OBJECTNESS
AND
OBJECT DETECTION

A Project Report
Submitted for the Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY

IN

Electrical Engineering

submitted by

PRIYAM DEY (12EE01012)
GANDRETI SANTHOSH BABU (12EE01007)

under the guidance of

Dr. Niladri Bihari Puhan

School of Electrical Sciences
Indian Institute of Technology Bhubaneswar

May 2016

Indian Institute of Technology Bhubaneswar

CERTIFICATE

Certified that the project work titled OBJECTNESS AND OBJECT DETECTION was carried out by Mr. Priyam Dey, Roll No. 12EE01012 and Mr. Gandreti Santhosh Babu, Roll No. 12EE01007, bonafide students of Indian Institute of Technology Bhubaneswar in partial fulfillment for the award of Bachelor of Technology in Electrical Engineering during the year 2015-16. It is certified that all the corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Name & Signature of the Guide　　　　　　　　Name & Signature of the Head of School

Project evaluation date:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

PRIYAM DEY
12EE01012

GANDRETI SANTHOSH BABU
12EE01007

Date:

DEDICATED

TO OUR

LOVELY PARENTS

ACKNOWLEDGEMENT

First and above all, we thank our parents for providing us with all the opportunities in our life and making us what we are today. We would never have able to finish our dissertation without the guidance of professors, help from friends and support from our families.

We take this opportunity to express our deepest gratitude to our guide, **Dr. Niladri Bihari Puhan** for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. His years of experience in image processing, as well as his immense passion for his field of study are sure to inspire us throughout our career. Working with him was a great learning experience.

TABLE OF CONTENTS

CHAPTER 4: EVALUATION AND RESULTS

ABSTRACT

Object detection is one of the important and highly focused area of computer vision in recent years. It is also a prerequisite step in a wide range of higher-level vision tasks, including activity or event recognition, full scene content understanding, etc. The objectness measure acts as a *class-generic* object detector. It quantifies how likely it is for an image window to contain an object of any class, such as cars and dogs, as opposed to backgrounds, such as grass and water.

In this project, we propose two novel objectness measures which quantifies the presence of an object in an image window:

1) A symmetry-based measure, named as ***Quadrant Symmetry*** measure;

2) Feature-based dictionary learning approach for objectness, named as ***Dictionary of Objects***.

Proposed method 1 uses gradient-based histograms to capture the symmetry of edge gradients in four quadrants of an image window and uses histogram intersection distance for computing the similarities between the histograms. Proposed method 2 trains an object dictionary based on gradient features and Haar features, separately, in an image window using KSVD and Spherical K-means clustering. Using the dictionary, it scores the image windows using projections over the dictionary.

We evaluated our proposed methods on the *PASCAL VOC 2007 test dataset*, famous standard dataset for object detection. The best detection rate (***DR***) of the methods are:

1) ***Quadrant Symmetry*** measure: 83% with 5000 proposal windows;

2) ***Dictionary of Objects*** measure: 93.88% for 5000 proposal windows.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Introduction to object detection and objectness

Object detection is a process of detecting objects in an image, may it be a car, bicycle, person or an animal. This task of detecting objects is a very important and well-known computer vision problem. It is also a pre-processing step for many higher-level vision tasks like activity and event recognition. Typically only a small number of instances of the object are present in an image, but there is a very large number of possible locations and scales at which they can occur. So, any algorithm for object detection has to be fast as well as accurate.

Goals to be achieved in object detection:

(i) ***Object categorization*** which determines whether or not any instance of the categories of interest is present in a given image. Object categorization can be performed at three different levels: image-level categorization, which determines whether or not any instance of the categories of interest is present in a testing image and hence is a subtask of object class detection as discussed at the beginning of this section. The second level is region-level categorization, which can be used in

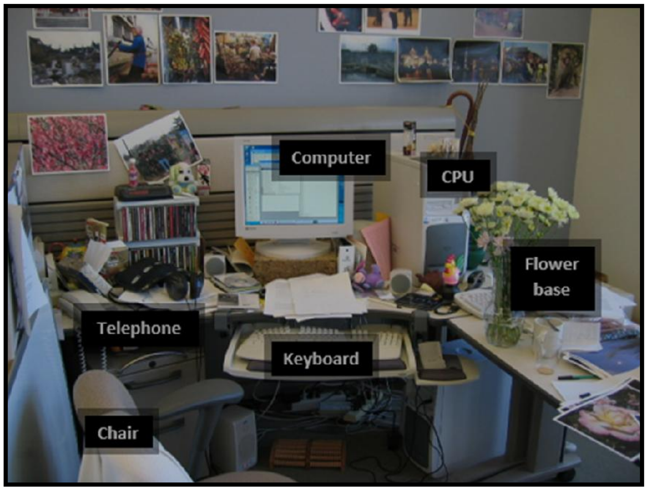


Fig.1. Determination of presence of objects in an image

practice to implement object class detection. The third level is pixel-level categorization that determines to which one of the predefined categories a given pixel belongs. It can also be applied to implement detection, in particular, to locate unstructured objects such as grass and road.

(ii) ***Object localization*** which determines the positions and extents of all the objects that are found present. If only one class is of interest, say the pedestrian or the face, then it is a special case of class-specific detection. A more general case is multiclass detection, by which all instances of multiple predefined categories are detected.

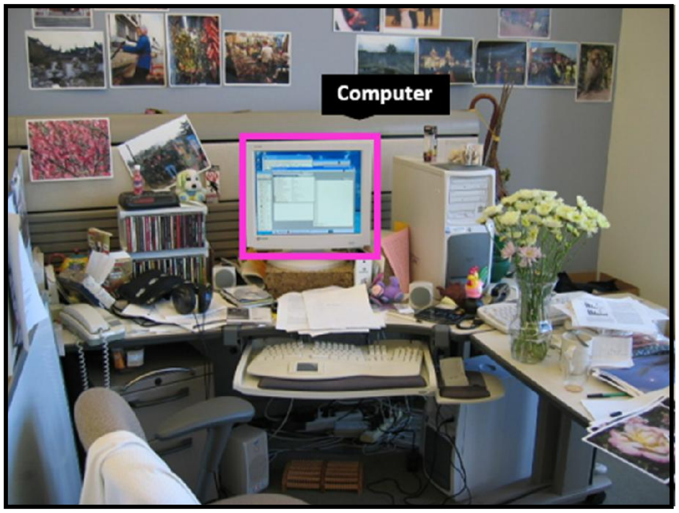


Fig.2. Determining the extent of an object (flower) in the image

Another closely related task is **figure-ground segmentation**, which aims at accurately separating foreground objects from their background in a given image and thus is essentially a kind of category-level object segmentation. It focuses only on visible parts of objects and so their results are inaccurate for specifying the real extents and even locations of occluded objects. In contrast, object detectors attempt to localize whole object even if it is partially occluded.

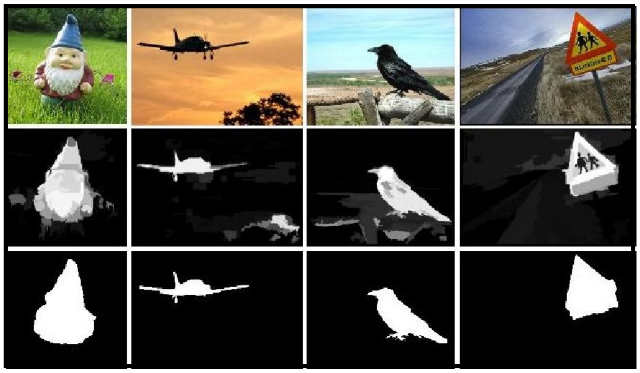


Fig.3. Figure-ground segmentation

In contrast to figure-ground segmentation, there is another type of segmentation called **semantic image segmentation** which segments all objects in a given image irrespective of whether they are foreground or background, i.e., even if object are part of background, they will be segmented. Thus it is a more general vision task compared with figure-ground segmentation, which can be considered as one of its subtasks.

Now, after looking at the brief description of object detection, we move onto objectness.

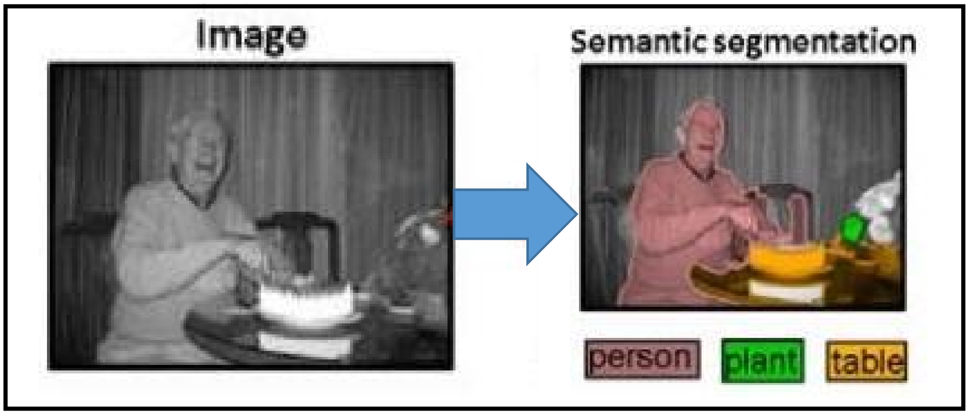


Fig.4. Semantic segmentation

What is Objectness?

The objectness measure, as mentioned in **[2]**, acts as a *class-generic* object detector. It quantifies how likely it is for an image window to contain an object of any class, such as cars and dogs, as opposed to backgrounds, such as grass and water.

Talking about objects, how are they defined? According to **[2]**, it can be stated that any object possess at least three distinctive features:

- a well-defined closed boundary
- a different appearance from its surroundings
- salient (stand out distinctively) in the image.

Objects can even possess some or all of these characteristics in an image at the same time. Based on our knowledge built on principles of object detection and definition of an object, we formulate our first measure. We argue that any object we observe in nature is symmetric in some way or the other. It is mostly compact with a closed boundary and its gradients on its boundary emerge in all directions. Based on this notion, we propose a symmetry-based objectness measure, named as ***Quadrant Symmetry*** measure, which exploits the symmetry of an object through spatial localization of edge gradients. This measure is obtained by taking into account the histogram of gradient angles of canny edge pixels in 4 quadrants of a window. The quadrants are not fixed but rotates by some angle. We compute symmetry score for each rotated quadrant system and take the maximum of all these scores, thereby finding coordinate axes in which the object is most symmetric. We compute the symmetry score for a proposal window by forming histogram of gradient angles in four quadrants, and finding the similarities between these histograms by calculating histogram intersection distance between them. For efficient computation of quadrant histograms, we transform the solution space to the different coordinate system so as to reduce the computational load. In this way, we capture the symmetry in an object by exploiting its gradient map and edge map.

Sparse representation of images has been a recent area of growing interest. It finds applications in many problems in image processing. We feel that like texts, most of the objects we see in nature can be represented using a visual dictionary. So, moving in that direction, we formulate an objectness measure using what we call a ***Dictionary of Objects***. This dictionary is formed by first extracting gradient and Haar features from object windows and then training it using ***K-SVD*** algorithm **[8]** and ***Spherical K-means clustering*** algorithm, which is also an unsupervised learning method with unit sparsity.

1.2 Motivation

Object detection, as mentioned before, is a very important and essential pre-processing step to most of the higher-level vision tasks. So, many critical problems such as event detection, especially, abnormal event detection, which is an essential part of surveillance, activity monitoring, robot navigation, targeting in missile detection, obstacle detection for blind people, which are heavily dependent on object detection, can be solved to make life more secure and better. Motivated by this thought, we took up the problem of object detection. Also, objectness measure for object detection has been an innovative as well as a promising field for solving the problem of object detection efficiently. So, we address the problem of object detection using objectness measure.

1.3 Objective

- Our objective is to develop a novel objectness measure which can accurately categorize possible object regions in an image and localize them properly using rectangular bounding boxes.
- The measure should be fast as well as accurate to detect possible object regions.

1.4 Organization of the thesis

The remainder of the thesis is organized as follows -

Chapter 2 provides the description of various kinds of techniques used in the field of object detection. This chapter is mainly focused on the survey of existing state-of-the-art methods of object detection using objectness measure.

Chapter 3 describes the proposed methodologies for objectness measure using object detection. It describes the methods in detail both from algorithmic and implementation point of view.

Chapter 4 shows the results of experimental evaluation of the proposed methodologies on a well-famous and standard dataset for object detection, *PASCAL VOC 2007* dataset. Our proposed methods are compared with the existing state-of-the-art methods of objectness measure for object detection.

Chapter 5 draws conclusion of the simulation results we obtained for our proposed methods with a projection on the future scope.

CHAPTER 2: LITERATURE REVIEW

We started this project by reading about object detection and its various techniques, key challenges involved in it. Then we focused on the techniques based on objectness and evaluated the existing objectness-based object detection techniques. So, first we will discuss core techniques used in object detection.

2.1 Core techniques for object detection

1. **Appearance modelling techniques:** Appearance models are very important to represent an object of a real world. It is used to map low-level features of an image to some high level semantics.

   a) ***Description of Relevant Visual Cues:*** Visual features commonly used for building categorical representations for recognition can be subdivided into three groups according to their different levels of locality, that is, pixel-, patch- and region-level feature descriptions.

   - Pixel-level Feature Description: gray-scale, its color vector.
   - Patch-level Feature Description: The term "patch" often refers to a small local sub-window surrounding some point of interest in the image plane or scale pyramid. Since it is usually quite small relative to the image size, patch-level descriptors are called local feature descriptors as well. Different descriptors often emphasize different image properties like pixel intensities, colors, textures, edges, etc. Examples of patch level descriptors are SIFT descriptor, Filter-Bank responses, etc.

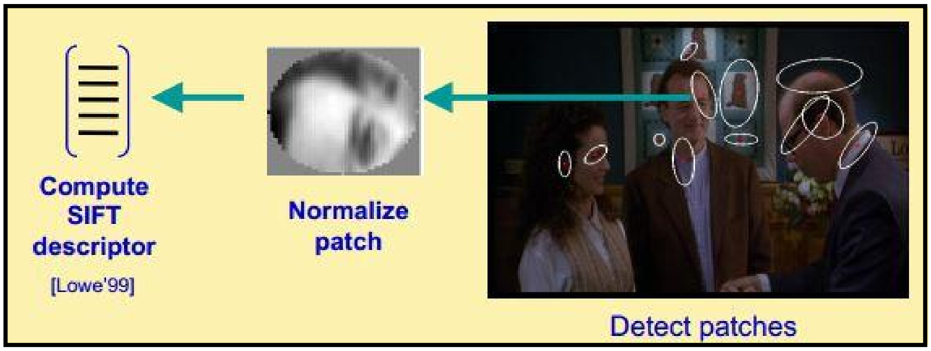


Fig.5. Appearance modelling using patch-level descriptors

- Region-level Feature Description: ‘Region’ is a set of connected pixels in an image. Thus, a region can be a regularly or irregularly shaped segment in an image, or can even be the whole image. A region-level description is commonly developed with the goals of: (i) capturing the (most) discriminating visual properties of the target categories or their components while (ii) keeping sufficient robustness against possible intra-class variations. Examples are Bag-Of-Words (BoW) or Bag-Of-Features, Histogram of Oriented Gradients (HOG), shape features, self-similarity features, etc.

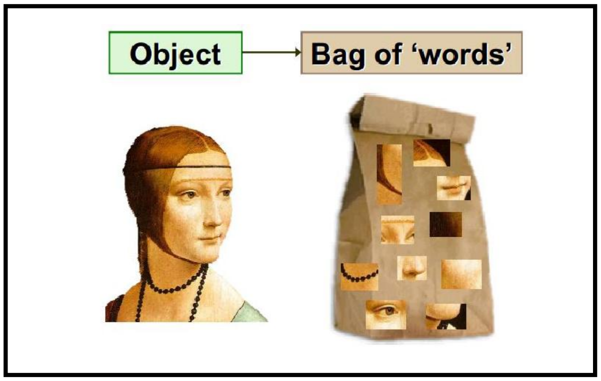


Fig.6. Appearance modelling using regional-level descriptors

b) ***Structured Models:*** Three different types of models are commonly used in the literature to describe structured objects, namely, window-based, part-based, and mixed models.

- Window-based model: The basic idea underlying window-based or global models is to describe the appearance of an object in an image using a (mid-level) descriptor computed in a window surrounding the object. To compute such a window-based representation, the key steps include: (i) specifying the window shape in advance, (ii) selecting appropriate features and the corresponding descriptions, and (iii) concatenating feature descriptors computed in the given image window to form a single window-based appearance descriptor.

- Part-based model: A typical part-based model consists of two components: a set of parts and the geometric relations among them, called part topology. Parts commonly refer to some (relatively) rigid components of an object, for example, the head or forearm of a human body. They are often described in terms of their appearance properties such as colors, textures, or gradients, as well as their geometric properties such as height, width, and shape. Part topology is usually described using the relative locations of parts and the (possible) connections among them, for example, an upper leg is connected with a lower leg. Examples of this model are star-structured models, tree-structured models, and Grammar-based models.

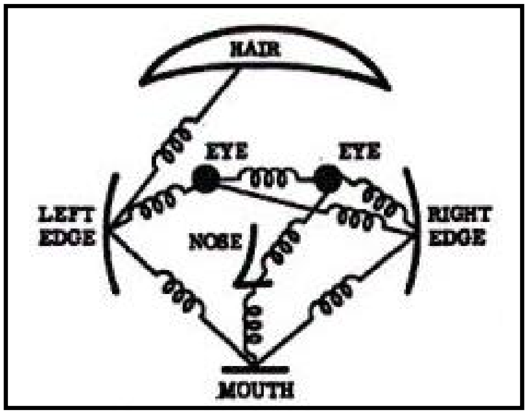


Fig.7. Appearance modelling of face using part-based model

2. **Localization strategies**: The task of localization is to search in the input image for regions that best match the learned appearance models of the categories of interest.

   a) ***Localization through Subwindow Search***: The subwindow search strategy, also called sliding window, performs object categorization over all possible sub windows in the input image to locate potential objects.

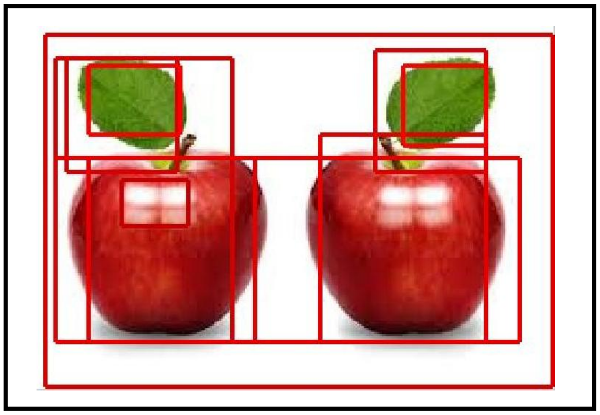


Fig.8. Detection of two apples (objects) and their leaves in an image using subwindow search

   b) ***The Voting strategy***:

   - Designed exclusively for part-based models, particularly star-structured ones. Since such a model consists of two key components, namely a set of parts and their topological relations, it's matching score S(c, R) is equal to the sum of the matching scores of the parts and the topological conformity measure.
   - The voting strategy maximizes S(c,R) in two stages: The first stage finds the best matching locations in the input image for all model parts, and hence maximizes the sum of the matching scores of the parts. The second stage maximizes the topological conformity measure by searching for the best topological hypotheses in the Hough voting space cast by the parts detected in the first stage. Thus, the voting strategy is essentially a greedy approach.

c) ***Localization through segmentation***:

- A successful segmentation automatically leads to a perfect localization.
- The most straightforward way is to localize objects through a semantic segmentation process, in which a pixel-level unstructured representation is used to perform pixel-wise categorization.
- Due to the locality of the adopted representation, the segmentation results often suffer "holes" inside objects and very inaccurate object boundaries.
- Hence, additional constraints are imposed under a probabilistic framework such as the CRF (Conditional Random Field) or the MRF (Markov Random Field) to ensure the label consistency between neighboring pixels as well as to incorporate contextual cues.
- Another method is to perform semantic segmentation and object localization simultaneously through region-wise categorization. Input image is first over-segmented into a set of pieces, for example, superpixels, or small coherent regions. The appearance of each of these regions is summarized using some mid-level representation. Then, a region-level categorization is performed using the SVM with, for e.g., a multilayer perceptron.

3. **Supervised Classification methods:** The significance of supervised classification methods is: (i) Parameters contained in different types of appearance models are usually estimated during classifier training. (ii) Categorization and detection results of the testing images are commonly predicted by trained classifiers.

   a) ***Parametric vs. Nonparametric methods***: A classifier is called non parametric if its classifier function contains no parameter; otherwise it is called parametric. More accurately, nonparametric classifiers are distribution free, that is, they make no assumption about the distribution of sample descriptors or the conditional distribution of categorical labels given sample descriptors. Thus, they do not need parameter estimation or inference. Nonparametric classifiers commonly used for object categorization and detection include the Nearest-Neighbor (NN) and the K-Nearest-Neighbor (KNN) methods.

b) ***Probabilistic vs. Non-probabilistic Methods***: Classifiers in which classification can be performed by the so-called Maximum A Posteriori (MAP) criterion are commonly called probabilistic models, including the Bayesian framework, the MRF, the Hidden Markov Model (HMM), the CRF, etc. The remaining classifiers are commonly called non probabilistic ones, including AdaBoost (or boosting), etc.

c) ***Generative vs. Discriminative Methods***: In general, there are two methods to obtain the posterior probability for classification. The first one is to use a parametric model. This kind of model is called a discriminative model. Non-probabilistic models belong to this family. The second method is to model the joint distribution instead with parametric models, which are called generative models in the literature and also known as informative models or sampling paradigms previously.

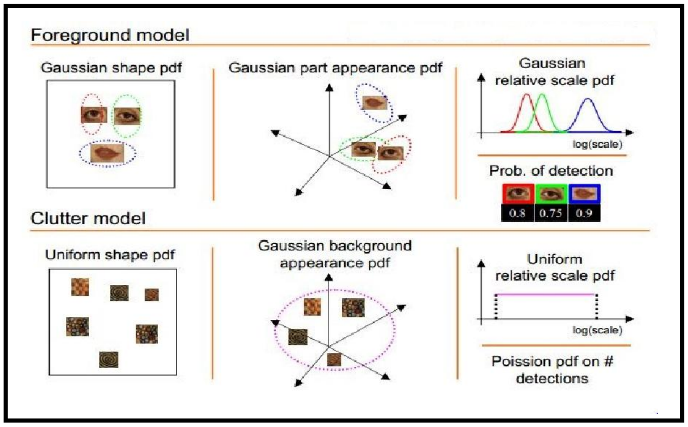


Fig.9. Intra-class variation of chair (object)

2.2 Key challenges in object detection

1) ***Robustness related*:** This consists mainly of the usually very large intra-class appearance variations and potentially small inter-class appearance differences.

2) ***Computational-complexity and scalability-related*:** The challenges in this group include the existence of a large number (or rather tens of thousands) of different categories and their high-dimensional appearance descriptions.

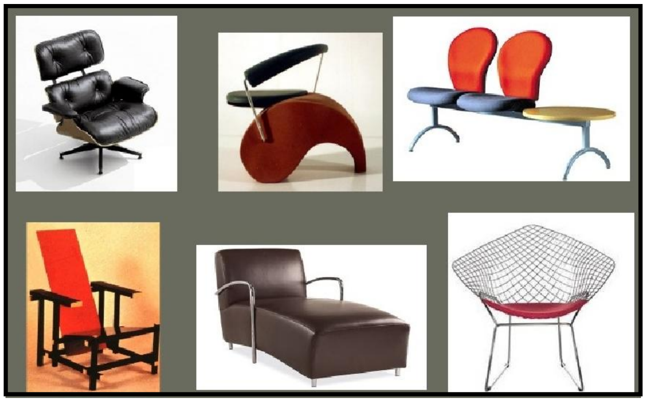


Fig.10. Intra-class variation of chair

2.3 Object detection using objectness

1. ***Alexe et. al* [2]** uses an objectness measure which is trained explicitly to distinguish objects with a well-defined boundary in space, such as cows and telephones, from amorphous background elements, such as grass and road. The measure combines in a Bayesian framework several image cues measuring characteristics of objects, such as appearing different from their surroundings and having a closed boundary. The visual cues used are multi-scale saliency, color contrast, edge density, and super-pixel straddling:

- <u>Multi-scale Saliency (MS)</u>:

Saliency map $I(p)$ is given by:

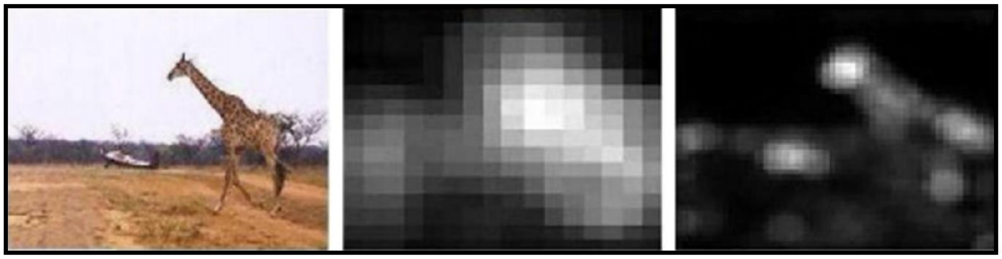
$$I(p) = g(p) * F^{-1}[\ exp(\ R(f) + P(f)\ )\ ]^2 \qquad \textbf{(1)}$$

where $g(p)$ is a Gaussian filter for smoothing, $R(f)$ is Residual spectrum of image with magnitude spectrum $L(f)$ obtained as $R(f) = L(f) - h(f)^* L(f)$, and $P(f)$ is the phase spectrum of the image.

Initially, the images are needed to be of the same size and for this, the images are scaled to various sizes and for each scale s, the Saliency maps '$I_{\mathrm{MS}}{}^{s}$' are generated. Keeping a threshold value '$\theta_{\mathrm{MS}}{}^{s}$' for each scale, saliency of a window w at scale s is defined as

$$\mathrm{MS}(\mathrm{w}, \theta_{\mathrm{MS}}{}^{s}) = \sum \cdot_{\left\{\mathrm{p}\epsilon\mathrm{w} \middle| I_{\mathrm{MS}}^{s(\mathrm{p})} \geq \theta_{\mathrm{MS}}^{s}\right\}} [I_{\mathrm{MS}}{}^{s}(\mathrm{p}) * \frac{\left|\left\{\mathrm{p}\epsilon\mathrm{w} \middle| I_{\mathrm{MS}}^{s}(\mathrm{p}) \geq \theta_{\mathrm{MS}}^{s}\right\}\right|}{|\mathrm{w}|}] \qquad \textbf{(2)}$$

where **| . |** indicates the number of pixels in that region. This MS cue measures the uniqueness characteristics of an image.



(a) (b) (c)

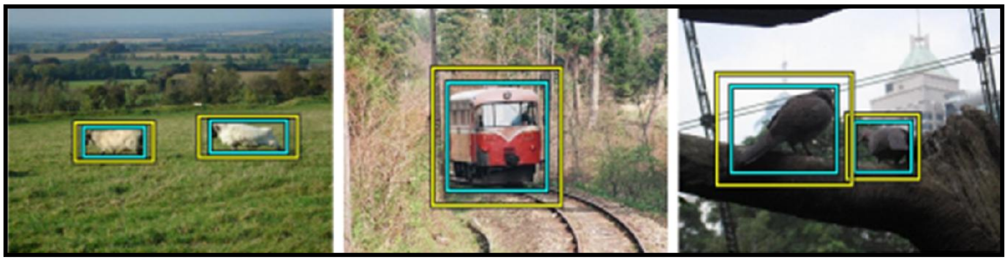
Fig.11. (a) An image containing two objects, (b) Its saliency map at a particular scale, (c) Its saliency map at another scale

- Color Contrast (CC):

  This cue determines the dissimilarity of the window to its immediate surrounding pixels. This is done by enlarging the window the window by a factor 'θ_{CC}' equally in all directions and comparing the pixels in the rectangular ring 'Surr(w, θ_{CC})' thus produced with the pixels in the window 'w'. Chi-square distance between the LAB histograms are considered to score the windows, taking into account the stand-out nature of the window from its surrounding portion.

$$\mathrm{CC}(\, w, \theta_{CC}) = \chi^2 (\, h(w) \,, h(\, \mathrm{Surr}(w, \theta_{CC})) \,) \qquad \textbf{(3)}$$

  To reduce the time complexity, integral histograms are generated beforehand for the image so that histograms of the any windows can easily be derived in no time.



(a) (b) (c)

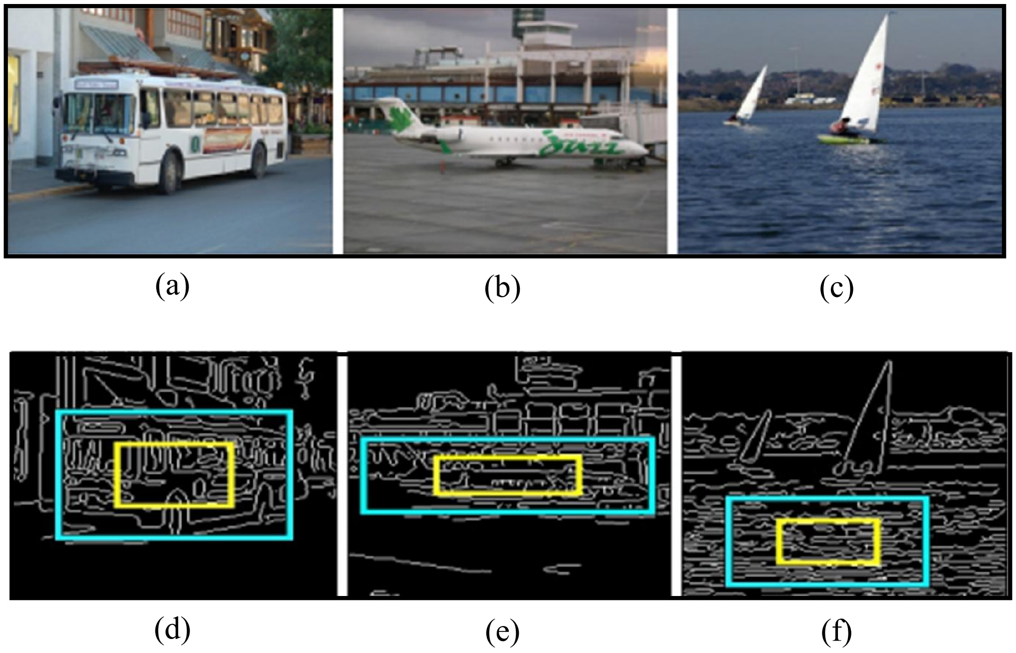
Fig.12. Examples of color contrast in images: (a) and (b) objects (sheep) with high contrast, (c) objects with low contrast

- Edge Density (ED):

  This cue scores those windows that are tightly packed with objects by taking into account the density of edges in the border pixels of the window of consideration. For this case, the window is shrunken by a factor of θ_{ED} in all directions creating a rectangular ring inside the window. Using Canny Edge detector, binary edge-map of the image is generated and density of edgels (pixels classified as edge pixels) in the inner ring compared to the perimeter of the ring is considered for scoring the windows. Mathematical expression for scoring the windows using the present cue is

$$\mathrm{ED}\ (\mathrm{w},\ \theta_{\mathrm{CC}}) = \frac{\sum_{\{p \epsilon Inn(\mathrm{w},\theta_{\mathrm{CC}})\}} I_{ED}(p)}{Len(Inn(\mathrm{w},\theta_{\mathrm{CC}}))} \qquad \textbf{(4)}$$

where I_{ED} is the canny edge detector output. This cue when used alone could not give good results, whereas when used along with other cues gave better results.


(a) (b) (c)

(d) (e) (f)

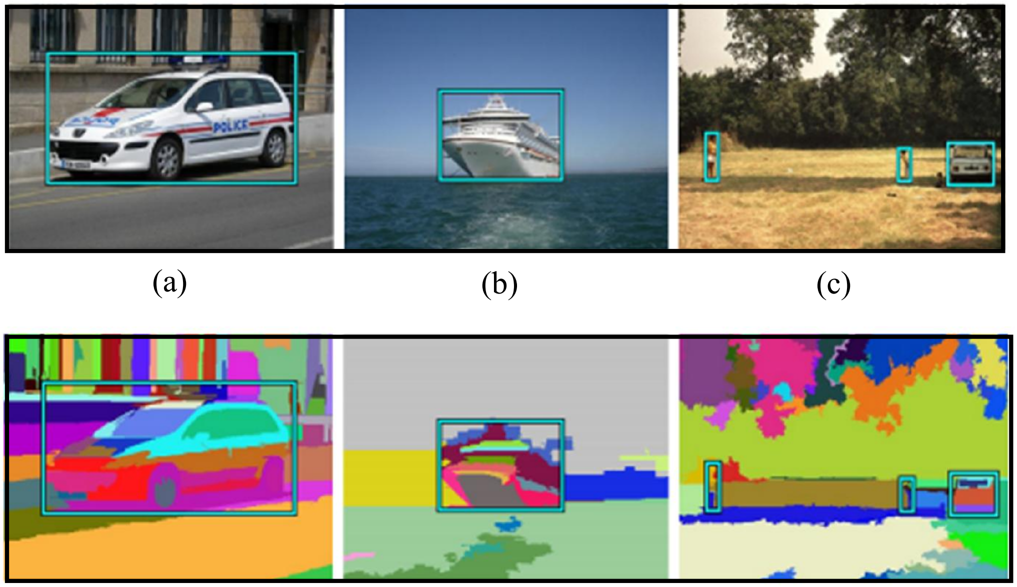
Fig.13. Examples of edge densities of objects in images: (a) and (b) objects (sheep) with high density edge maps (d) and (e), respectively, (c) object with low density edge map (f).

- Super-pixel Straddling (SS):

This cue scores windows with respect to the straddling of a superpixel across a window. In other words, it takes into account the ratio of number of pixels corresponding to the object inside and outside the window. Windows that tightly packed the object with least straddling is scored the highest. Superpixels are generated using graph-based segmentation algorithm thus segmenting the whole image into several sets in which pixels of any two sets are unique. Considering this kind of segmentation, a window 'w' is scored as follows.

$$\mathrm{SS}(w, \theta_{\mathrm{SS}}) = 1 - \sum_{\{s \epsilon S(\theta_{SS})\}} \frac{min(|s \backslash w|,|s \cap w|)}{|w|} \quad \textbf{(5)}$$

Where $S(\theta_{SS})$ is a set of superpixels obtained using segmentation scale θ_{SS} and the notations $| s \backslash w |$ and $| s \cap w |$ implies the number of superpixels $s \epsilon \mathrm{S}(\theta_{\mathrm{SS}})$ outside and inside the window w. This cue is better that ED cue but the time complexity is too high. In order to address this issue, Integral images of the superpixel map are used.



(a) (b) (c)

(d) (e) (f)

Fig.14. Examples of super-pixels of objects in images: (a) and (b) objects (sheep) with high edge density, (c) object with low edge density

These visual cues are trained using the training images to find the class-conditional density functions. Now for a given test image, the posterior probability of a test window w is given by -

$$p(obj|C) = \frac{p(C|obj)p(obj)}{p(C)} \tag{6}$$

$$= \frac{p(obj)\prod_{cue \in C} p(cue|obj)}{\sum_{c \in \{obj,bg\}} p(c)\prod_{cue \in C} p(cue|c)} \tag{7}$$

2. ***Cheng et. al* [3]** observed that generic objects with well-defined closed boundary can be discriminated by looking at the norm of gradients, with a suitable resizing of their corresponding image windows in to a small fixed size. Based on this observation and computational reasons, they proposed to resize the window to 8 × 8 and use the norm of the gradients as a simple 64D feature to describe it, for explicitly training a generic objectness measure. Methodology of their work is as follows:

- image is first resized to predefined quantized window sizes :

  ({Wo,Ho}, where Wo, Ho ∈{10,20,40,80,160,320}).

- Each window is then scored with a linear model **w** ∈ R^{64} (of size 8 × 8, to be learned),

  $$s_l = \langle \mathbf{w}, \mathbf{g}_l \rangle \quad , \quad l = (i,x,y) \tag{8}$$

  s_l = *filter score*, $\mathbf{g}_l$ = *NG feature, l = location, i = size, (x,y) = position* of a window and ⟨.,.⟩ is a dot product.

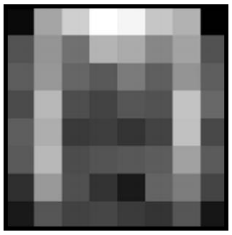


Fig.15. Example of learned weight matrix **w** of size 8 × 8

- Now, after scoring all the quantized windows, a small set of proposals from each size *i* is shortlisted using non-maximal suppression (NMS).
- Also, some window sizes (e.g. 10 × 500) are less likely than others to contain object instance (e.g. 100 × 100). Thus, final objectness score o_l is defined as-

  $$o_l = v_i \,.\, s_l + t_i \qquad \textbf{(9)}$$

  where v_i *,* $t_i \in \mathrm{R}$ *are separately learnt coefficient and a bias terms for each quantized size i.*

3. ***Zitnick et. al*** **[11]** observed that the number of contours that are wholly contained in a bounding box is indicative of the likelihood of the box containing an object. They proposed a simple box objectness score that measures the number of edges that exist in the box minus those that are members of contours that overlap the box's boundary. Brief methodology of their work is as follows:

- Finding edge groups and affinities between them: Edge groups are formed by considering the 8-connected components of an edge pixel. After the edge groups are formed, affinities are found for a set of edge groups $s_i \in S$ as -

  $$a(s_i, s_j) = |\cos(\theta_i - \theta_{ij})\cos(\theta_j - \theta_{ij})|^{\gamma} \qquad \textbf{(10)}$$

  where θ_{ij} is the angle between x_i and x_j. The value of γ is used to adjust the affinity between each pair of neighboring groups.

- Bounding box scoring: Once the set of edge groups S and their affinities are found, first the cumulative sum of the magnitudes m_i of edge pixels in a group s_i is found as -

  $$m_i = \sum\nolimits_{p \in s_i} m_p \ , \qquad i = 1,2, \ldots\ldots, length(S) \qquad \textbf{(11)}$$

For each group s_i, they defined a continuous value $w_b(s_i) \in [0,1]$ that indicates whether s_i is wholly contained in b, $w_b(s_i) = 1$, or not, $w_b(s_i) = 0$. Let S_b denote the set of edge groups that overlap with the bounding box b's boundary. For all $s_i \in S_b$, $w_b(s_i)$ is set to 0. Similarly, $w_b(s_i) = 0$ for all s_i for which $\bar{x}_i \notin b$. For remaining edge groups for which $\bar{x} \in b$ and $s_i \notin S_b$, they computed $w_b(s_i)$ as follows -

$$w_b(s_i) = 1 - \max_T \prod_j^{|T|-1} a(t_j, t_{j+1}) \tag{12}$$

where T is an ordered path of edge groups with a length of $|T|$ that begins with some $t_1 \in S_b$ and ends at $t_{|T|} = s_i$. If no such path exists, then they defined $w_b(s_i) = 1$. Now using the values of w_b, they defined their bounding box score using-

$$h_b = \frac{\sum_i w_b(s_i) m_i}{2(b_w + b_h)^\kappa} \tag{13}$$

where b_w and b_h are the bounding box's width and height. κ is used to offset the bias of larger windows having more edges on average.

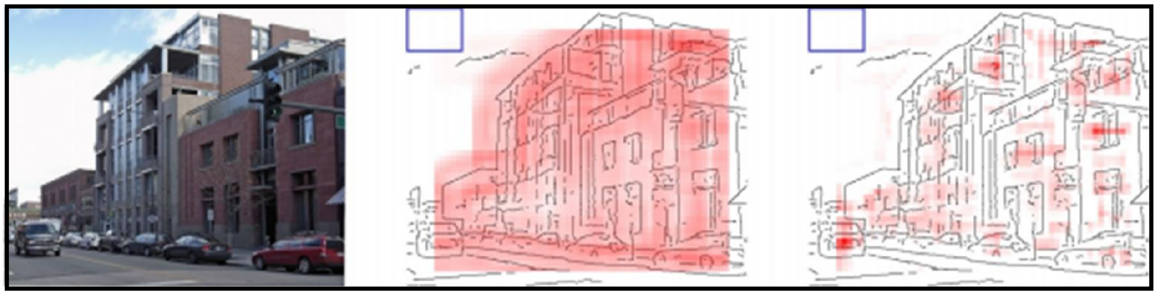

(a) (b) (c)

Fig.16. (a) RGB Image and its edge magnitude map (b) with no contour removal, and (c) with contour removal

CHAPTER 3: PROPOSED METHODOLOGIES

3.1 Proposed Method 1 for objectness: Quadrant Symmetry measure

In this section, we describe our proposed method for objectness measure, namely, ***Quadrant Symmetry*** measure. Any object we observe in nature is symmetric in some way or the other and it is compact with a closed boundary and its gradients on its boundary emerge in all directions. Based on this observation, we formulated our first objectness measure. We first describe the algorithm of the method, its implementation from a Cartesian coordinate system point of view and then in the subsequent section, we explain its coordinate system transformation for a more efficient computation of the objectness score.

3.1.1 Algorithm:

1. First of all, edge magnitude map of the input RGB image $\boldsymbol{M}(x,y)$ is extracted using **[10]**.

2. As for the gradient angles at those pixel positions, we first find the gradients $\boldsymbol{g_y}$ and $\boldsymbol{g_x}$ of the image in *y*- and *x*- directions using Sobel masks and find the gradient angle map as follows :-

$$\boldsymbol{\theta}(x,y) = \tan^{-1}\left(\frac{\boldsymbol{g_y}}{\boldsymbol{g_x}}\right) \tag{14}$$

3. Finally, to generate the edge map $\boldsymbol{E}(x,y)$, we apply non-maximum suppression to $\boldsymbol{M}(x,y)$ and threshold it to obtain the edge map.

$$\boldsymbol{E}(x,y) = NMS\big(\boldsymbol{M}(x,y)\big) > \boldsymbol{Threshold} \tag{15}$$

where NMS is Non-Maximum Suppression.

For our experiments, we chose the $\boldsymbol{Threshold}$ value as 0.1, because it gives very clean edge map consisting of object boundaries mostly, devoid of most of the background, like grasses, tree leaves, etc.

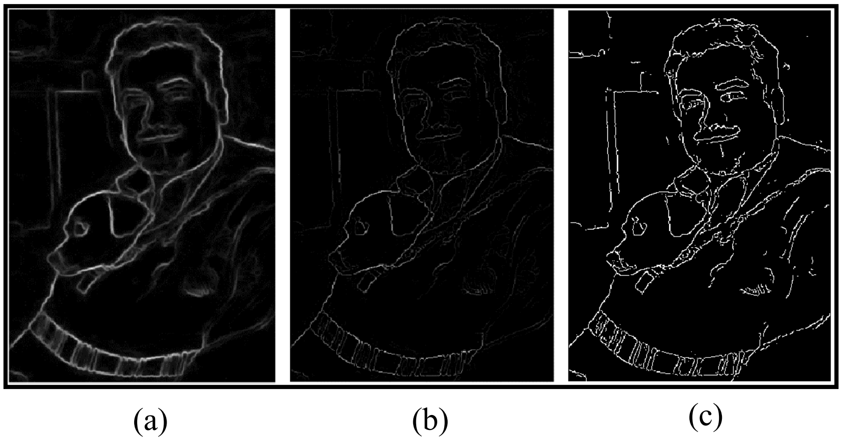
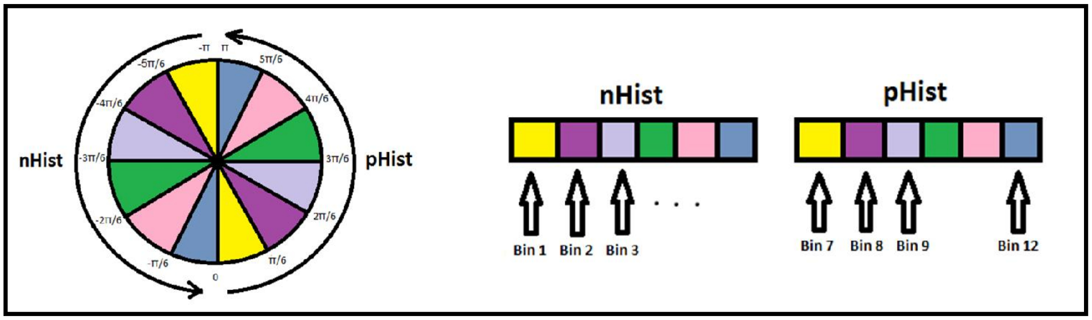


Fig.17. (a) Edge magnitude map of an image; (b) NMS sampled image and (c) Edge map

4. Gradient angles obtained are in the range$[-\pi, \pi]$. Now, we remove the gradient angle values from those pixel positions where $\boldsymbol{E}(x, y) = 0$ and store a value *v* there which is outside the gradient range, say for e.g., 5.

$$\boldsymbol{\theta}(x, y) = \boldsymbol{E}(x, y).\boldsymbol{\theta}(x, y) + v.\left(1 - \boldsymbol{E}(x, y)\right). \tag{16}$$

5. Now for each proposal window in the image, we divide the entire gradient angle range into $\boldsymbol{B}$ equal sectors or **bins**. The whole gradient angle range will be divided into 2x$\boldsymbol{B}$ bins. The first $\boldsymbol{B}$ bins will span the gradient angle range$(-\pi, 0]$, and the next $\boldsymbol{B}$ bins will span the range$(0, \pi]$, as shown in the figure below. The 1st bin will cover the range$\left(-\pi, -\frac{(B-1)\pi}{B}\right]$, the 2nd bin$\left(-\frac{(B-1)\pi}{B}, -\frac{(B-2)\pi}{B}\right]$, and so on. The (***B-1)***th bin will cover$\left(-\frac{\pi}{B}, 0\right]$, the $\boldsymbol{B}$th bin$\left(0, \frac{\pi}{B}\right]$, ... and the last (***2B)***th bin$\left(\frac{(B-1)\pi}{B}, \pi\right]$.



(a) (b)

Fig.18: (a) Visual representation of gradient angle wheel and (b) corresponding positive angle (pHist) and negative angle (nHist) histograms for ***B***=6.

6. Then we determine the sector/bin in which gradient angle of each pixel in the window falls into. All such edge pixels belonging to same sector/bin are summed up. This is done for every edge pixel and finally, a histogram ***H*** of such bins is formed for the proposal window and normalized. Mathematically,

$$pHist(i) = \sum_{\substack{0 \leq x \leq M \\ 0 \leq y \leq N \\ \frac{(i-1)\pi}{B} < \boldsymbol{\theta}(x,y) \leq \frac{i\pi}{B}}} \boldsymbol{E}(x,y) \quad \textbf{(17)}$$

$$nHist(i) = \sum_{\substack{0 \leq x \leq M \\ 0 \leq y \leq N \\ -\frac{(B-i-1)\pi}{B} < \boldsymbol{\theta}(x,y) \leq -\frac{(B-i)\pi}{B}}} \boldsymbol{E}(x,y) \quad \textbf{(18)}$$

where *M* and *N* are no. of rows and columns in the window, $i \in [1, \boldsymbol{B}]$.

$$\boldsymbol{H}(m) = \begin{cases} nHist(m), & 1 \leq m \leq \boldsymbol{B} \\ pHist(m - B), & \boldsymbol{B} + 1 \leq m \leq 2\boldsymbol{B} \end{cases} \quad \textbf{(19)}$$

where $\boldsymbol{E}(x, y)$ is the edge map, $\boldsymbol{\theta}(x, y)$ is the gradient angle map, $m \in [1, 2\boldsymbol{B}]$.

7. Now we have the histogram ***H*** at our disposal. Next step is to check that about which axis the current image window is most symmetric. To do that, we find a Chi-Square distance score ***S*** of 1st half (1st ***B*** elements) of ***H*** with its 2nd half. Mathematically,

$$\boldsymbol{S}\ (Chi - Square\ distance) = \sum_{i=1}^{B/2} \frac{[\boldsymbol{H}(i) - \boldsymbol{H}(i + B/2)]^2}{[\boldsymbol{H}(i) + \boldsymbol{H}(i + B/2)]} \quad \textbf{(20)}$$

By doing this, we are measuring the symmetry of the image window about 0° axis, i.e., vertical axis. In the next step, by circular shifting the histogram ***H***, i.e., keeping the 1st element at the end, we calculate the Chi-Square distance of this shifted histogram in the same way as mentioned above. This will give us the symmetry score ***S*** of the image window along $+\pi/B$ axis. Like this, we keep on circular shifting the histogram and get a total of ***B*** symmetry scores. We take the maximum of all the scores ***S*** so that we can get to know that the gradient angle, corresponding to the max-score, is the angle of maximum symmetry for this window and we will call it θ_{max}. If there are more than one angle of maximum symmetry, then we consider them all for the next step.

8. Once θ_{max} is found out, our next step is to find the axis ***l*** which has the slope $\tan(\theta_{max})$. To do that, we need to find the point through which the axis will pass because a line can be

characterized by its slope and a point through which it passes. We consider all the bins which constitute the 1st half of ***H*** when gradient angle $\theta = \theta_{max}$ and tag them as set ***A***. For all the bins in the 2nd half, tag given is ***B***. Now we consider all the edge points which constitute the set ***A*** and take the average of the *x*- and *y*- coordinates of these points. We get a point $P_1 \equiv (x_1, y_1)$. Repeating the same for edge pixels in set B, we get another point$P_2 \equiv (x_2, y_2)$. So, if $p \equiv (x, y)$ is an edge pixel, $P_i \equiv (x_i, y_i)$ is obtained as

$$x_i = \begin{cases} \frac{1}{N_A} \sum_{x|p\epsilon A} x & if\ i = 1 \\ \frac{1}{N_B} \sum_{x|p\epsilon B} x & if\ i = 2 \end{cases} \tag{21}$$

and

$$y_i = \begin{cases} \frac{1}{N_A} \sum_{y|p\epsilon A} y & if\ i = 1 \\ \frac{1}{N_B} \sum_{y|p\epsilon B} y & if\ i = 2 \end{cases} \tag{22}$$

where N_A represents no. of edge pixels in set ***A***, N_B represents no. of edge pixels in set ***B***. Now the point *P* through which the axis ***I*** will pass through is simply -

$$P = \frac{1}{2} \sum_{i=1}^{2} P_i \tag{23}$$

9. Now we have the point *P* through which axis ***I*** will pass along with its slope $\tan(\theta_{max})$. Now we consider another axis ***I_p*** which is perpendicular to the axis ***I***, i.e., slope is $-\frac{1}{\tan(\theta_{max})}$ and passing through the same point P. These two lines form a rotated coordinate system for the edge pixels in the image window, with the angle of rotation as θ_{max}.

10. We have 4 quadrants for this rotated coordinate system, $Q_i, i = 1,2,3,4$. We have edge pixels distributed in these 4 quadrants. Now for edge pixel assignment in the respective quadrants, we do the following -

Equation of axis ***I***: $a_1x + b_1y + c_1 = 0$ **(24)**
Equation of axis ***I***

An edge pixel $p \equiv (x, y)$ will be assigned to quadrant -

a) Q_1 if $a_1x + b_1y + c_1 < 0 \quad and \quad a_2x + b_2y + c_2 < 0$
b) Q_2 if $a_1x + b_1y + c_1 > 0 \quad and \quad a_2x + b_2y + c_2 < 0$ **(25)**
c) Q_3 if $a_1x + b_1y + c_1 < 0 \quad and \quad a_2x + b_2y + c_2 > 0$
d) Q_4 if $a_1x + b_1y + c_1 > 0 \quad and \quad a_2x + b_2y + c_2 > 0$

11. We need to form 4 quadrant histograms $H_i, i = 1,2,3,4$, corresponding to four quadrants $Q_i, i = 1,2,3,4$, respectively. These histograms will be similar to the previous histogram ***H***, with the difference is that these histograms are now quadrant-localized. Each histogram is 2***B*** length long. So, we will take all the edge pixels present in the quadrantQ_i, assign them to different bins according to the gradient angle distribution as mentioned in step 5, and finally normalize them. Once these histograms are formed, we are ready to calculate the final symmetry score for the image window. We will use histogram intersection distance as the metric for measuring symmetry score. That will be done as follows-

$$Score\ S1 = HistIntDist(H_1, rev(H_2))$$
$$Score\ S2 = HistIntDist(H_3, rev(H_4))$$ **(26)**

$HistIntDist(M, N)$ is the histogram intersection distance between two histograms M and N which is given by -

$$HistIntDist(M, N) = \sum_{i=1}^{L} \min(M(i), N(i)) \quad where\ L = length\ of\ M\ or\ N \qquad \textbf{(27)}$$

So, final symmetry score $SymmScore$ for the proposal window is given by -

$$SymmScore = S1 + S2 \qquad \textbf{(28)}$$

3.1.2 Efficient implementation: Coordinate System transformation

Note that for determining QS score, we need to do to every pixel in the window to get its gradient angle as well as to assign it to a quadrant histogram. This is very much time consuming and computationally expensive. To do it alternatively, we transform the Cartesian coordinate space to another coordinate space, whose x-axis is gradient angle of an edge pixel, and y-axis is angle of the position of an edge pixel from the centroid of all the edge pixels. This technique is inspired from [6]. We use discrete mapping to map all the edge pixels to the new coordinate space. Our new coordinate space is stored as a $2\boldsymbol{B}\ \times 2\boldsymbol{B}$ matrix, i.e., we divide the gradient angle range into $2\boldsymbol{B}$ bins, and angular position range, which is $[0, 2\pi]$, into $2\boldsymbol{B}$ bins as well. Some examples of the original and transformed coordinate system are shown below.

Now, instead of first finding the angle of maximum symmetry and then dividing the window space into 4 quadrants at that angle, we directly form the quadrant histograms from the transformed matrix for each edge normal angle and then calculate the scores for each of them. Out of these scores, we choose the maximum one. This is much faster and way easier to determine the maximum symmetry score than in Cartesian space.

To start with, we first consider symmetry about vertical axis. For that, we consider the first $\boldsymbol{B}/\boldsymbol{2}$ rows. Summing them up column wise, we will get the quadrant histogram for angular positions $\left(0, \frac{\pi}{2}\right]$. For the next histogram, we sum up next $\boldsymbol{B}/\boldsymbol{2}$ rows, which accounts for the

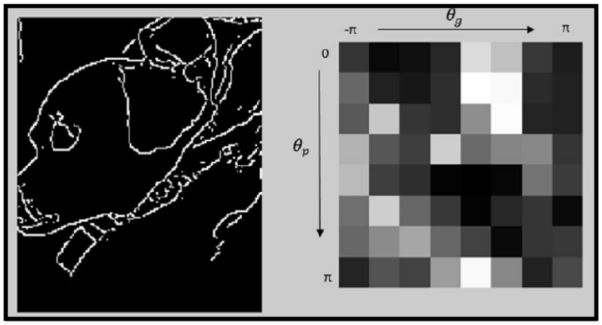
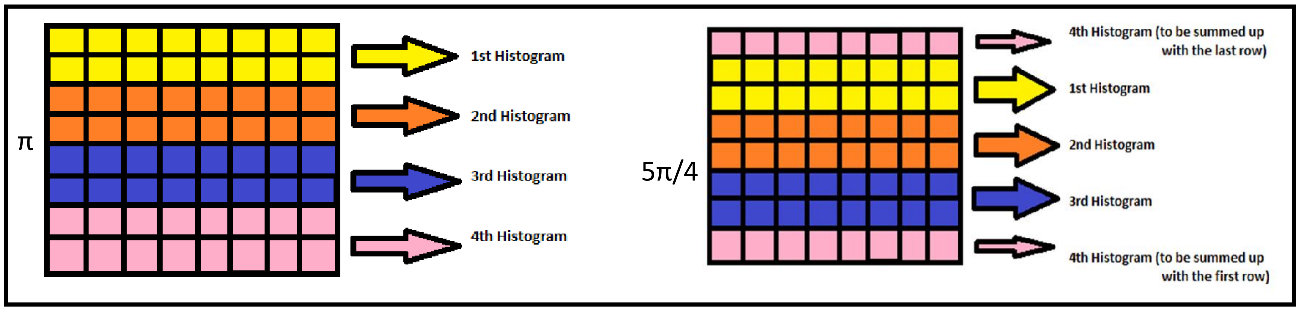


Fig.19: Mapping of Cartesian edge image to the new coordinate space. Note that the transformed space is of size $2\boldsymbol{B} \times 2\boldsymbol{B}$ (here, $\boldsymbol{B}$=4). θ_p stands for angular position of an edge pixel and θ_g stands for the gradient angle of an edge pixel.

angular positions$\left(\frac{\pi}{2}, \pi\right]$. Likewise, we get all 4 histograms. We use histogram intersection distance to compute the symmetry score.

For the next symmetry score, we compute the histograms by circularly shifting them downwards by one row and summing them up again and calculating the score. In this way, all the scores can be determined by simple circular shifting of the rows of the transformed matrix and adding them up row wise to get the histograms.


π
1st Histogram
2nd Histogram
3rd Histogram
4th Histogram
5π/4
4th Histogram (to be summed up with the last row)
1st Histogram
2nd Histogram
3rd Histogram
4th Histogram (to be summed up with the first row)

(a) (b)

Fig.20: Visual representation for finding quadrant histograms using transformed matrix. ($\boldsymbol{B}$=4). (a) is for score calculation about vertical axis, (b) is circular shifted version of the left image, score calculation about 5π/4 axis.

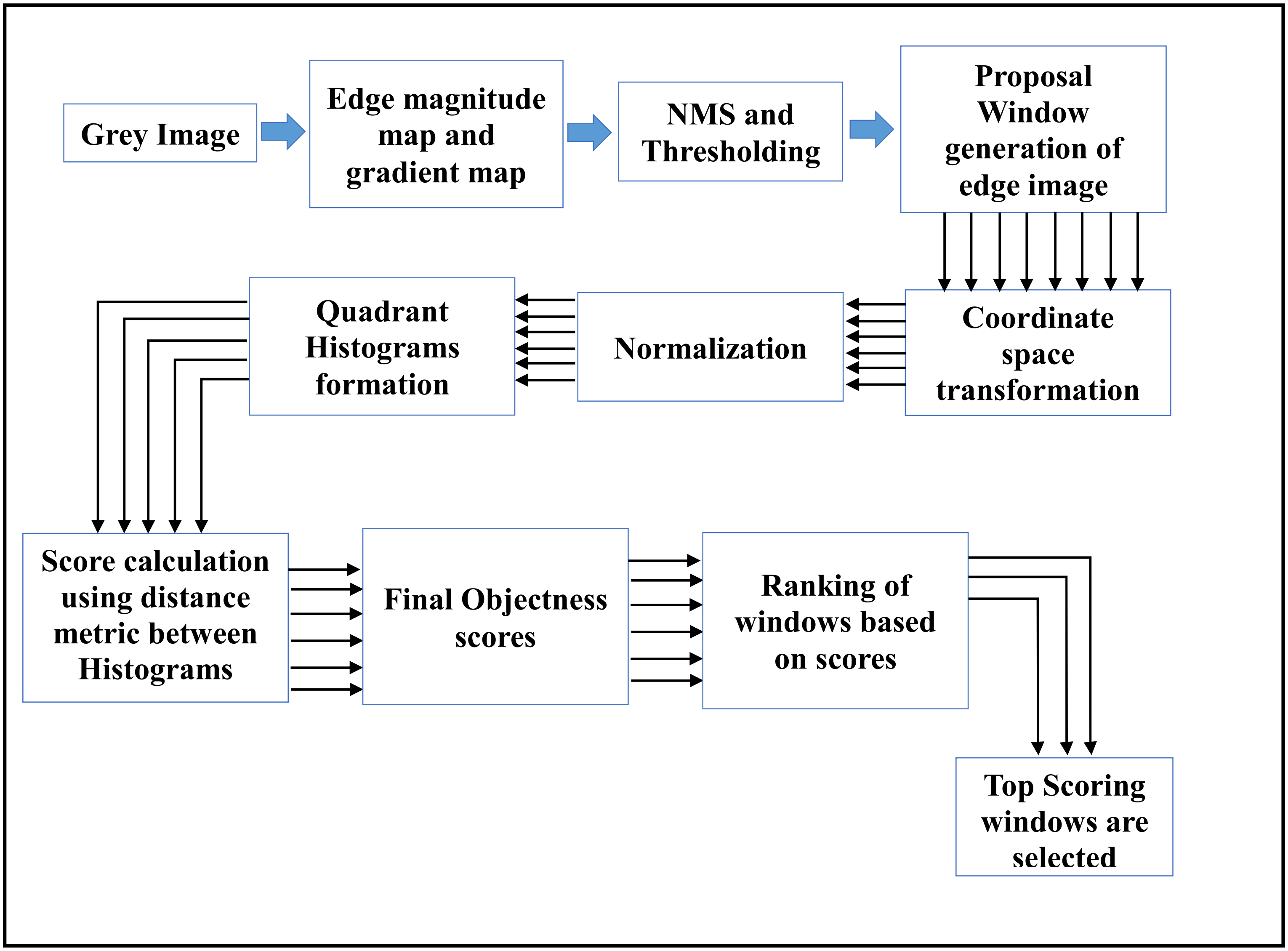

Grey Image
Edge magnitude map and gradient map
NMS and Thresholding
Proposal Window generation of edge image
Coordinate space transformation
Normalization
Quadrant Histograms formation
Score calculation using distance metric between Histograms
Final Objectness scores
Ranking of windows based on scores
Top Scoring windows are selected

Fig.21: Flowchart of the algorithm for proposed method 1

3.2 Proposed Method 2 for objectness: Dictionary of Objects

In this section, we will describe our proposed method 2, i.e., objectness measure using ***Dictionary of Objects***. As mentioned before, we feel that like texts, most of the objects we see in nature can be represented using a visual dictionary. So, inspired by that thought, we formulated this objectness measure. First we describe the feature extraction for the method, and then the algorithm to learn dictionaries for object classes using two methods: **K-SVD** and **Spherical K-means clustering**.

3.2.1 Feature extraction for the dictionaries: Gradient and Haar Grid

- We use gradient grid as our first feature for this method. Furthermore, we modified this feature by using Haar grid instead of gradient grid because Haar grid boosts the boundary edges of the objects and suppresses internal edges, as it compares a group of pixels around any point and not just its immediate neighbors.
- Given an RGB image, gradient/Haar map is obtained for all the three planes and norm of gradients is derived from them. This normed gradient/Haar map is now considered for scoring the windows.

$$G(x,y) = \sqrt{G_R(x,y)^2 + G_G(x,y)^2 + G_B(x,y)^2} \quad \textbf{(29)}$$

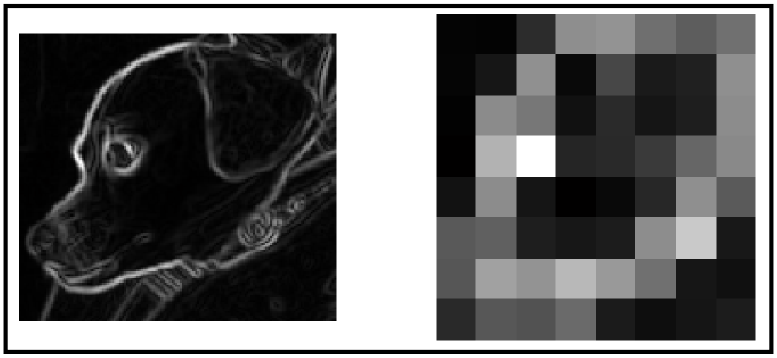
where G_R, G_G, G_B are gradient maps of RGB planes of the image respectively. This takes care of color contrasts while deriving the gradients.

- Let 'w' be the desired window characterized by the parameters [x_{min}, x_{max}, y_{min}, y_{max}]. This window 'w' is further divided into a grid of size 8×8 and in each subdivided window, sum of gradient magnitudes inversely weighted by the length and breadth of itself are computed thus resulting in a 64 dimensional feature vector.

$$Y(p,q) = \frac{|G(w,p,q)|}{(l(p,q) + b(p,q))/2} \quad \textbf{(30)}$$

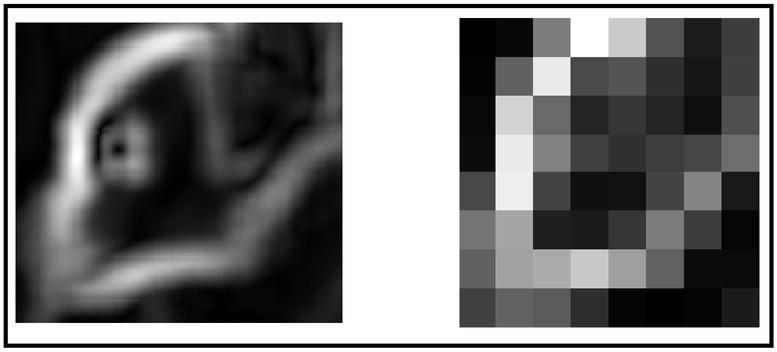
where G is the gradient/Haar map of the image, w is the chosen window, p and q determine the desired subwindow and $l(p, q)$ and $b(p, q)$ are the length and breadth of the subwindow.

- These features are used for training the dictionary atoms using the linear dictionary learning algorithms.



(a) (b)

Fig.22: (a) Normed Gradient map of a proposal window, (b) 8×8 gradient grid of gradients.



(a) (b)

Fig.23: (a) Normed Haar map of a proposal window, (b) 8×8 Haar grid of gradients.

3.2.2 Dictionary formation using K-SVD Algorithm

KSVD is a linear dictionary learning algorithm that develops dictionary atoms (also known as *basis vectors*) of given sample of feature-vectors.

Let *Y* be a matrix containing features such that each column corresponds to a feature. For a given dictionary *D*, all the input vectors are decomposed on these atoms thus resulting in coefficient matrix *X*, where each i^{th} row of the matrix corresponds to the coefficients of i^{th} feature vector of *Y* such that multiplication of matrices *D* and *X* would approximate to matrix *Y*.

KSVD algorithm contains the following steps:

- **Initialization:** Dictionary is initialized randomly and Sparsity threshold parameter 'T' is chosen.

- **Updating the coefficients:** Given the Dictionary *D*, *X* is obtained by selecting those atoms which have maximum projection. This is done using OMP algorithm.

- **Updating the Dictionary:** Given coefficient matrix *X*, *D* is obtained by using SVD.

Pseudo code for KSVD algorithm:

```
for $i < N$ do
    $\mathbf{X} = \mathrm{OMP}(\mathbf{D}, \mathbf{Y}, \mathrm{l}, T)$
    for $j \leq K$ do
        idx $\Leftarrow$ all non zero indices of $\mathbf{x}_j^T$
        if idx is empty then
            $\mathbf{E} \Leftarrow \mathbf{Y} - \mathbf{DX}$
            $p \Leftarrow$ position where error vector has maximum norm
            $\mathbf{x}_j^T \Leftarrow 0$  % Deselect j − th atom
            $\boldsymbol{d_j} \Leftarrow \frac{y_p}{\|y_p\|_2}$
            else
            $\mathbf{lst} \Leftarrow \{\mathbf{x}_k | k \in \mathbf{idx}\}$
            $\mathbf{x}_j \Leftarrow 0$    % Deselect j − th atom
            $\mathbf{E}_k^R \Leftarrow \mathbf{Y}_{\mathbf{lst}} - \mathbf{DX}$
            $[\mathbf{U}, \mathbf{S}, \mathbf{V}^T] \Leftarrow \mathrm{SVD}(\mathbf{E}_k^R)$
            $\mathbf{x}_{j,p}^r \Leftarrow \mathbf{S}_{1,1}\, \mathbf{V}_1 \;\; \forall\, p \in \mathbf{idx}$
            $\mathbf{d_j} \Leftarrow \frac{\mathbf{U_1}}{\|\mathbf{U_1}\|_2}$
        end if
    end for
end for
```

```
function $[\mathbf{x}] = \mathrm{OMP}(\mathbf{y}, \mathbf{D}, T, \epsilon)$
$\mathbf{r}_0 = \mathbf{y}$
for $k < T$ do
    p $\Leftarrow \mathbf{D}^T \mathbf{r}_{k-1}$  % Project dictionary on error
    $\mathbf{l}_k \Leftarrow$ add to list index where column $|\mathbf{p}|_i$ is maximum
    $\mathbf{D}_\mathrm{k} \Leftarrow$ atoms from $\mathbf{D}$ which have entries in $\mathbf{l}_k$
    $\mathbf{x}_k \Leftarrow \mathbf{D}_k^\dagger \mathbf{y}$  % $\mathbf{x} = argmin_{\mathbf{x}} \|\mathbf{y} - \mathbf{Dx}\|$
    $\mathbf{r}_k \Leftarrow \mathbf{y} - \mathbf{D_k x_k}$  % Calculate error of representation
    if $\|\mathbf{r}_\mathrm{k}\|_2^2 < \epsilon$ then
        break;
    end if
end if
Return $\mathbf{x} \Leftarrow \mathbf{x}_\mathrm{k}$
```

$\mathbf{y} \Leftarrow$ observed signal to encode of size $\mathrm{n} \times 1$
$\mathbf{D} \Leftarrow$ dictionary of atoms of size $\mathrm{n} \times K$
$T \Leftarrow$ sparsity threshold
$\epsilon \Leftarrow$ error tolerance
$K \Leftarrow$ number of atoms in dictionary
$N \Leftarrow$ number of k − svd iterations to run

3.3.3 Dictionary formation using Spherical K-means clustering

Spherical k-means is a clustering algorithm where centroids (also called Dictionary atoms) are formed by taking into the angle of orientation from the origin. It is a way of dictionary learning with unit sparsity.

In this method, centroids are constrained to be on a sphere of unit radius, thus making them normal but not necessarily orthogonal. For each data point, dot product of itself with all the centroids are calculated. Then, the data point is assigned to the cluster to whose centroid it is nearer to. In other words, data point is assigned to that cluster whose centroid gives the maximum value of dot product. Let *X* be the matrix in which each column is a feature vector for a single sample, and let D be the dictionary where each column represents the centroids of respective clusters. S is a variable matrix that stores the information of the cluster to which the data point belongs. Then the algorithm iterates as follows.

$$X(j,i) = \begin{cases} D_j^T Y_j & if\ j ==\ argmax_l |D_l^T Y_l| \\ 0 & otherwise \end{cases}$$

$$D = Y.X^T + D \tag{31}$$

$$D_j = \frac{D_j}{\left\|D_j\right\|_2}$$

3.3.4. Window scoring using trained dictionaries

Gradient map of image is formed using Sobel masks and 100000 uniformly distributed proposal windows of various sizes are chosen and for each window 'w', 8×8 normed gradient features are computed as mentioned above. These are reshaped to a column vector as follows.

$$y(w, (8(q-1)+p)) = G(p,q) \tag{32}$$

- Scoring windows using dictionary from **K-SVD** algorithm-

  Dictionary is trained using a few object windows and is used for testing.

$$D = KSVD(Y, K, T, \in) \tag{33}$$

Dot product of feature vector and each dictionary atom is calculated. Since the dictionary atoms are orthonormal, norm of the coefficients obtained would result in the component of the window with respect to the space spanned by dictionary atoms and is used as score for the windows.

$$S(w) = \ ||D^T.y(w)||_2 \tag{34}$$

- Scoring windows using dictionary from **Spherical K-means** algorithm-

  Dictionary is trained using a few object windows and is used for testing.

$$D = SphK(Y, K) \tag{35}$$

Dot product of feature vector with each dictionary atom is calculated. Since each dictionary atom itself represents a group of desired features (of object window), the dot product would imply how near is the feature near to that of a group of object windows. So maximum of the coefficients is chosen as score for the window.

$$S(w) = Max\{D^T.Y(w)\} \tag{36}$$

- NMS sampling of the scored windows is done (with respect to either individual scores or weighted sum of the scores) to obtain high scoring windows.

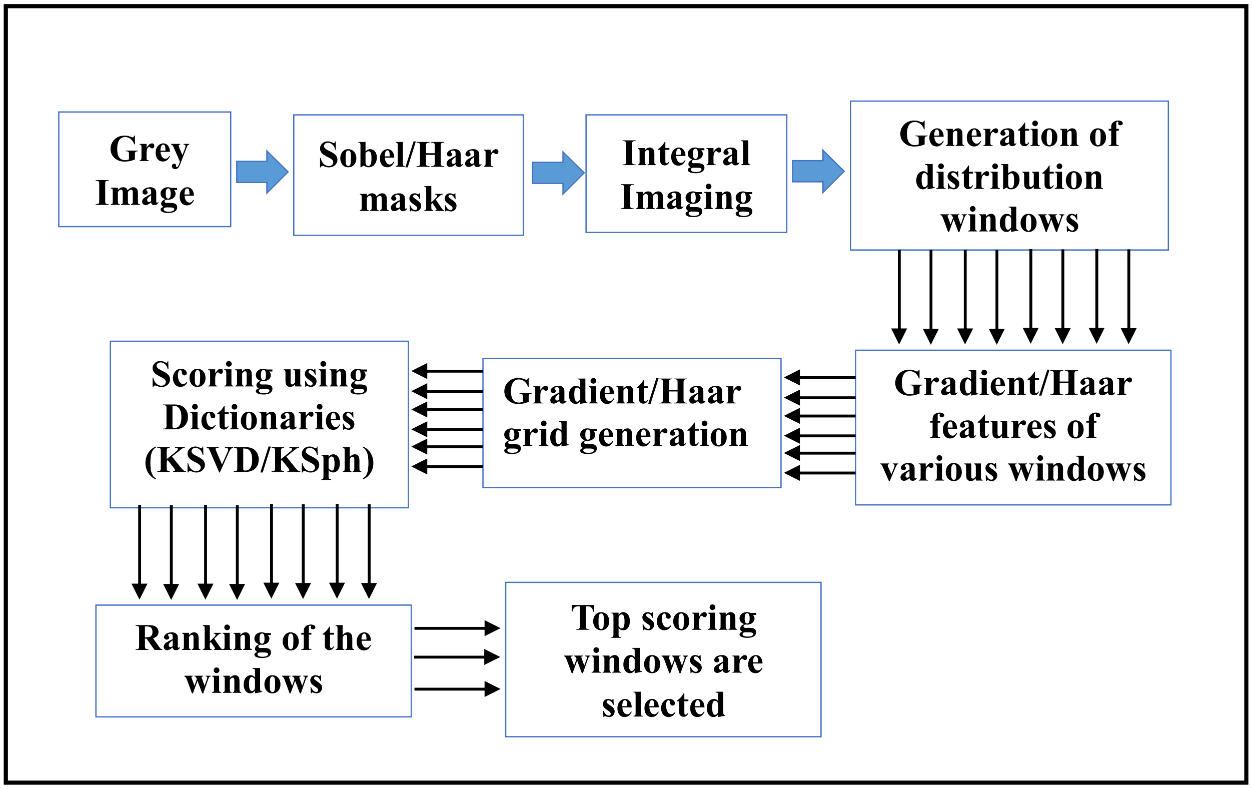

Grey Image
Sobel/Haar masks
Integral Imaging
Generation of distribution windows
Gradient/Haar features of various windows
Gradient/Haar grid generation
Scoring using Dictionaries (KSVD/KSph)
Ranking of the windows
Top scoring windows are selected

Fig.24: Flowchart of the algorithm for proposed method 2

CHAPTER 4: EVALUATION AND RESULTS

4.1 Experimental evaluation

We evaluated the performance of our proposed objectness measures on the popular *PASCAL VOC 2007* dataset **[13]**, which is a standard dataset for evaluation of class-specific object detectors **[14**, **15**, **16]**. We show our results on the test part of the dataset, which consists of 4952 test images with all the objects from 20 categories annotated. The large number of objects and the variety of classes make this dataset best suited to our evaluation, as we want to find all objects in the image, irrespective of their classes. This dataset is very challenging: the objects appear against heavily cluttered backgrounds and vary greatly in location, scale, appearance, viewpoint and illumination.

DR-#WIN curve: We evaluate the performance using the famous DR (Detection Rate) - #WIN (No. of proposal windows) curve.

$$\text{Detection rate (DR)} = \frac{Total\ no.of\ Detected\ proposal\ windows\ in\ M\ images}{Total\ No.of\ ground\ truth\ windows\ in\ M\ images} \quad \textbf{(37)}$$

$$\text{Criteria for proposal window detecting an object} = \frac{PWA \cap GTWA}{PWA \cup GTWA} \quad \textbf{(38)}$$

where *PWA= Proposal Window Area, GTWA=Ground Truth Window Area*

- **DR-#WIN** curve for ***Quadrant Symmetry***:

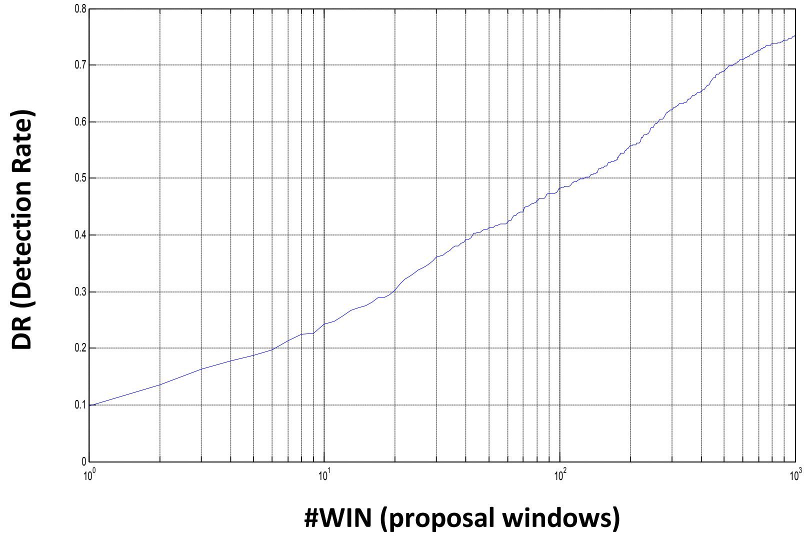


Fig.25: DR-#WIN curve using ***Quadrant Symmetry***

- **DR-#WIN** curve 1 for ***Dictionary of Objects***: Using KSVD Algorithm

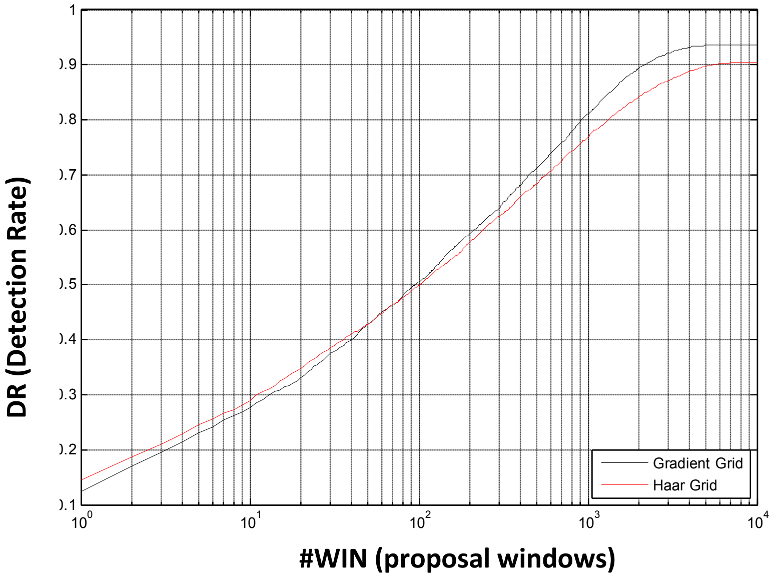


Fig.26: DR-#WIN curve for ***Dictionary of Objects***
using KSVD

- **DR-#WIN** curve 2 for ***Dictionary of Objects***: Using Sph. K-means Algorithm (K-Sph.)

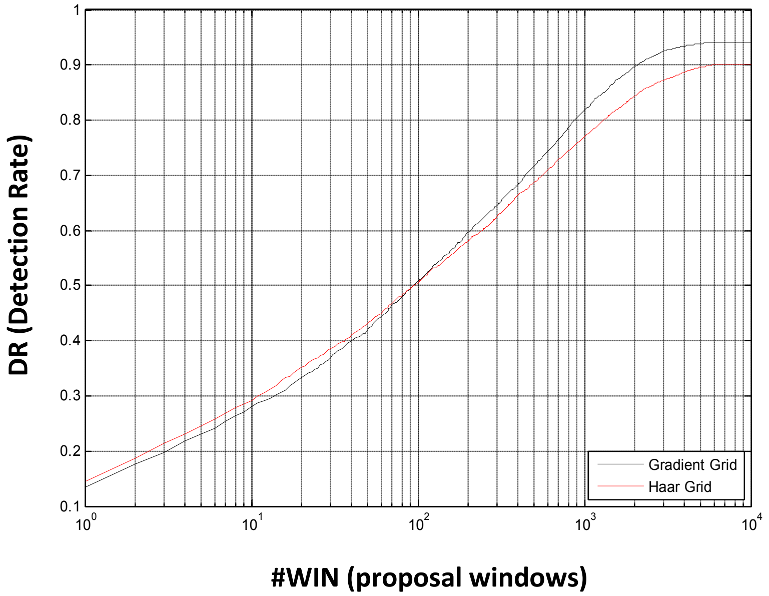


Fig.27: DR- #WIN curve for ***Dictionary of Objects*** using K-Sph.

- We have also observed the outputs in ***Dictionary of Objects*** - objectness measure with different normalization procedures of the feature grids of the windows. We normalized the grids in 3 ways:
  1. No normalization - we just summed up the intensity values in each sub-grid.
  2. Dividing each sub-grid in the feature grid by mean of the intensity values in that sub-grid.
  3. Dividing each sub-grid in the feature grid by mean of length and breadth of the sub-grid.
- We observed that normalization procedure 3 gave the best results compared to simple summing up or averaging of intensities of pixels in each sub-grid. The corresponding DR-#WIN curves for the different normalization procedures are plotted for 200 test images.

- **DR-#WIN** curve for ***Dictionary of Objects***: KSVD with normalization procedure 1

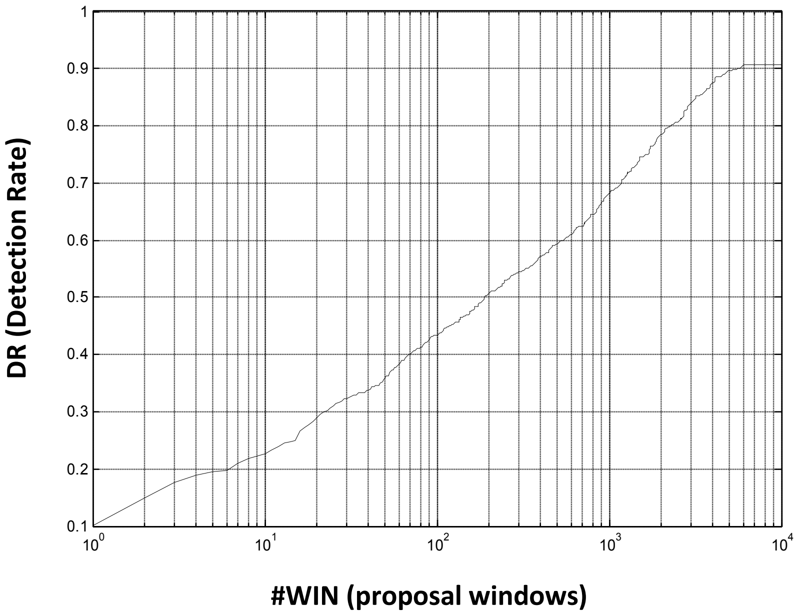


Fig.28: DR-#WIN curve: KSVD using normalization procedure 1

- **DR-#WIN** curve for ***Dictionary of Objects***: KSVD with normalization procedure 2

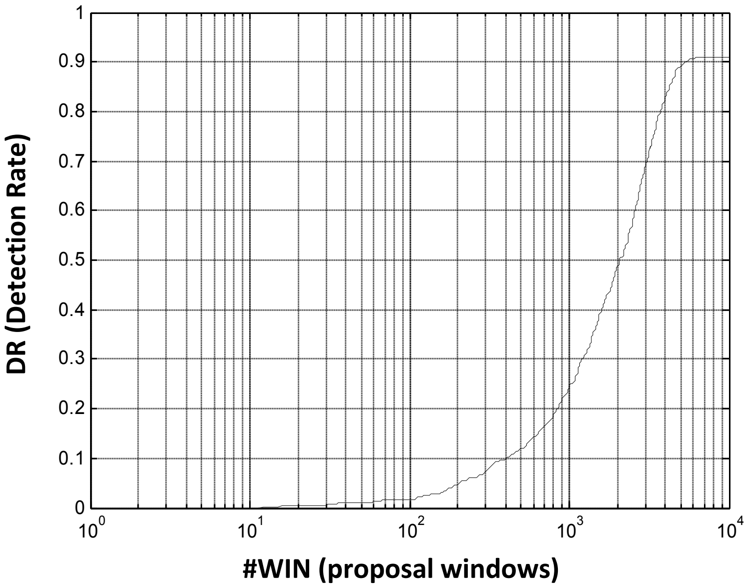


Fig.29: DR-#WIN curve: KSVD using normalization procedure 2

- **DR-#WIN** curve for ***Dictionary of Objects***: KSVD with normalization procedure 3

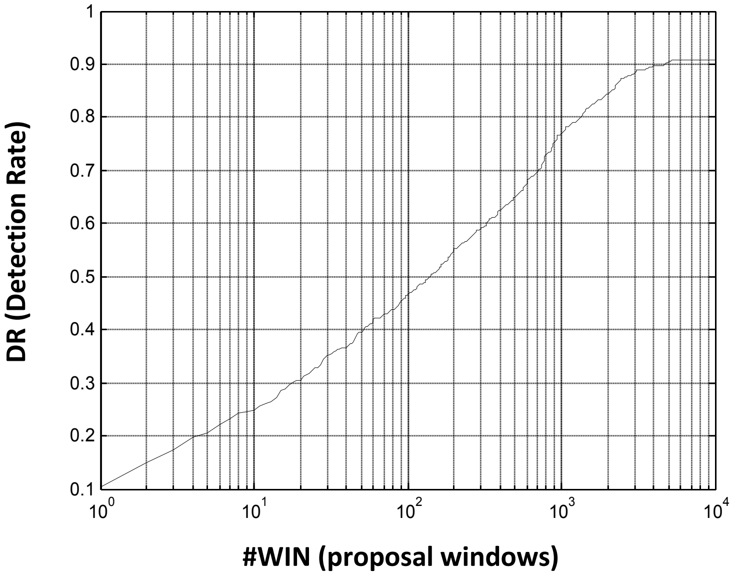


Fig.30: DR-#WIN curve: KSVD using normalization procedure 3

4.2 Comparison with state-of-the-art methods

The table below compares the various state-of-the-art methods of objectness with our proposed methods in terms of detection rate for a given number of proposal windows for IoU (Intersection over Union) of 0.5.

Method \ #WIN→	**1**	**10**	**100**	**1000**	**10000**
Objectness **[2]**	12%	40%	70%	90%	-
BING **[3]**	25%	38%	80%	96%	99.5%
Edge Boxes **[11]**	12%	40%	77%	95%	-
Quadrant Symmetry	12%	32%	55%	76%	83%

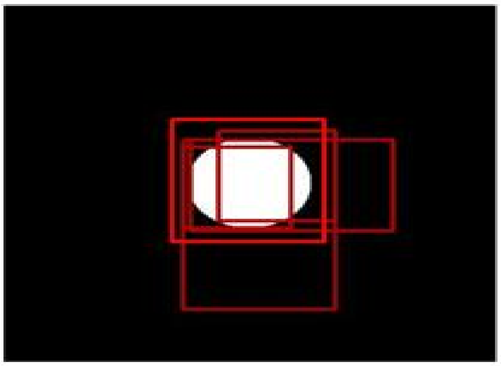
Method \ #WIN→	**1**	**10**	**100**	**1000**	**10000**
Gradient Grid: KSVD	12.5%	27.95%	50.98%	81.28%	93.49%
Gradient Grid: K-Sph	13.5%	28.1%	50.9%	81.67%	93.88%
Haar Grid: KSVD	14.53%	29.04%	49.86%	76.66%	89.74%
Haar Grid: K-Sph	14.55%	29.25%	50.55%	76.87%	90.07%

Table I: Comaprison of proposed methods with state-of-the-art methods

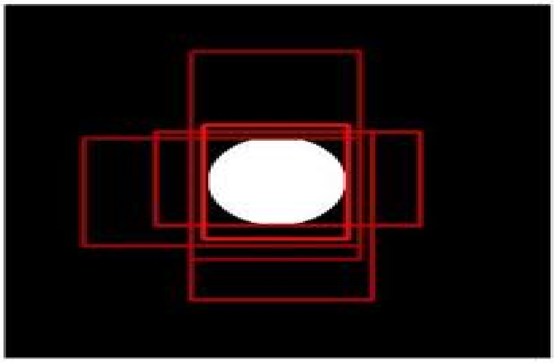
4.3 PASCAL VOC 2007 examples

In this section we show some image results of *PASCAL VOC 2007* dataset for object detection using our proposed objectness measures.

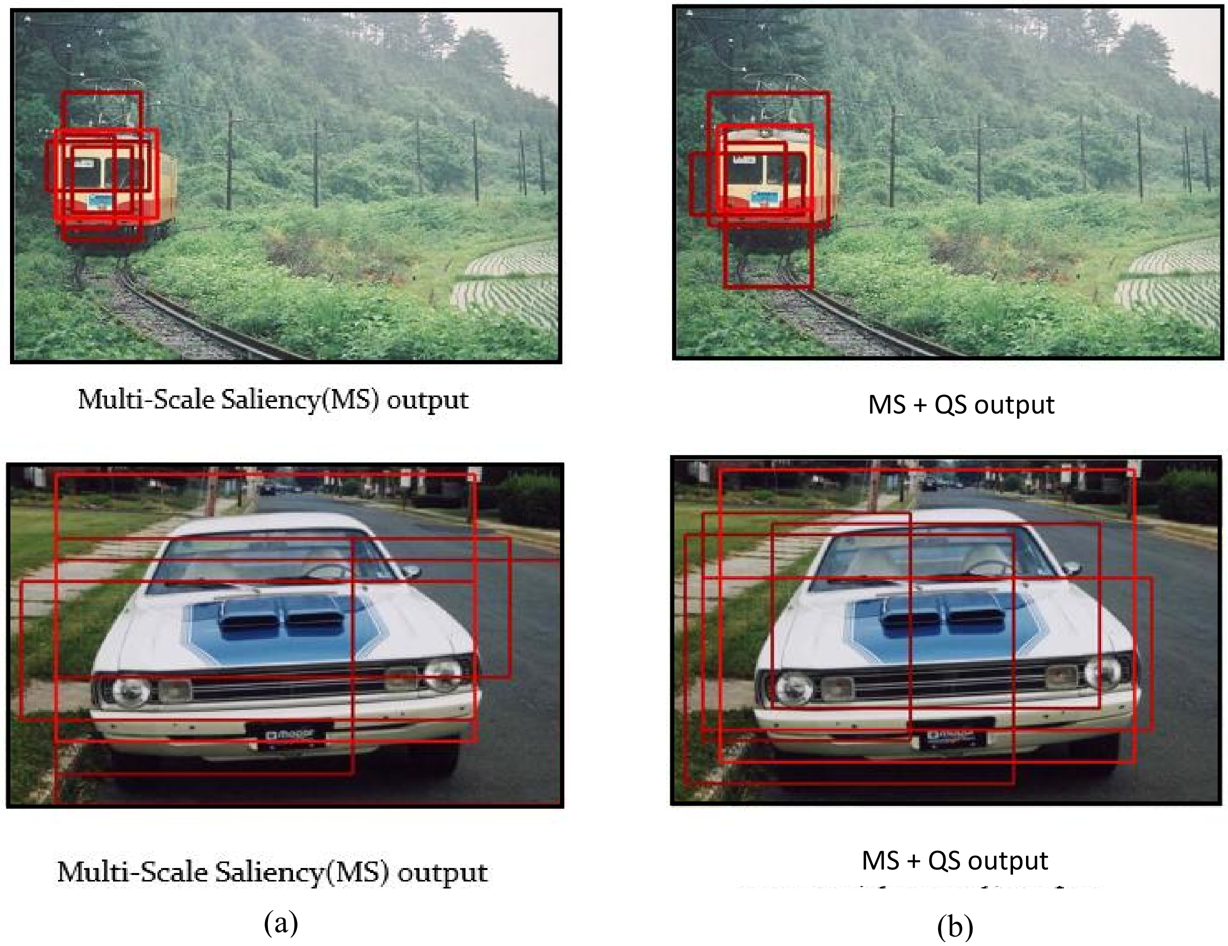
- Image results for ***Quadrant Symmetry*** measure:



Multi-scale saliency (MS) output



MS + QS output



Multi-Scale Saliency(MS) output

MS + QS output

Multi-Scale Saliency(MS) output

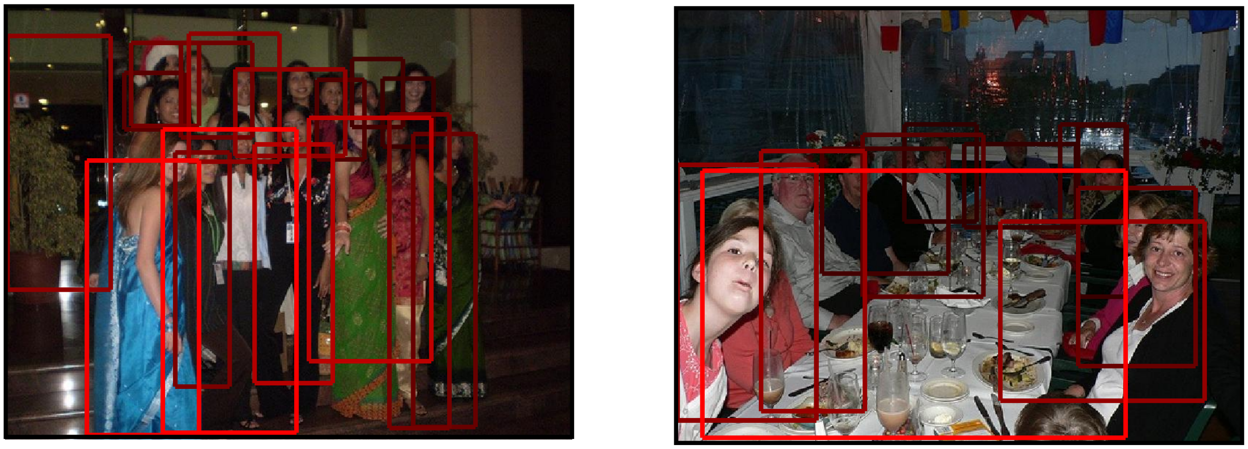
MS + QS output

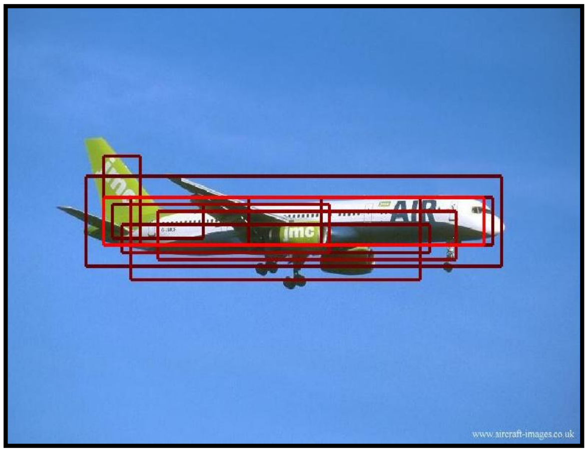
(a)

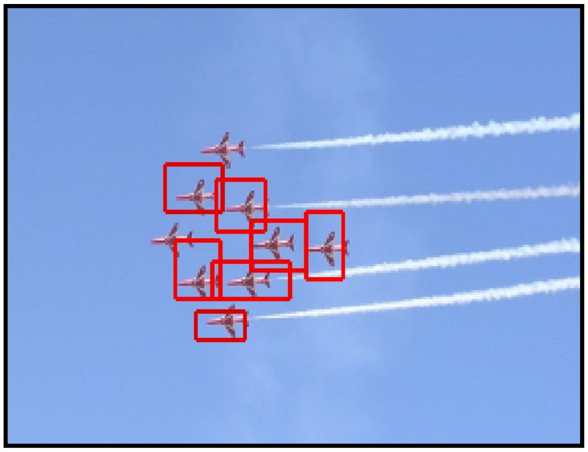
(b)

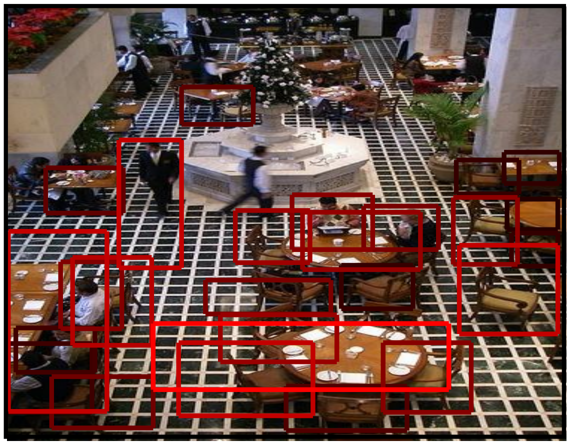
Fig.31: Image results using (a) Multi-scale saliency (***MS***) measure, (b) ***MS*** + ***QS*** measure

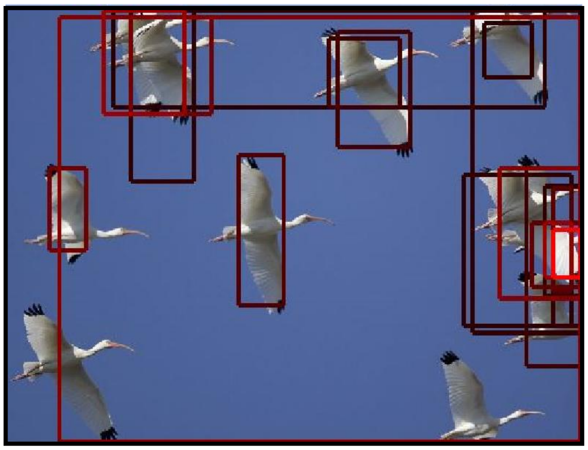
- Image results for ***Dictionary of Objects*** measure:

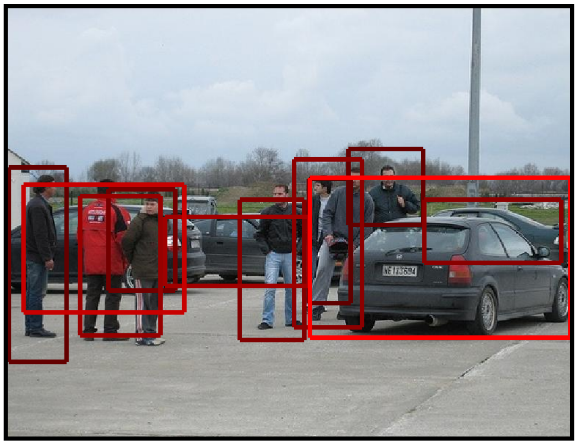


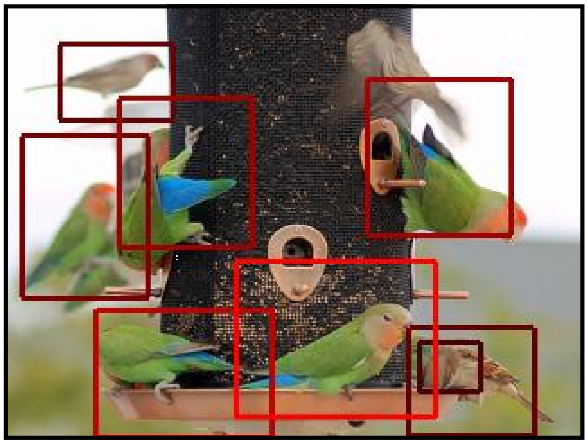

AIR
jmc
www.aircraft-images.co.uk











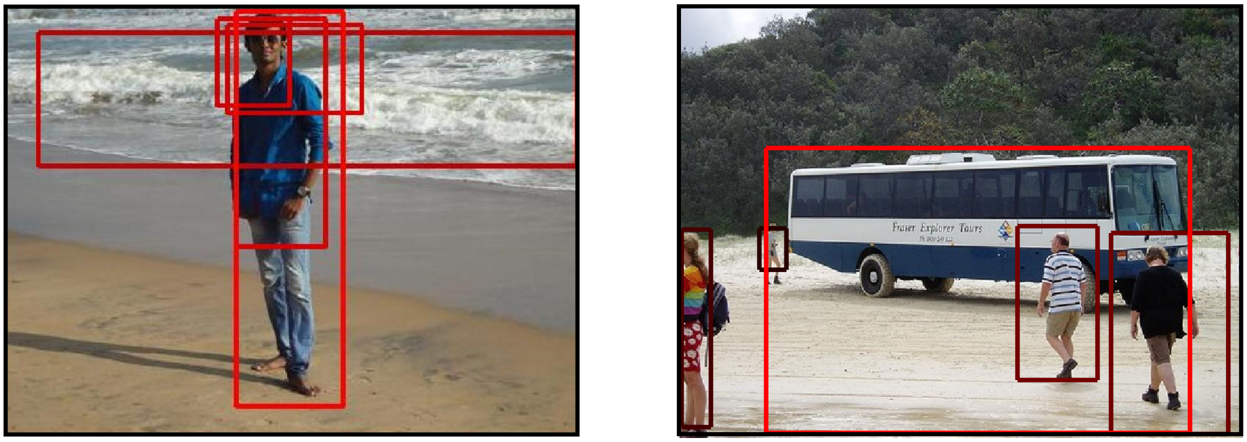


Fig.32: Results of some images using ***Dictionary of Objects*** measure

4.4 Conclusion and Future Work

We presented two objectness measures to distinguish object windows from background windows. Measure 1, i.e., ***Quadrant Symmetry*** measure, uses quadrant histograms of gradient angles in a proposal window to quantize the symmetry of an object. We also presented an efficient implementation of this measure to evaluate the windows in a much faster way. Detection rate for this measure on *PASCAL VOC 2007* dataset is 83% for 5000 proposal windows.

Measure 2, i.e., ***Dictionary of Objects*** measure uses gradient/Haar grid as a feature to construct the dictionary and this dictionary is trained for the object windows. Using this trained dictionary of objects, we compute the objectness score by measuring the similarity of the feature of a test window with this trained dictionary. We tested this measure for both gradient and Haar grids (as features) separately and the best detection rate is 93.88% for 5000 proposal windows, which is for gradient grid with algorithm as spherical K-means clustering for training the dictionary.

As our future work, we plan to improve our symmetry measure by employing it along with closed contourness measure **[7]** for the objects, because, as mentioned before, objects also possess a closed boundary around them and that can be merged with our symmetry measure to get much better object proposals.

As for our proposed measure 2, it has shown a very good efficacy compared to the symmetry measure. We will introduce different scales for the Haar grid feature so that it can cover the objects in a more tight fashion. We also plan to introduce more perceptual features which represent objects in a more proper way.

Finally, we also explored the field of manifold learning and we feel that objects and non-objects may lie in some higher dimensional space, separated in a non-linear fashion. So, manifold learning techniques like LLE, ISOMAPs, Laplacian Eigenmaps, which are non-linear dimensionality reduction techniques, can be employed to reduce the dimensions of this higher-dimensional space, so that classifiers like SVM or AdaBoost can be used to classify them properly in lower dimensions.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In CVPR, 2010.

[2] B. Alexe, T. Deselaers, and V. Ferrari. 2010. Measuring the objectness of image windows, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'12).

[3] M. M. Cheng, Z. Zhang, Wen-Yan Lin, and P. Torr. BING: Binarized Normed Gradients for Objectness Estimation at 300 fps, IEEE CVPR 2014.

[4] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In CVPR, 2007.

[5] J. F. Canny. A computational approach to edge detection. IEEE Trans. on PAMI, 8(6):679–698, 1986.

[6] R. C. Gonzalez, R. E. Woods. Digital Image Processing. Prentice Hall Publication.

[7] C. Lu, S. Liu, J. Jia, C. K. Tang, Contour Box: Rejecting Object Proposals Without Explicit Closed Contours. In ICCV, 2015

[8] M. Aharon, A. Bruckstein, K-SVD: An Algorithm for Designing Over complete Dictionaries for Sparse Representation - IEEE Transactions on Signal Processing, Vol. 54, No. 11, November 2006.

[9] Priyam Chatterjee, Denoising using the K-SVD Method - submitted in partial requirement for EE 264 - Image Processing and Reconstruction Instructor - Prof. Peyman Milanfar, Spring 2007.

[10] P. Doll´ar and C. L. Zitnick**.** Structured forests for fast edge detection. In ICCV, 2013.

[11] L. Zitnick and P. Dollar. Edge boxes: Locating object proposals from edges. In ECCV, 2014

[12] S. He, R. W. H. Lau, Oriented Object Proposals, ICCV, 2015.

[13] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 Results, 2007.

[14] N. Dalal and B. Triggs. Histogram of Oriented Gradients for Human Detection. In CVPR, volume 2, pages 886–893, 2005.

[15] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. IEEE Trans. on PAMI, 32(9):1627–1645, 2010.

[16] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In ICCV, 2009.

[17] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, A.W.M. Smeulders. Selective search for object recognition. IJCV 2013.

[18] P.A. Viola, M.J. Jones. Robust real-time face detection. IJCV 57(2) (2004) 137–154.

[19] S. He, Rynson W.H. Lau. Oriented Object Proposals. The IEEE International Conference on Computer Vision (ICCV), 2015, pp. 280-288.

[20] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, November 2004, Volume 60, Issue 2, pp 91-110.