Sparselet Models for Efficient Multiclass Object Detection

Hyun Oh Song[1] , Stefan Zickler[2] , Tim Althoff[1] , Ross Girshick[3] , Mario Fritz[4] , Christopher Geyer[2] , Pedro Felzenszwalb[5], Trevor Darrell[1]

[1]UC Berkeley, [2]iRobot, [3]University of Chicago, [4]MPI Informatics, [5]Brown University

Motivation

- High detection accuracy : DPM
- Hypothesis pruning : Cascade / Coarse-to-fine
- What if we want to detect hundreds or thousands of object classes? Sparselets

Intuition

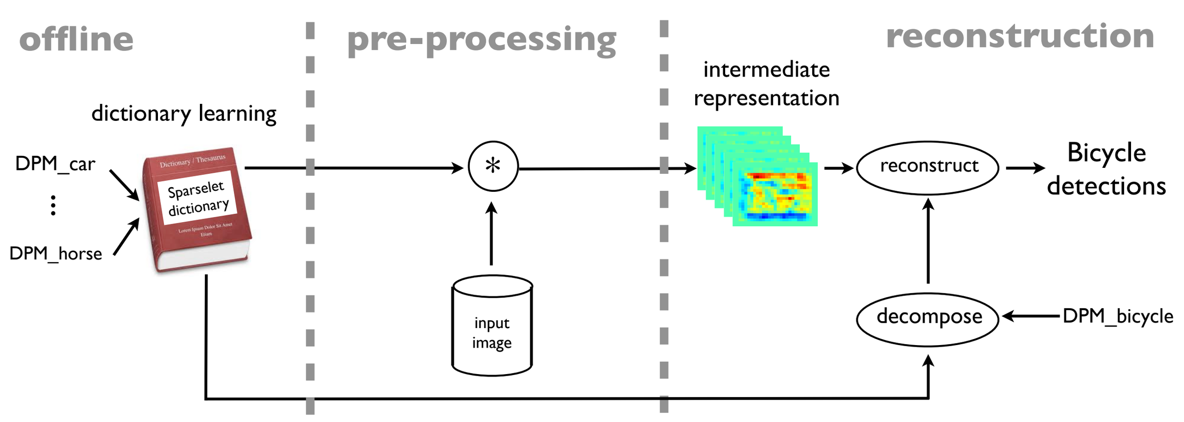
- As the number of object categories grows, individual model filters are increasingly likely to be redundant with respect to each other.

Bottleneck in DPM inference

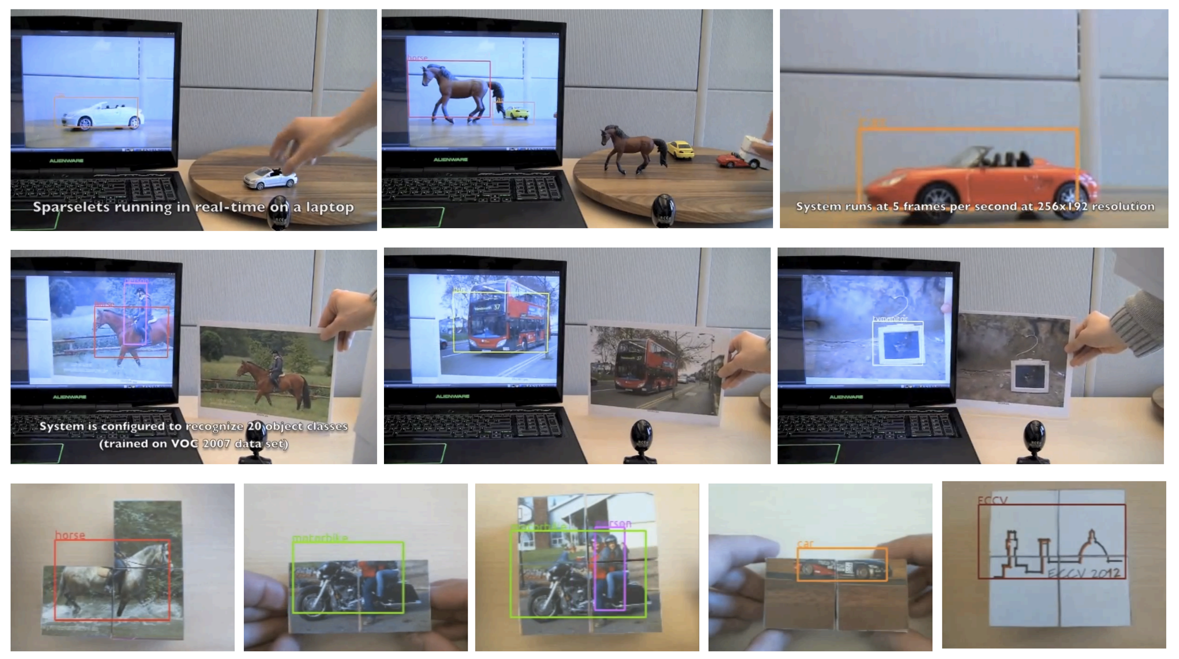
$$\text{score}(x,z) = w_c + \boxed{\sum_{i=0}^{N} \mathbf{w}_{ci}^{\intercal} \psi_{ci}(x,\rho_i)} + \sum_{i=1}^{N} \mathbf{d}_{ci}^{\intercal} \boldsymbol{\delta}_{ci}(\rho_0,\rho_i)$$

- Filter evaluation takes 60~70 % of total computation time
- Per every pixel in image pyramid, algorithm computes 1000~3000 convolutions (@ 20 classes)

Overall Concept



Real-time multiclass DPM detection on a laptop



Sparselets

Set of model filters $\mathcal{W} = \{\mathbf{w_1}, ..., \mathbf{w_N}\}$

Set of sparselet $\mathcal{S} = \{\mathbf{s_1}, ..., \mathbf{s_K}\}$

$$\min_{\alpha_{ij}, s_j} \sum_{i=1}^{N} ||\mathbf{w}_i - \sum_{j=1}^{K} \alpha_{ij} \mathbf{s}_j||_2^2$$

$$\text{subject to} \quad ||\boldsymbol{\alpha}_i||_0 \le \epsilon \quad \forall i = 1, ..., N$$

$$||\mathbf{s}_j||_2^2 \le 1 \quad \forall j = 1, ..., K$$

Sparse reconstruction of filter response

$$\Psi * \mathbf{w}_i \approx \Psi * \left(\sum_{\substack{j=1 \\ \forall \alpha_{ij} \neq 0}}^{K} \alpha_{ij} \mathbf{s}_j \right) = \underbrace{\sum_{\substack{j=1 \\ \forall \alpha_{ij} \neq 0}}^{K}}_{\text{Sparsity}} \alpha_{ij} \underbrace{(\Psi * \mathbf{s}_j)}_{\text{Cached}}$$

Matrix Factorization point of view

$$\begin{bmatrix} \text{—} \ \Psi * \mathbf{w}_1 \ \text{—} \\ \text{—} \ \Psi * \mathbf{w}_2 \ \text{—} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \text{—} \ \Psi * \mathbf{w}_N \ \text{—} \end{bmatrix} \approx \boxed{\begin{bmatrix} \text{—} \ \boldsymbol{\alpha}_1 \ \text{—} \\ \text{—} \ \boldsymbol{\alpha}_2 \ \text{—} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \text{—} \ \boldsymbol{\alpha}_N \ \text{—} \end{bmatrix}} \begin{bmatrix} \text{—} \ \Psi * \mathbf{s}_1 \ \text{—} \\ \text{—} \ \Psi * \mathbf{s}_2 \ \text{—} \\ \vdots \\ \text{—} \ \Psi * \mathbf{s}_K \ \text{—} \end{bmatrix}$$

80 ~ 99 % Sparse

Sparselet DPM

$$\text{score}(x,z) = \underbrace{w_c}_{\text{Component bias}} + \underbrace{\sum_{i=0}^{N} \mathbf{w}_{ci}^{\intercal} \psi_{ci}(x,\rho_i)}_{\text{Filter evaluation}} + \underbrace{\sum_{i=1}^{N} \mathbf{d}_{ci}^{\intercal} \boldsymbol{\delta}_{ci}(\rho_0,\rho_i)}_{\text{Deformation cost}}$$

$$= w_c + \sum_{i=0}^{N} \sum_{\substack{j=1 \\ \forall \alpha_{ij} \neq 0}}^{d} \alpha_{ij} \left(\mathbf{s}_j^{\intercal} \psi_{ci}(x,\rho_i) \right) + \sum_{i=1}^{N} \mathbf{d}_{ci}^{\intercal} \boldsymbol{\delta}_{ci}(\rho_0,\rho_i)$$

Complexity per pixel

$$\text{Speedup} = \frac{\text{Convolution with all model filters}}{\text{Convolution with sparselets} + \text{Sparse reconstruction}}$$

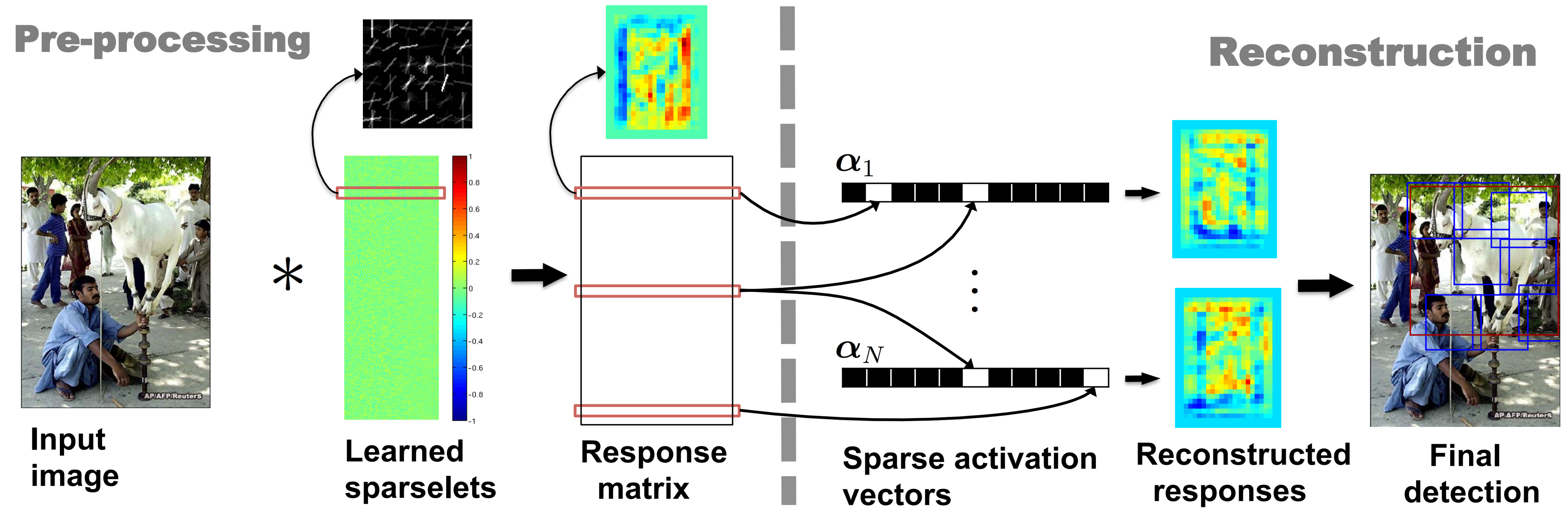
$$= \frac{Nm}{|S|m + \underbrace{N\mathbb{E}[||\boldsymbol{\alpha}_i||_0]}_{\text{N lookups}}} \qquad m: \ \text{Convolution filter size}$$

As N grows to a large number,

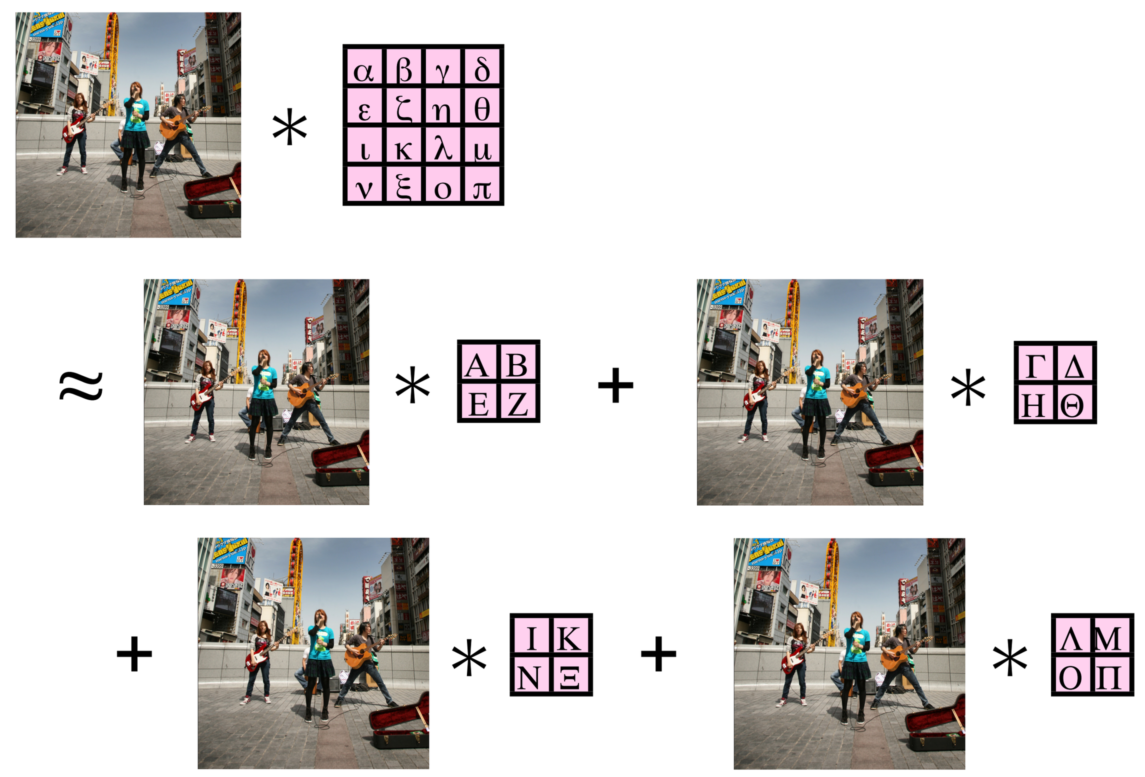
$$\text{Speedup} = \frac{m}{\mathbb{E}[||\boldsymbol{\alpha}_i||_0]}$$

Sparsity dominates as N grows!

Sparselet in a slide



Subsparselets = Mini Parts



Parts of an object detector, for any class, can be constructed by tiling sparse linear combinations of sparselet mini-parts

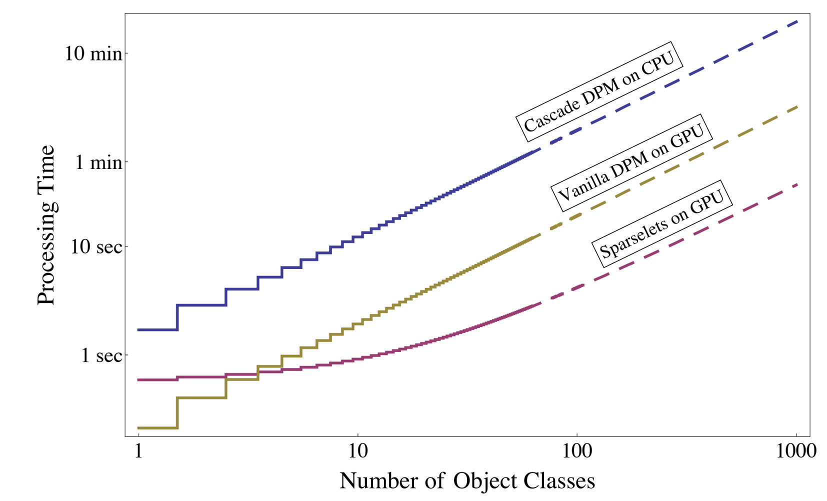
$h_s \times w_s$	$\lvert S \rvert$	$h_s w_s \lvert S \rvert$	ϵ	$(h_F/h_s)(w_F/w_s)$	**bicycle**	**car**	**cat**	**person**
6 × 6	128	4608	112	1	1.0645	1.0349	0.8521	1.1939
3 × 3	512	4608	28	4	0.3116	0.3360	0.2552	0.4573
2 × 2	1152	4608	13	9	0.2298	0.2706	0.1763	0.4007
1 × 1	4608	4608	3	36	0.1062	0.1200	0.0820	0.1635

- Empirically, filter reconstruction error always decreases as we decrease sparselet size $|S|$ (@ fixed computation time)
- However, the space required to store the intermediate representation is proportional to the sparselet dictionary size. This means we have computation time VS memory bandwidth tradeoff.

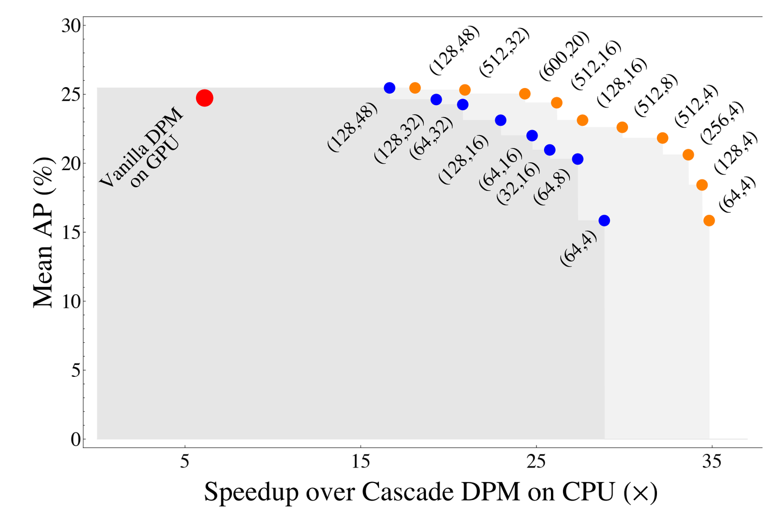
Acknowledgments

H. Song was supported by Samsung Scholarship Foundation. S. Zickler and C. Geyer were supported by DARPA contract W911NF-10-C-0081. P. Felzenszwalb and R. Girshick were supported in part by NSF grant IIS-0746569. T. Darrell was supported by DARPA contract W911NF-10-2-0059, by NSF awards IIS-0905647, IIS-0819984, and support from Toyota and Google.

Experiment



Comparison of cascade algorithm on CPU vs. vanilla DPM on GPU vs. sparselets accelerated DPM on GPU as number of object classes grows.



Speedup vs. Mean AP. Blue dots are for "online" results measuring end-to-end time. Orange dots are for "post-hoc" case. The tuple in parenthesis denote ($|S|$, ε).

Conclusion

- We introduced sparse intermediate representations that enable real-time multiclass object detection on a laptop computer.
- Sparselets exploit the intrinsic redundancy among model filters and can generalize to previously unseen categories from other domains.
- Our model is well suited to a parallel implementation, and we report a new GPU DPM implementation with state of the art performance one to two orders of magnitude faster than the fastest current deformable part model implementations.