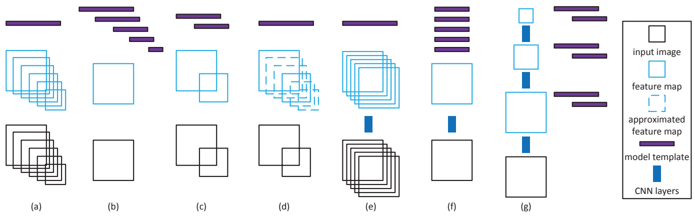
A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection

Zhaowei Cai[1], Quanfu Fan[2], Rogerio Feris[2], and Nuno Vasconcelos[1]
[1]UC San Diego, [2]IBM Watson Research

I. Introduction

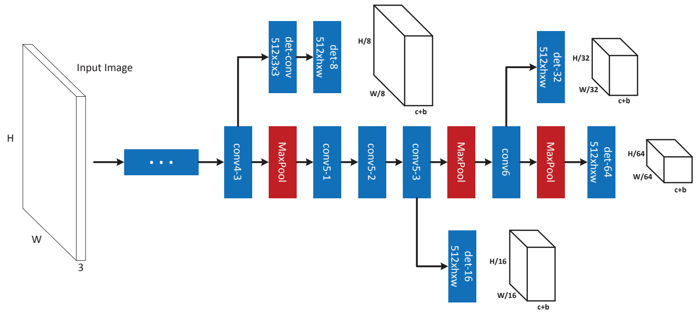
- Motivations:
  - There is an inconsistency between the sizes of objects, which are variable, and filter receptive fields, which are fixed, in Faster-RCNN framework.
  - Multi-scale detection is not well addressed in CNN based object detection frameworks.
  - The original input images are usually upsampled to boost performance, which exponentially increases the memory and computation costs of the detector.
- Contributions:
  - This work proposes a unified multi-scale deep CNN, denoted the multi-scale CNN (MS-CNN), for fast object detection.
  - To ease the inconsistency between the sizes of objects and receptive fields, object detection is performed with multiple output layers, each focusing on objects within certain scale ranges.
  - Feature upsampling (implemented by a deconvolutional layer) is used as an alternative to input upsampling, which improves detection accuracy but adds trivial computation and no parameter.

II. Multi-scale Object Detection



– Inspired by previous evidence on the benefits of the strategy of (c) over that of (b), we propose a new multi-scale strategy (g). This can be seen as the deep CNN extension of (c), but only uses a single scale of input.

III. Multi-scale Object Proposal Network



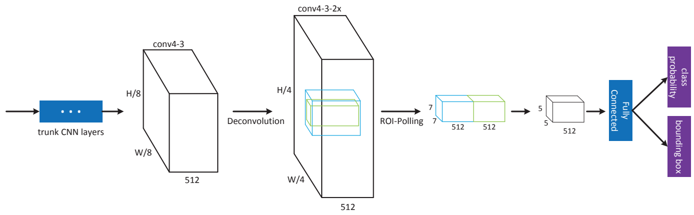
– Each detection branch detects objects that match its scale, and the combination of those branches forms a strong multi-scale detector.

– objective function:

$$\mathcal{L}(\mathbf{W}) = \sum_{m=1}^{M} \sum_{i \in S^m} \alpha_m l^m(X_i, Y_i|\mathbf{W})$$

where $$l(X, Y|\mathbf{W}) = L_{cls}(p(X), y) + \lambda[y \geq 1] L_{loc}(b, \hat{b})$$

IV. Object Detection Network



– unified objective function:

$$\mathcal{L}(\mathbf{W}, \mathbf{W}_d) = \sum_{m=1}^{M} \sum_{i \in S^m} \alpha_m l^m(X_i, Y_i|\mathbf{W}) + \sum_{i \in S^o} \alpha_o l^o(X_i, Y_i|\mathbf{W}, \mathbf{W}_d)$$

– Trunk CNN layers are shared with proposal sub-network.

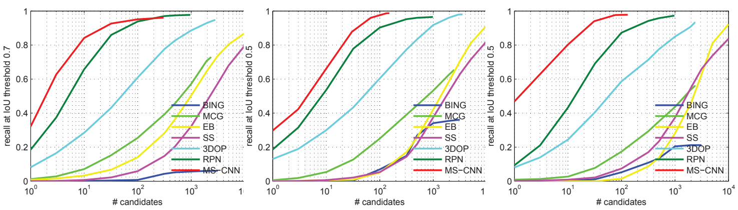
– ROI pooling is applied to the top of the "conv4-3" layer.

– A deconvolutional layer is used to upsample feature maps as an alternative of input upsampling, avoiding issues such as large memory requirements, slow training and testing.

– Object and context regions are stacked together immediately after ROI pooling, followed by an extra convolutional layer to compress redundant information and avoid parameters increase.

V. Experimental Results

- Datasets
  - KITTI: 7,481 images (1250×375) for training and 7,518 for testing, no testing ground truth is available.
  - Caltech: 32,077 images (640×480) for training and 4,024 for testing.
- Proposal comparison
  - achieves a recall about 98% with only 100 proposals of high quality.



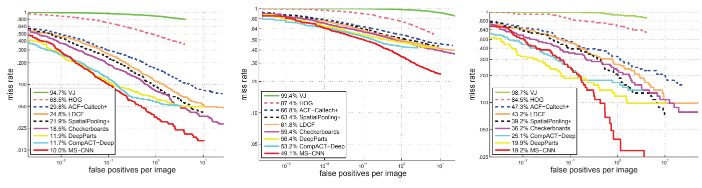
- Ablation study
  - input size, feature upsampling, context embedding

Model	Time	# params	Car	Pedestrian
h384	0.11s	471M	80.63	68.37
h576	0.22s	471M	88.14	70.77
h768	0.41s	471M	88.88	72.26
h576-2x	0.23s	471M	89.12	72.49
h576-ctx	0.24s	863M	88.88	71.45
h576-ctx-c	0.22s	297M	89.13	72.13

- Comparison on KITTI
  - set a new record for the detection of pedestrians and cyclists, and ranked top 1 for cars among published works.

Methods	Time	Car	Pedestrian	Cyclist
Faster-RCNN	2s	81.84	65.90	63.35
Regionlets	1s	76.45	61.15	58.72
3DOP	3s	88.64	67.47	68.94
SDP+RPN	0.4s	88.85	70.16	73.74
Mono3D	4.2s	88.66	66.68	66.36
MS-CNN	0.4s	**89.02**	**73.70**	**75.46**

- Comparison on Caltech
  - achieves state-of-the-art performance, high detection rate, robust to small and occluded pedestrians.



- Real-time running speed
  - up to 10 fps on KITTI (1250×375) and 15 fps on Caltech (640×480) images.
- Reproducible research
  - https://github.com/zhaoweicai/mscnn