Measuring Objectness

Xu Zhou

Colorado School of Mines

April 22, 2015

Introduction

In recent years, most state-of-the-art object class detectors use the sliding-window method, but they are usually specialized for one object class.

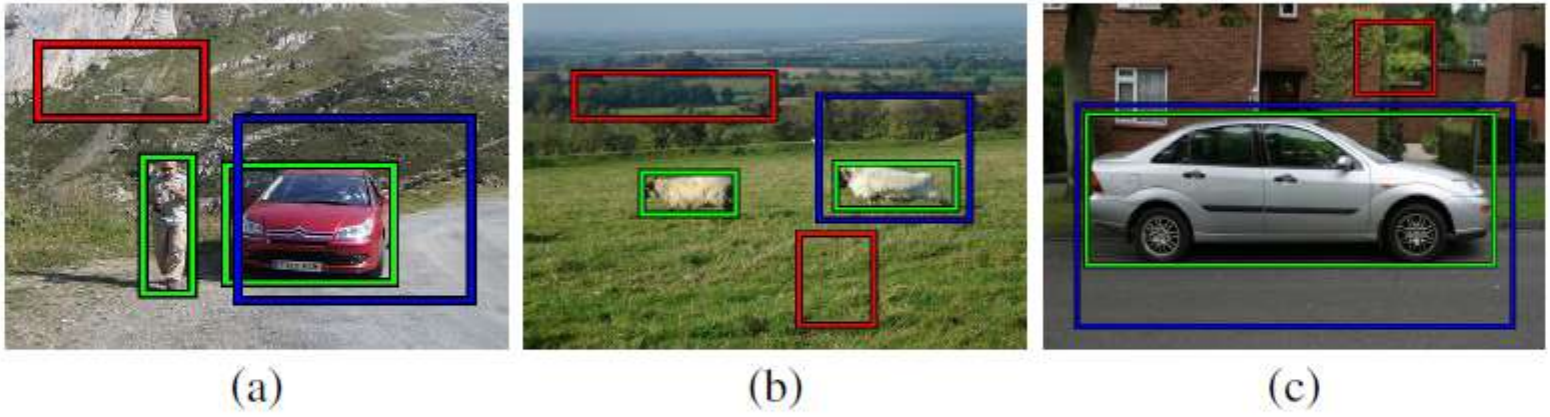
The goal for this project is to define and train a measure of objectness generic over classeses, i.e. quantifying how likely it is for an image window to cover an object of any class.

In order to define the objectness measure, we argue that any object has at least one of three distinctive characteristics:

(1) a well-defined closed boundary

(2) a different appearance from its surroundings

(3) sometimes it is unique within the image and stands out as salient



Related Methods

- Interest points

  Interest point detectors (IPs) focus on individual points

- Class-specific saliency

  These works are defining the salient as the visual characteristics that best distinguish a particular object class (e.g. cars) from others

- Generic saliency

  This definition measures the saliency of pixels as the degree of uniqueness of their neighborhood wrt the entire image or the surrounding area.

Objectness Cues

Since objects in an image are characterized by a closed boundary in 3D space or a different appearance from their immediate surrounding and sometimes by uniqueness, we will present five image cues to measure these characteristics.

- Multi-scale Saliency (MS)
- Color Contrast (CC)
- Edge Density (ED)
- Superpixels Straddling (SS)
- Location and Size (LS)

Multi-scale Saliency (MS)

We can define the global saliency measure based on the spectral residual of the FFT, which favors regions with an unique appearance within the entire image f. (*)

The saliency map I of an image f is obtained at each pixel p as

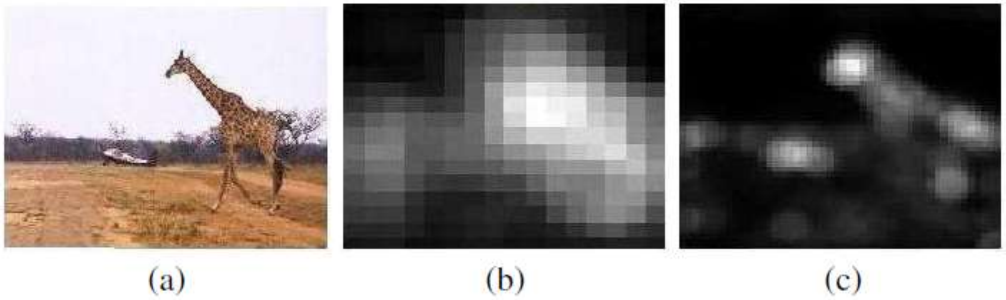
$$I(p) = g(p) * \mathcal{F}^{-1}[\exp(\mathcal{R}(f) + P(f))]$$

where $\mathcal{F}$ is FFT, $\mathcal{R}(f)$ and $P(f)$ are the spectral residual and phase spectrum of the image f, and g is a Gaussian filter used for smoothing the output.

For each scale s, we will extend the above formula to

$$MS(w, \theta_{MS}^{s}) = \sum_{\{p \in w | I_{MS}^{s}(p) \geq \theta_{MS}^{s}\}} I_{MS}^{s}(p) \times \frac{|\{p \in w | I_{MS}^{s}(p) \geq \theta_{MS}^{s}\}|}{|w|}$$

where θ_{MS}^{s} is the scale-specific thresholds and | · | indicates the number of pixels.



(*) X. Hou and L. Zhang, "Saliency detection: A spectral residual approach", IEEE conference on Computer Vision and Pattern Recognition, pp. 1-8, 2007

Color Contrast (CC)

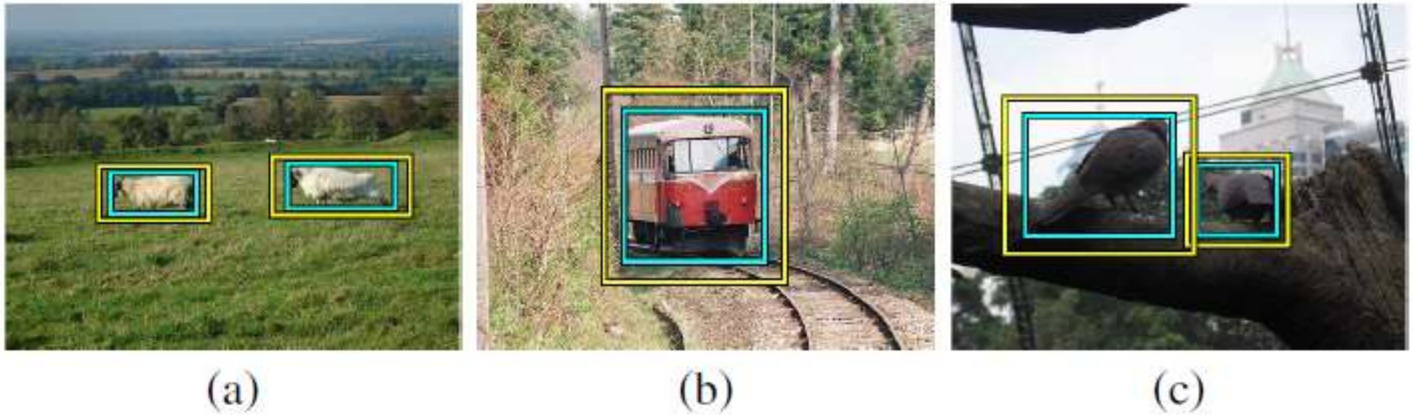
CC is a measure of the dissimilarity of a window to its immediate surrounding area.

The surrounding $Surr(w, \theta_{CC})$ of a window w is a rectangular ring obtained by enlarging the window by a factor θ_{CC} in all directions, so that

$$\frac{|Surr(w, \theta_{CC})|}{|w|} = {\theta_{CC}}^2 - 1$$

Then the CC between a window and its surrounding can be computed by the Chi-square distance between their LAB histograms h.

$$CC(w, \theta_{CC}) = \chi^2(h(w), h(Surr(w, \theta_{CC})))$$



Edge Density (ED)

Edge Density is a measure of the density of edges near the window borders.

The inner ring $Inn(w, \theta_{ED})$ of a window w can be obtained by shrinking it by a factor θ_{ED}in all directions, so that

$$\frac{|Inn(w, \theta_{ED})|}{|w|} = \frac{1}{{\theta_{ED}}^2}$$

The ED of a window w is computed as the density of edgels[1] in the inner ring

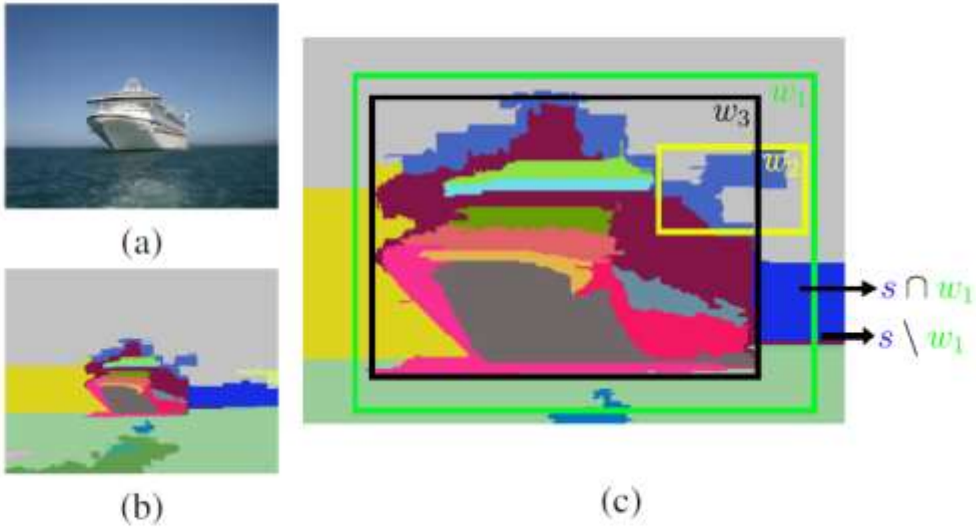
$$ED(w, \theta_{ED}) = \frac{\sum_{p \in Inn(w, \theta_{ED})} I_{ED}(p)}{Len(Inn(w, \theta_{ED}))}$$

where, the binary edgemap $I_{ED}(p) \in \{0,1\}$ is obtained using the Canny detector, and $Len(\cdot)$ measures the perimeter of the inner ring.

[1] an edgel is a pixel classified as edge by an edge detector

Superpixels Straddling (SS)

Superpixels segment an image into small regions of uniform color or texture and a key property of superpixels is to preserve object boundaries: all pixels in a superpixel belong to the same object (ideally), hence an object is typically oversegmented into several superpixels, but none straddles its boundaries.



(a) (b) (c)

Define SS cue measures for all superpixels s the degree by which they straddle w

$$SS(w, \theta_{SS}) = 1 - \sum_{s \in S(\theta_{SS})} \frac{\min(|s \backslash w|, |s \cap w|)}{|w|}$$

where $S(\theta_{SS})$ is the set of superpixels determined by the segmentation scale θ_{SS}.

Location and Size (LS)

Although windows covering objects vary in size and location within an image, some windows are more likely to cover objects than others: an elongated window located at the top of the image is less probable a priori than a square window in the image center.

We compute the probability using kernel density estimation in the 4D space $\mathcal{W}$ of all possible windows in an image. The space $\mathcal{W}$ is parametrized by the (x,y) coordinates of the center, the width and the height of a window.

Then we will use a large training set of N windows covering objects to compute the probability $p_{\mathcal{W}}$

$$p_{\mathcal{W}}(w, \theta_{LS}) = \frac{1}{Z}\sum_{i=1}^{N} \frac{1}{(2\pi)^2|\theta_{LS}|^{\frac{1}{2}}} e^{-\frac{1}{2}(w-w_i)^T(\theta_{LS})^{-1}(w-w_i)}$$

where the normalization constant Z ensures that $p_{\mathcal{W}}$ is a probability, i.e. $\sum_{w\in\mathcal{W}} p_{\mathcal{W}}(w) = 1.$

Learn parameters of CC, ED, SS

- For every image I in T (training dataset from PASCAL VOC 07), we generate 100000 random windows uniformly distributed over the entire image. Windows covering[1] an annotated object are considered positive examples ($\mathcal{W}^{obj}$), the others negative ($\mathcal{W}^{bg}$).
- Then for any value of θ, we can build the likelihoods for the positive $p_\theta(CC(w, \theta_{CC})|obj)$ and negative classes $p_\theta(CC(w, \theta_{CC})|bg)$, as histograms over the positive/negative training windows.
- After that, we can find the optimal

$$\theta^* = arg \max_\theta \prod_{w \in \mathcal{W}^{obj}} p_\theta(CC(w, \theta)|obj) \; = arg \max_\theta \prod_{w \in \mathcal{W}^{obj}} \frac{p_\theta(CC(w, \theta)|obj) \cdot p(obj)}{\sum_{c \in \mathcal{W}^{obj}} \, p_\theta(CC(w, \theta)|c) \cdot p(c)}$$

where the priors are set by relative frequency:

$$p(obj) = |\mathcal{W}^{obj}|/(\left|\mathcal{W}^{obj}\right| + |\mathcal{W}^{bg}|), p(bg) = 1 - p(obj)$$

[1] The widespread PASCAL criterion of considering a window w to cover an object is $|w \cap o|/|w \cup o| > 0.5$

Learn parameter of MS

- Optimize the localization accuracy of the training object windows $\mathcal{O}$ at each scale s.
- After computing the saliency map I_{MS}^{s} and the MS score of all windows, non-maximum suppression on 4D score space will result in a set of local maxima windows $\mathcal{W}_{max}^{s}$. Based on those, we can find the optimal θ_{MS}^{s*} by maximizing

$$\theta_{MS}^{s*} = \underset{\theta_{MS}^{s}}{\operatorname{argmax}} \sum_{o \in \mathcal{O}} \max_{w \in \mathcal{W}_{max}^{s}} \frac{|w \cap o\,|}{|w \cup o\,|}$$

- This means θ_{MS}^{s*} will lead the local maxima of MS in images to most accurately cover the annotated objects.

Learn parameter of LS

- The covariance matrix θ_{LS} is considered diagonal $\theta_{LS} = diag(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$.
- We can learn the standard deviations σ_i using k-nearest neighbors approach.
- For each training window $w_i \in \mathcal{O}$ we compute its k-nearest neighbors in the 4D Euclidian space $\mathcal{W}$, and then derive the standard deviation of the first dimension over these neighbors. We set σ_1 to the median of these standard deviations over all training windows.

Bayesian cue integration

Since the proposed cues are complementary, using several of them at the same time appears promising.

- MS gives only a rough indication of where an object is as it is designed to find blob-like things.
- CC provides more accurate windows, but sometimes misses objects entirely.
- ED provides many false positives on textured areas.
- SS is very distinctive but depends on good superpixels, which are fragile for small objects.
- LS provides a location-size prior without analyzing image pixels.

To combine n cues $\mathcal{C} \subseteq \{MS, CC, ED, SS, LS\}$, we train a Bayesian classifier to distinguish between positive and negative n-uples of values (one per cue).

For each training image, we sample 100000 windows from the distribution given by MS cue (thus biasing towards better locations), and then compute other cues in $\mathcal{C}$ for them. Windows covering an annotated object are considered as positive examples $\mathcal{W}^{obj}$, all others are considered as negative $\mathcal{W}^{bg}$.

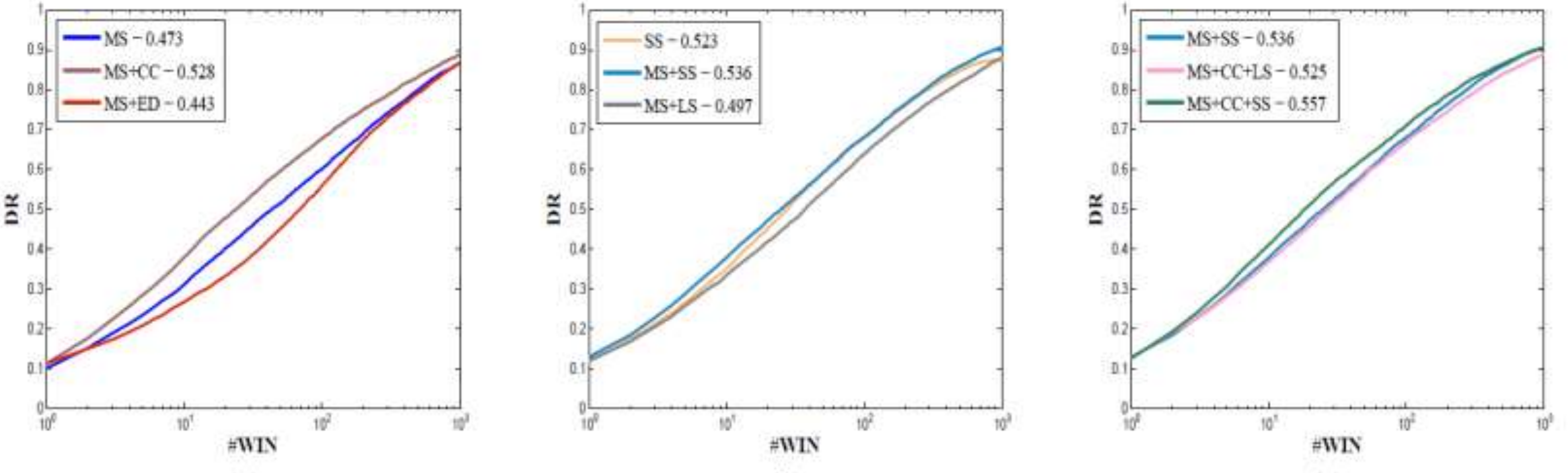
$$p(obj|\mathcal{C}) = \frac{p(\mathcal{C})p(obj)}{p(\mathcal{C})} = \frac{p(obj)\prod_{cue\in\mathcal{C}} p(cue|obj)}{\sum_{c\in\{obj,bg\}} p(c)\prod_{cue\in\mathcal{C}} p(cue|c)}$$

Experimental Results

- Use PASCAL VOC 07 Dataset
- It includes twenty classes of objects: {bird, horse, cat, cow, boat, sheep, dog, aeroplane, bicycle, bottle, bus, chair, diningtable, car, motorbike, person, pottedplant, sofa, train, tvmonitor}.
- We will use the first 6 classes for training.
- The remaining classes will be used for testing.

Experimental Results

- Evaluation criteria: DR-#WIN curves.
- Performance is evaluated with curves measuring the detection-rate vs number of windows.

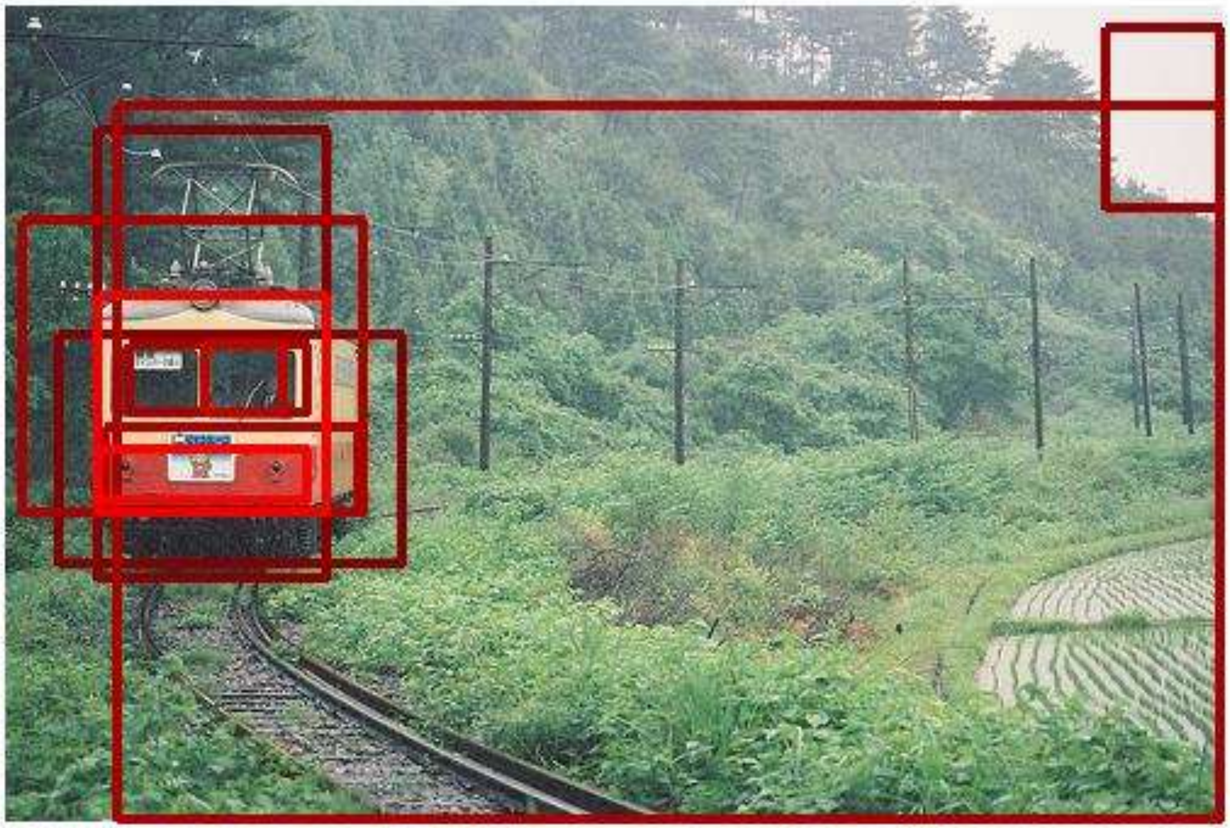


(a)

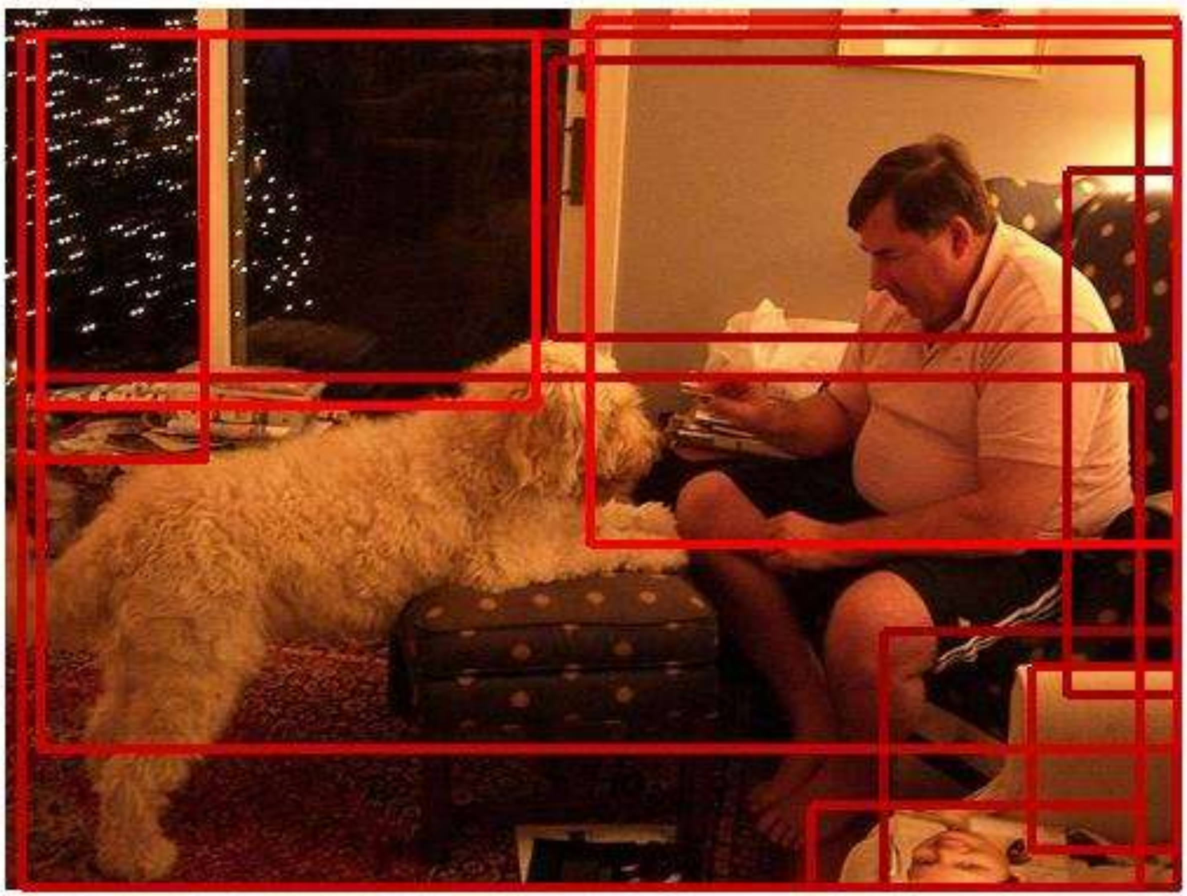
(b)

(c)

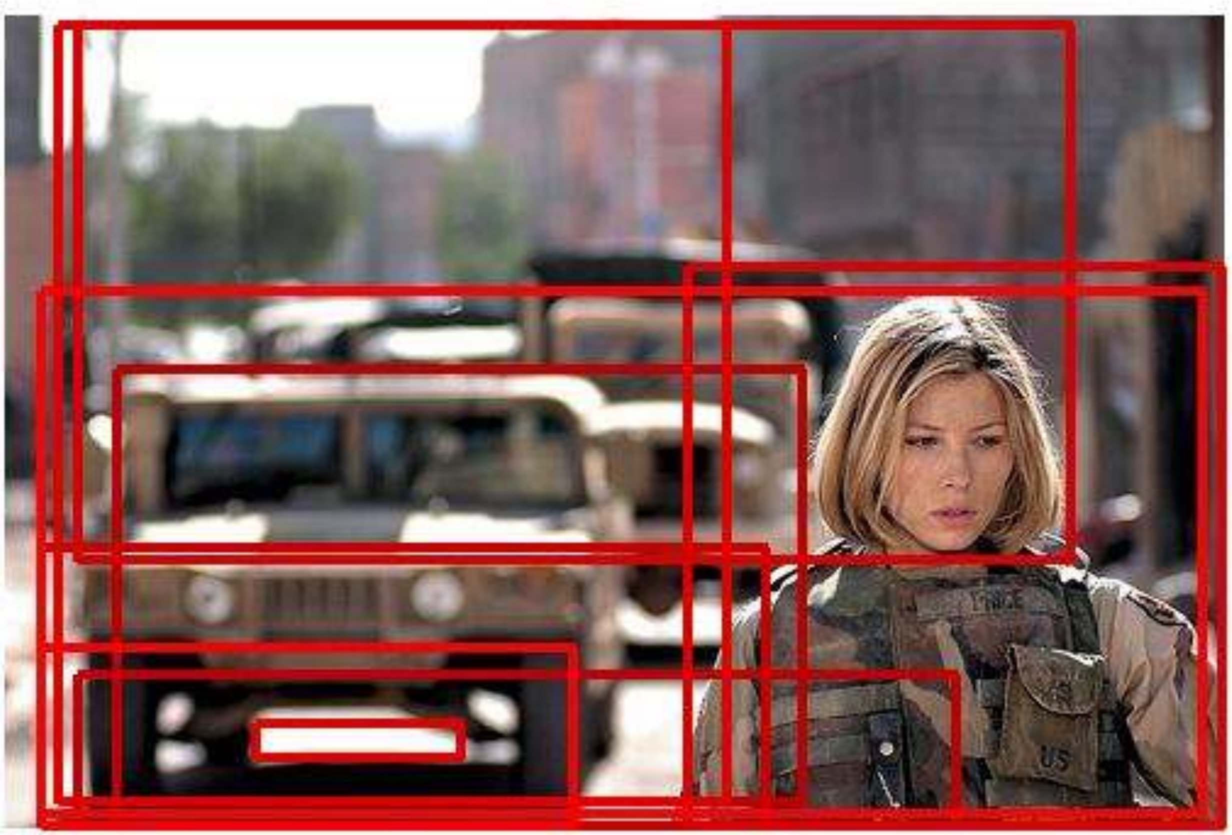
Experimental Results



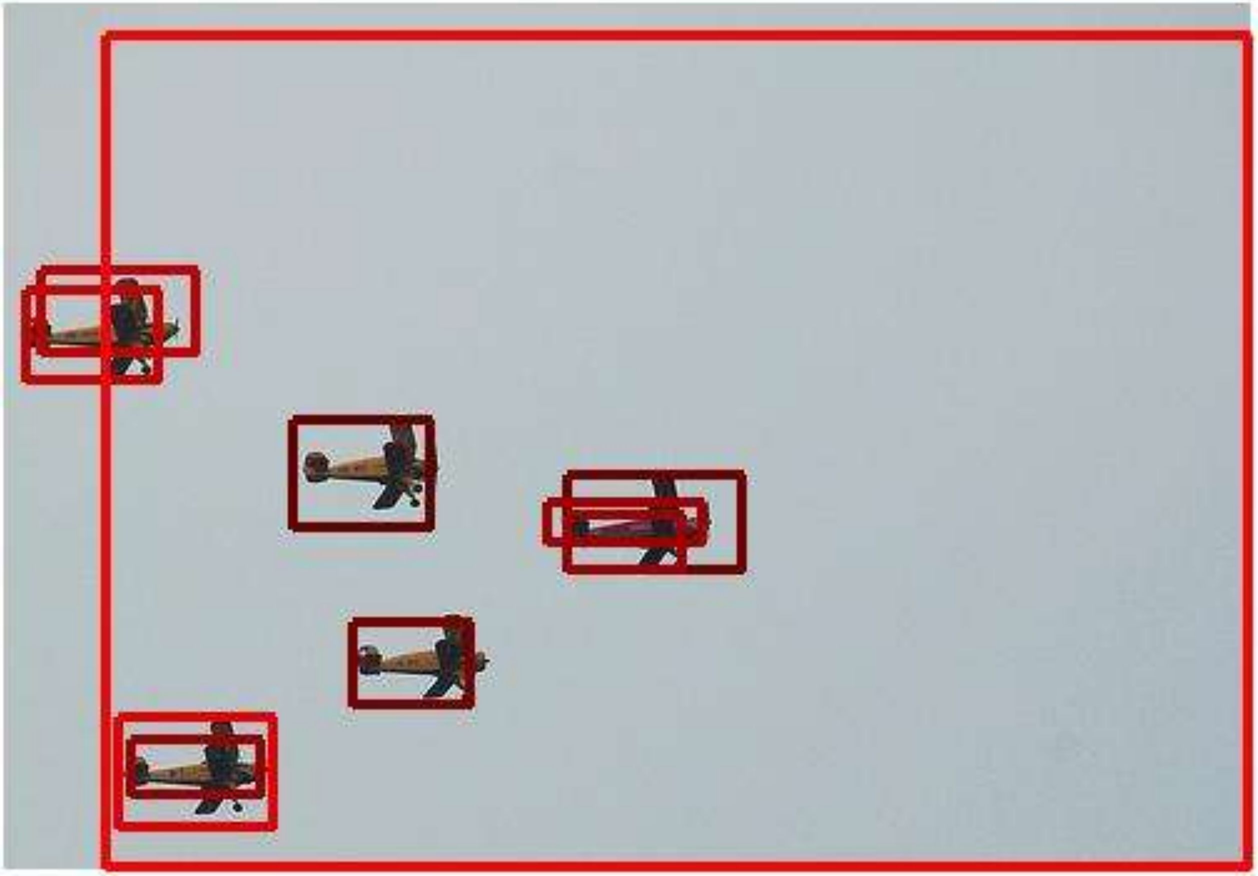
Experimental Results



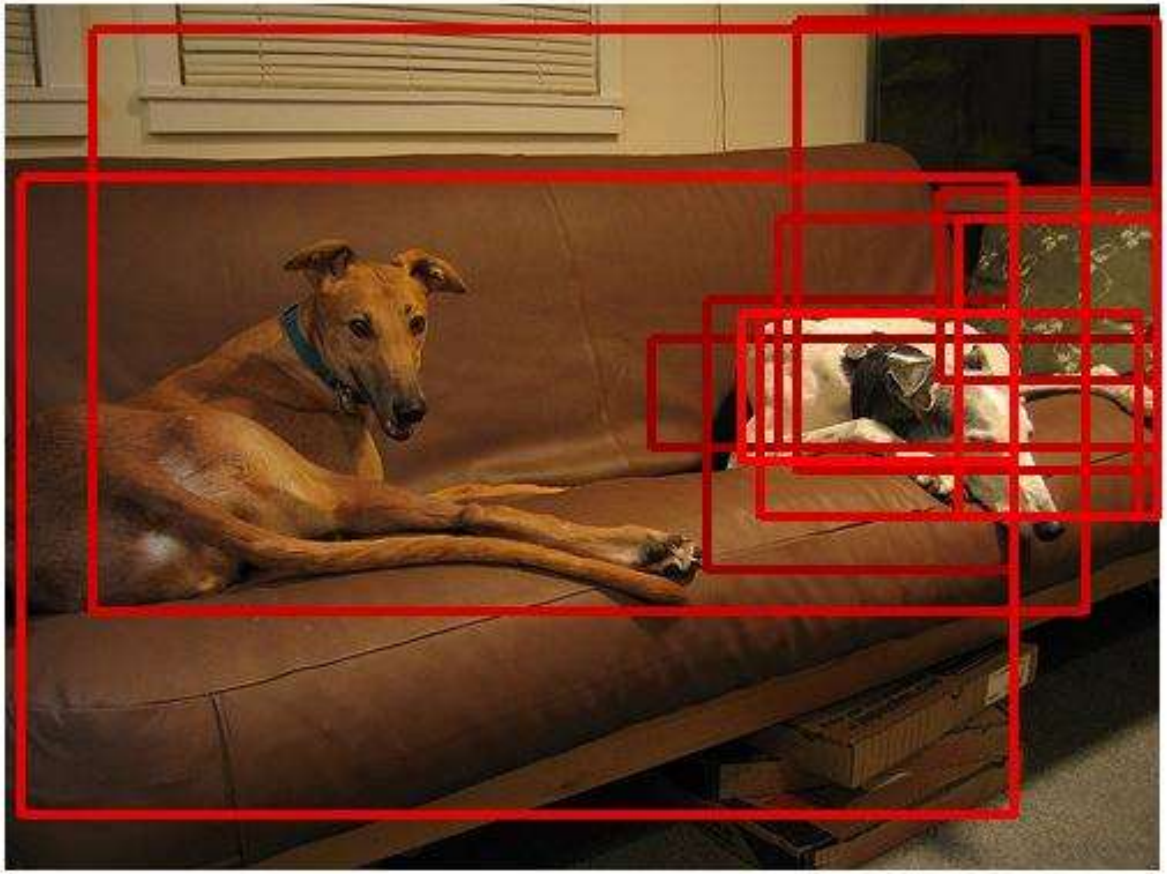
Experimental Results



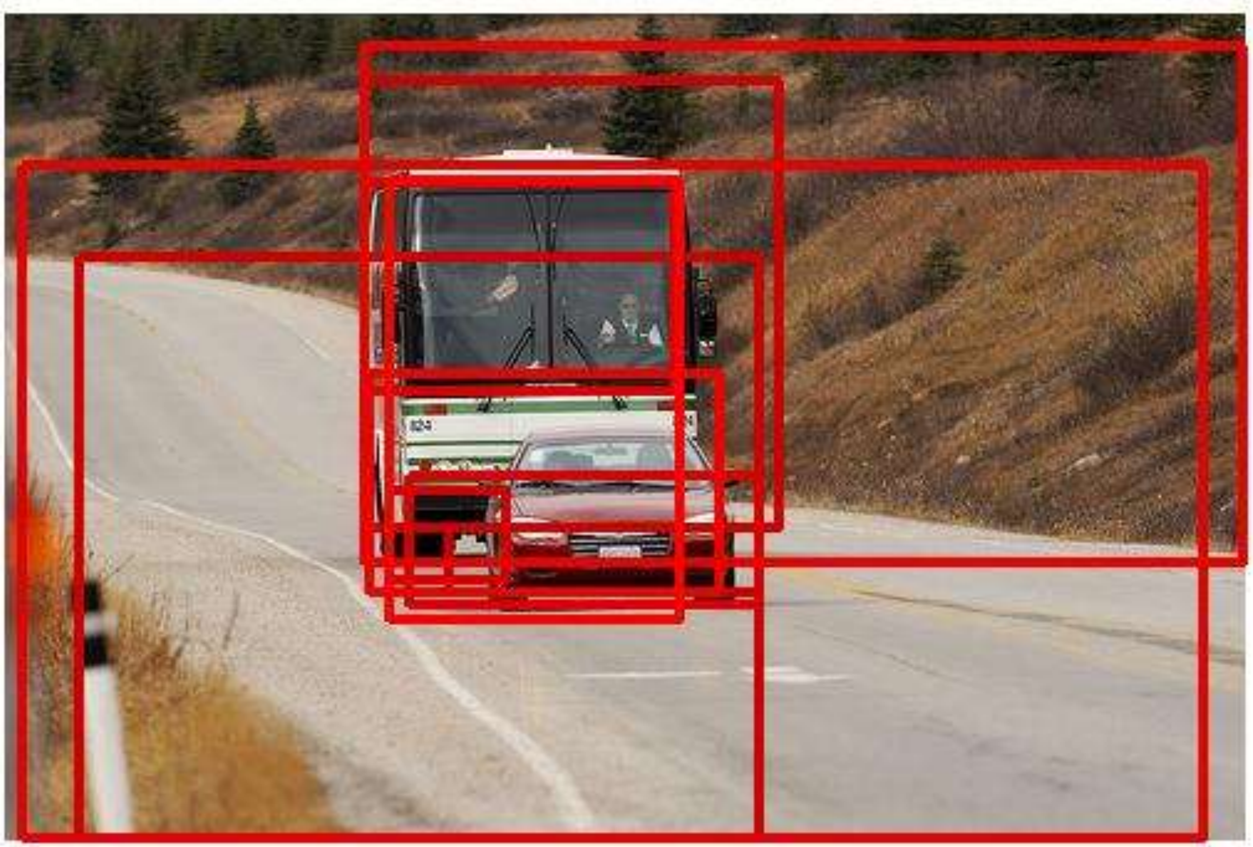
Experimental Results



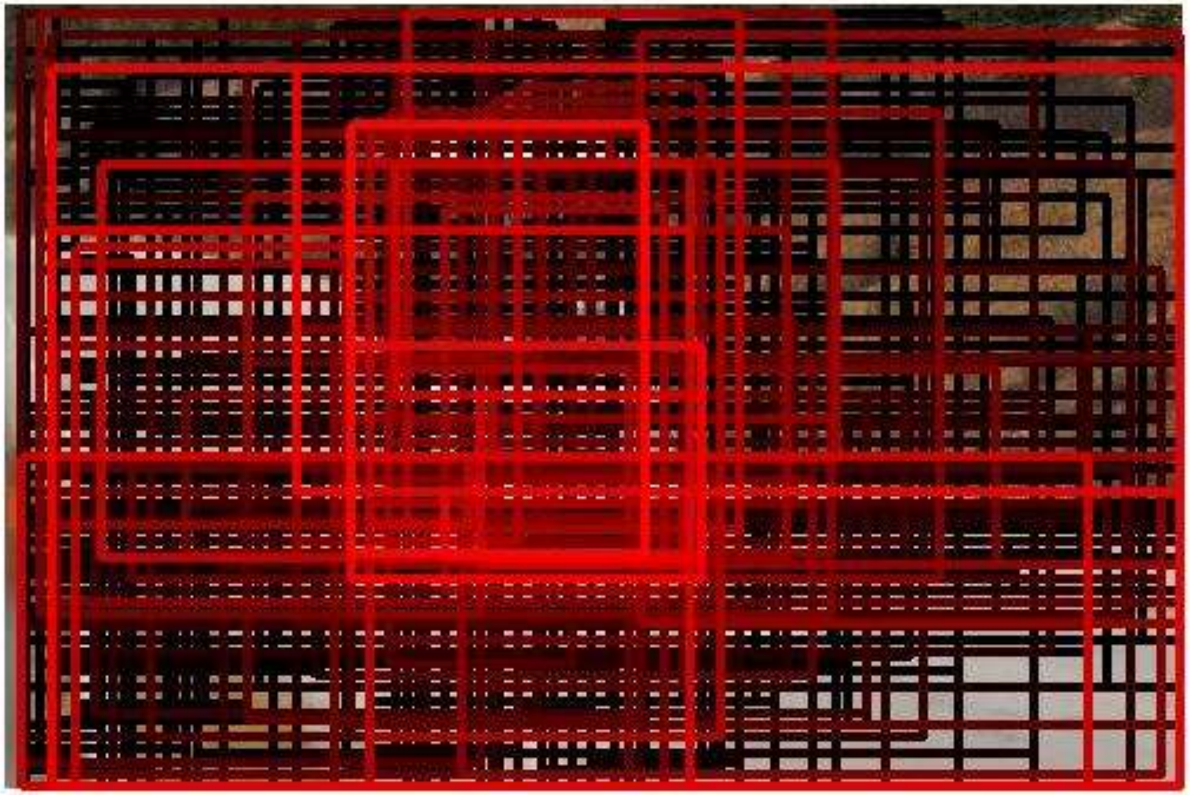
Experimental Results



Experimental Results



Experimental Results



Ongoing Work

- Analyze detection rate as a function of the number of windows sampled for various cues and combinations.
- After obtaining the detection rate, compare that with some other algorithms.
- Mark out the most possible objects.

Conclusion

- If we use more sampling window and combine several cues properly, we will get satisfied detection rate.

- The computation time of the total program is less than 4 seconds even with sampling 1000 windows.

Questions?

References

- B. Alex, T. Deselaers, V. Ferrari, “Measuring the objectness of image windows”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2189-2202, Sept 2012
- X. Hou and L. Zhang, “Saliency detection: A spectral residual approach”, IEEE conference on Computer Vision and Pattern Recognition, pp. 1-8, 2007
- P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation”, International Journal of Computer Vision, vol. 59, no. 2, pp. 167-181, 2004
- T. Liu, J. Sun, N. Zheng, X. Tang, and H. Shum, “Learning to detect a salient object”, IEEE conference on Computer Vision and Pattern Recognition, 2007