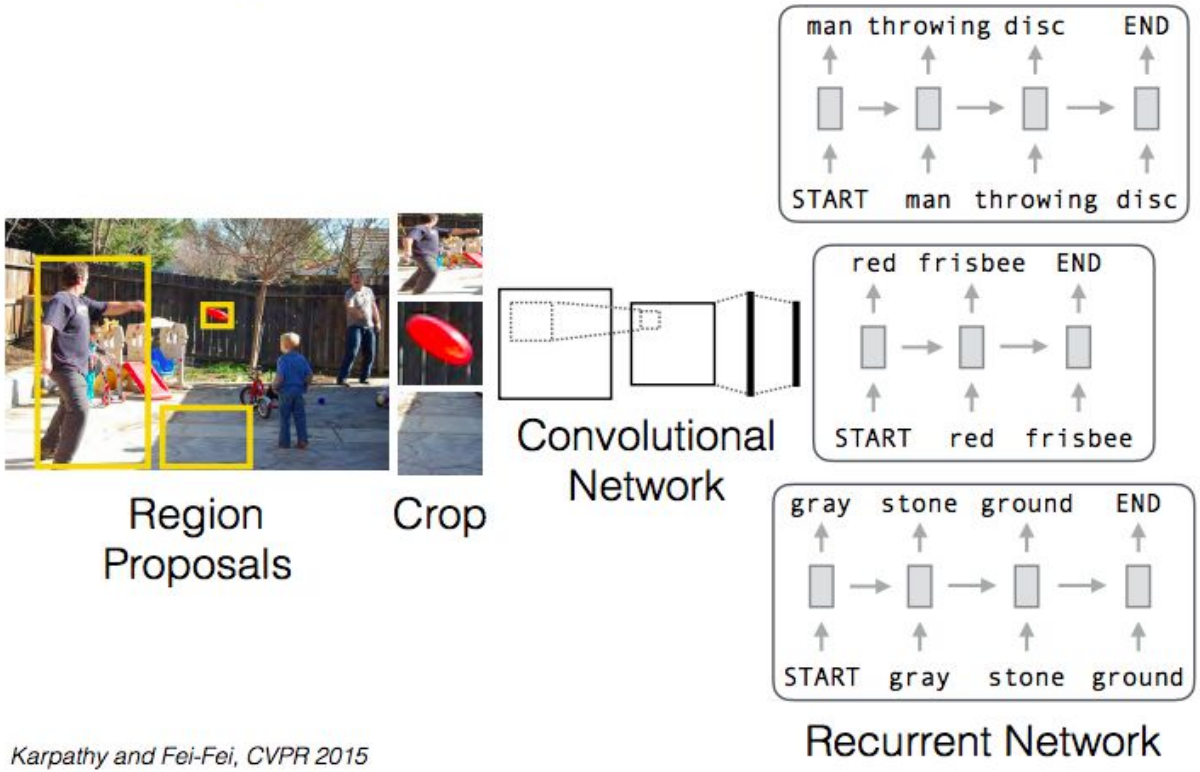
Bilinear interpolation

- Allows for backpropagation of error to previous layers.

$$f(x,y) = \frac{1}{(x_2-x_1)(y_2-y_1)}\left(f(Q_{11})(x_2-x)(y_2-y) + f(Q_{21})(x-x_1)(y_2-y) + f(Q_{12})(x_2-x)(y-y_1) + f(Q_{22})(x-x_1)(y-y_1)\right)$$

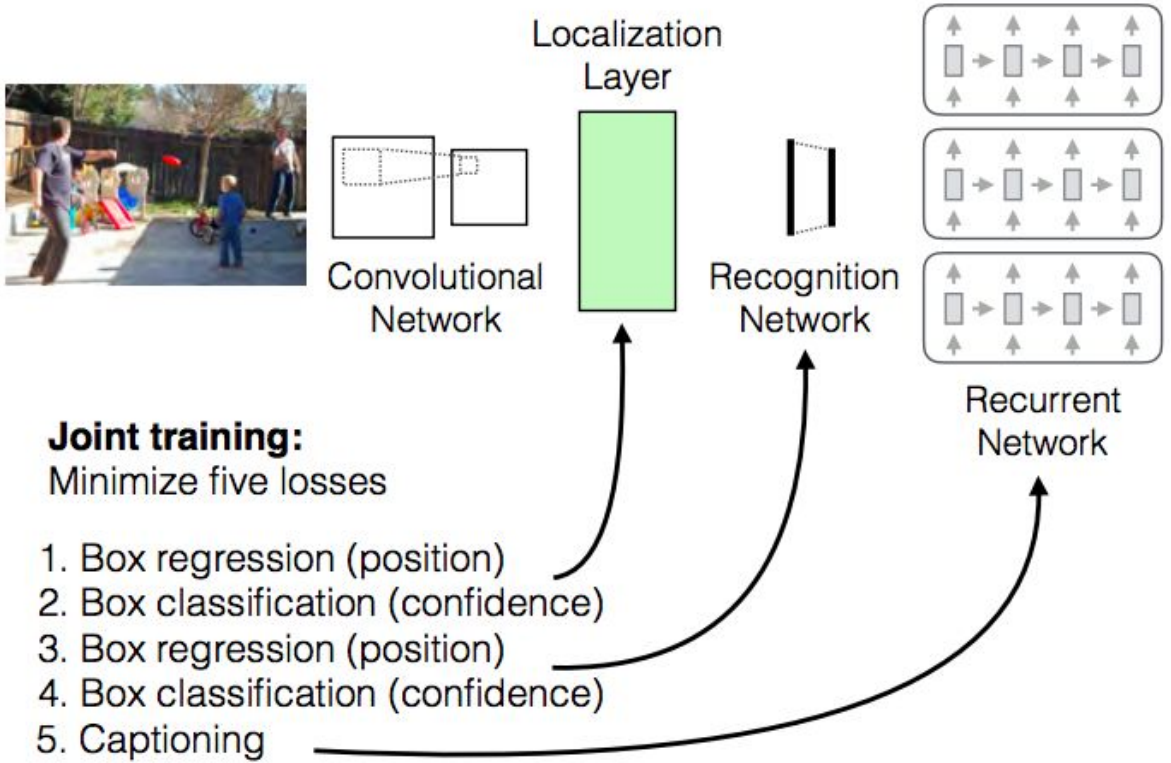


Recognition network



Karpathy and Fei-Fei, CVPR 2015

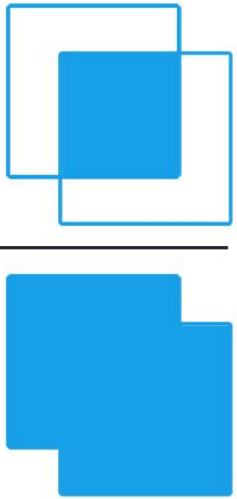
Loss function



Evaluation

- Use intersection over union (IoU) to measure localization accuracy for each dense caption box
- Use METEOR to measure natural language outputs compared to baseline
  - Metric was found to be most highly correlated with human judgments in settings with a low number of references

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



Results (Dense Captioning)

Region source	Language (METEOR)			Dense captioning (AP)			Test runtime (ms)			
	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**

- Outperforms all other models by far for individual regional description.
- Much faster in runtime when compared to other models.

Image Retrieval

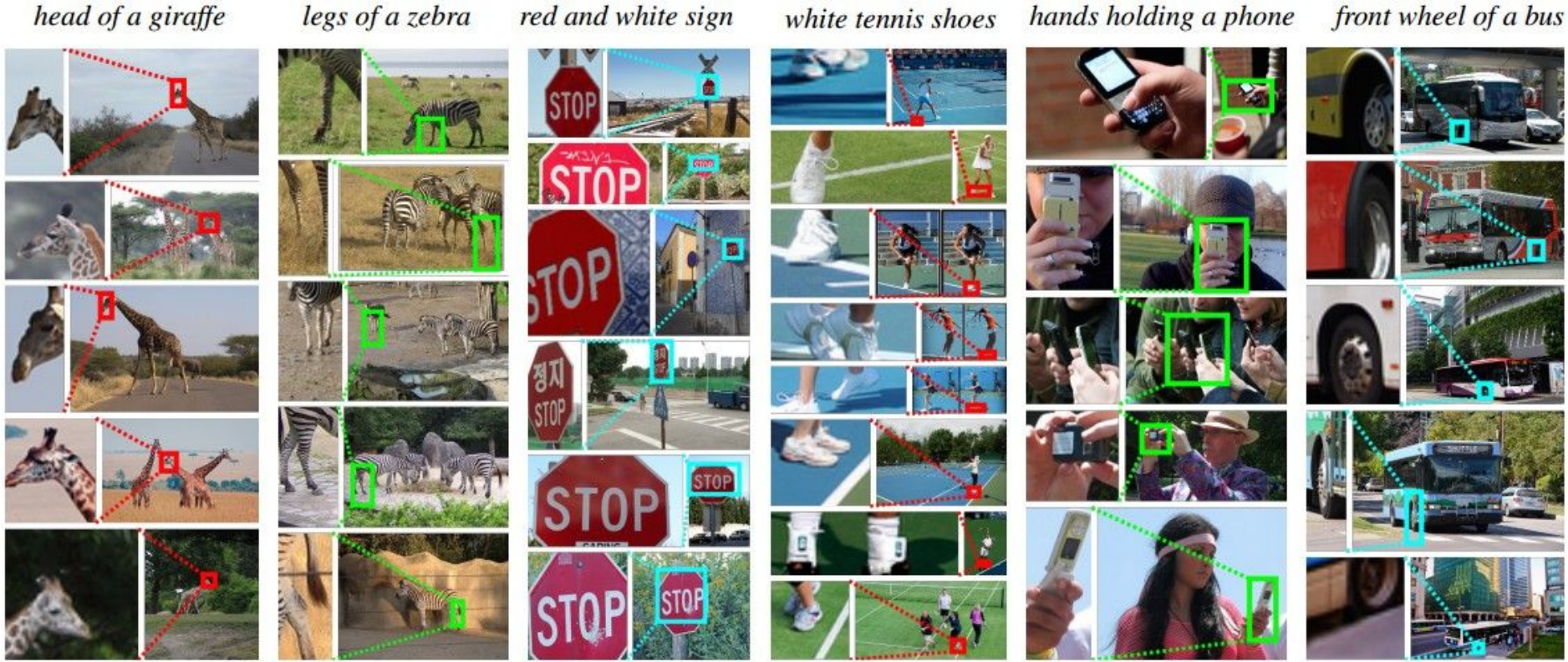
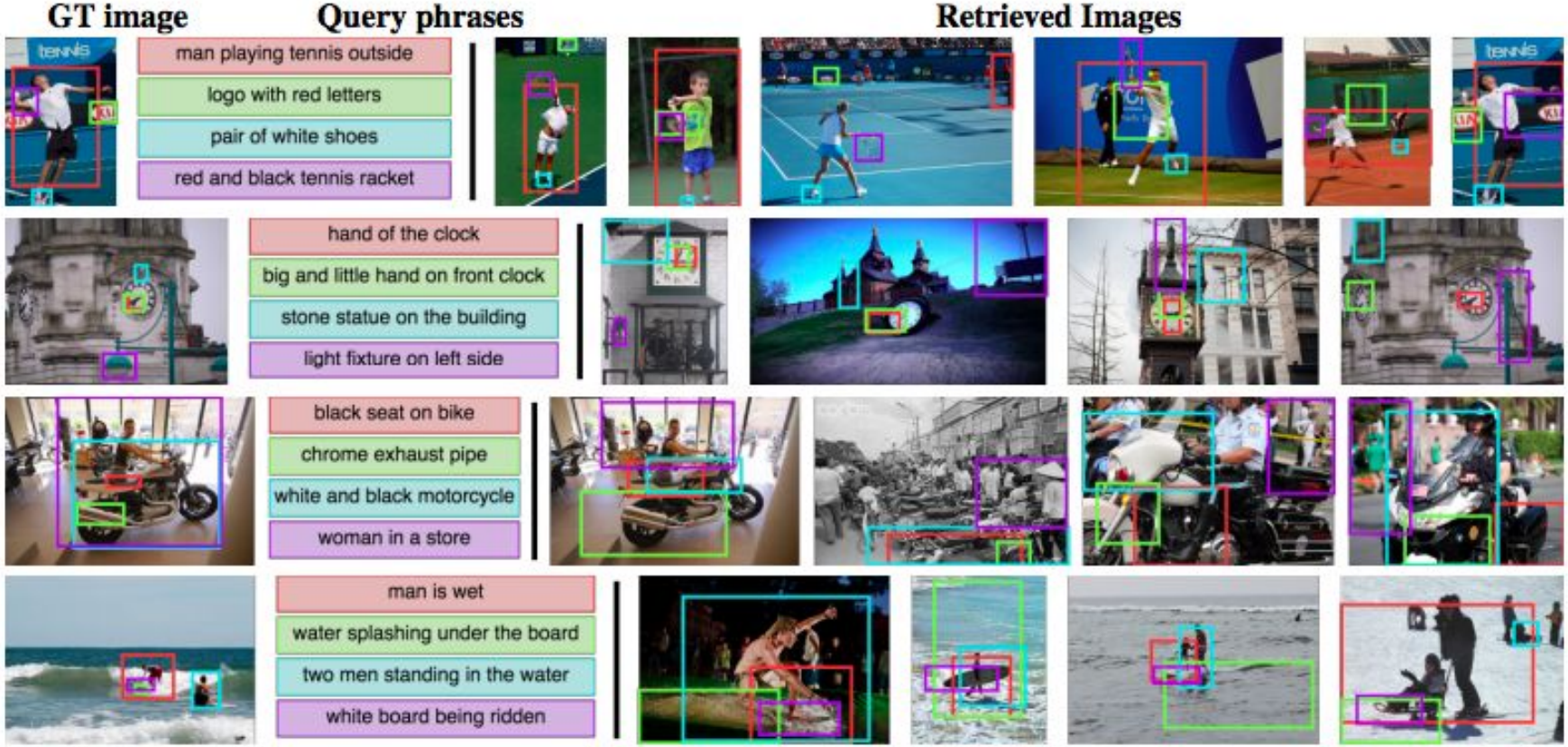


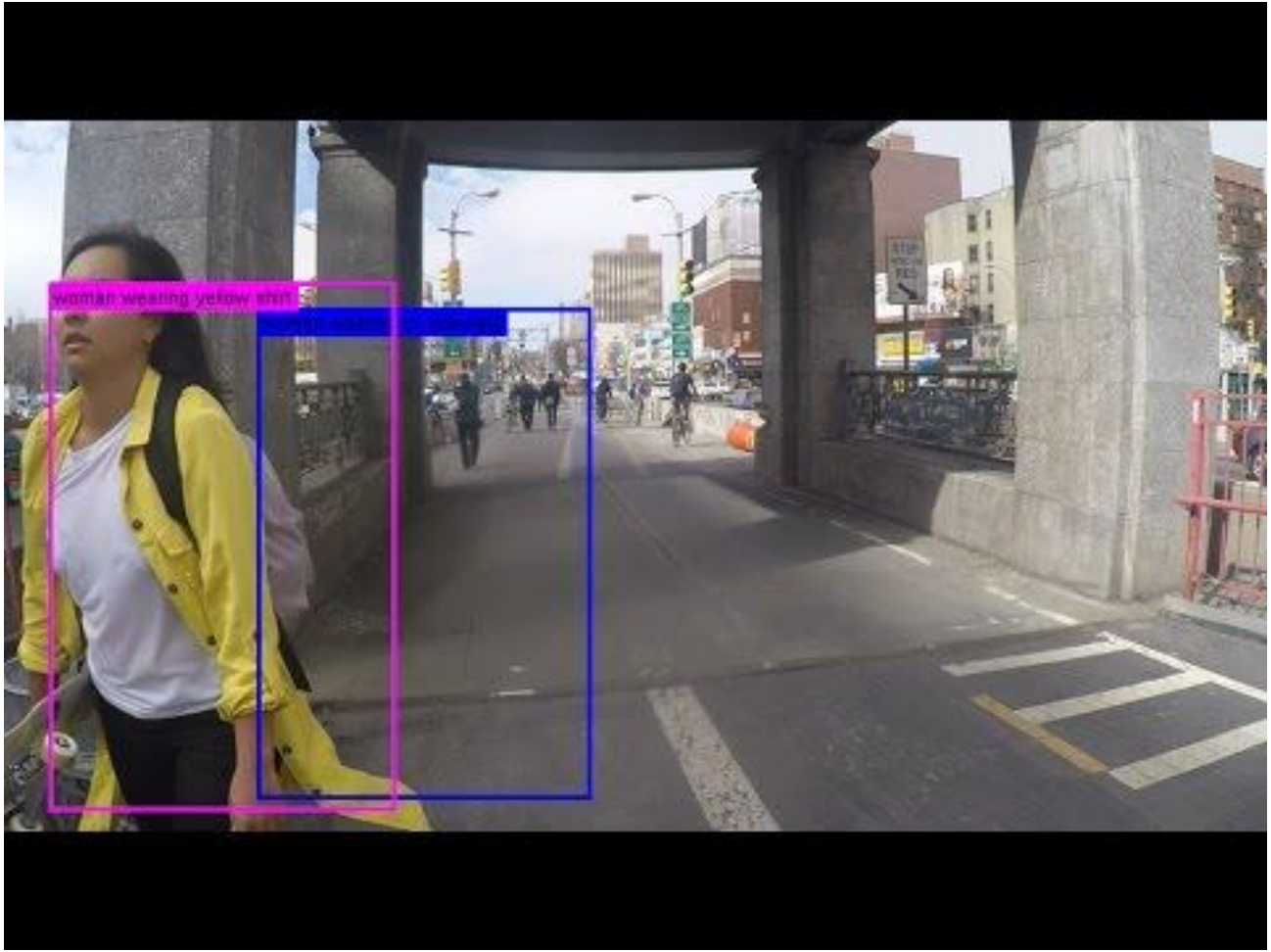
Image Retrieval



Results (Image Retrieval)

	Ranking				Localization			
	R@1	R@5	R@10	Med. rank	IoU@0.1	IoU@0.3	IoU@0.5	Med. IoU
Full Image RNN [21]	0.10	0.30	0.43	13	-	-	-	-
EB + Full Image RNN [21]	0.11	0.40	0.55	9	0.348	0.156	0.053	0.020
Region RNN [21]	0.18	0.43	0.59	7	0.460	0.273	0.108	0.077
Our model (FCLN)	**0.27**	**0.53**	**0.67**	**5**	**0.560**	**0.345**	**0.153**	**0.137**

- Performs much better than most other models when doing natural language queries to retrieve images.



[Link]

<END>