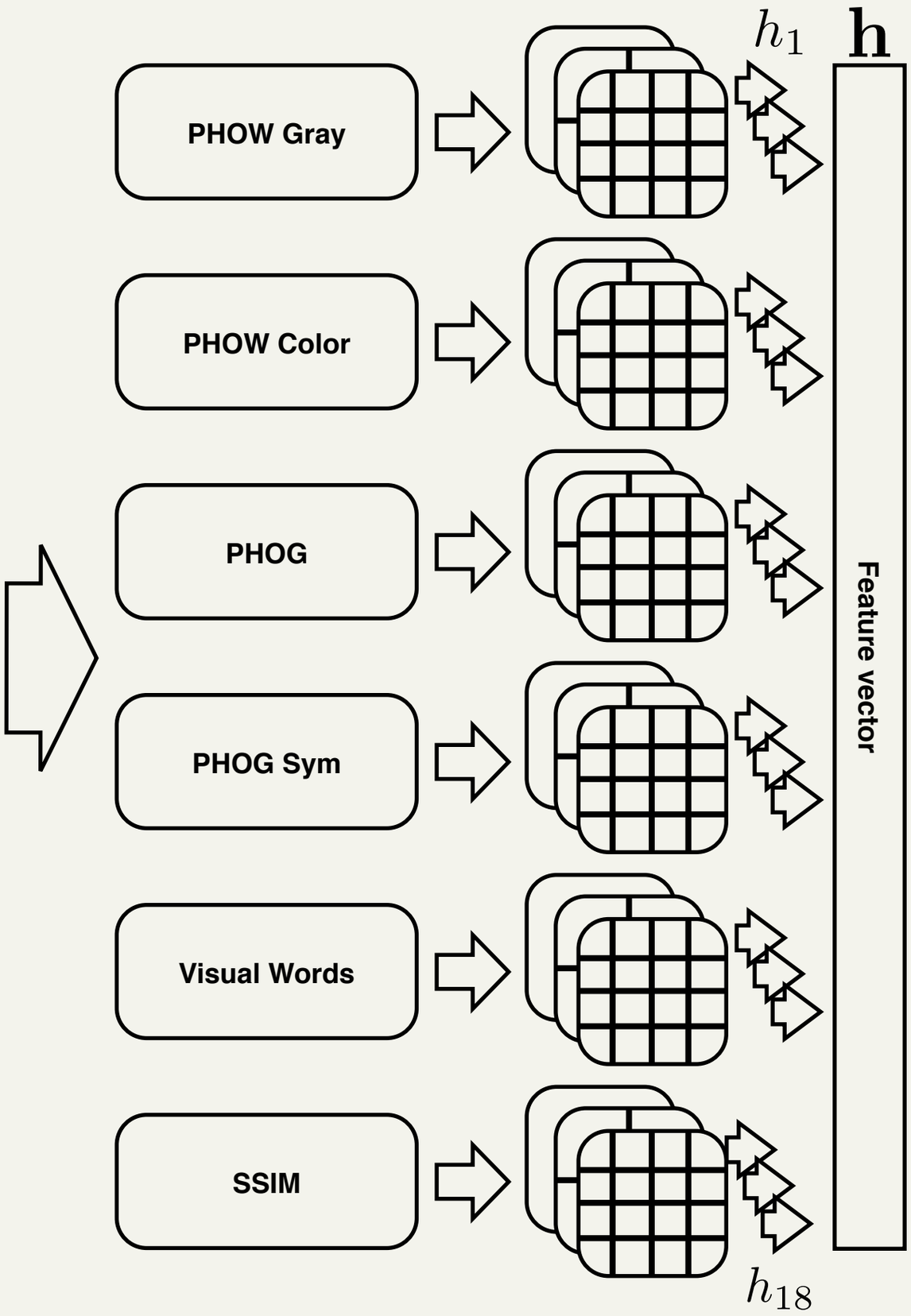
Multiple Kernels for Object Detection

Andrea Vedaldi
Varun Gulshan
Manik Varma
Andrew Zisserman

MK *classification*



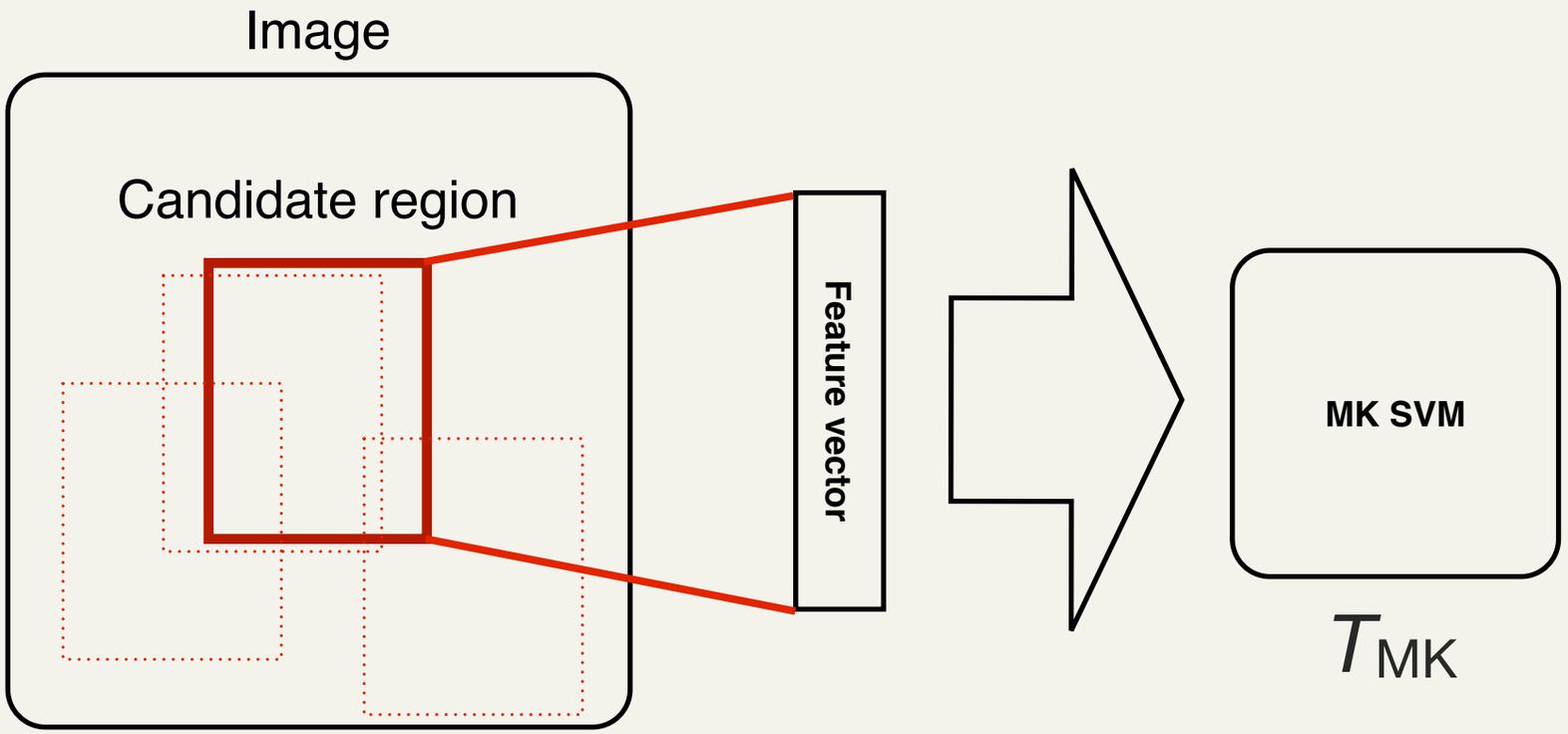
combine one kernel per histogram

$$K(\mathbf{h}, \mathbf{h}') = \sum_{i=1}^{F} d_i K(h_i, h_i')$$

[Varma Rai 2007]
[Gehler Nowozin 2009]

MK *detection*: challenges

- Goal: *sliding window MK classifier*

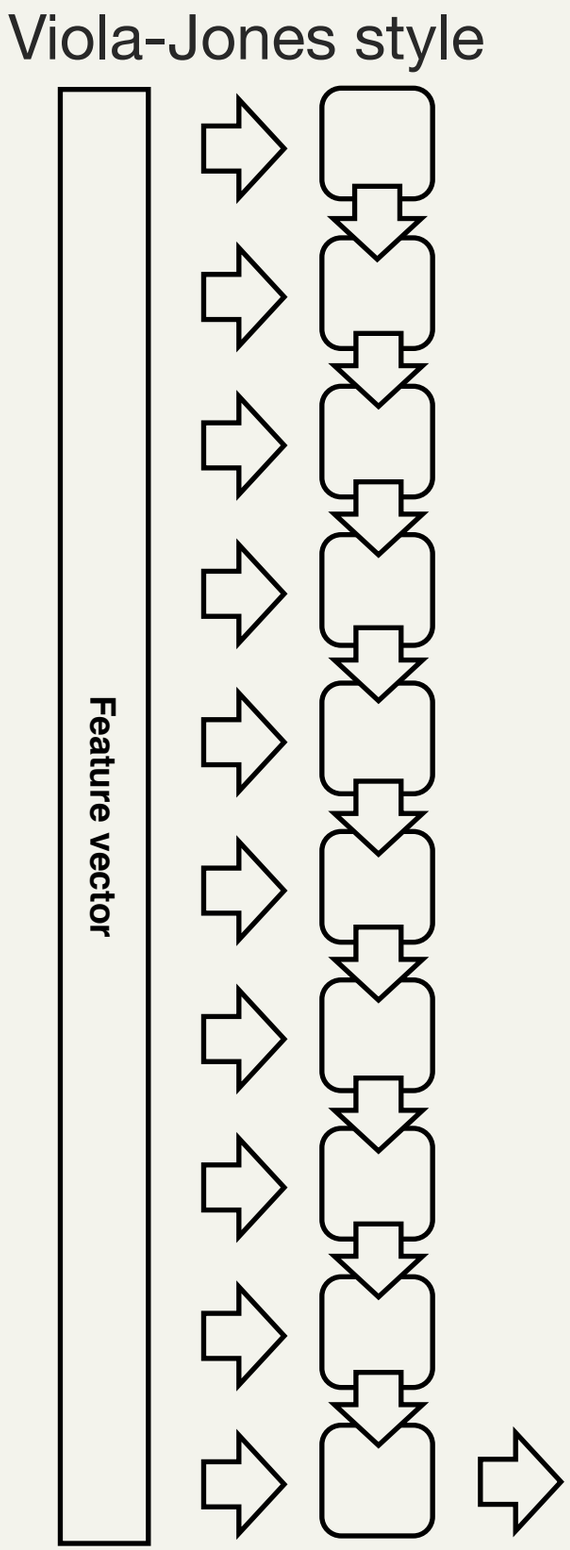


Time required:
T_{MK} × #windows

- Inference space is huge
  #windows = 100 millions
- T_{MK} = seconds

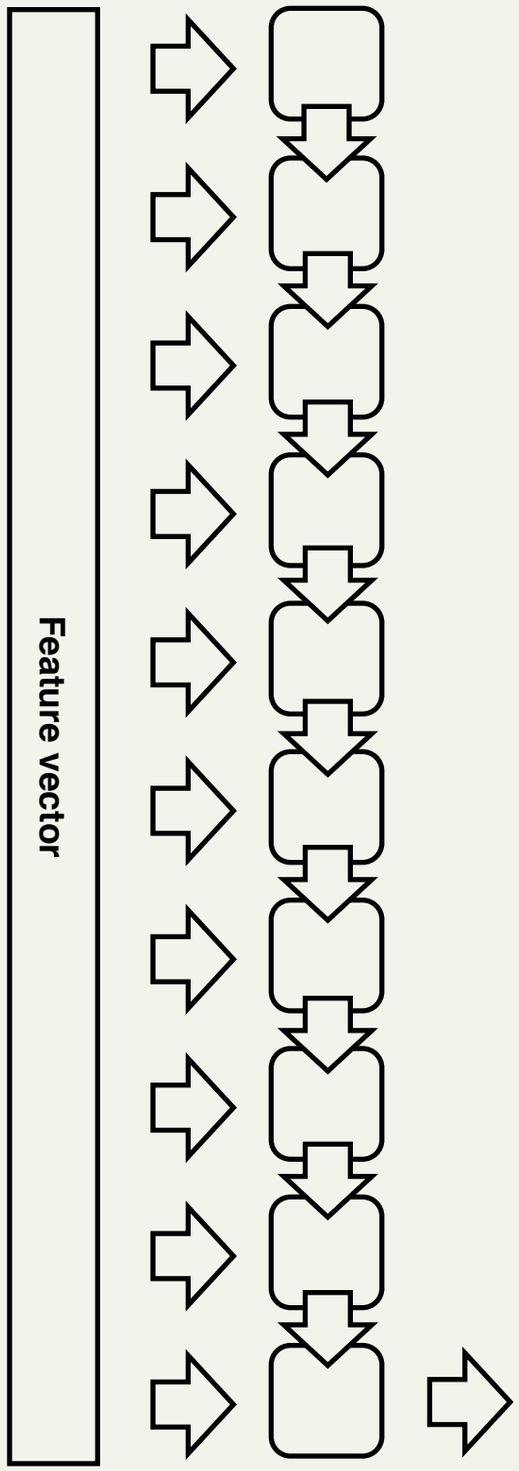
Excruciatingly slow (days/image)

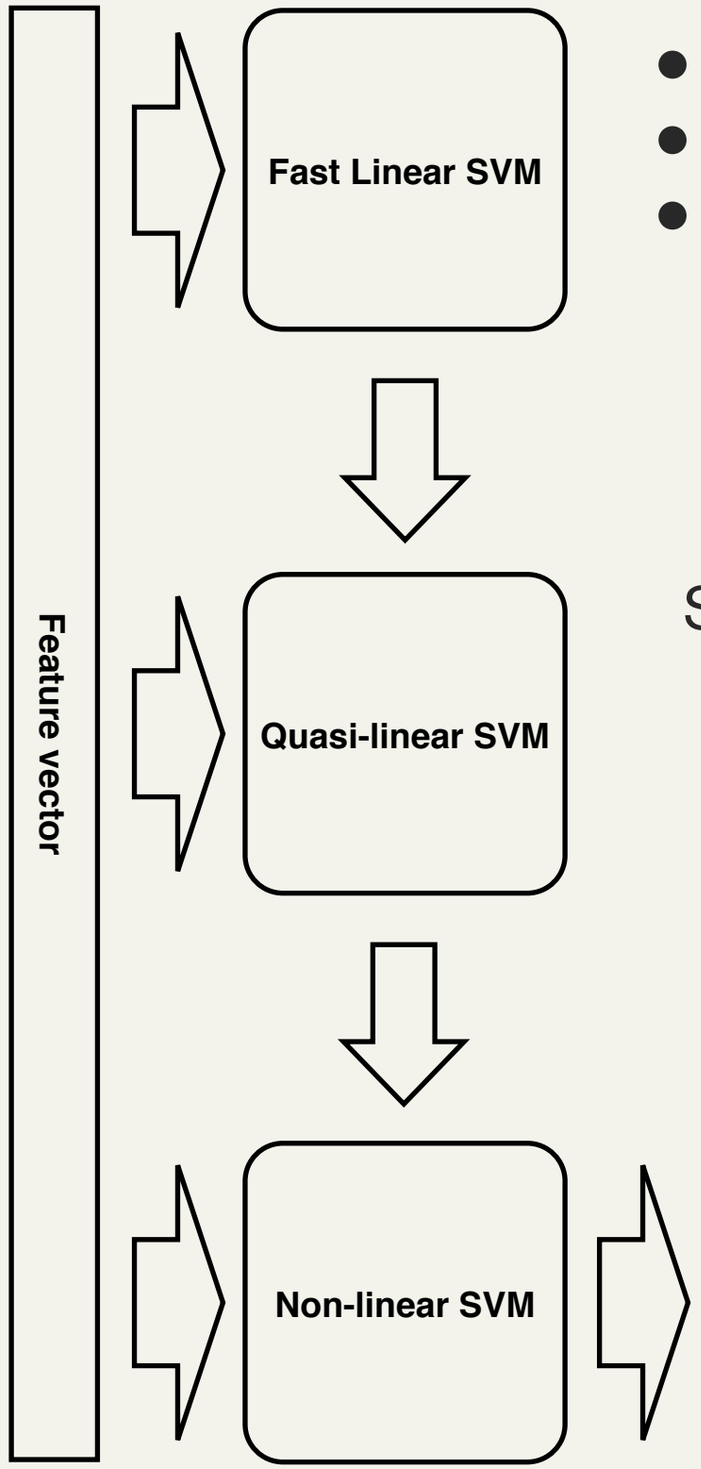
Cascade



Cascade

Viola-Jones style

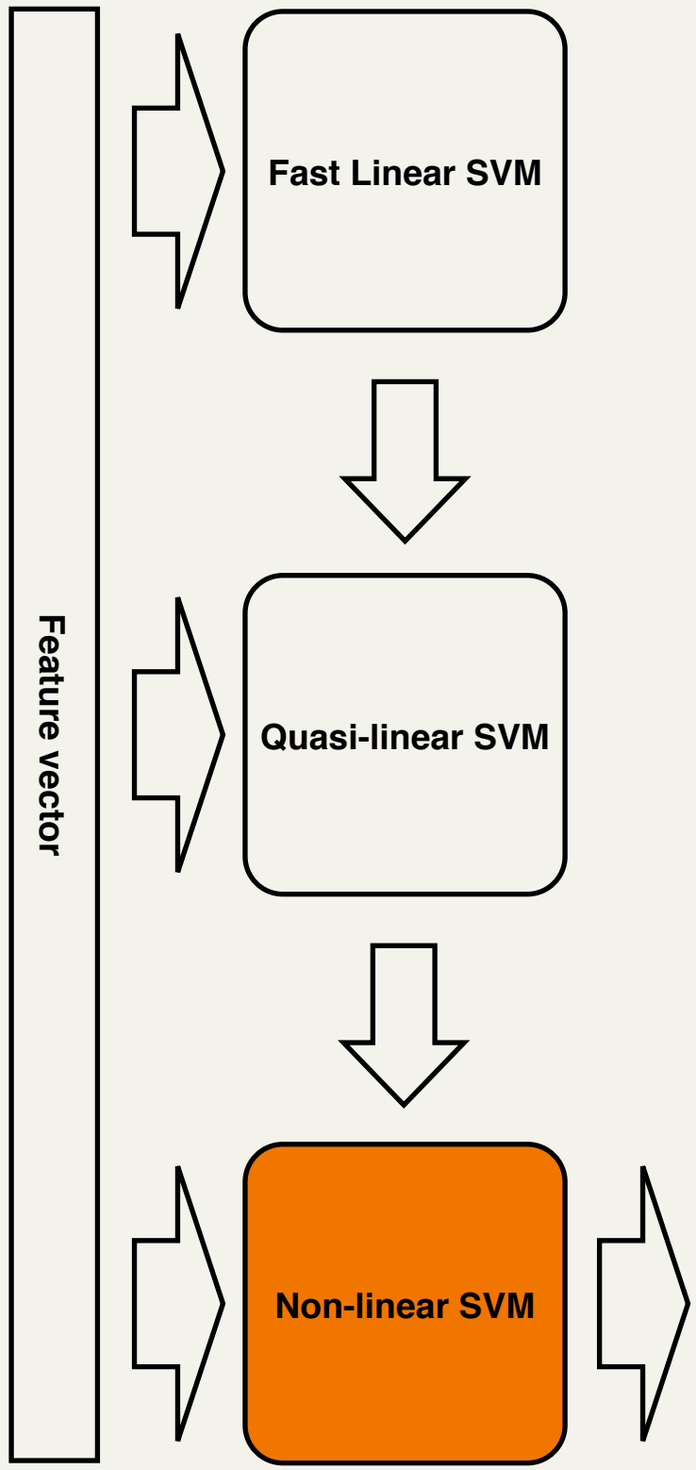




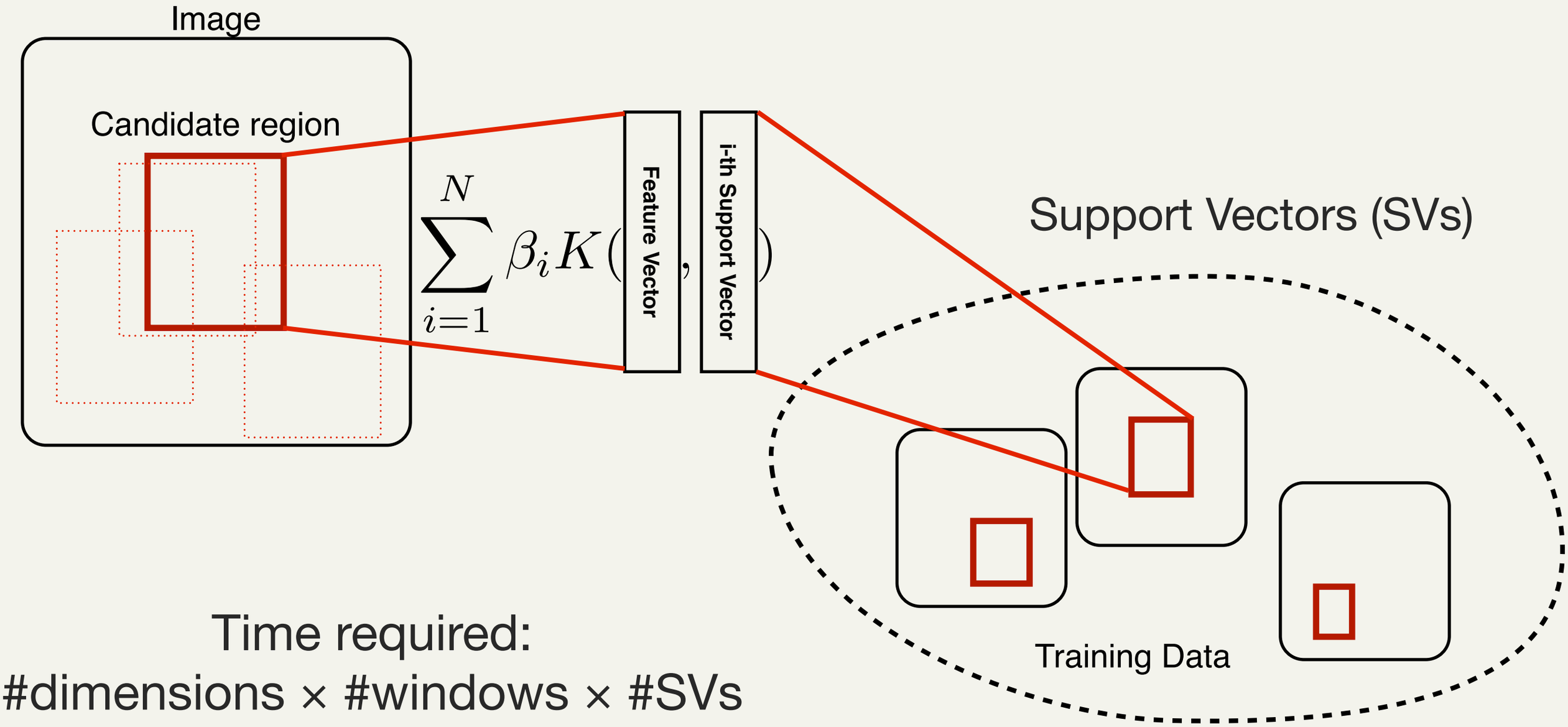
- all full MK SVMs
- all look at all features
- trade-off speed and power tradeoff by choosing the kernel structure

See also [Harzallah et al. 09]

Cascade

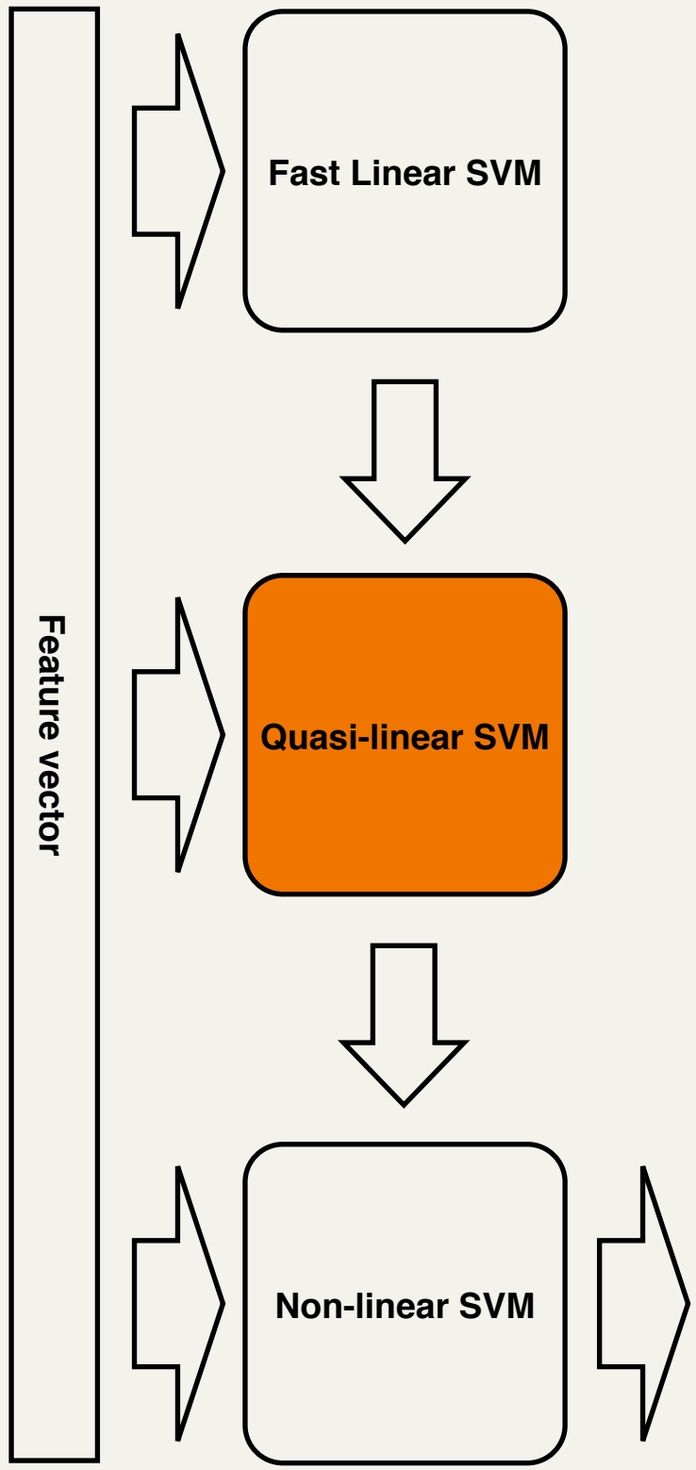

Feature vector
Fast Linear SVM
Quasi-linear SVM
Non-linear SVM

Non-linear sliding SVM

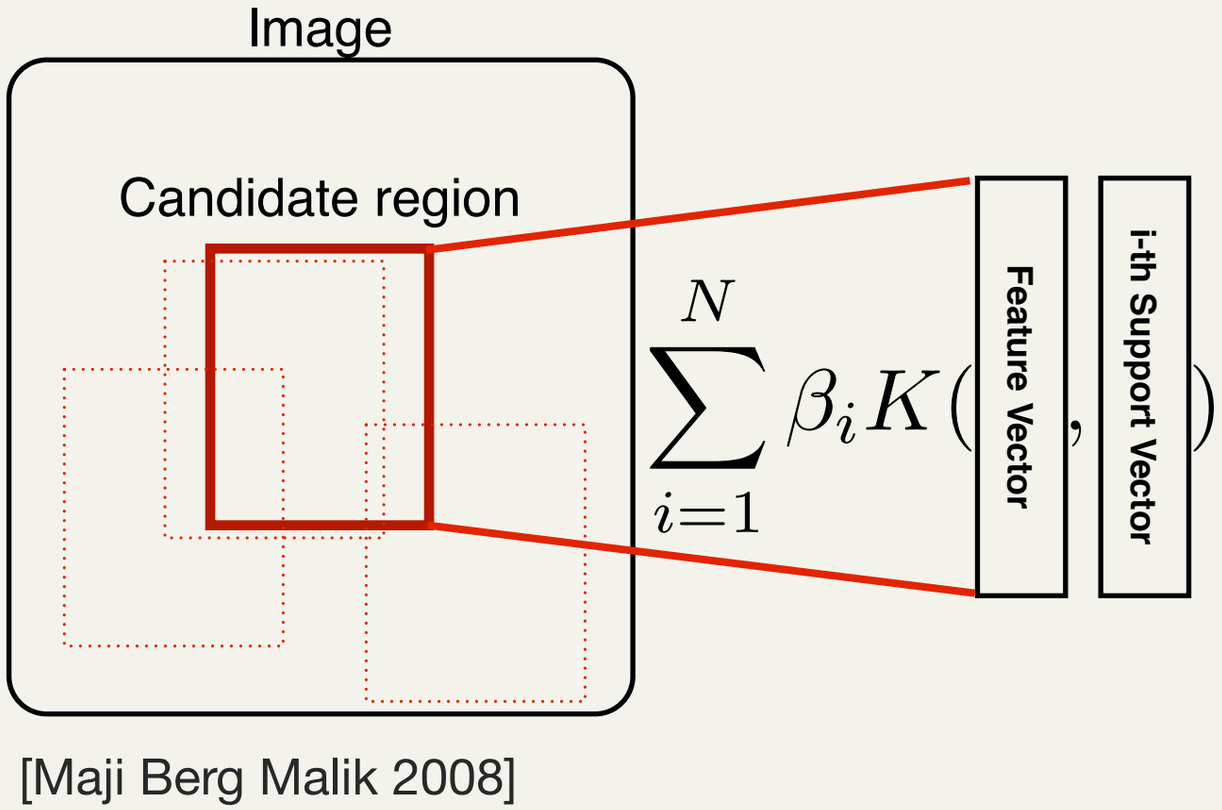


Time required:
#dimensions × #windows × #SVs

Cascade


Fast Linear SVM
Quasi-linear SVM
Non-linear SVM
Feature vector

Quasi-linear SVM



[Maji Berg Malik 2008]

Time required:

#dimensions × #windows × ~~#SVs~~

#dimensions × #windows

Quasi-linear (or additive) kernel decompose as:

$$K(x, y) = \sum_{j=1}^{d} k(x_j, y_j)$$

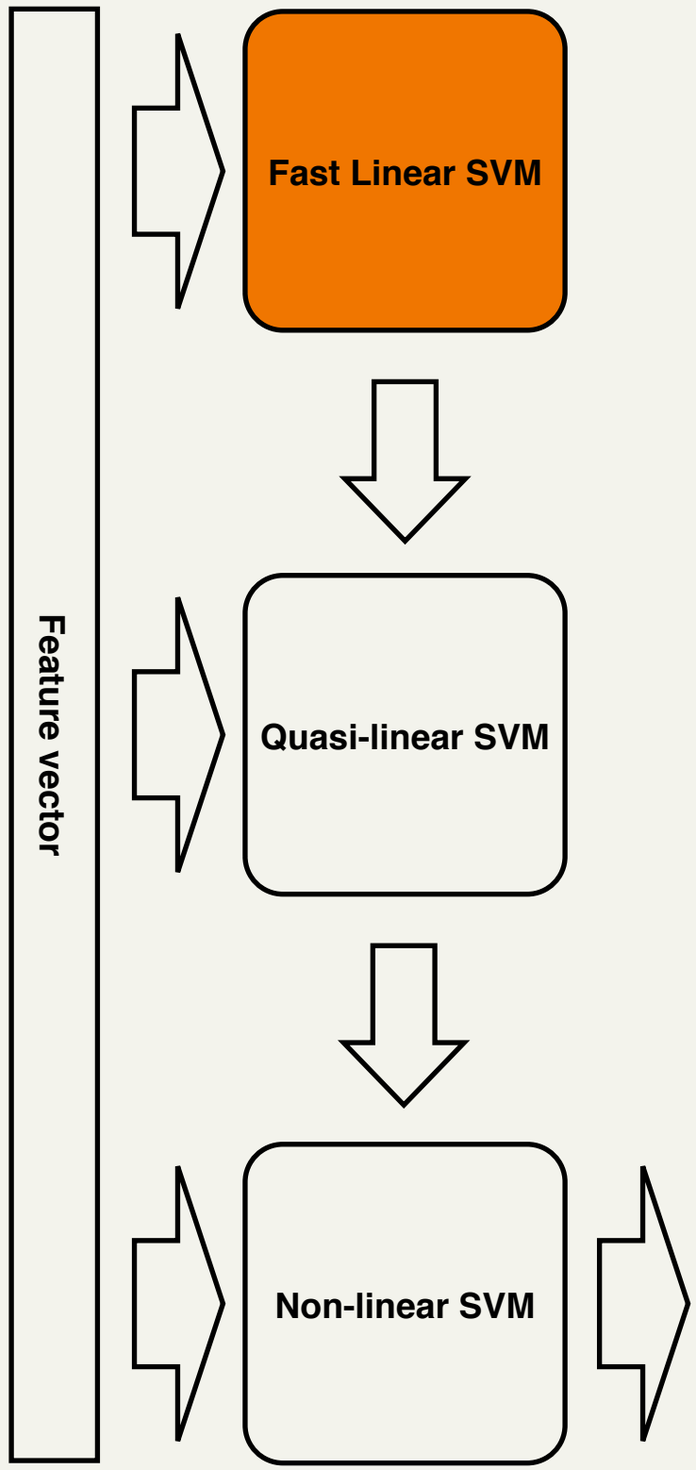
Thus SVM score rewrites:

$$\sum_{j=1}^{d} \sum_{i=1}^{N} \beta_j k(x_j, y_j)$$

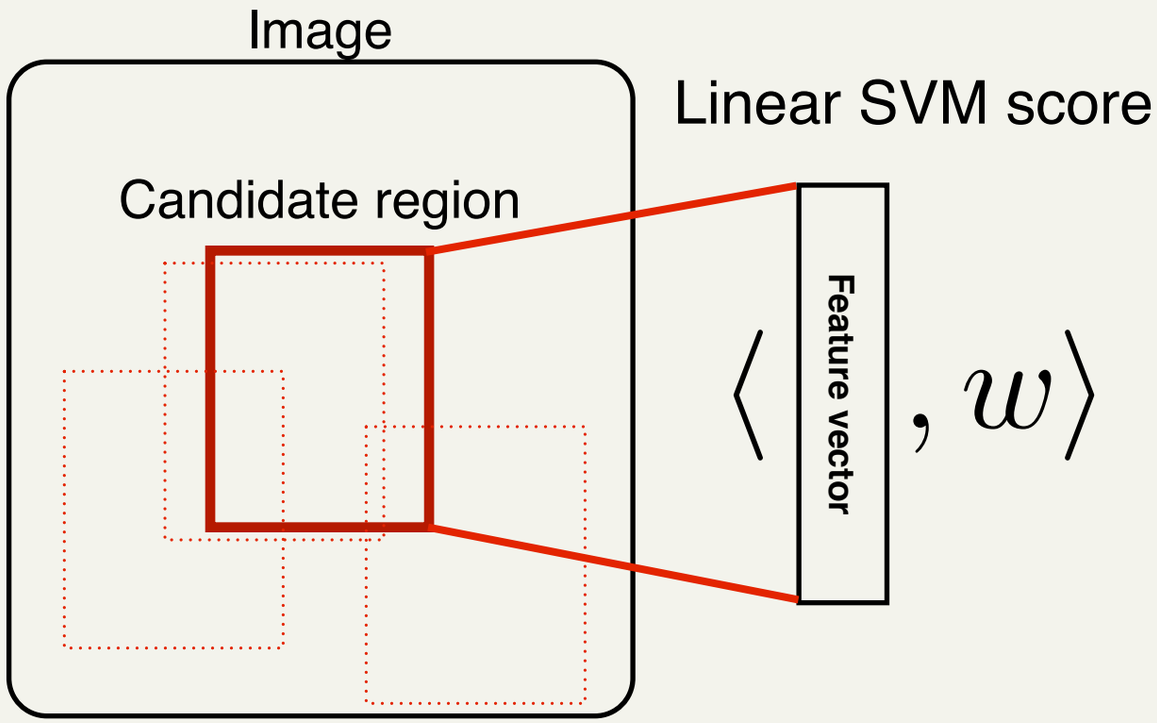
$$\psi_i(y_i)$$

Pre-compute look-up table.

Cascade


Feature vector
Fast Linear SVM
Quasi-linear SVM
Non-linear SVM

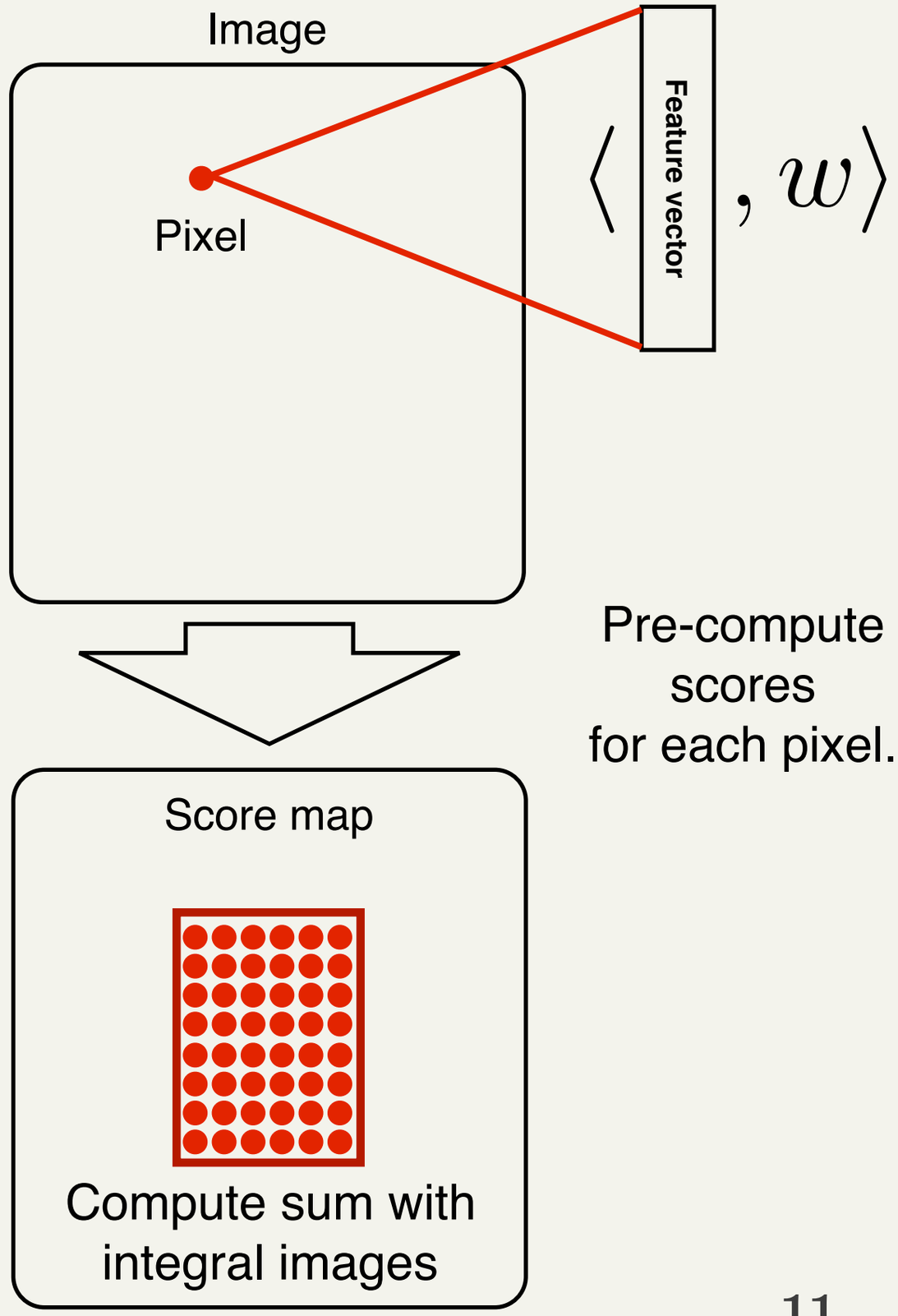
Fast linear SVM



Time required:
#dimensions × #windows × #SVs

#windows

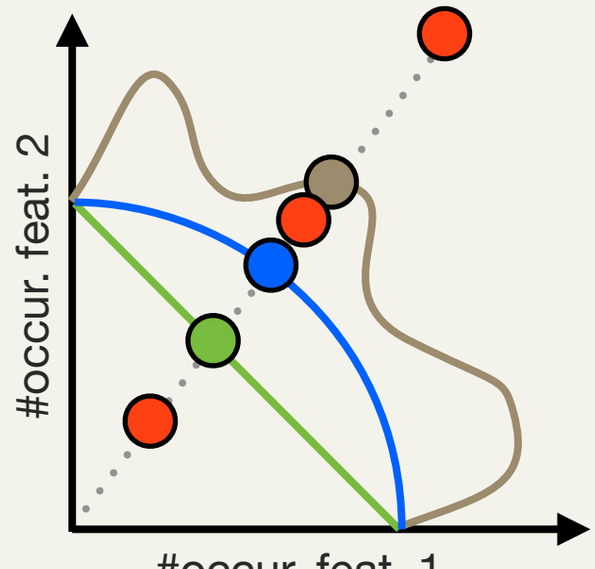
Additional speedup possible with branch and bound
[Lampert Blaschko Hofmann 2008]



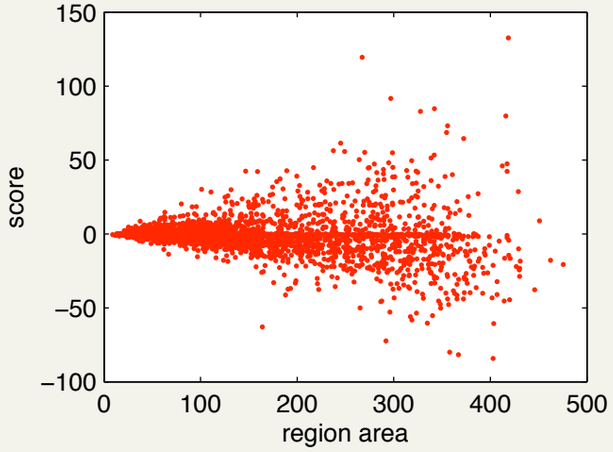
Pre-compute
scores
for each pixel.

Histogram *normalization*

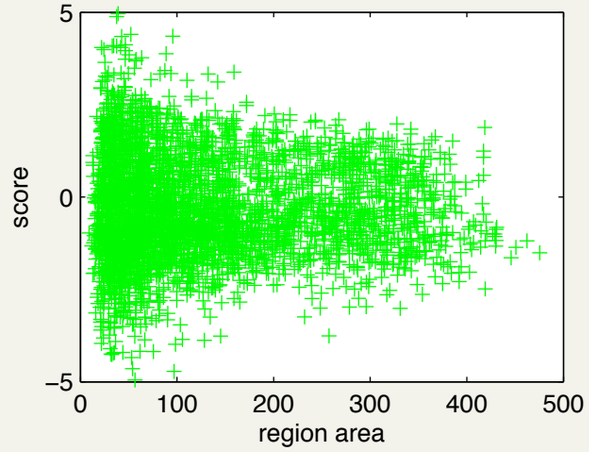
- Invariance to #features (region area)
- Kernel as similarity
  - *An image region should be most similar to itself*
  - $\forall h, h' : K(h,h) \geq K(h,h')$
    - l^2 norm for linear kernel
    - l^1 norm for intersection, χ^2, Hellinger kernels



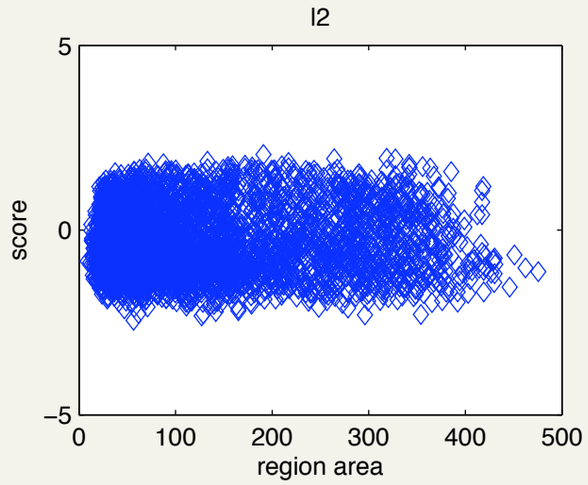
none

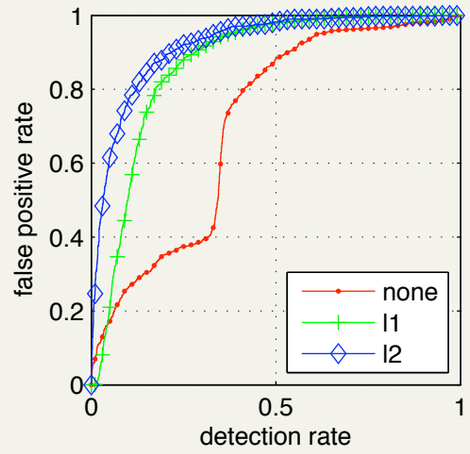


l1



l2



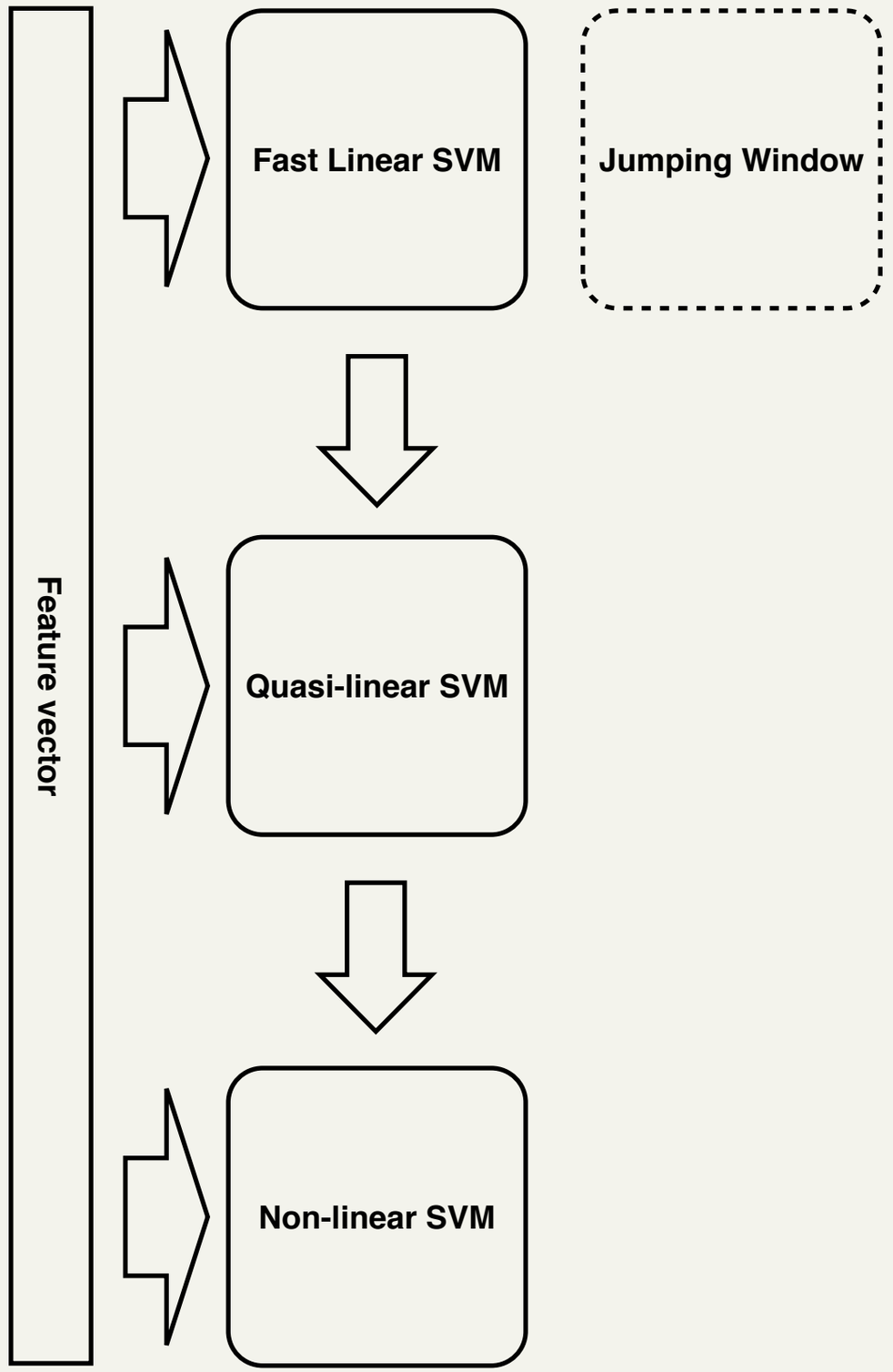


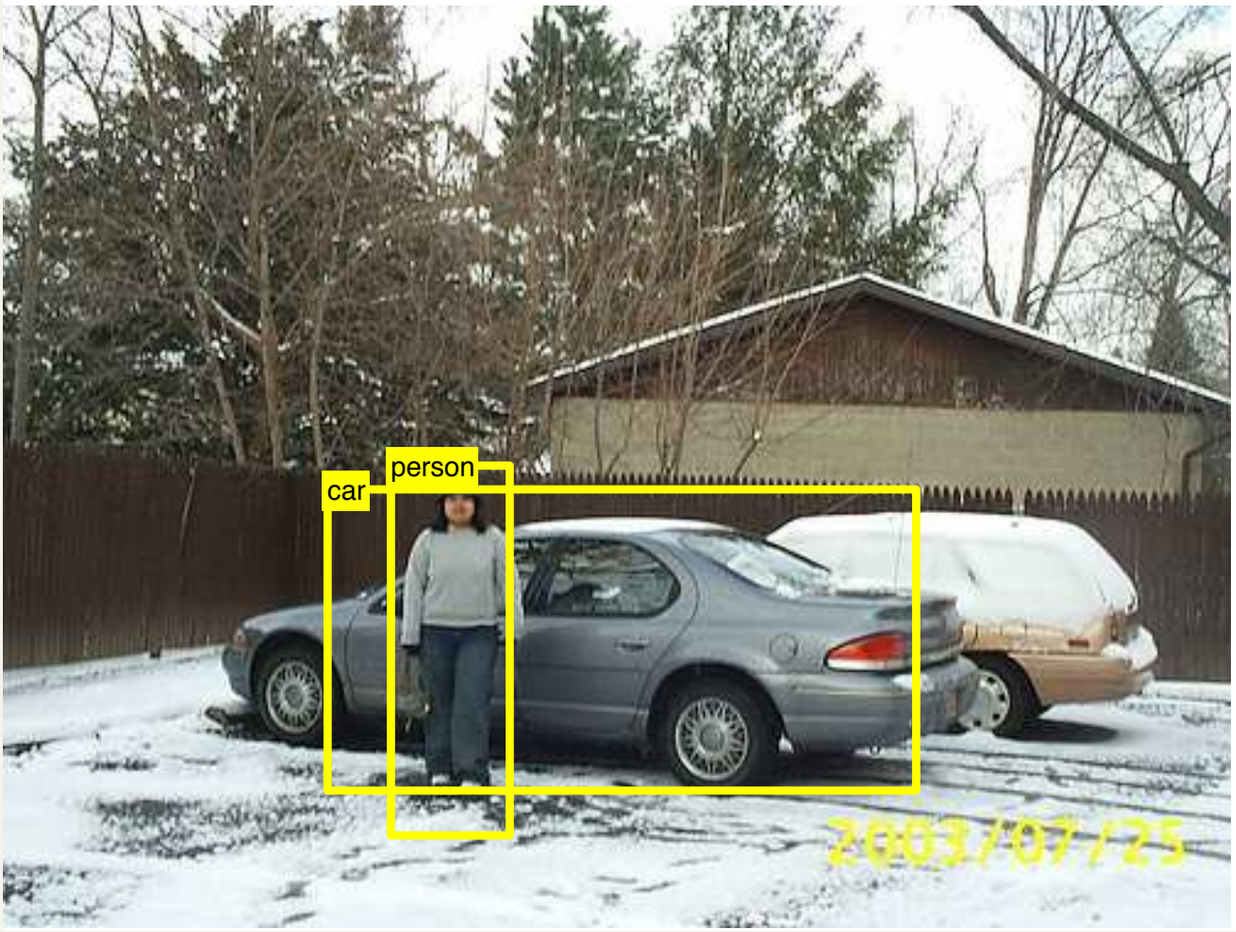
Scatter plot:
linear SVM score vs region area

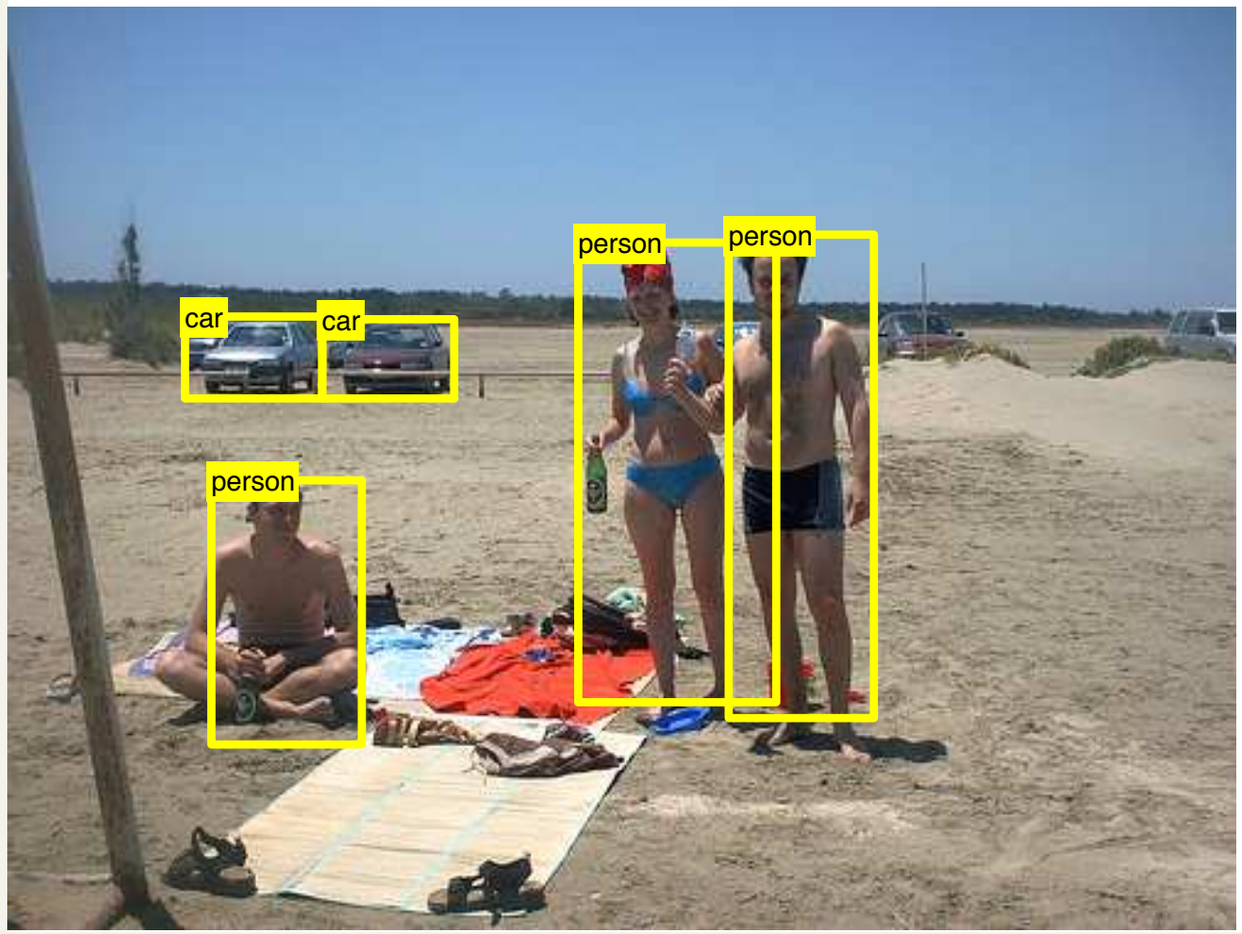
- Linear SVM works better with l2 normalization
- Fast linear SVM requires no or l1 normalization

SVMs overview

- **First stage**
  - linear SVM
  - (or jumping window)
  - time: #windows
- **Second stage**
  - quasi-linear SVM
  - χ^2 kernel
  - time: #windows × #dimensions
- **Third stage**
  - non-linear SVM
  - χ^2-RBF kernel
  - time:
    #windows × #dimensions × #SVs

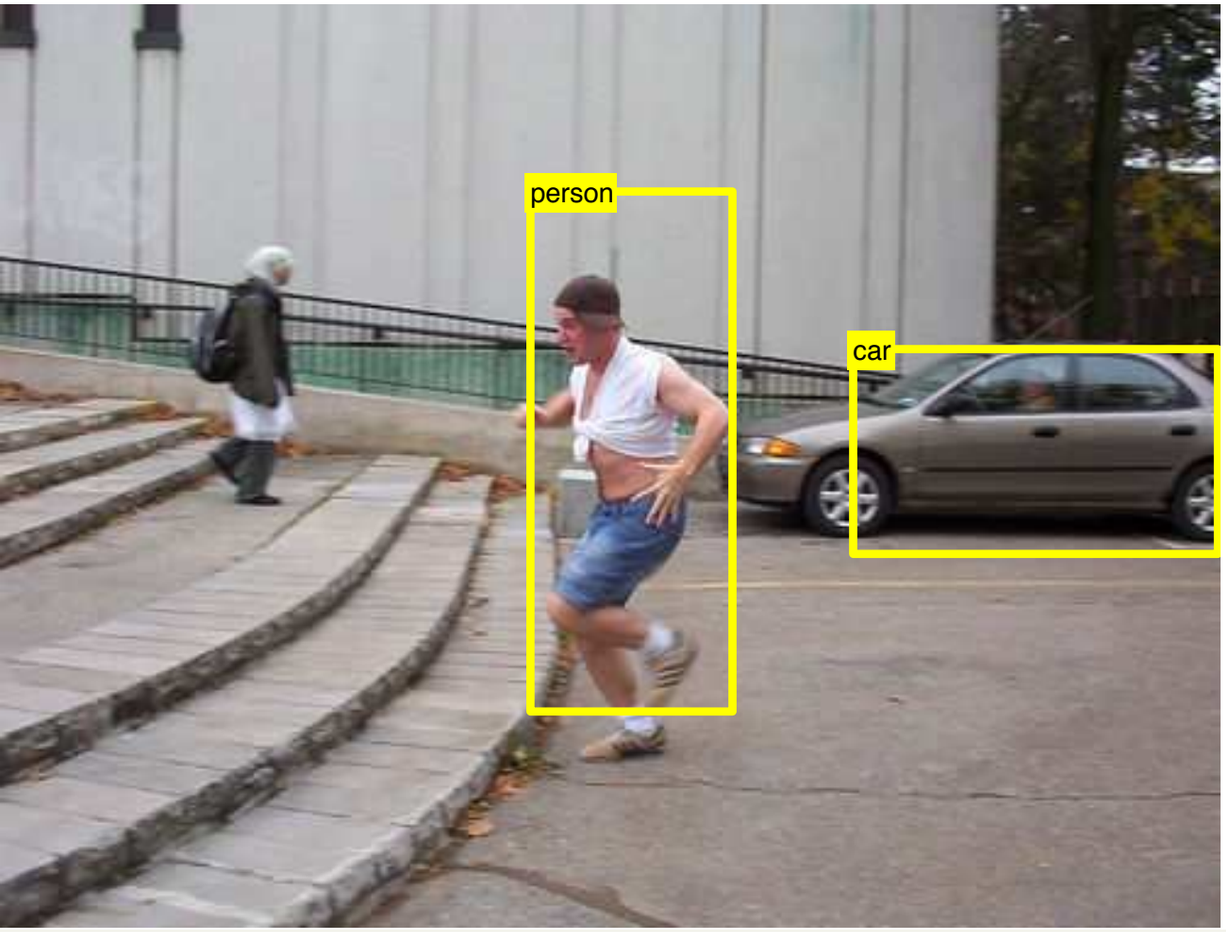


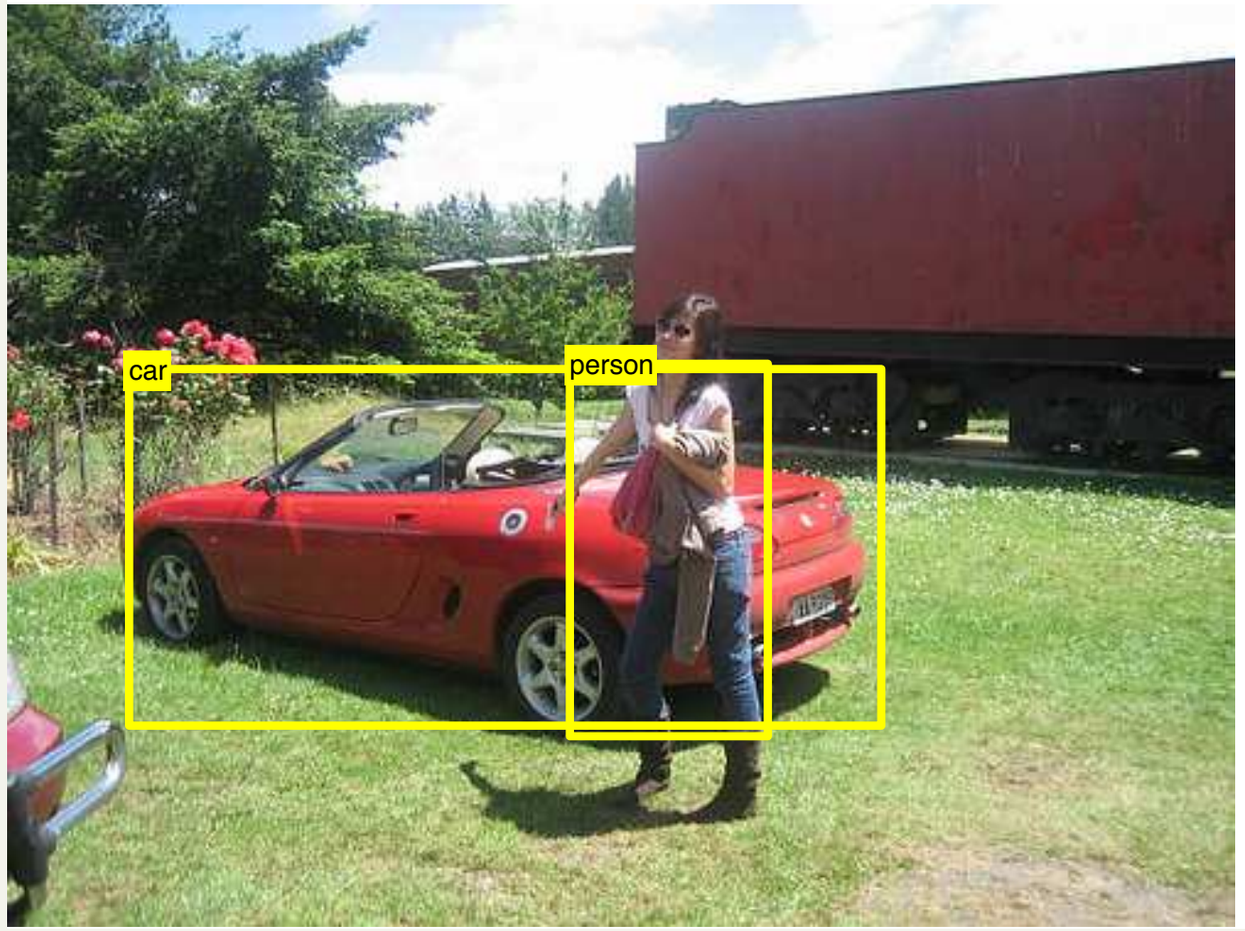

car
person
2003/07/25

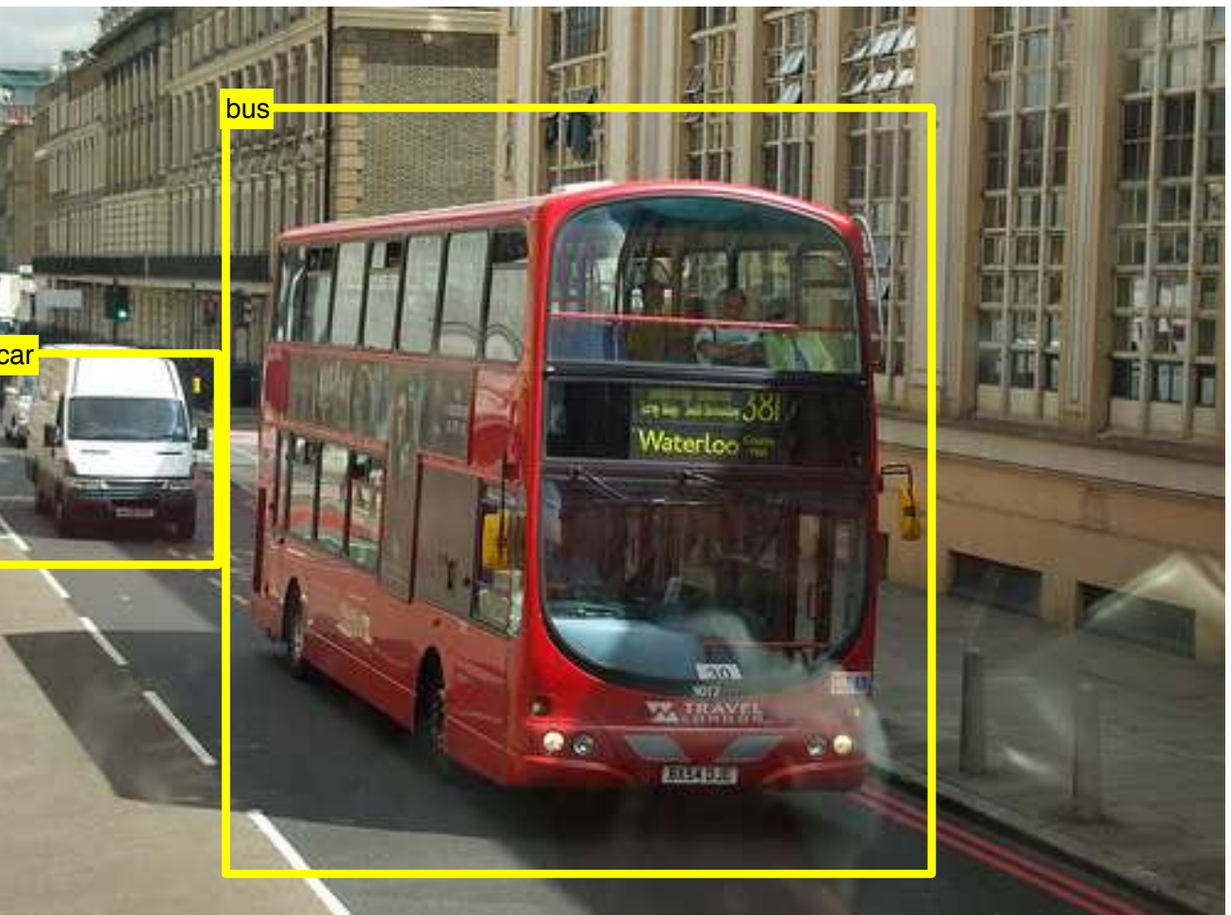

car
car
person
person
person


car
person
HONDA
Stessl

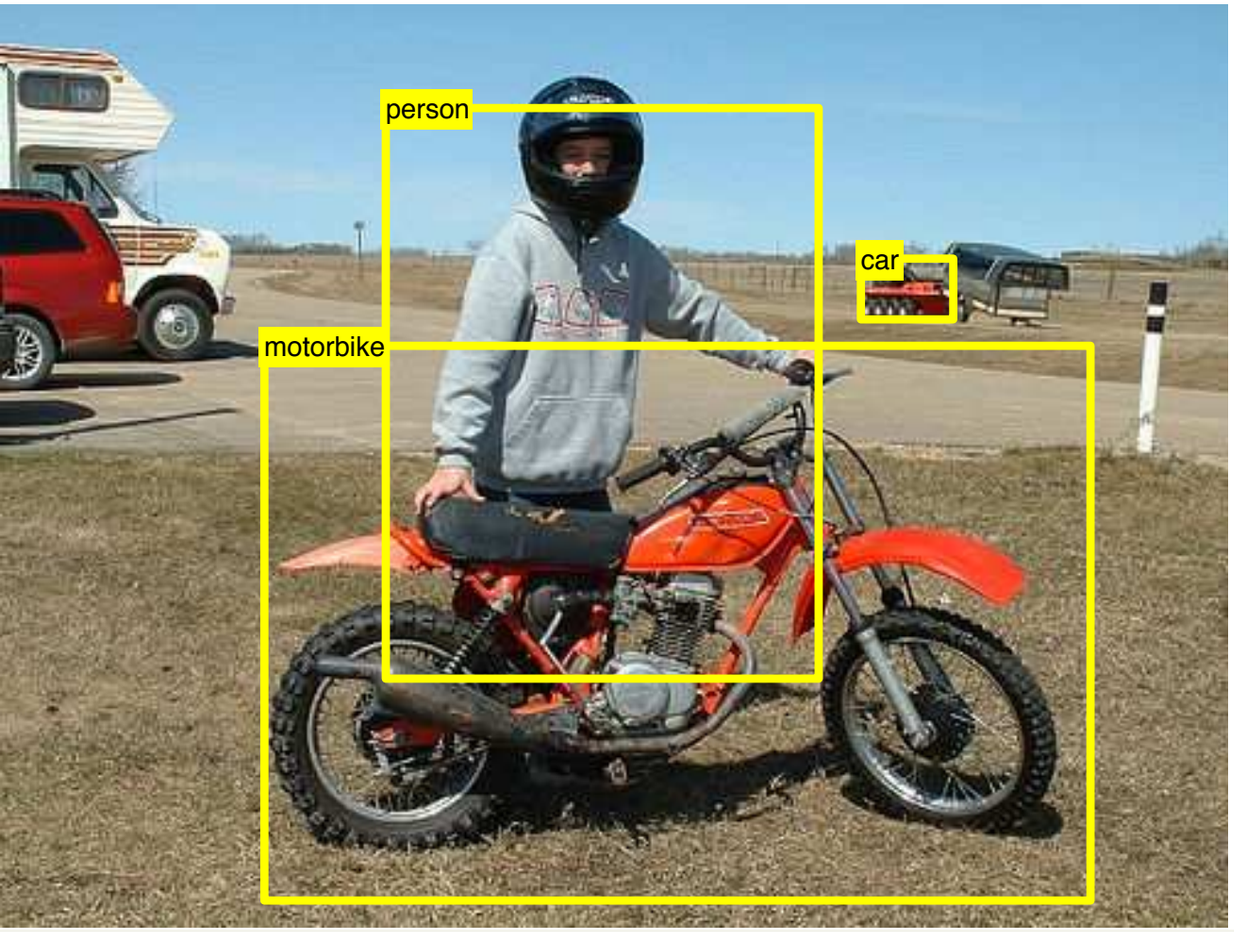

person
car
person

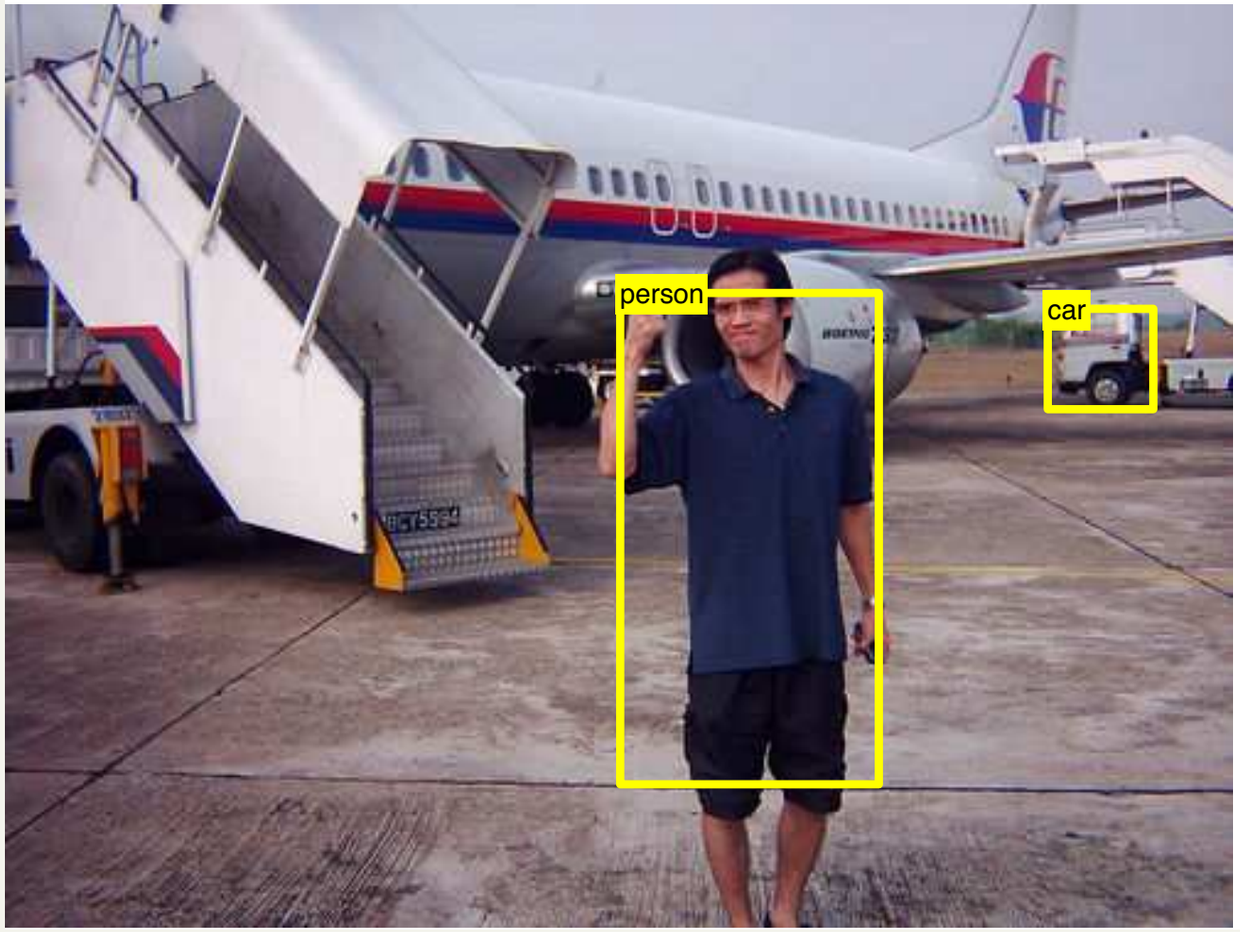

person
car

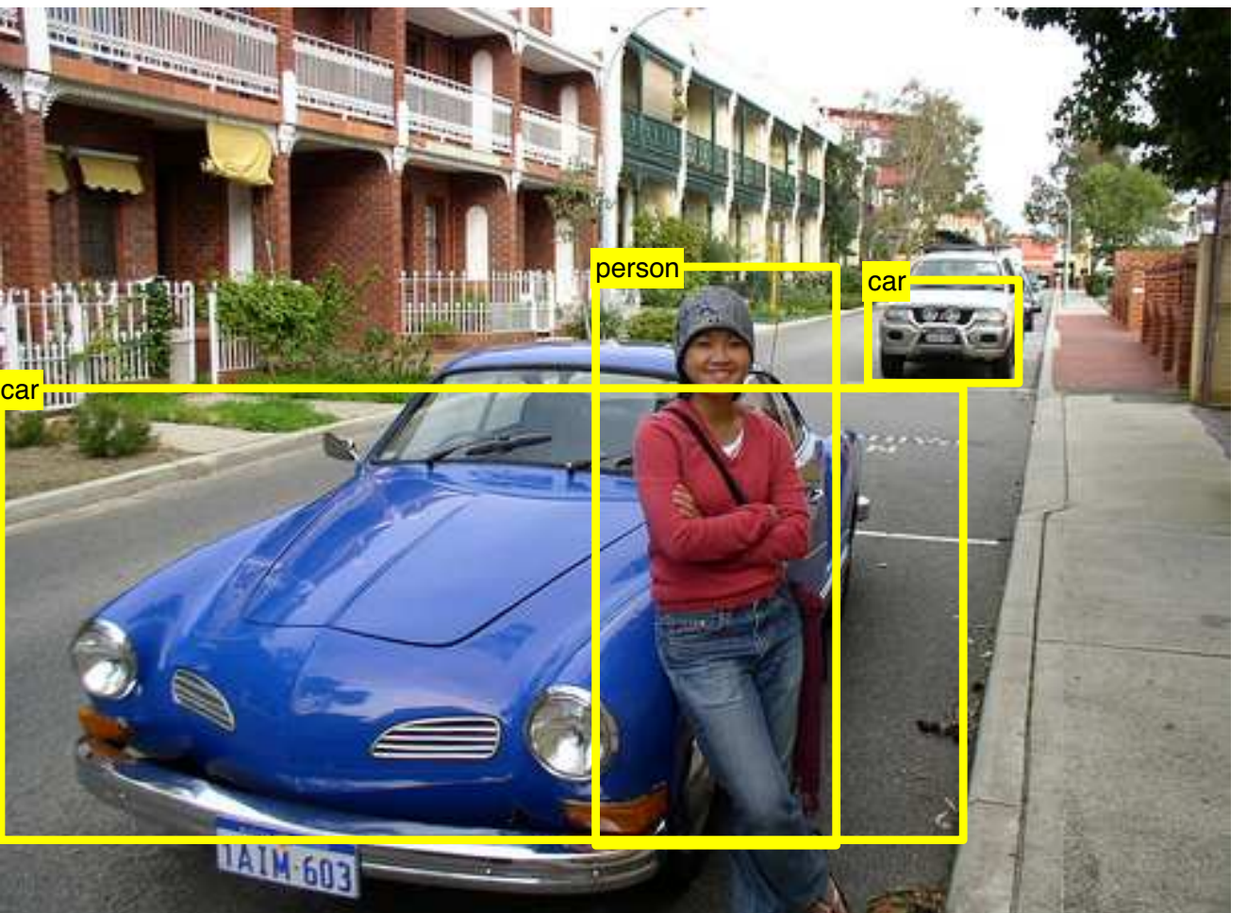

car
person

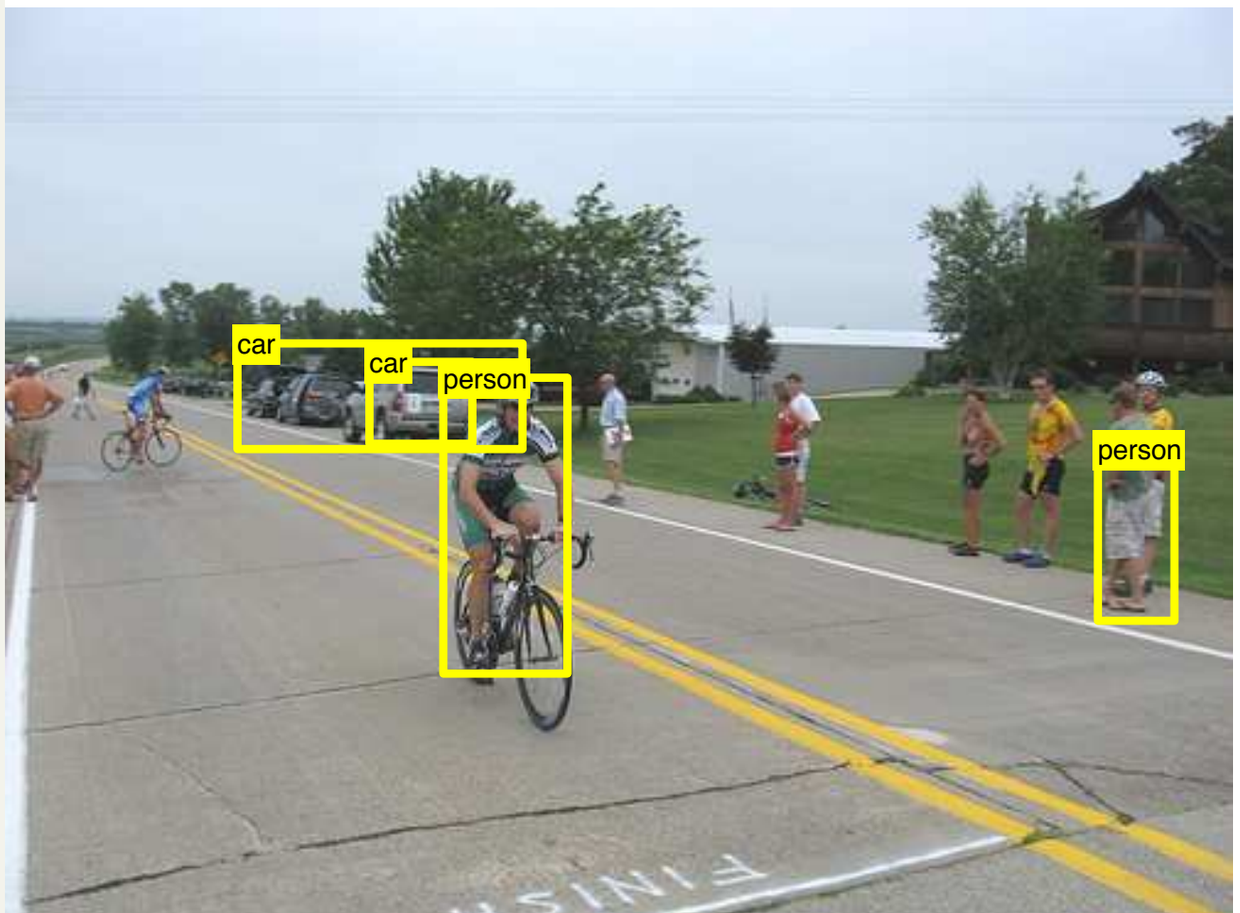

bus
car
Waterloo
381


car
person


person
car
motorbike

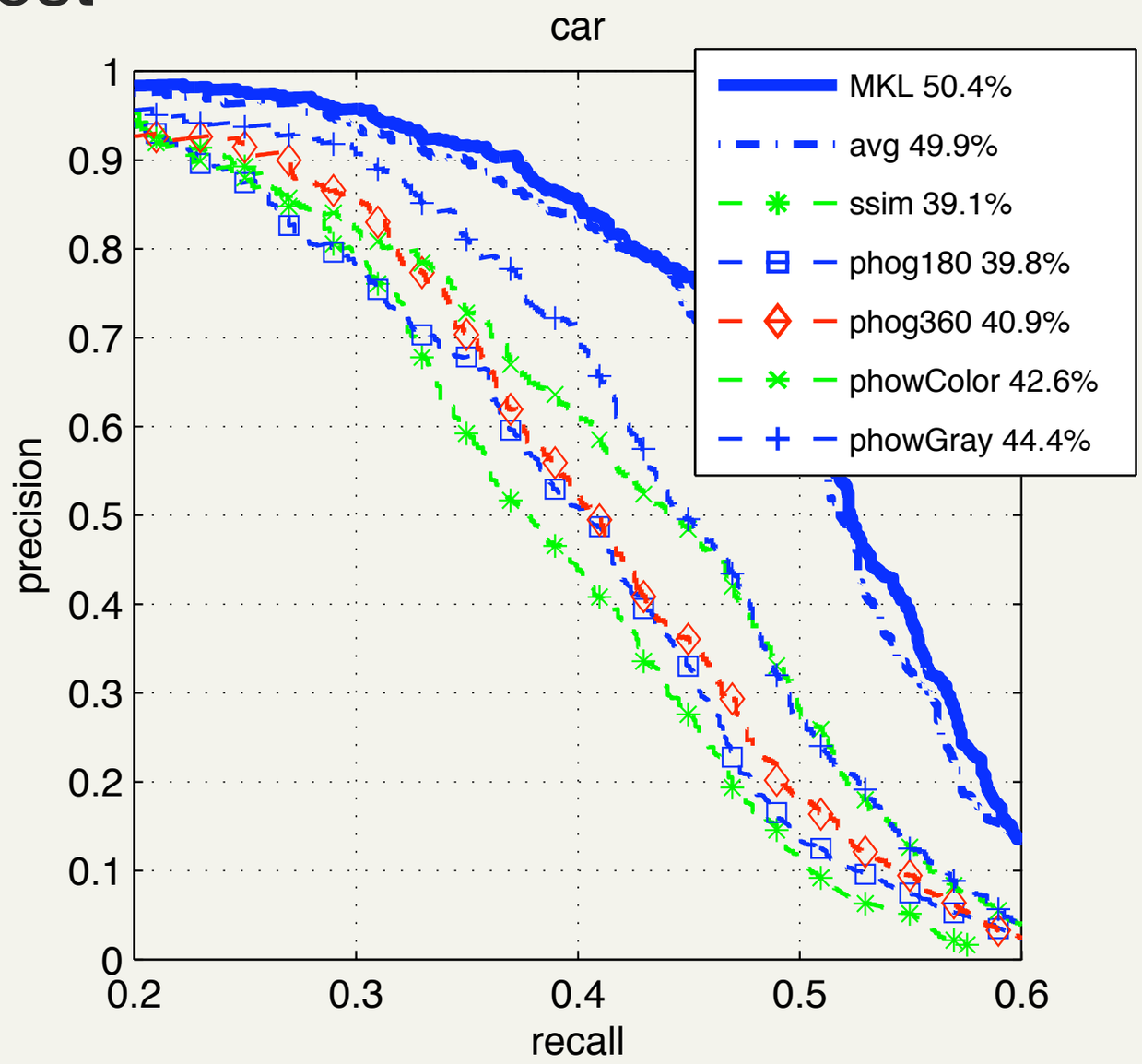

person
car


person
car
car
TAIM 603

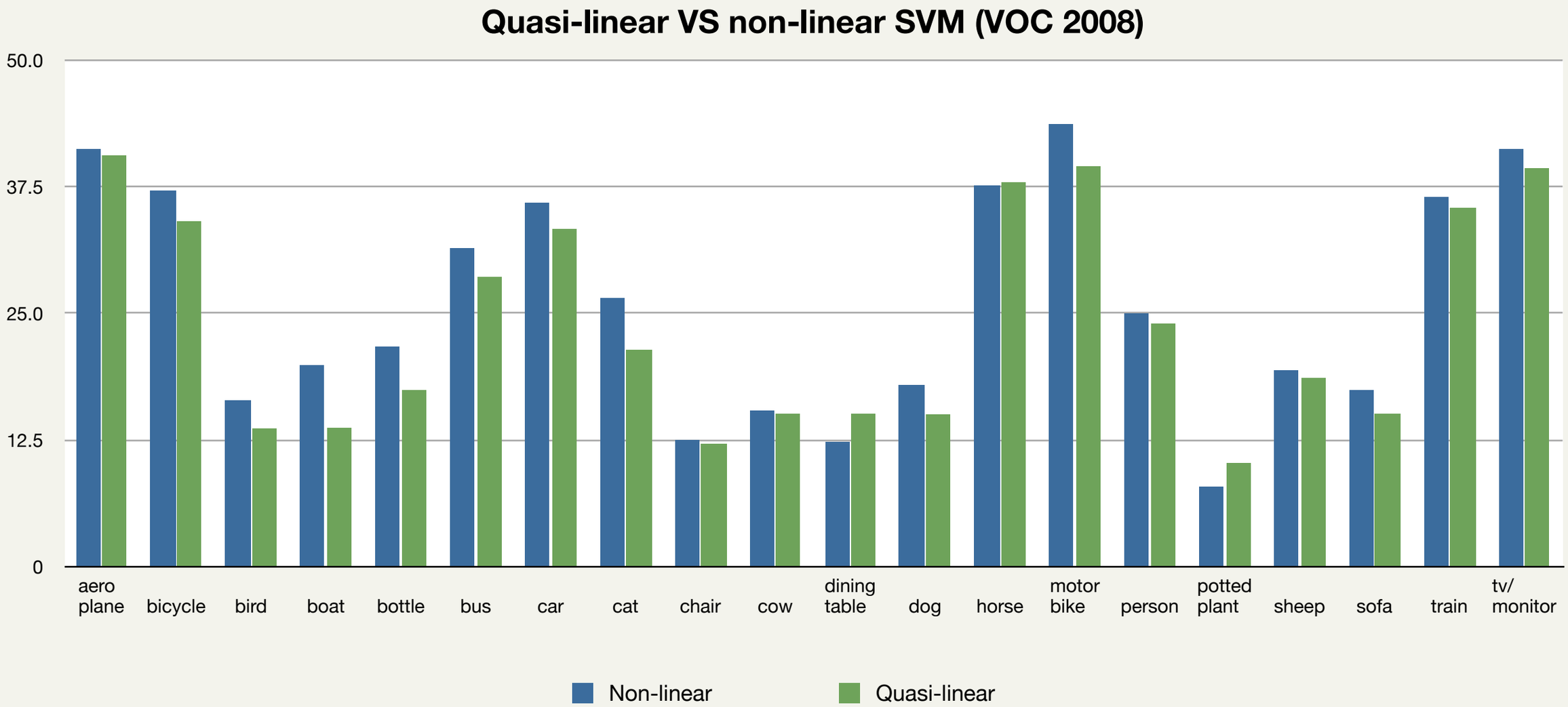

car
car
person
person

Single kernel *vs* multiple kernels

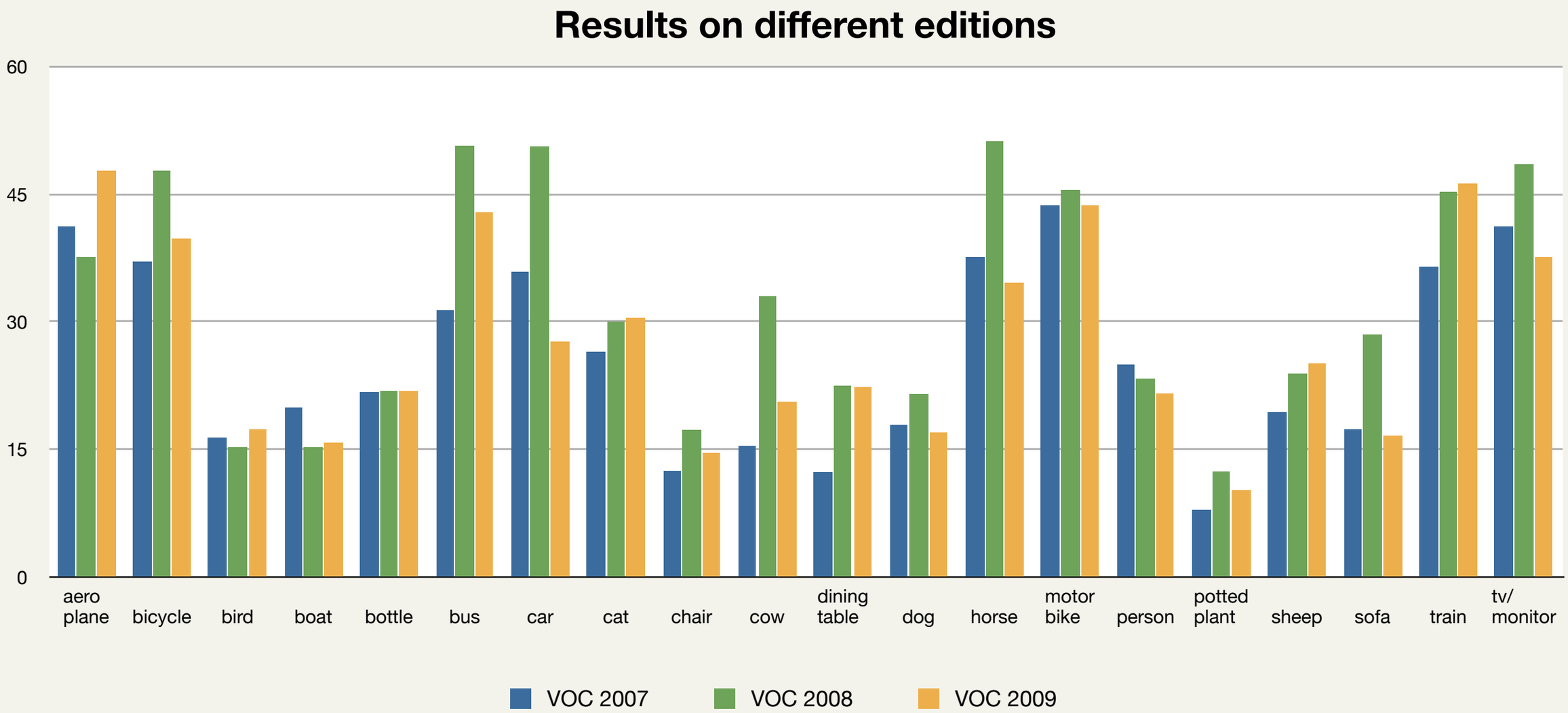
- *Multiple Kernels* ⇒ substantial boost
- Multiple Kernel *Learning* ⇒
  - marginal boost over averaging
  - sparse feature selection
- Consistent with [Gehler Nowozin 09]



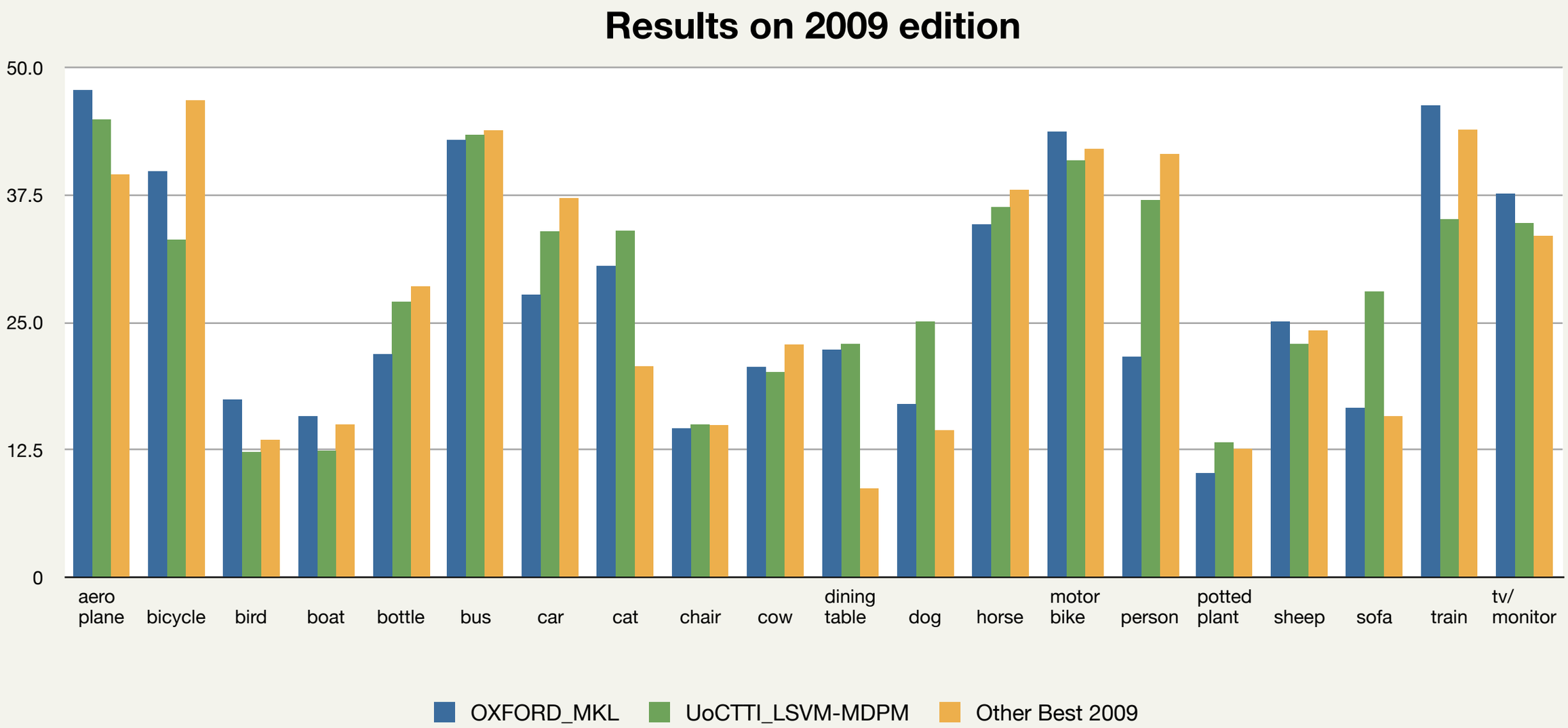
Quasi-linear *vs* non-linear kernels



2007 *vs* 2008 *vs* 2009



VOC 2009 results



Conclusions

- Hierarchy of kernel “structures”
  - trade-off speed and power with the same data/algorithm
- Histogram normalization
  - affects the results
  - should be selected based on the kernel
    - consistency criterion

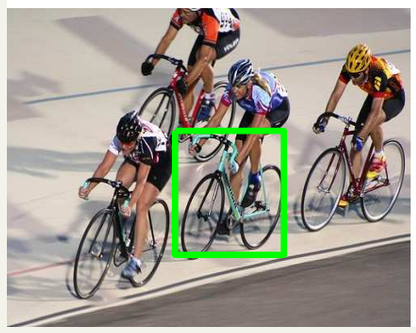
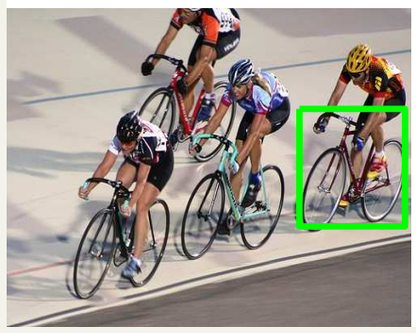
- MK
  - large boost from feature combination
  - sparse feature selection from MK *learning*

- MK *classification* code available
  http://www.robots.ox.ac.uk/~vgg/software/MKL/
- MK *detection* code will be available soon

Thank You!



aeroplane

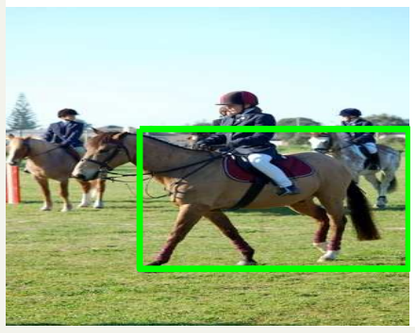

bicycle



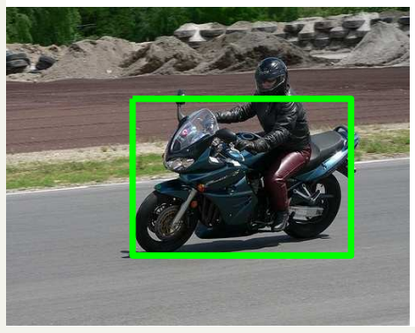
car



cow



horse



motorbike