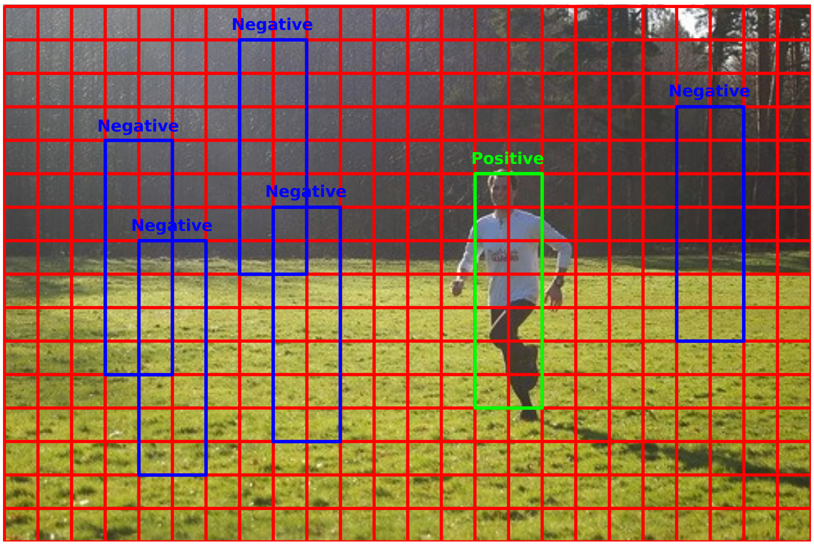
EXACT ACCELERATION OF LINEAR OBJECT DETECTORS

CHARLES DUBOUT AND FRANÇOIS FLEURET

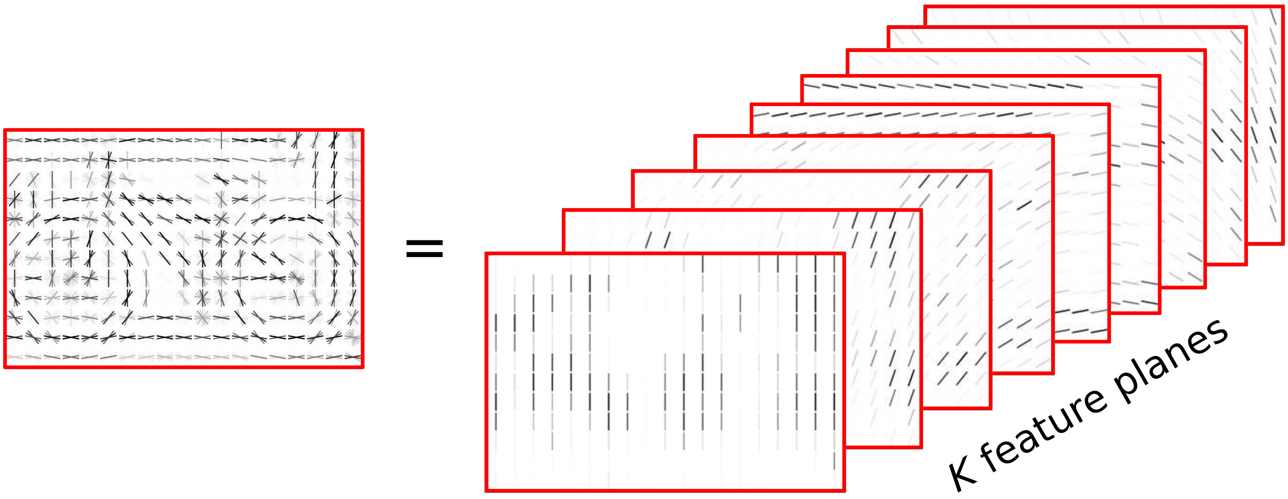
OCTOBER 1, 2012

THE SLIDING WINDOW TECHNIQUE



- Transforms a detection problem into a binary classification one
- Applies a binary classifier at every image position and scale
- Similar to sweeping the detection window across the image

HOG FEATURE PLANES

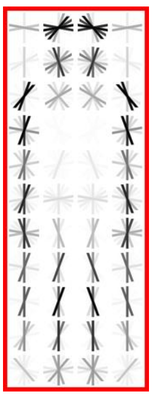


The HOG features can be seen as organized in planes, containing distinct features from each grid cell ($K = 32$).

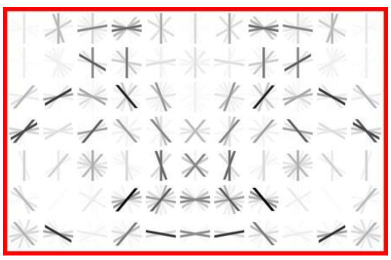
HOG AND LINEAR SVM

[Dalal & Triggs '05]

Pedestrian template



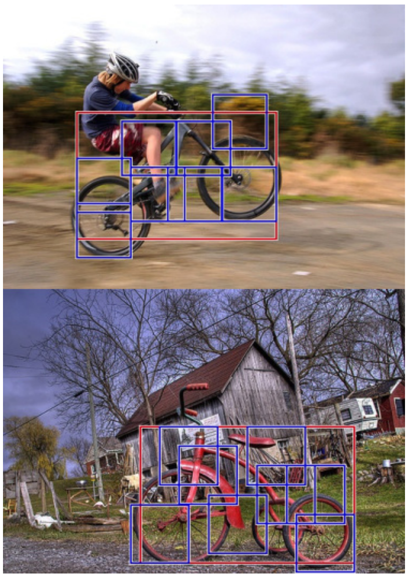
Bicycle template

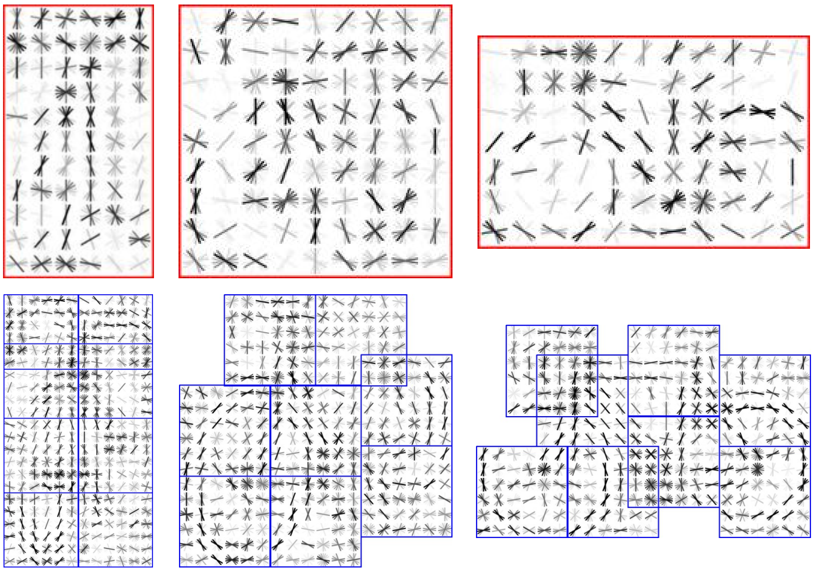


The detection score is linear: $S(x, y) = \langle w, HOG(x, y) \rangle$, where $HOG(x, y)$ is the vector of features extracted from the subwindow at (x, y), of same size as w.

Deformable Part Model

[Felzenszwalb & al. '08]





Deformable Part Model

If we define

$S_0(x, y)$ the root detection score at location (x, y)

$S_q(x, y), \; q = 1, \ldots, Q$ the part q detection score

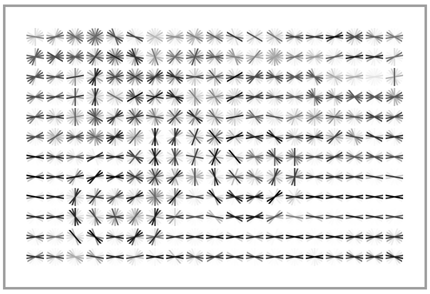
$D_q(x, y, x', y')$ the deformation cost for part q

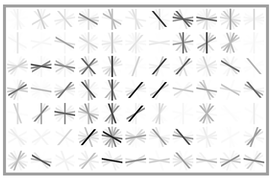
The total score for the deformable model at location (x, y) is

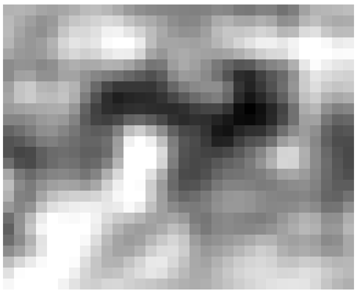
$$
\begin{aligned}
S(x, y) &= S_0(x, y) + \max_{(x_1, y_1, \ldots, x_Q, y_Q)} \sum_q S_q(x_q, y_q) - D_q(x, y, x_q, y_q) \\
&= S_0(x, y) + \sum_q \underbrace{\max_{x', y'} S_q(x', y') - D_q(x, y, x', y')}_{T_q(S_q)(x, y)}
\end{aligned}
$$

Deformable Part Model

Root detection

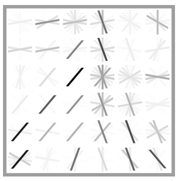


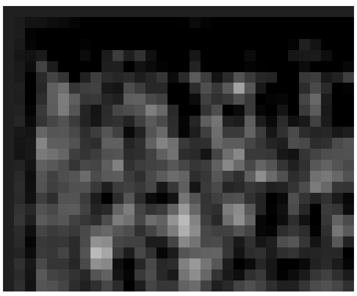
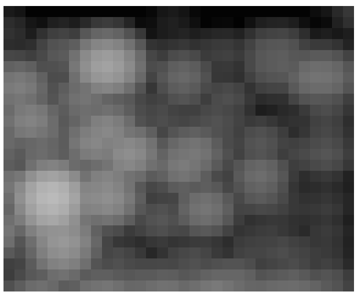


$S_0 =$ 

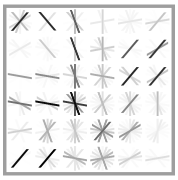
Deformable Part Model

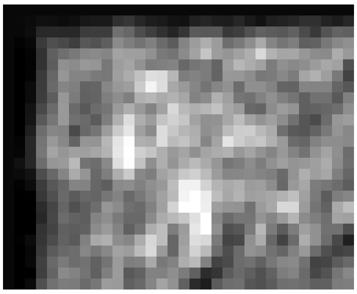
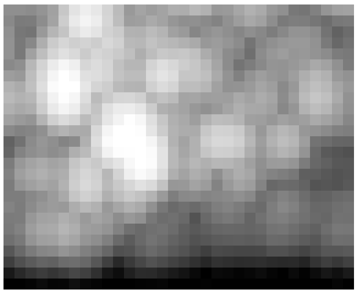
Part detection



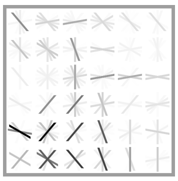
$S_1 =$  $T_1(S_1) =$ 

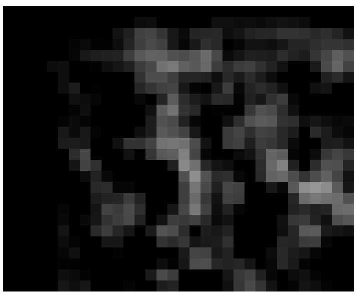
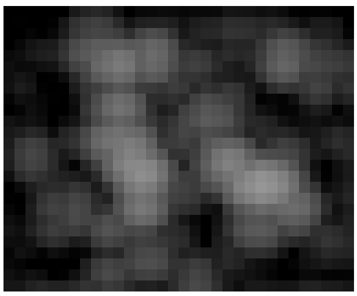
Deformable Part Model
Part detection

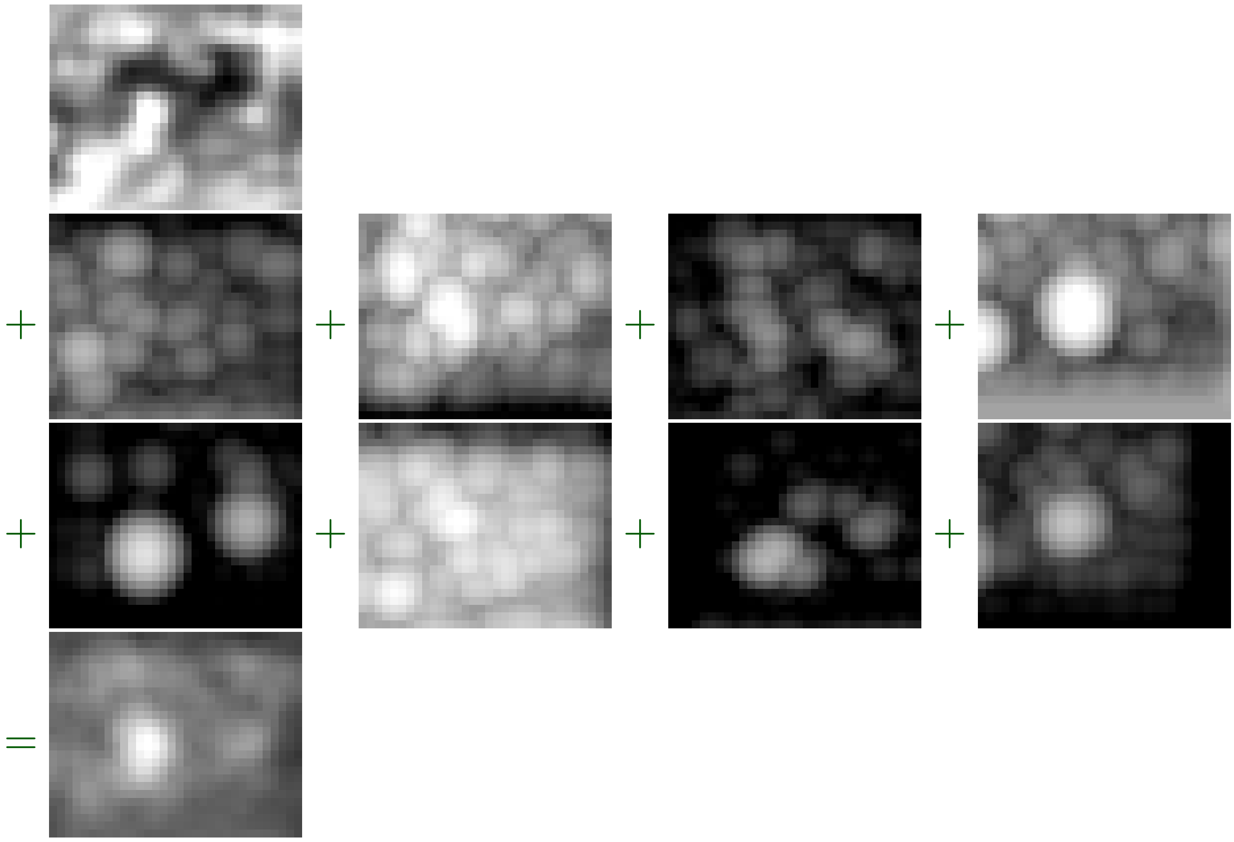
$S_2 =$  $T_2(S_2) =$ 

Deformable Part Model
Part detection

$S_3 =$  $T_3(S_3) =$ 

Deformable Part Model
Final score



Computational Challenge

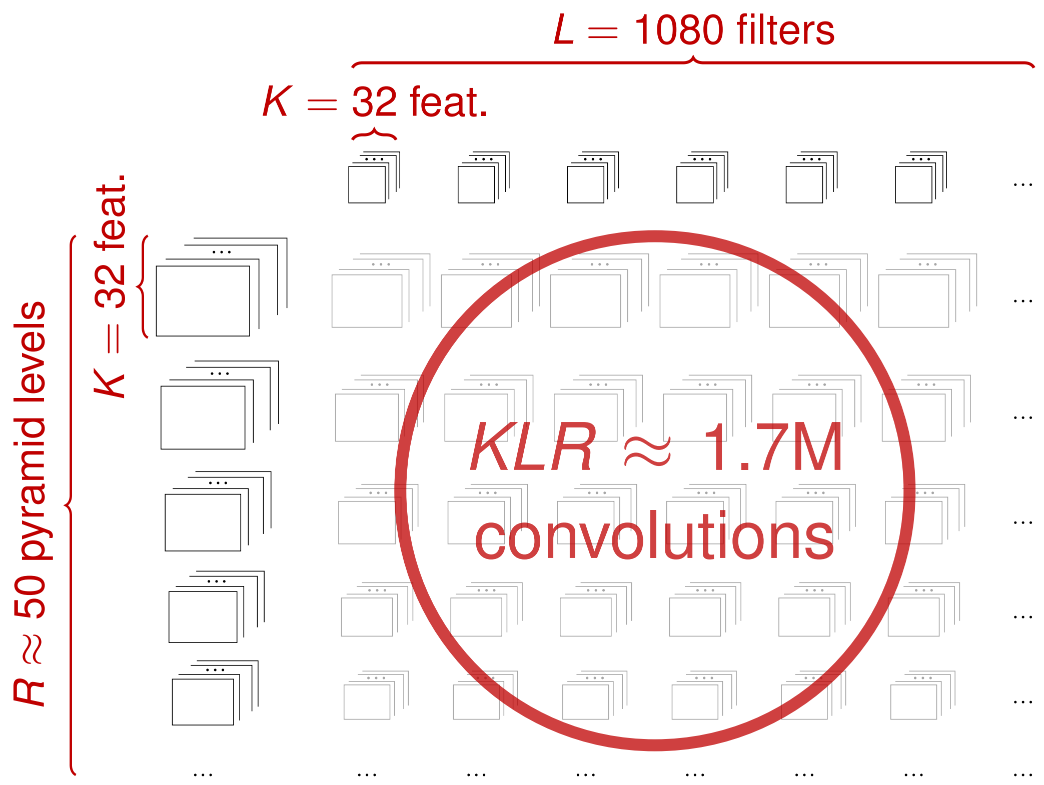
This process has to be repeated for every class of interest and every component of the model's mixture.

The core operation in this process is the convolution by linear filters to compute the root and part detection scores.

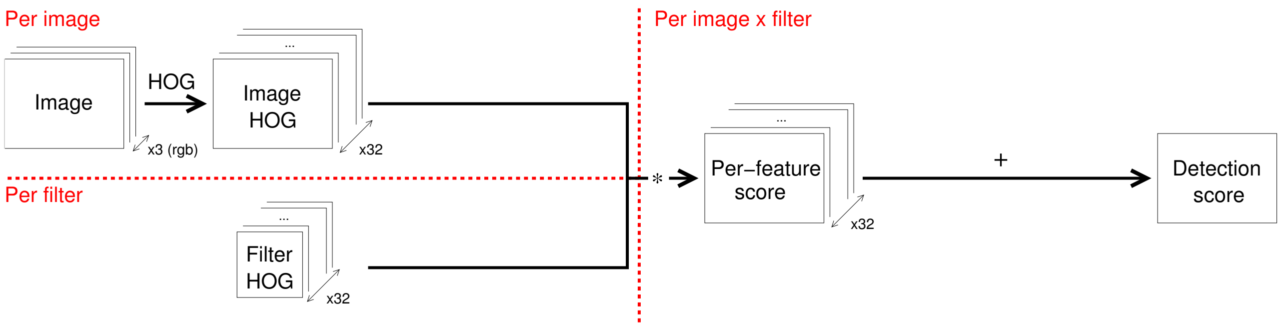
For 20 classes $\times$ 6 mixtures $\times$ 9 parts $=$

1080 linear detectors!

Computational Challenge



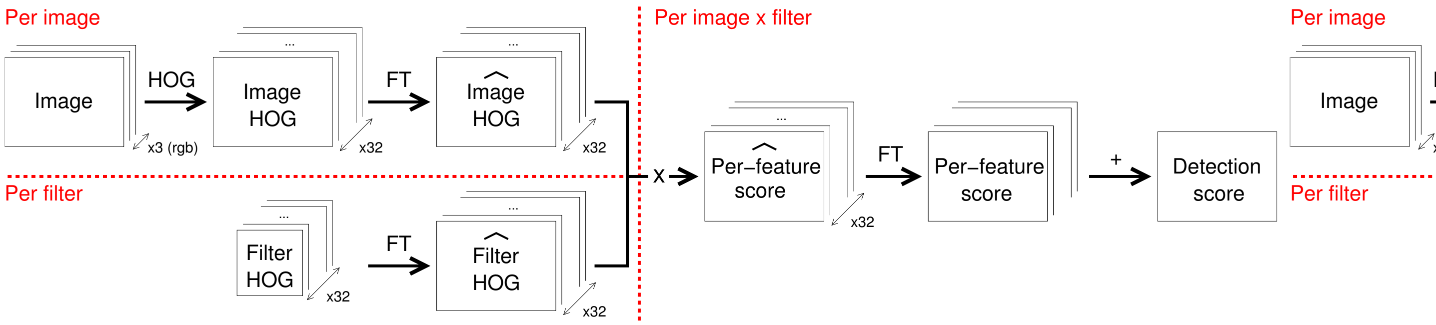
Standard convolution process



The computational cost to convolve a HOG image of size $M \times N$ with L filters of size $P \times Q$ across K features is:

$$C_{\text{std}} = \mathcal{O}(KLMNPQ)$$

Fourier based convolutions



The computational cost to convolve a HOG image of size $M \times N$ with L filters of size $P \times Q$ across K features is:

$$C_{\text{FFT}} = \underbrace{\mathcal{O}(KMN \log MN)}_{\text{Forward FFTs}} + \underbrace{\mathcal{O}(KLMN)}_{\text{Multiplications}} + \underbrace{\mathcal{O}(KLMN \log MN)}_{\text{Inverse FFTs}}$$

$$C_{\text{opt}} = \underbrace{\mathcal{O}(KMN \log MN)}_{\text{Forward FFTs}} + \underbrace{\mathcal{O}(KLMN)}_{\text{Multiplications}} + \underbrace{\mathcal{O}(\cancel{K}LMN \log MN)}_{\text{Inverse FFTs}}$$

$$\approx \mathcal{O}(KLMN)$$

Lets plug in typical numbers

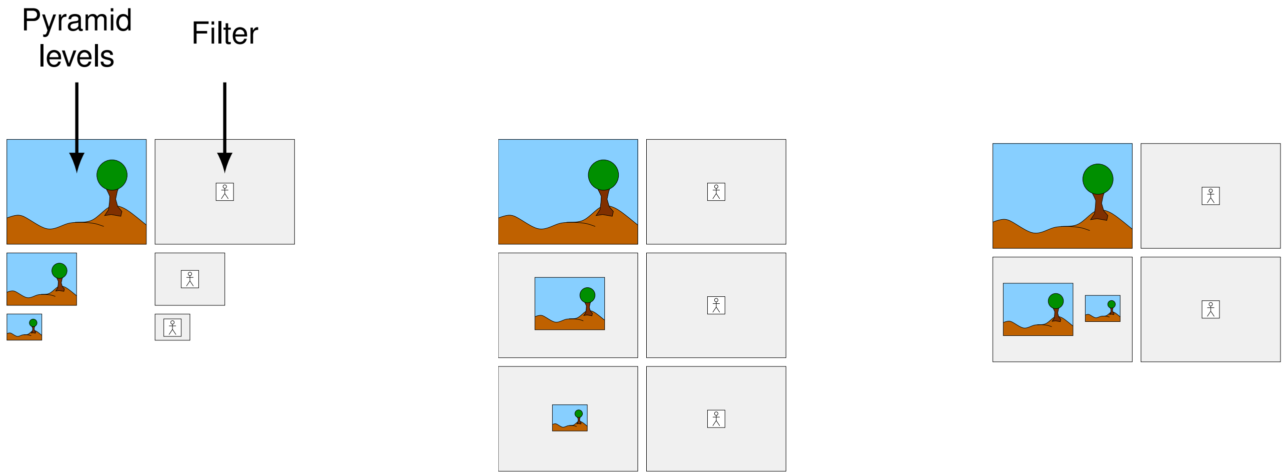
- $K = 32$ (number of HOG features)
- $L = 54$ (number of filters)
- $M \times N = 64 \times 64$ (size of the pyramid level)
- $P \times Q = 6 \times 6$ (size of the filters)

$$C_{\text{std}} \approx 2KLMNPQ \approx 490 \text{ MFlop}$$
$$C_{\text{FFT}} \approx 3KLMN + 2.5(K + KL)MN \log_2 MN \approx 230 \text{ MFlop}$$
$$C_{\text{opt}} \approx 4KLMN + 2.5(K + L)MN \log_2 MN \approx 37 \text{ MFlop}$$

A gain by a factor 13 compared to the standard process, and 6 compared to the standard Fourier one.

Patchworks of pyramid scales

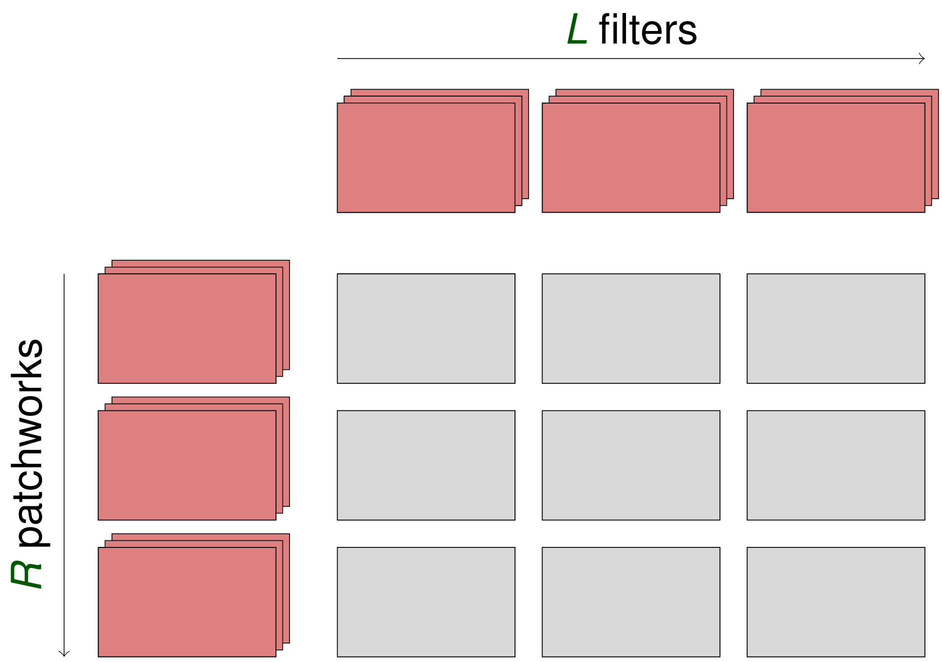
To use the FFT the image and the filter need to be of the same size.



Memory inefficient — Computationally inefficient — Best of both worlds

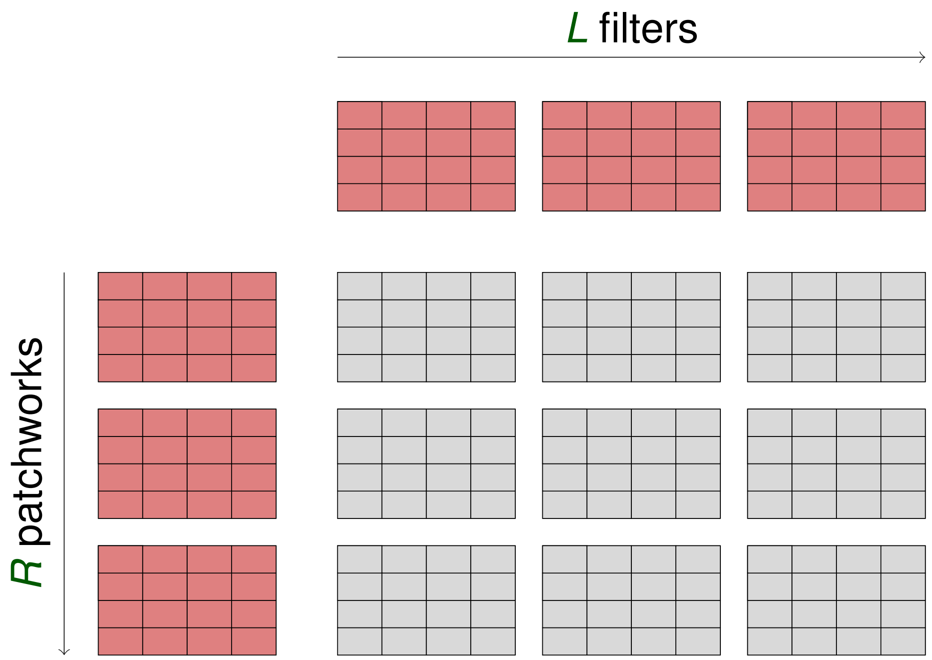
Cache violations

Naive strategy



Read 2 into cache $\Rightarrow$ compute 1.
Read $2LR$ into cache $\Rightarrow$ compute LR.

CACHE VIOLATIONS
FRAGMENT STRATEGY



Read $(L+R)\frac{\epsilon}{L+R} = \epsilon$ into cache $\Rightarrow$ compute $LR\frac{\epsilon}{L+R}$.
Read $L+R$ into cache $\Rightarrow$ compute LR.

RESULTS

Table: Pascal VOC 2007 challenge convolution time and speedup

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table
V4 (ms)	409	437	403	414	366	439	352	432	417	429	450
Ours (ms)	55	56	53	56	57	56	54	56	56	57	57
Speedup (x)	7.4	7.8	7.6	7.4	6.4	7.9	6.5	7.7	7.5	7.5	8.0

	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
V4 (ms)	445	439	429	379	358	351	425	458	433	**413**
Ours (ms)	57	59	57	54	54	55	57	58	55	**56**
Speedup (x)	7.8	7.5	7.6	7.0	6.6	6.4	7.4	7.9	7.9	**7.4**

- Error rate: identical to the baseline (32.3% AP)
- Numerical accuracy: better than the baseline ($1.8 \cdot 10^{-8}$ vs. $2.4 \cdot 10^{-8}$ MAE)

CONCLUSION

- Part-based models obtain state-of-the-art performance at the price of a huge number of convolutions
- The FT is linear, enabling one to do the addition of the convolutions across feature planes in Fourier space
- The computational cost becomes invariant to the filters' sizes, resulting in a big speedup ($\times$7.4 in our experiments)

ECCV 2012 "Spotlight" video.

THANK YOU!

francois.fleuret@idiap.ch
http://www.idiap.ch/~fleuret/