Brain Unleashing Series -

Beyond R-CNN detection: Learning to Merge Contextual Attribute

Shu Kong

CS, ICS, UCI

2015-1-29

Outline

1. RCNN is essentially doing classification, without considering contextual information of the object,
2. Our brains exploit the contextual information in action classification.
3. Semantic attributes of texture as contextual info,
4. Patch Match to model context - 3D chair detection,
5. Unleash the mind - can we learn to merge them?

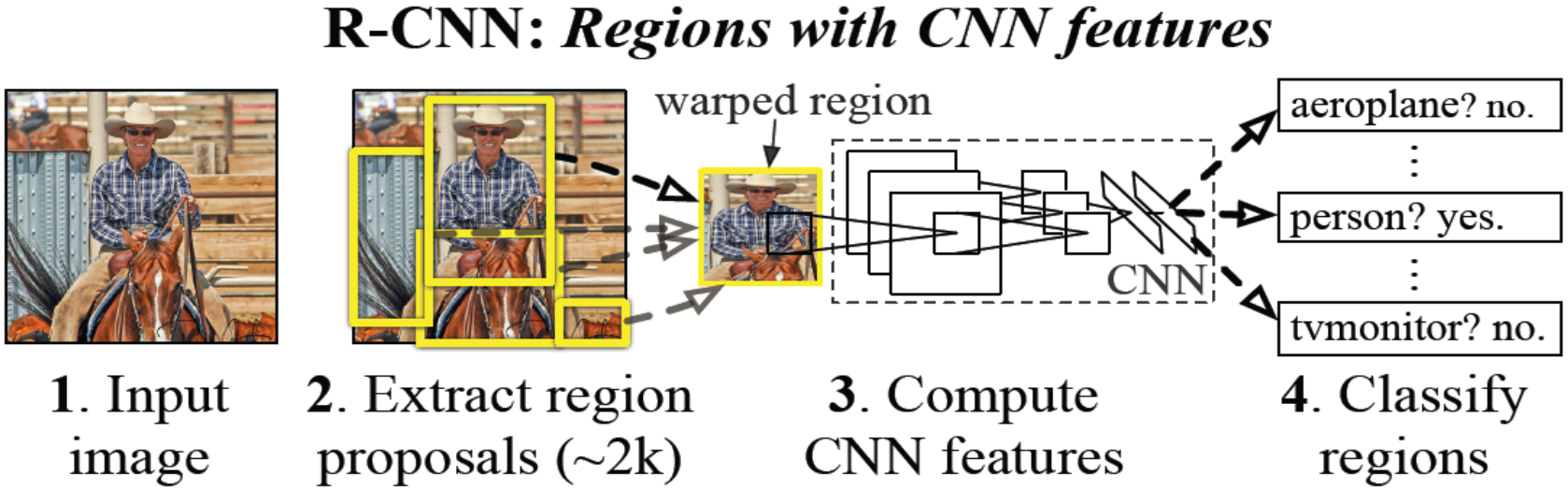
1. seeing is worse than believing – reading people's minds better than computer-vision methods recognize actions, eccv2014

2. describing Textures in the Wild, cvpr2014

3. Rich feature hierarchies for accurate object detection and semantic segmentation, cvpr2014

4. Seeing 3D chairs – exemplar part-based 2D-3D alignment using a large dataset of CAD models, CVPR2014

1. R-CNN



- R-CNN for detection is a successful application of CNN
- But it does not consider contextual information.
- Other people exploit contextual info and get better performance in LSVRC2014
  - For example, *CNN output of the whole image as a contextual information, or contextual SVM (by NUS)*

http://www.image-net.org/challenges/LSVRC/2014/results

Rich feature hierarchies for accurate object detection and semantic segmentation, cvpr2014

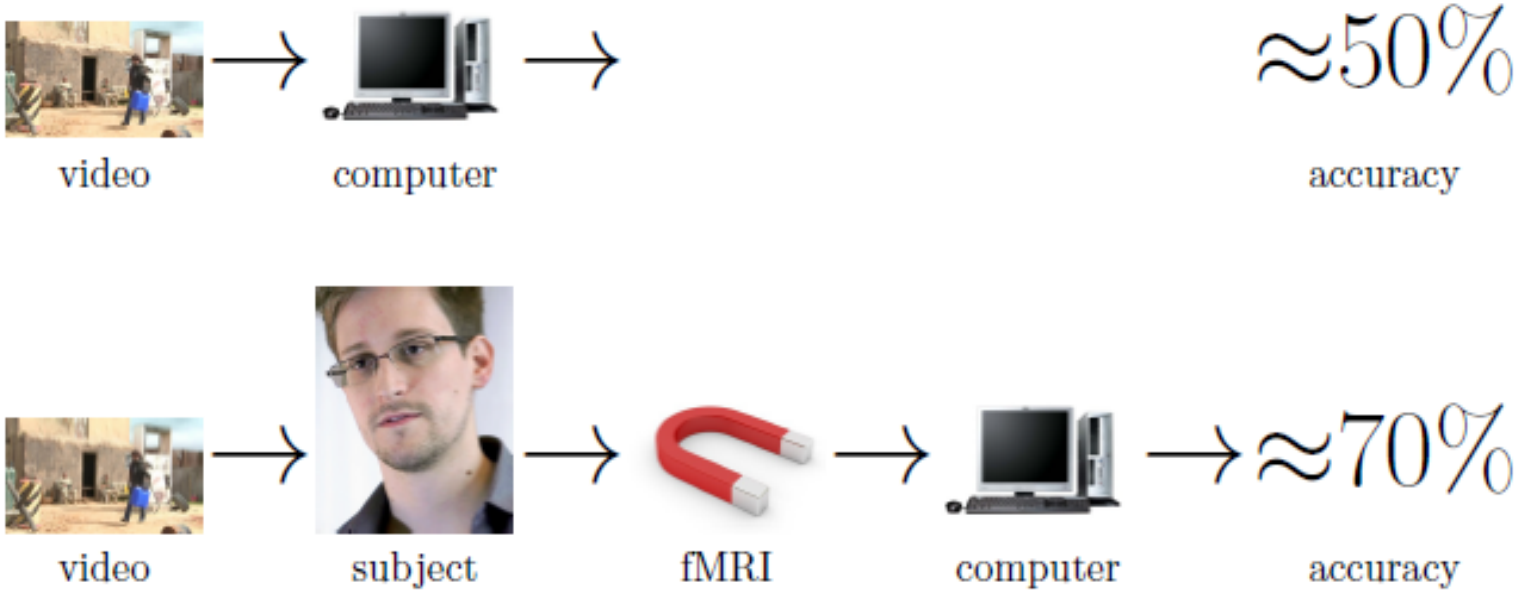
2. Action classificaiton: fMRI vs video (1)

An action dataset was collected for this study. It consists of six classes -- quite small in number, yet difficult to classification.

Human subjects perform classification while undergoing functional Magnetic Resonance Imaging (fMRI) -- geting fMRI data.

seeing is worse than believing - reading people's minds better than computer-vision methods recognize actions, eccv2014

2. Action classificaiton: fMRI vs video (2)



Keypoints in experiment settings

1. training and testing within subject and cross subject
2. Linear SVM as the classifier

seeing is worse than believing - reading people's minds better than computer-vision methods recognize actions, eccv2014

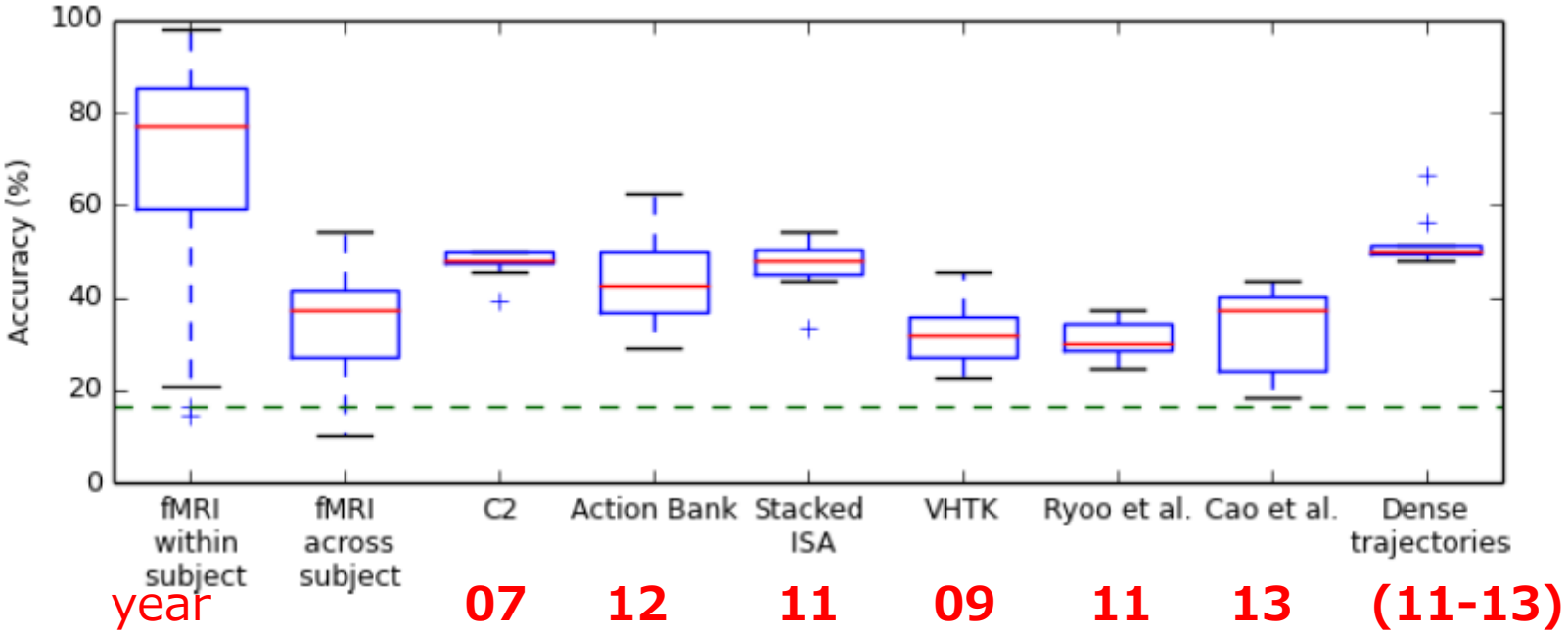
2. Action classificaiton: fMRI vs video (3)

analysis	subject	mean	stddev
fMRI within subject	mean	0.6973	
	stddev		0.2171
fMRI across subject	mean	0.3480	
	stddev		0.1068
C2 [12]		0.4740	0.0348
Action Bank [24]		0.4427	0.1112
Stacked ISA [16]		0.4688	0.0649
VHTK [18]		0.3255	0.0721
Ryoo's method* [23]		0.3125	0.0459
Cao's method [2]		0.3333	0.0964
Dense Trajectories [27–29]		0.5234	0.0634

1. The cross-subject average classification accuracy is lower than the within-subject average classification accuracy. This is because there is significant cross-subject anatomical variation.

seeing is worse than believing - reading people's minds better than computer-vision methods recognize actions, eccv2014

2. Action classificaiton: fMRI vs video (4)

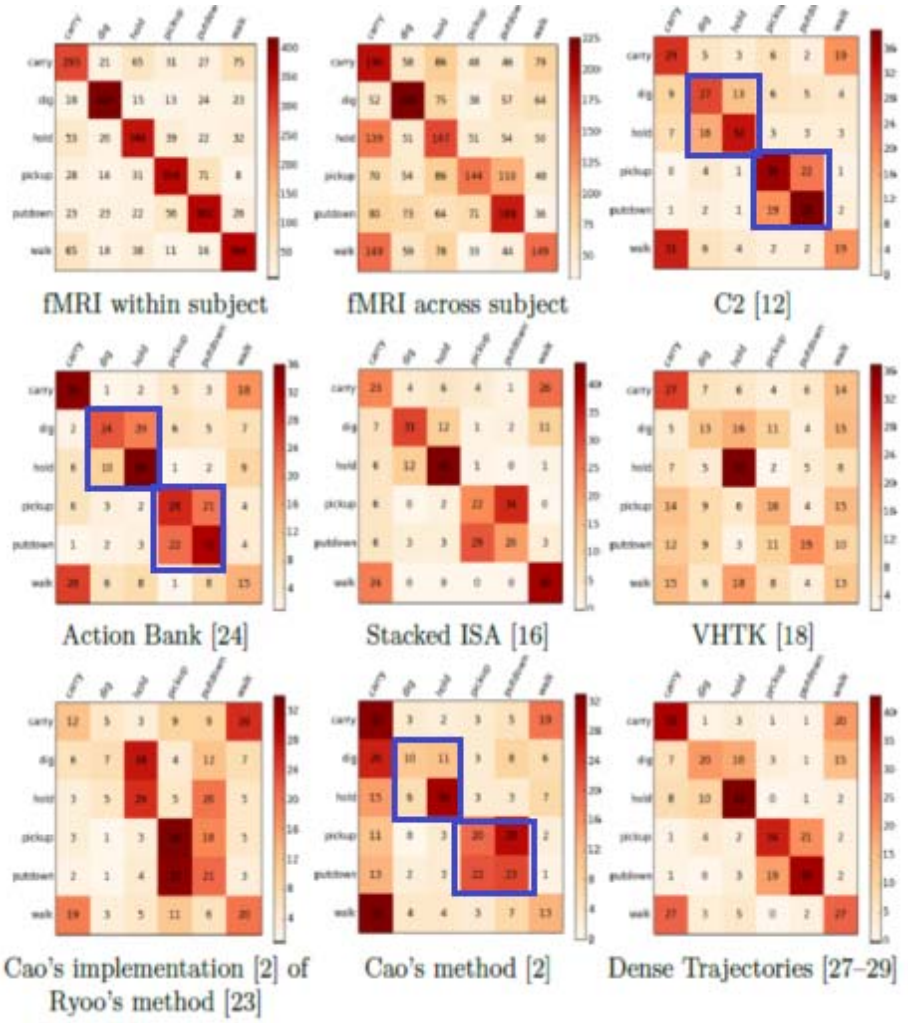


2. The newer methods perform worse than the older ones; it shows that the field is basically not progressing.

seeing is worse than believing - reading people's minds better than computer-vision methods recognize actions, eccv2014

2. Action classificaiton: fMRI vs video (5)

- CV methods confuse *carry:wall hold:dig* -- it suggests that the semantics of human perceptior may play a role in action classification.
- CV methods don't employ contextual information, except local features at very short spatial and/or temporal scales.
- Brain confuses less *pick-up:put down, dig:hold*. This indicates that even cross subject, the fM results appear to be using a degree of semantic inference. The reduced accuracy is due more to issues of registration.



seeing is worse than believing - reading people's minds better than computer-vision methods recognize actions, eccv2014

3. Semantic Attributes for Texture (1)

1. Collecting data: 5640 images from 47 texture classes (*balanced*)
2. **Attribute: 47 SVMs to score based on the following representations:**
    a. **SIFT-IFV (off-the-shelf)**
    b. **L2-normalized DeCAF**
    c. **47 attributes as a 47-dim mid-level texture descriptor**
3. complementary role for material recognition

Describing Textures in the Wild, cvpr2014

3. Semantic Attributes for Texture (2)

Attribute: 47 SVMs to score based on the following representations:

a) **SIFT-IFV (off-the-shelf)**

b) **L2-normalized DeCAF**

c) **47 attributes as a 47-dim mid-level texture descriptor** achieved by (a) and (c)

Therefore, image representation fed into the classifier can be various combinations of the three feature types above.

As the attribute presentation is only 47 dim, RBF classifier is really cheap.

3. Semantic Attributes for Texture (3)

Local descr.	Kernel Linear	Hellinger	add-χ^2	exp-χ^2
MR8	15.9±0.8	19.7 ± 0.8	24.1 ± 0.7	30.7 ± 0.7
LM	18.8 ± 0.5	25.8 ± 0.8	31.6 ± 1.1	39.7 ± 1.1
$\text{Patch}_{3\times3}$	14.6 ± 0.6	22.3 ± 0.7	26.0 ± 0.8	30.7 ± 0.9
$\text{Patch}_{7\times7}$	18.0 ± 0.4	26.8 ± 0.7	31.6 ± 0.8	37.1 ± 1.0
LBP^u	8.2 ± 0.4	9.4 ± 0.4	14.2 ± 0.6	24.8 ± 1.0
LBP-VQ	21.1 ± 0.8	23.1 ± 1.0	28.5 ± 1.0	34.7 ± 1.3
SIFT	**34.7 ± 0.8**	**45.5 ± 0.9**	**49.7 ± 0.8**	**53.8 ± 0.8**

Table 1: Comparison of local descriptors and kernels on the DTD data, averaged over ten splits.

Conclusion:

Study of low-leve features demonstrates SIFT is the best for this task.

Describing Textures in the Wild, cvpr2014

3. Semantic Attributes for Texture (4)

Dataset			SIFT					DeCAF	IFV + DeCAF	Previous Best
Source	Splits	Metric	IFV	BoVW	VLAD	LLC	KCB			
CUReT	20	acc.	**99.5**±0.4	98.1±0.9	98.8±0.6	97.1±0.4	97.7±0.6	97.9±0.4	**99.8**±0.1	99.4±n/a [36]
UMD	20	acc.	99.2±0.4	98.1±0.8	99.3±0.4	98.4±0.7	98.0±0.9	96.4±0.7	**99.5**±0.3	**99.7**±0.3 [34]
UIUC	20	acc.	97.0±0.9	96.1±2.4	96.5±1.8	96.3±0.1	91.4±1.4	94.2±1.1	**99.0**±0.5	**99.4**±0.4 [34]
KT	20	acc.	**99.7**±0.1	98.6±1.0	99.2±0.8	98.1±0.8	98.5±0.8	96.9±0.9	**99.8**±0.2	**99.4**±0.4 [34]
KT-2a$^{\alpha}$	4	acc.	**82.2**±4.6	**74.8**±5.4	**76.5**±5.2	**75.7**±5.6	**72.3**±4.5	**78.4**±2.0	**84.7**±1.5	73.0±4.7 [33]
KT-2b$^{\beta}$	4	acc.	**69.3**±1.0	58.4±2.2	63.1±2.1	57.6±2.3	58.3±2.2	**70.7**±1.6	**76.2**±3.1	66.3 [36]
FMD	14	acc.	**58.2**±1.7	49.5±1.9	52.6±1.5	50.4±1.6	45.1±1.9	**60.7**±2.0	**65.5**±1.3	57.1 $^{\gamma}$[31]
DTD	10	acc.	**61.2**±1.0	55.5±1.1	59.7±1.1	54.7±1.1	53.2±1.6	54.8±0.9	**66.7**±0.9	–
DTD	10	mAP	**63.5**±1.0	54.9±0.9	61.3±0.8	54.3±1.0	52.5±1.3	55.0±1.1	**69.4**±1.2	–
DTD-J$^{\delta}$	10	mAP	**63.5**±0.9	56.1±0.8	61.1±0.7	54.8±1.0	53.2±0.8	48.9±1.1	**68.9**±0.9	–

Conclusion:

1. IFV is better than other encoding methods
2. DeCAF is good enough
3. IFV+DeCAF is the best

Describing Textures in the Wild, cvpr2014

3. Semantic Attributes for Texture (5)

Conclusion:

Attributes is a complementary role for material recognition

Combination of DeCAF, IFV and attributes gives the best results.

CNN can do attribute learning, being used as complementary feature.

Feature	KTH-2b	FMD
DTD_{LIN}^{IFV}	62.9±3.8	49.8±1.3
DTD_{RBF}^{IFV}	66.0±4.3	52.4±1.3
$DTD_{LIN}^{IFV+DeCAF}$	71.2±0.6	55.9±2.3
$DTD_{RBF}^{IFV+DeCAF}$	72.0±0.5	58.0±1.8
$DTD_{RBF}^{IFV} + DTD_{RBF}^{IFV+DeCAF}$	73.8±1.3	61.1±1.4
DeCAF	70.7±1.6	60.7±2.1
IFV_{RGB}	58.8±2.5	47.0±2.7
$IFV_{SIFT} + IFV_{RGB}$	67.5±3.3	63.3±1.9
$DTD_{RBF}^{IFV} + IFV_{SIFT}$	70.2±2.4	60.1±1.6
$DTD_{RBF}^{IFV} + IFV_{RGB}$	70.9±3.5	61.3±2.0
Combined	74.6±3.0	65.4±2.0
$IFV_{SIFT} + DTD_{RBF}^{IFV}$	70.2±2.4	60.0±1.9
$IFV_{SIFT} + DTD_{RBF}^{IFV+DeCAF}$	75.6±1.8	65.5±1.2
$DeCAF + DTD_{RBF}^{IFV}$	75.4±1.8	64.6±1.6
$DeCAF + DTD_{RBF}^{IFV+DeCAF}$	73.7±1.8	64.1±1.5
IFV_{SIFT} +DeCAF + DTD_{RBF}^{IFV}	**77.3±2.3**	66.7±1.7
IFV_{SIFT} +DeCAF + $DTD_{RBF}^{IFV+DeCAF}$	76.4±2.8	66.9±1.6
Combined	**77.1±2.4**	**67.1±1.5**
Prev. best	66.3 [36]	57.1 [31]

Table 3: **DTD for material recognition.** Combined with IFV_{SIFT} and IFV_{RGB}, the DTD_{RBF}^{IFV} features achieve a significant improvement in classification performance on the challenging KTH-TIPS-2b and FMD compared to published state of the art results. See the text for the details on the notation and the methods.

Describing Textures in the Wild, cvpr2014

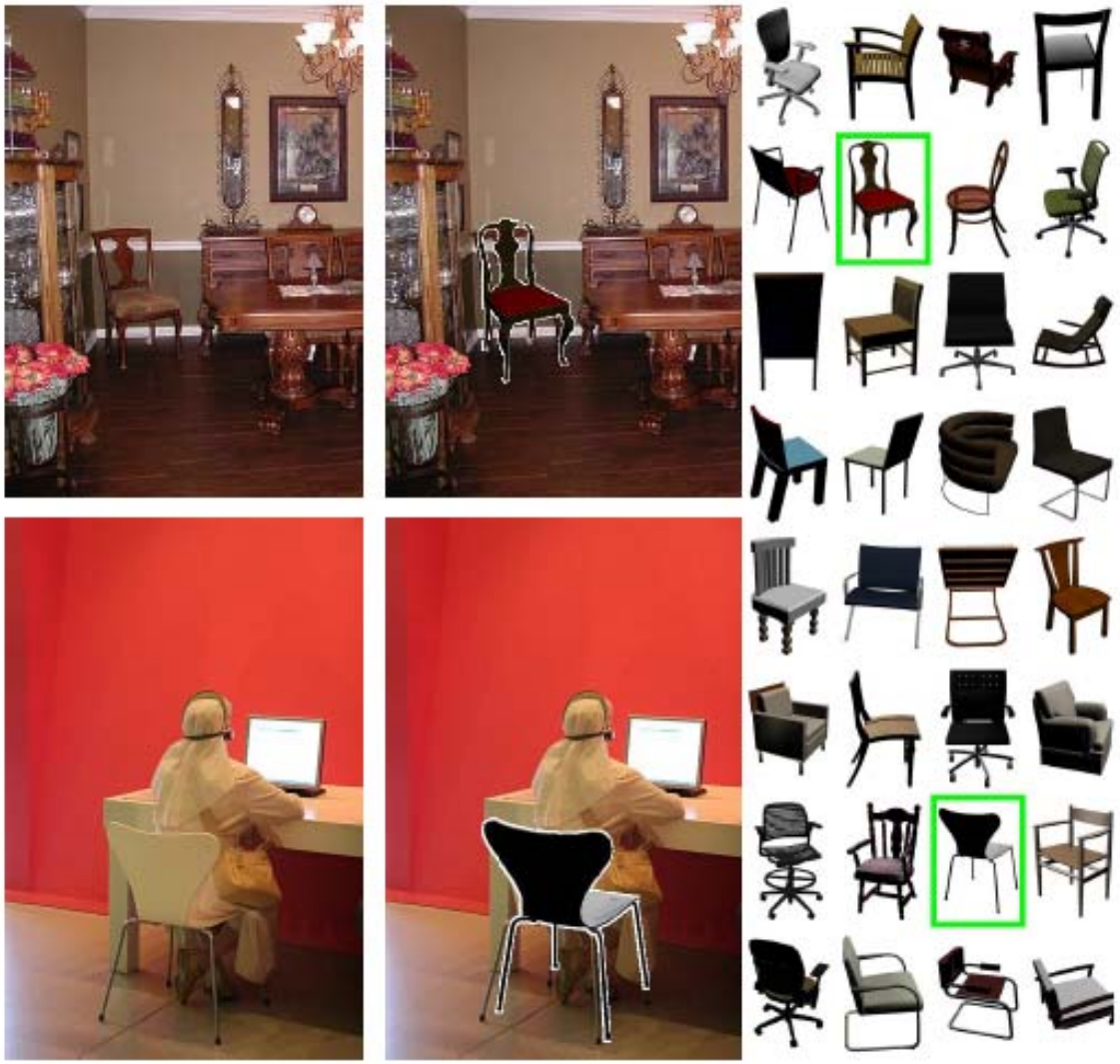
4. Patch Match for Detection (1)

Motivation

We need a tool to do deeper reasoning about the scene, e.g. "who's sitting on chair?"

Selling points

1. data collection
2. no training object model for fine detection
3. marriage between part-based discriminative models and exemplar-based matching



(a) Input images (b) Aligned output (c) 3D chair models

Seeing 3D chairs - exemplar part-based 2D-3D alignment using a large dataset of CAD models

4. Patch Match for Detection (2)

Data collection:

1393 chair styles

62 rendered/synthesized images for each,

1393*62 = 86366 in total

Seeing 3D chairs - exemplar part-based 2D-3D alignment using a large dataset of CAD models

4. Patch Match for Detection (3)

Building **Discriminative Visual Element Detectors**:

- HOG descriptor to represent patch
  - patch *q* in a rendered 3D view
  - find patch x^* in image to maximize the score:

$$S_q(x) = w_q^T x$$

  - using LDA to get w_q,

$$w_q = \Sigma^{-1}(q - \mu_n)$$

$$\mu_n = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_n)(x_i - \mu_n)^T$$

where a large set of HOG descriptors $\{x_i\}$ are negative training data sampled independently.

rewrite -- ranks patches using dot product between whitened q and x as the similarity measure

$$S_q(x) = \Phi(q)^T\Phi(x) - \Phi(q)^T\Phi(0) \qquad \Phi(x) = \Sigma^{-\frac{1}{2}}(x - \mu_n)$$

Seeing 3D chairs - exemplar part-based 2D-3D alignment using a large dataset of CAD models

4. Patch Match for Detection (3)

From millions of patches, select a few of most discriminative ones in each view and each style

Discriminative Patch Selection

1. densely compute squared whitened norm response at multiple spatial scales, $\|\Phi(q)\|^2$
2. select 10 visual elements after **non-maximum suppression** with interaction area to union ratio of 0.25,
3. another trick:

   They set to zero components of w_q corresponding to spatial bins with sum of the absolute value across the HOG channels less than 0.01. This effectively downweights the background (white) pixels in the rendered views and focuses the detector weights on the foreground chair.

4. Patch Match for Detection (4)

Visual Element Detector Calibrating

$$S'_q(x) = a_q\, S_q(x) + b_q$$

Calibration of matching scores across different visual elements is important for the quality of the final detection outputs.

Recover from two points by setting

$$S'_q(x_n) = 0 \qquad S'_q(\mu_n) = -1$$

where negative patch x_n yeilds a false positive rate of 0.01%, μ_n is the mean HOG feature vector.

This is a good compromise between representing the tail of the score distribution and the amount of time to scan the negative data.

4. Patch Match for Detection (5)

Matching visual element spatial configuration

star model, multi-scale, NMS

starting from the most confident detection

...(other details)

Q:

Is it easier to use Patch Match to learn background attributes?

5. Ask ourselves

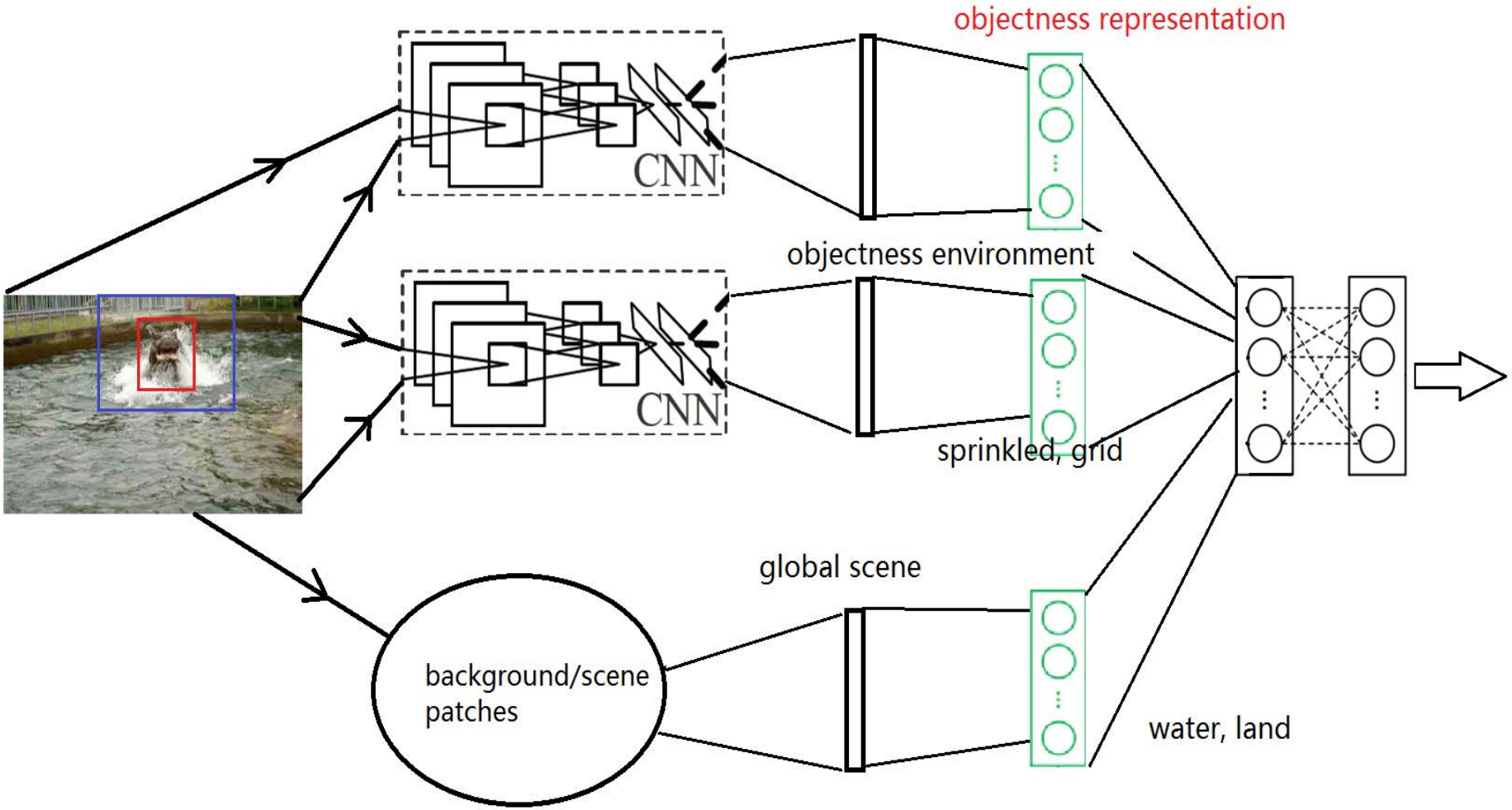
1. Is contextual info really necessary for detection? How about cluttered background?
2. How to leverage the weights of contextual info according to how confidence we can classify the object region.
3. How to design a contextual representation?
4. Can we build a few more layers to merge DeCAF, texture attributes and background attributes to boost detection?

1. seeing is worse than believing - reading people's minds better than computer-vision methods recognize actions, eccv2014

2. describing Textures in the Wild, cvpr2014

3. Rich feature hierarchies for accurate object detection and semantic segmentation, cvpr2014

5. Ask ourselves



Thank you

Abstract

We will briefly review the R-CNN [1], which actually does classification over thousands of objectness regions extracted from the image. We will see what it missed -- interaction between objects and context within the image. When people make use contextual information in addition to CNN, performance is improved [2]. This is also recently supported by an interesting study [3], which compares the action classification performance between state-of-the-art CV methods and linear SVM over the fMRI data. The conclusions in the paper are very interesting, but we emphasize the most "trivial" yet convincing one -- human brain exploits semantic inference for action classification, which is absent in CV methods for action classification. So, exploiting the contextual information will be a reasonable step to improve detection. But how can we represent, extract and utilize the contextual information? To answer these questions, I will present two other papers which are seemingly unrelated to the questions. The first one is [4], which presents how to represent/learn/use texture attribute to improve texture and material classification; the second one is [5] which uses patch match techniques for chair detection in a finer way. Based on these two papers, we will try to answer the questions -- how can we represent, learn and use the contextual information to boost detection?