Structured Regression for Efficient Object Detection

Christoph Lampert
www.christoph-lampert.org

Max Planck Institute for Biological Cybernetics, Tübingen

December 3rd, 2009

- [**C.L.**, Matthew B. Blaschko, Thomas Hofmann. CVPR 2008]
- [Matthew B. Blaschko, **C.L.** ECCV 2008]
- [**C.L.**, Matthew B. Blaschko, Thomas Hofmann. PAMI 2009]

Category-Level Object Localization

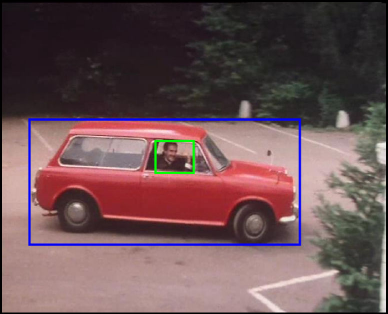
Category-Level Object Localization

What objects are present? **person**, **car**

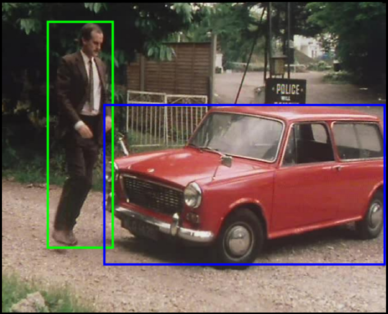
Category-Level Object Localization

Where are the objects?

Object Localization $\Rightarrow$ Scene Interpretation



A man inside of a car
$\Rightarrow$ He's driving.



A man outside of a car
$\Rightarrow$ He's passing by.

Algorithmic Approach: Sliding Window

$\mathbf{f(y_1) = 0.2}$

$\mathbf{f(y_2) = 0.8}$

$\mathbf{f(y_3) = 1.5}$

Use a (pre-trained) classifier function f:

- Place candidate window on the image.
- Iterate:
  - Evaluate f and store result.
  - Shift candidate window by k pixels.
- Return position where f was largest.

Algorithmic approach: Sliding Window

$\mathbf{f(y_1) = 0.2}$

$\mathbf{f(y_2) = 0.8}$

$\mathbf{f(y_3) = 1.5}$

Drawbacks:

- single scale, single aspect ratio
  $\longrightarrow$ repeat with different window sizes/shapes
- search on grid
  $\longrightarrow$ speed–accuracy tradeoff
- computationally expensive

New view: Generalized Sliding Window

Assumptions:

- Objects are rectangular image regions of arbitrary size.
- The score of $\mathbf{f}$ is largest at the correct object position.

Mathematical Formulation:

$$\mathbf{y}^{\text{opt}} = \underset{\mathbf{y}\in\mathcal{Y}}{\mathbf{argmax}}\ \mathbf{f}(\mathbf{y})$$

with $\mathcal{Y} = \{$*all rectangular regions in image*$\}$

New view: Generalized Sliding Window

Mathematical Formulation:

$$\mathbf{y}^{\text{opt}} = \underset{\mathbf{y}\in\mathcal{Y}}{\mathbf{argmax}}\ \mathbf{f}(\mathbf{y})$$

with $\mathcal{Y} = \{$*all rectangular regions in image*$\}$

- **How to choose/construct/learn the function** f **?**
- **How to do the optimization efficiently *and* robustly?** (exhaustive search is too slow, $\mathcal{O}(\mathbf{w^2h^2})$ elements).

New view: Generalized Sliding Window



Mathematical Formulation:

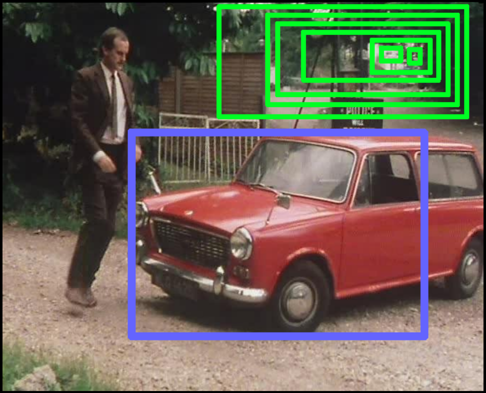
$$\mathbf{y}^{\text{opt}} = \operatorname*{\mathbf{argmax}}_{\mathbf{y}\in\mathcal{Y}} \ \mathbf{f}(\mathbf{y})$$

with $\mathcal{Y} = \{$*all rectangular regions in image*$\}$

- **How to choose/construct/learn the function f ?**
- **How to do the optimization efficiently *and* robustly?**
  (exhaustive search is too slow, $\mathcal{O}(\mathbf{w^2 h^2})$ elements).

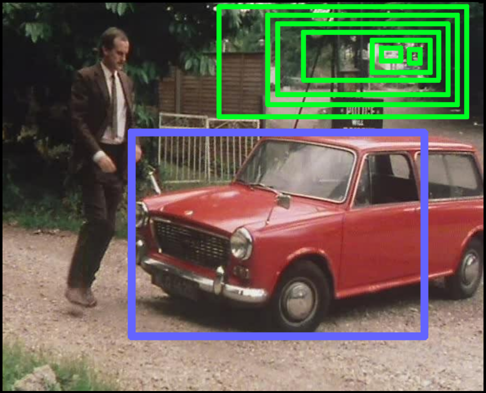
New view: Generalized Sliding Window

Use the problem's *geometric structure*:



New view: Generalized Sliding Window

Use the problem's *geometric structure*:



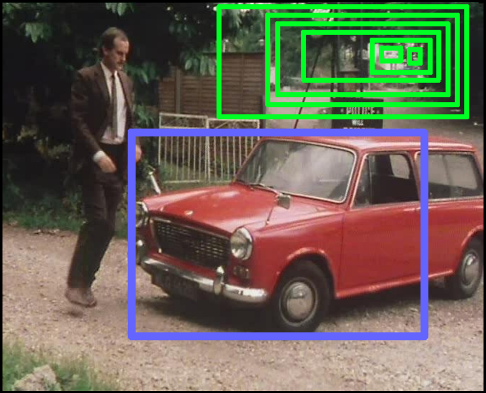
- Calculate scores for *sets of boxes* jointly.
- If no element can contain the maximum, discard the box set.
- Otherwise, split the box set and iterate.

$\longrightarrow$ Branch-and-bound optimization

- finds global maximum $\mathbf{y}^{\text{opt}}$

New view: Generalized Sliding Window

Use the problem's *geometric structure*:



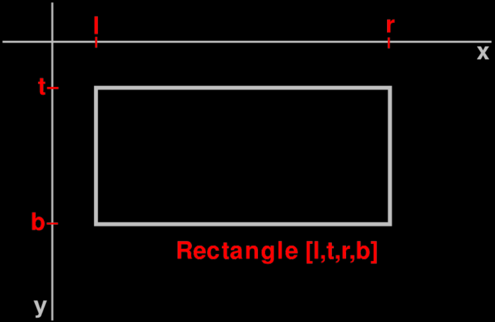
- Calculate scores for *sets of boxes* jointly.
- If no element can contain the maximum, discard the box set.
- Otherwise, split the box set and iterate.

$\longrightarrow$ Branch-and-bound optimization

- finds global maximum $\mathbf{y}^{\mathrm{opt}}$

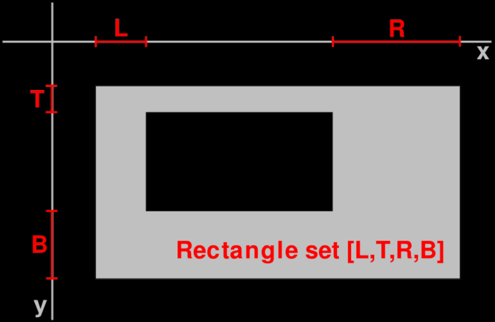
Representing Sets of Boxes

- Boxes: $[\mathbf{l}, \mathbf{t}, \mathbf{r}, \mathbf{b}] \in \mathbb{R}^4$.



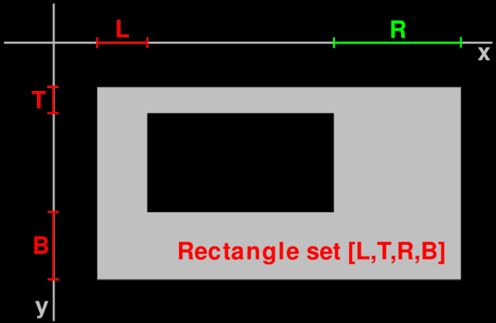
Representing Sets of Boxes

- Boxes: $[\mathbf{l}, \mathbf{t}, \mathbf{r}, \mathbf{b}] \in \mathbb{R}^4$. Boxsets: $[\mathbf{L}, \mathbf{T}, \mathbf{R}, \mathbf{B}] \in (\mathbb{R}^2)^4$



Representing Sets of Boxes

- Boxes: $[\mathbf{l}, \mathbf{t}, \mathbf{r}, \mathbf{b}] \in \mathbb{R}^4$. Boxsets: $[\mathbf{L}, \mathbf{T}, \mathbf{R}, \mathbf{B}] \in (\mathbb{R}^2)^4$

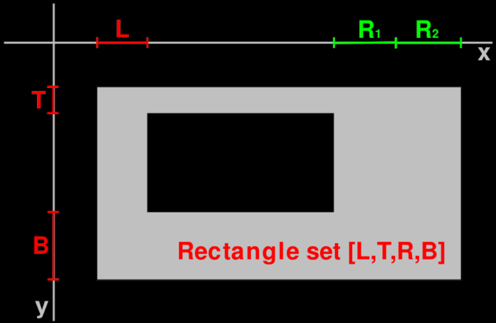


Splitting:

- Identify largest interval.

Representing Sets of Boxes

- Boxes: $[\mathbf{l}, \mathbf{t}, \mathbf{r}, \mathbf{b}] \in \mathbb{R}^4$. Boxsets: $[\mathbf{L}, \mathbf{T}, \mathbf{R}, \mathbf{B}] \in (\mathbb{R}^2)^4$

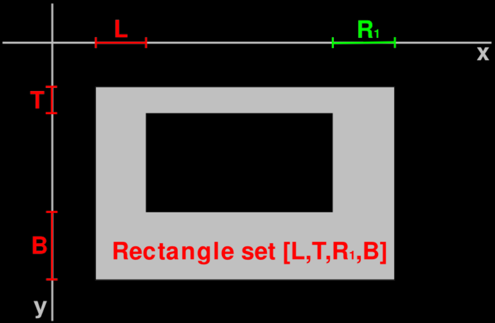


Splitting:

- Identify largest interval. Split at center: $\mathbf{R} \mapsto \mathbf{R_1} \cup \mathbf{R_2}$.

Representing Sets of Boxes

- Boxes: $[\mathbf{l}, \mathbf{t}, \mathbf{r}, \mathbf{b}] \in \mathbb{R}^4$. Boxsets: $[\mathbf{L}, \mathbf{T}, \mathbf{R}, \mathbf{B}] \in (\mathbb{R}^2)^4$

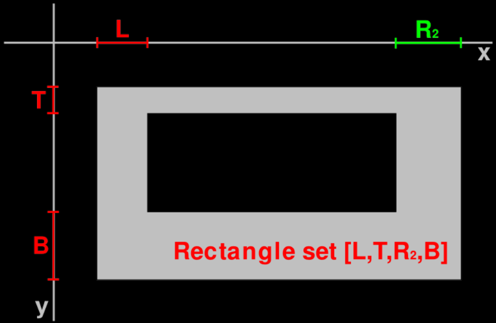


Splitting:

- Identify largest interval. Split at center: $\mathbf{R} \mapsto \mathbf{R_1} \cup \mathbf{R_2}$.
- New box sets: $[\mathbf{L}, \mathbf{T}, \mathbf{R_1}, \mathbf{B}]$

Representing Sets of Boxes

- Boxes: $[\mathbf{l},\mathbf{t},\mathbf{r},\mathbf{b}] \in \mathbb{R}^4$. Boxsets: $[\mathbf{L},\mathbf{T},\mathbf{R},\mathbf{B}] \in (\mathbb{R}^2)^4$

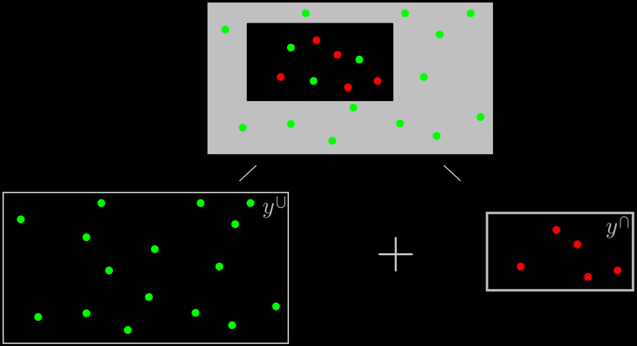


Splitting:

- Identify largest interval. Split at center: $\mathbf{R} \mapsto \mathbf{R_1} \cup \mathbf{R_2}$.
- New box sets: $[\mathbf{L},\mathbf{T},\mathbf{R_1},\mathbf{B}]$ and $[\mathbf{L},\mathbf{T},\mathbf{R_2},\mathbf{B}]$.

Calculating Scores for Box Sets

Example: Linear Support-Vector-Machine $\mathbf{f}(\mathbf{y}) := \sum_{\mathbf{p_i} \in \mathbf{y}} \mathbf{w_i}$.

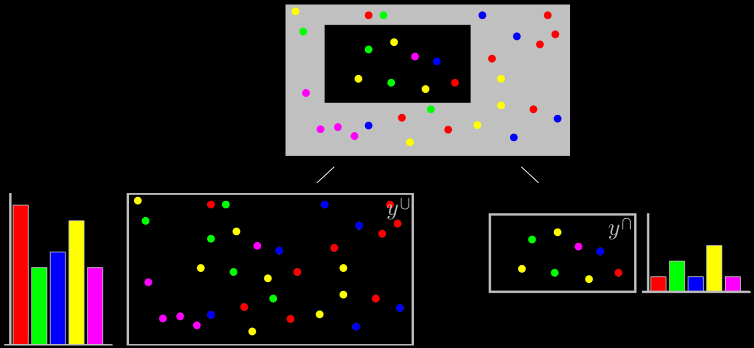


$$\mathbf{f}^{\text{upper}}(\mathbf{Y}) = \sum_{\mathbf{p_i} \in \mathbf{y}^{\cap}} \min(0, \mathbf{w_i}) + \sum_{\mathbf{p_i} \in \mathbf{y}^{\cup}} \max(0, \mathbf{w_i})$$

Can be computed in $\mathcal{O}(1)$ using *integral images*.

Calculating Scores for Box Sets

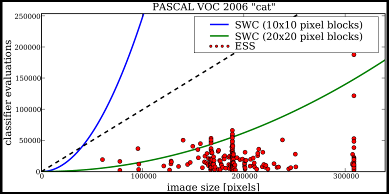
Histogram Intersection Similarity: $f(\mathbf{y}) := \sum_{j=1}^{J} \min(h'_j, h^{\mathbf{y}}_j)$.

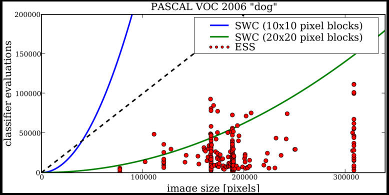


$$f^{\text{upper}}(\mathbf{Y}) = \sum_{j=1}^{J} \min(h'_j, h^{\mathbf{y}^{\cup}}_j)$$

As fast as for a single box: $\mathcal{O}(J)$ with *integral histograms*.

Evaluation: Speed (on PASCAL VOC 2006)





Sliding Window Runtime:

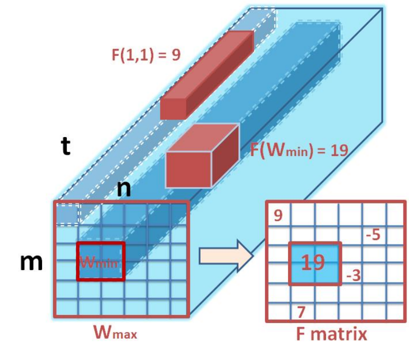
- always: $\mathcal{O}(\mathbf{w^2h^2})$

Branch-and-Bound (ESS) Runtime:

- worst-case: $\mathcal{O}(\mathbf{w^2h^2})$
- empirical: not more than $\mathcal{O}(\mathbf{wh})$

Extensions:

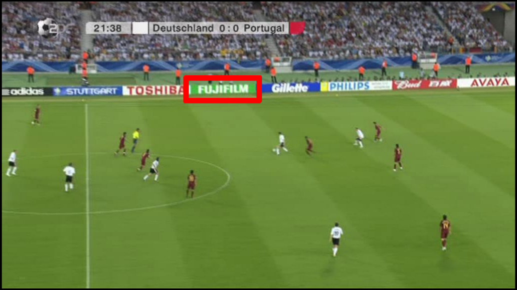
Action classification: $(y,t)^{\text{opt}} = \mathbf{argmax}_{(y,t)\in\mathcal{Y}\times T} f_x(y,t)$



- J. Yuan: *Discriminative 3D Subvolume Search for Efficient Action Detection*, CVPR 2009.

Extensions:

Localized image retrieval: $(x, y)^{\text{opt}} = \mathbf{argmax}_{y \in \mathcal{Y},\, x \in \mathcal{D}}\, f_x(y)$



- C.L.: *Detecting Objects in Large Image Collections and Videos by Efficient Subimage Retrieval*, ICCV 2009

Extensions:

Hybrid – Branch-and-Bound with Implicit Shape Model

- A. Lehmann, B. Leibe, L. van Gool: *Feature-Centric Efficient Subwindow Search*, ICCV 2009

Generalized Sliding Window

$$\mathbf{y}^{\mathrm{opt}} = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} \ \mathbf{f}(\mathbf{y})$$

with $\mathcal{Y} = \{$*all rectangular regions in image*$\}$

- **How to choose/construct/learn f ?**
- **How to do the optimization efficiently *and* robustly?**

Traditional Approach: *Binary Classifier*

Training images:

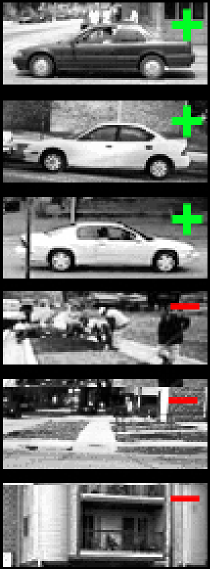
- $\mathbf{x}_1^+, \dots, \mathbf{x}_n^+$ show the object
- $\mathbf{x}_1^-, \dots, \mathbf{x}_m^-$ show something else

Train a classifier, e.g.

- support vector machine,
- boosted cascade,
- artificial neural network,...

Decision function $\mathbf{f} : \{\mathbf{images}\} \to \mathbb{R}$

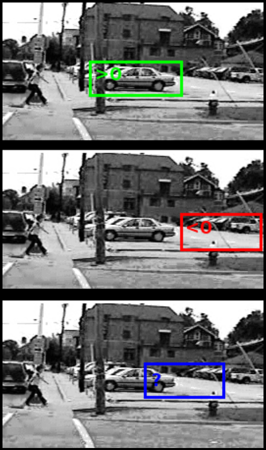
- $f > 0$ means "*image shows the object.*"
- $f < 0$ means "*image does not show the object.*"



Traditional Approach: *Binary Classifier*

Drawbacks:

- Train distribution
  $\neq$ test distribution
- No control over partial detections.
- No guarantee to even find training examples again.



Object Localization as Structured Output Regression

Ideal setup:

- function

$$g : \{\textit{all images}\} \longrightarrow \{\textit{all boxes}\}$$

  to predict object boxes from images
- train and test in the same way, end-to-end

$g_{car}\Bigg($ 

$\Bigg) =$ 

Object Localization as Structured Output Regression

Ideal setup:

- function

$$g : \{\textit{all images}\} \longrightarrow \{\textit{all boxes}\}$$

  to predict object boxes from images
- train and test in the same way, end-to-end

Regression problem:

- training examples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$
  - x_i are images, y_i are bounding boxes
- Learn a mapping

$$g : \mathcal{X} \longrightarrow \mathcal{Y}$$

  that generalizes from the given examples:
  - $g(x_i) \approx y_i$, for $i = 1, \ldots, n$,

Structured Support Vector Machine

SVM-like framework by *Tsochantaridis et al.*:

- Positive definite kernel $k : (\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y}) \rightarrow \mathbb{R}$.
  $\boldsymbol{\varphi} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{H}$: (implicit) feature map induced by k.
- $\boldsymbol{\Delta} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$: loss function
- Solve the *convex* optimization problem

$$\mathbf{min}_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$

  subject to margin constraints for $i=1,\dots,n$:

$$\forall y \in \mathcal{Y} \setminus \{y_i\} : \boldsymbol{\Delta}(y, y_i) + \langle w, \boldsymbol{\varphi}(x_i, y)\rangle - \langle w, \boldsymbol{\varphi}(x_i, y_i)\rangle \leq \xi_i,$$

- unique solution: $w^* \in \mathcal{H}$

- I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun: Large Margin Methods for Structured and Interdependent Output Variables, Journal of Machine Learning Research (JMLR), 2005.

Structured Support Vector Machine

- w^* defines *compatiblity* function

$$F(x,y) = \langle w^*, \boldsymbol{\varphi}(x,y) \rangle$$

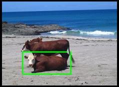
- best prediction for x is the *most compatible* y:

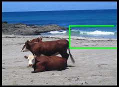
$$g(x) := \operatorname*{\mathbf{argmax}}_{y \in \mathcal{Y}} F(x,y).$$

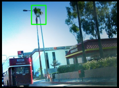
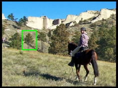
- evaluating $g : \mathcal{X} \to \mathcal{Y}$ is like generalized Sliding Window:
  - for fixed x, evaluate quality function for every box $y \in \mathcal{Y}$.
  - for example, use previous branch-and-bound procedure!

Joint Image/Box-Kernel: Example

Joint kernel: how to compare one (image,box)-pair (x, y) with another (image,box)-pair (x', y')?

$k_{joint}\Big($  ,  $\Big) = k\Big($  ,  $\Big)$ is large.

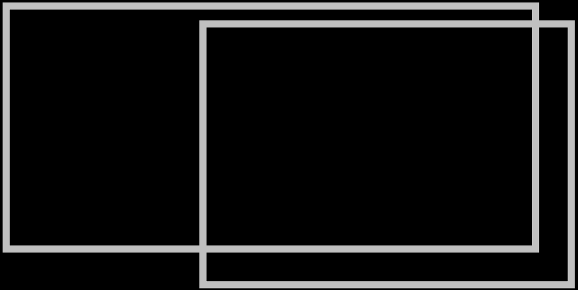
$k_{joint}\Big($  ,  $\Big) = k\Big($  ,  $\Big)$ is small.

$k_{joint}\Big($  ,  $\Big) = k_{image}\Big($  ,  $\Big)$

could also be large.

Loss Function: Example

Loss function: how to compare two boxes y and y'?



$$\begin{aligned}\Delta(y, y') &:= 1 - \textit{area overlap} \text{ between } y \text{ and } y' \\ &= 1 - \frac{\text{area}(y \cap y')}{\text{area}(y \cup y')}\end{aligned}$$

Structured Support Vector Machine

- S-SVM Optimization: $\mathbf{min}_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$
  subject to for $i=1,\ldots,n$:

$$\forall y \in \mathcal{Y} \setminus \{y_i\} : \ \boldsymbol{\Delta}(y, y_i) + \langle w, \boldsymbol{\varphi}(x_i, y)\rangle - \langle w, \boldsymbol{\varphi}(x_i, y_i)\rangle \leq \xi_i,$$

Structured Support Vector Machine

- S-SVM Optimization: $\mathbf{min}_{w,\xi}\ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$

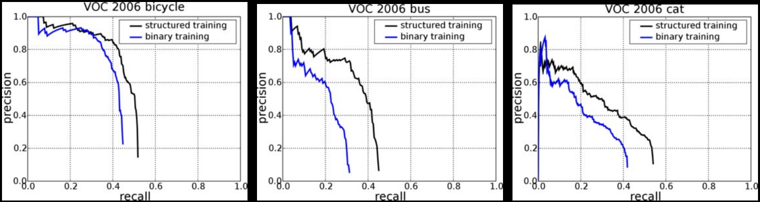
  subject to for $i=1,\ldots,n$:

$$\forall y \in \mathcal{Y}\setminus\{y_i\}:\ \boldsymbol{\Delta}(y,y_i) + \langle w, \boldsymbol{\varphi}(x_i,y)\rangle - \langle w, \boldsymbol{\varphi}(x_i,y_i)\rangle \le \xi_i,$$

- Solve via *constraint generation*:
- Iterate:
  - Solve minimization with working set of contraints
  - Identify $\mathbf{argmax}_{y\in\mathcal{Y}}\,\boldsymbol{\Delta}(y,y_i) + \langle w, \boldsymbol{\varphi}(x_i,y)\rangle$
  - Add violated constraints to working set and iterate
- Polynomial time convergence to any precision ε
- Similar to bootstrap training, but with a margin.

Evaluation: PASCAL VOC 2006

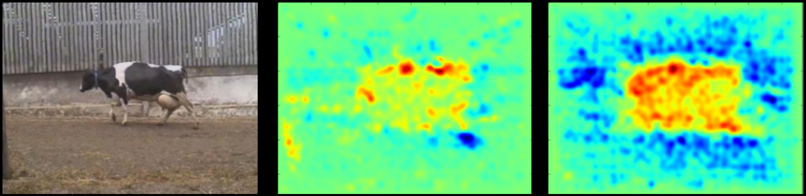
Example detections for VOC 2006 `bicycle`, `bus` and `cat`.



Precision–recall curves for VOC 2006 `bicycle`, `bus` and `cat`.

- Structured regression improves detection accuracy.
- New best scores (at that time) in 6 of 10 classes.

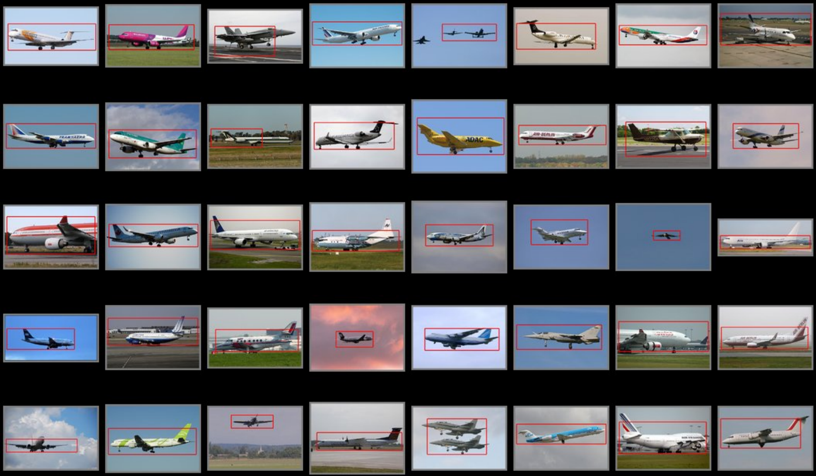
Why does it work?



Learned weights from binary (center) and structured training (right).

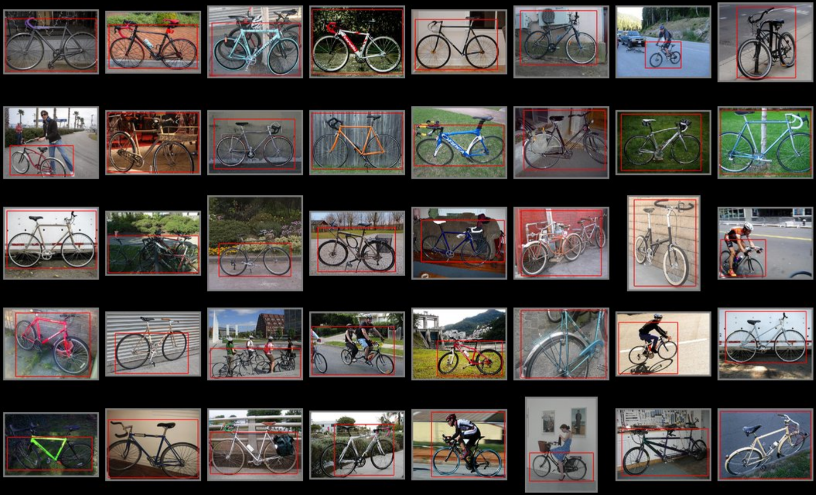
- Both methods assign positive weights to object region.
- Structured training also assigns negative weights to features surrounding the bounding box position.
- Posterior distribution over box coordinates becomes more peaked.

More Recent Results (PASCAL VOC 2009)



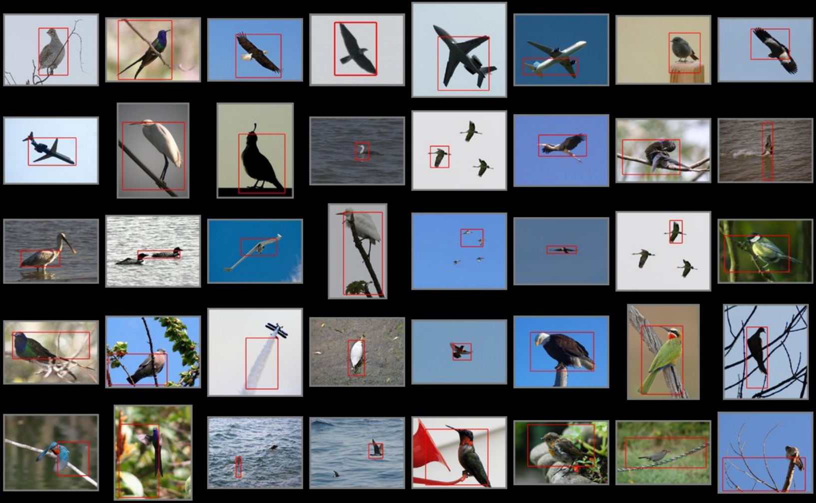
aeroplane

More Recent Results (PASCAL VOC 2009)



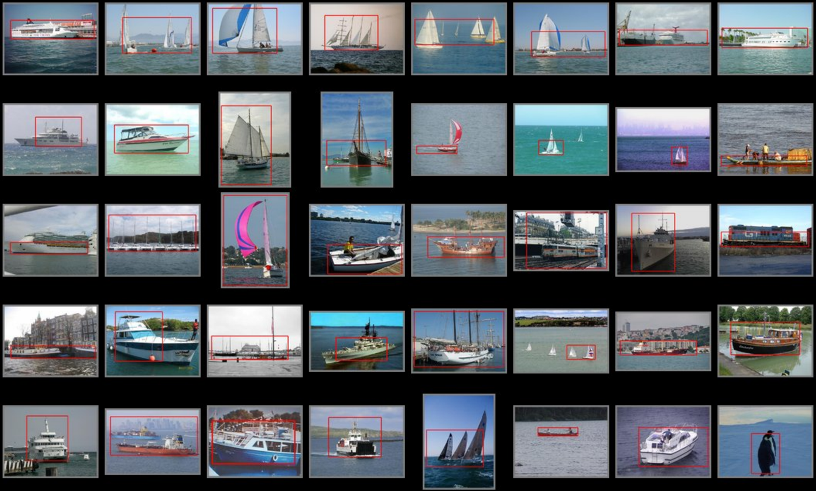
bicycle

More Recent Results (PASCAL VOC 2009)



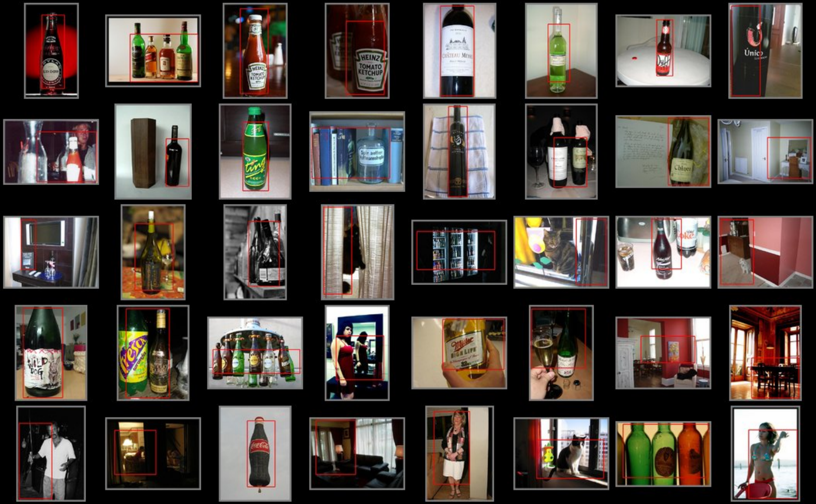
bird

More Recent Results (PASCAL VOC 2009)



boat

More Recent Results (PASCAL VOC 2009)

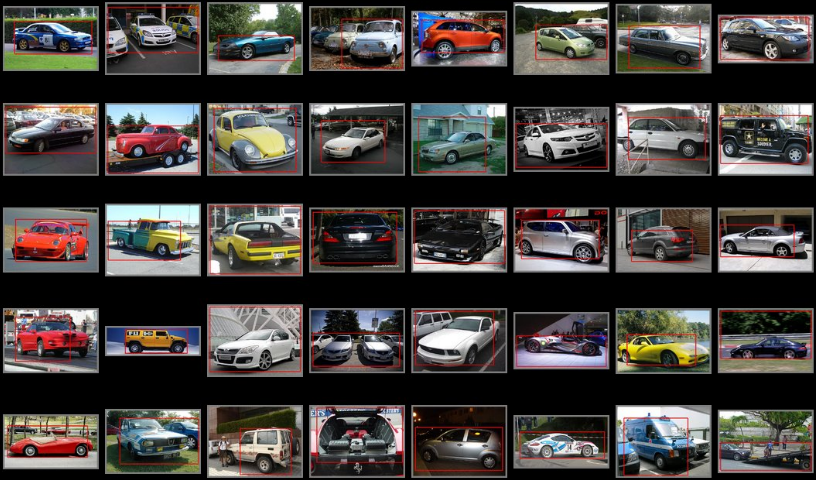


bottle

More Recent Results (PASCAL VOC 2009)

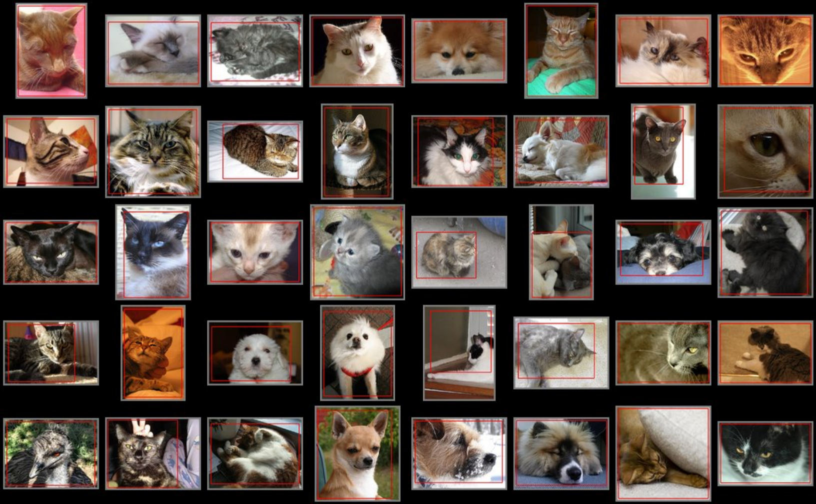
bus

More Recent Results (PASCAL VOC 2009)



car

More Recent Results (PASCAL VOC 2009)



cat

More Recent Results (PASCAL VOC 2009)

chair

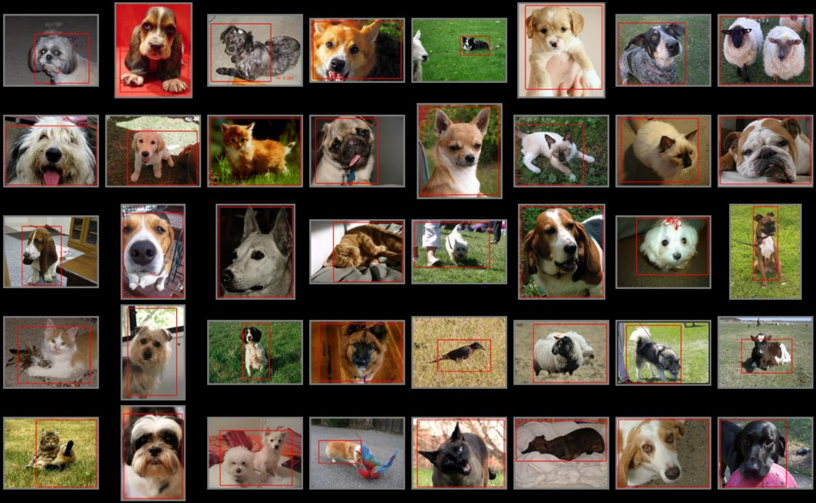
More Recent Results (PASCAL VOC 2009)

cow

More Recent Results (PASCAL VOC 2009)

diningtable

More Recent Results (PASCAL VOC 2009)

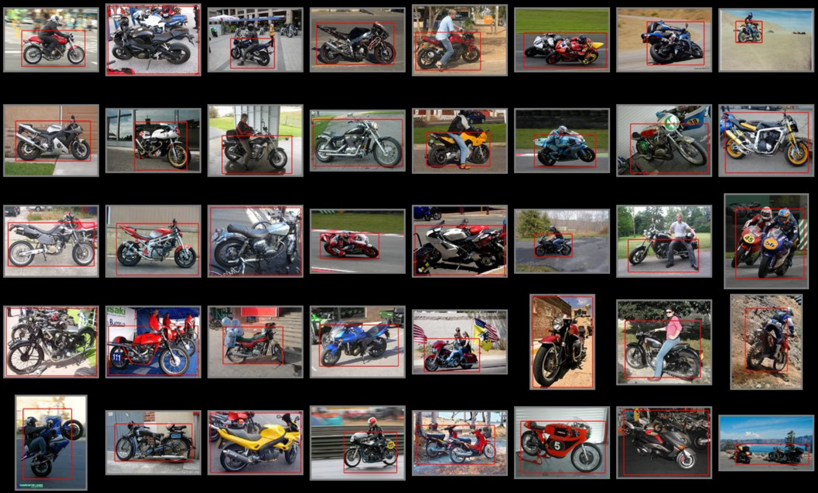


dog

More Recent Results (PASCAL VOC 2009)

horse

More Recent Results (PASCAL VOC 2009)

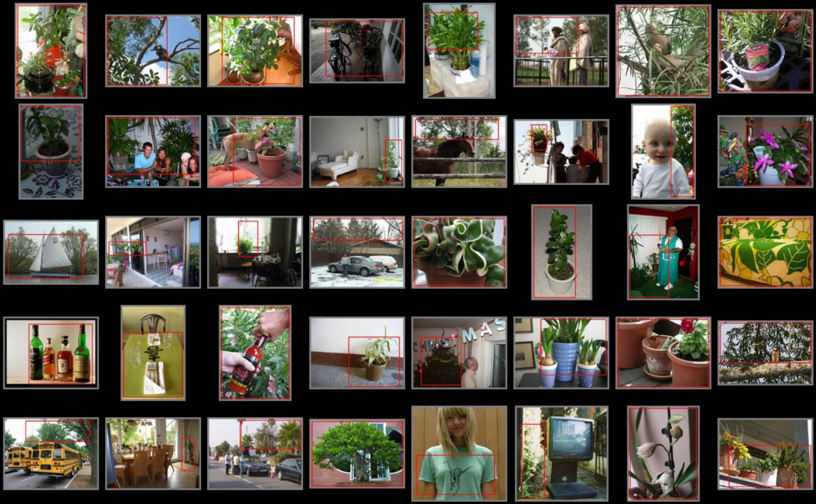


motorbike

More Recent Results (PASCAL VOC 2009)

person

More Recent Results (PASCAL VOC 2009)



pottedplant

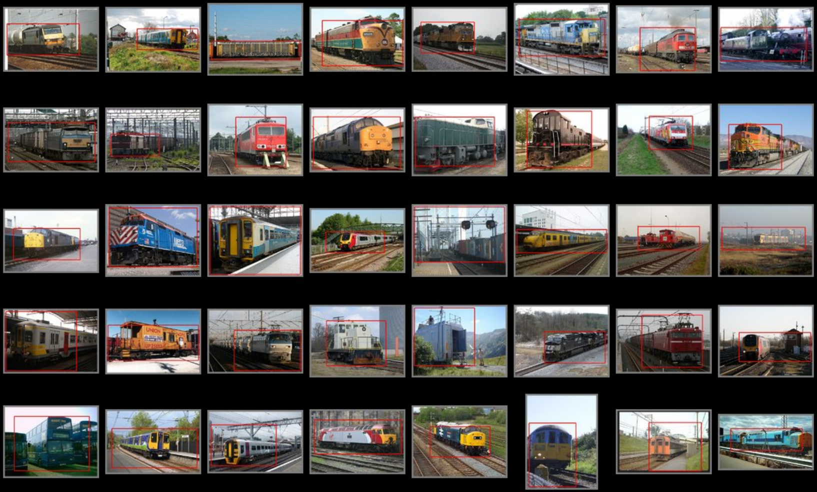
More Recent Results (PASCAL VOC 2009)

sheep

More Recent Results (PASCAL VOC 2009)

sofa

More Recent Results (PASCAL VOC 2009)



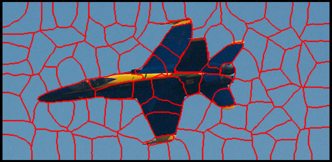
train

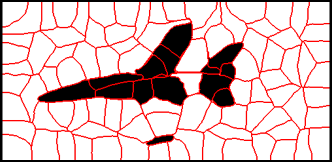
More Recent Results (PASCAL VOC 2009)

tvmonitor

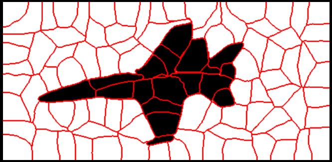
Extensions:

Image segmentation with connectedness constraint:





CRF segmentation



connected CRF segmentation

- S. Nowozin, C.L.: *Global Connectivity Potentials for Random Field Models*, CVPR 2009.

Summary

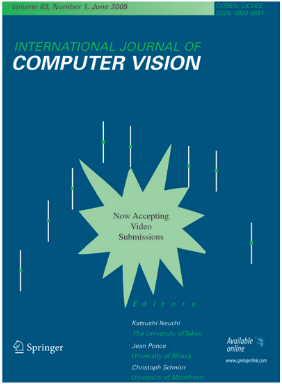
Object Localization is a step towards image *interpretation*.

Conceptual approach instead of *algorithmic*:

- *Branch-and-bound* evaluation:
  - don't slide a window, but solve an $\mathbf{argmax}$ problem, $\Rightarrow$ **higher efficiency**
- *Structured regression* training:
  - solve the prediction problem, not a classification proxy. $\Rightarrow$ **higher localization accuracy**
- Modular and kernelized:
  - easily adapted to other problems/representations, e.g. image segmentations

Springer
the language of science

springer.com



Special Issue

Structured Prediciton and Inference

Matthew B. Blaschko
University of Oxford
blaschko@robots.ox.ac.uk

Christoph H. Lampert
Max Planck Institute for Biological Cybernetic
chl@tuebingen.mpg.de

Paper submission:
*** Feb 26, 2010**

Estimate Online Publication:
*** Fall, 2010**

Topics of interest include, but are not limited to:

- Training for structured output learning
    - Probabilistic vs. max-margin training
    - Generative vs. discriminative training
    - Semi-supervised or unsupervised learning
    - Dealing with label noise
- Inference methods for structured output learning
    - Exact vs. approximate inference techniques
    - Pixel, voxel, and superpixel random field optimization
    - Priors and higher order clique optimization
    - Approaches that scale to large amounts of training and test data
- Computer vision applications of structured output learning
    - Segmentation
    - Stereo reconstruction
    - Relationship between scene components
    - Hierarchical models