Investigations of an “Objectness” Measure for Object Localization

by

Lewis Richard James Coates

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science
in
Computer Science

Thesis Committee:
Melanie Mitchell, Chair
Feng Liu
Bart Massey

Portland State University
2016

Abstract

Object localization is the task of locating objects in an image, typically by finding bounding boxes that isolate those objects. Identifying objects in images that have not had regions of interest labeled by humans often requires object localization to be performed first. The *sliding window method* is a common naïve approach, wherein the image is covered with bounding boxes of different sizes that form *windows* in the image. An object classifier is then run on each of these windows to determine if each given window contains a given object. However, because object classification algorithms tend to be computationally expensive, it is helpful to have an effective filter to reduce the number of times those classifiers have to be run.

In this thesis I evaluate one promising approach to object localization: the *objectness* algorithm proposed in [1, 2]. Specifically, I verify the results given in [1, 2] and further explore the weaknesses and strengths of their "objectness" approach. I then test the generality of this approach by applying the objectness algorithm to a new set of images. My results demonstrate the effectiveness and generality of this approach.

Acknowledgements

A special thanks to Sage Imel and his team for their tireless efforts.

Contents

List of Figures

1 The Task of Object Localization

Object localization is the task of finding objects in images, as opposed to the task of object classification, which is identifying what object is contained in an image, without localizing the object. The overall goal of object localization is to isolate the minimal bounding box of that object of interest, a so-called "window." The definition for "of interest," may vary by domain. The algorithms studied in this thesis attempt to be agnostic as to the exact nature of what is being searched for.

This is related to the machine learning task of classification. In classification a thing (e.g. an image, etc.) is presented to a classifier algorithm. In the case of binary classification, the algorithm then declares either "true," or "false." This is then used to determine some semantic fact about the thing, traditionally whether or not this thing has particular properties (e.g. an image has a person in it, or represents something man made).

A common mechanism in the computer vision literature is the *sliding window search*, which first breaks up an image into subsections, called windows, and then applies an object classifier to determine if the object of interest is in that window. Searching every window of every shape and size on an image can be expensive from a computation time perspective, so object localization acts as a filter, only classifying windows that are likely to contain an object.

The task of object localization can be difficult due to the variety of objects that can be searched for, the fact that orientation and scale of objects in an image are not known beforehand, and the variety and size of images that need to be searched. For example, consider the task of looking for a beer mug in an image. There could be zero, one, or many mugs; the background could be dark or light; the photo could have been taken inside or outside, etc. The appearance of the beer mug in the context of

the overall image will be quite different in these different cases, and finding all beer mugs, while avoiding false positives, is a requisite for correctness.

Computational limitations have pressured development of several techniques to make object localization more efficient. These include, notably, "objectness," salience detection, and context-driven detection. Each of these techniques tries to limit the number of searched windows in order to be tractable. *Salience* focuses on identifying *points of interest* and selects bounding boxes windows around these points. *Context-driven* detection uses prior knowledge of the location of objects to guess where other objects will be, for example, on a road, cars will likely be next to each other. *Objectness* attempts to use several cues and combines them into a probability that a window contains an object, without any knowledge of what the category of that object is [1]. This thesis focuses on the objectness algorithm proposed by Alexe et al.[1], hereafter referred to as the "objectness algorithm." This objectness algorithm is a fast way of evaluating a particular window for whether or not the window contains an object. This algorithm is run on a sliding window search in place of a classifier, which is feasible due to it's performance characteristics.

1.1 Setting

The objectness algorithm computes low level features, data points representing the image, on an image and uses these features to generate a list of proposed windows that are likely to contain objects. These windows will be compared with windows that are known to contain objects. The difference or similarity in these windows will be calculated and fed back into the training algorithm, and features that correspond to the presence of an object will be weighted strongly, while features that don't will be weighted weakly.

1.2 Terminology

For the purpose of this thesis, I use the following terminology:

Ground Truth: Known true values or otherwise "correct" answers. Here, windows are true objects, usually annotated by humans.

Intersection over Union (IOU): The area two rectangles (for our purposes) intersect divided by the total area covered by the union of two rectangles. Used to determine how well a proposed bounding box overlaps with a ground-truth one.

Detection Rate: The fraction of ground-truth objects in the test set found by the method in question. Also known as recall.

Window: A rectangle or bounding box positioned on an image. A window proposed by the objectness algorithm can be compared with a ground truth object window using the intersection-over-union measure. For the purposes of my project, an intersection over union of greater than 50% or 0.5 is considered a "true positive," and less is not. The 0.5 value is commonly used.

True Positive: A window that is proposed and has an IOU of at least 0.5 with a ground truth window.

False Positive: A window that is proposed that does not have an IOU of at least 0.5 with a ground truth window.

Signal-to-Noise Ratio: The number of true positives divided by the total number of proposed windows. This is analogous to precision.

2 Specific Goals and Results of this Thesis

In this project, I verify the results of a previous published paper on object localization, and I apply that project's code to a new, more difficult object-localization task.

In particular, in this thesis, I study an algorithm and code base developed by Alexe et al.[1, 2] that finds windows which are likely to have objects (as opposed to simply containing background), and apply their work to a new dataset.

Alexe et al. designed their algorithm to be trained on a set of objects and then find any objects in an image, even if the training set doesn't contain the same kind of objects. That is, the algorithm is meant to find locations in the image that exhibit the features of objectness in general. For the purposes of this algorithm, objects are items such as people, animals, cars, etc. One of my goals was to verify that the code does work as intended, and doesn't require domain-specific images to act as a training set.

My work and results consist of the following:

- First I verified that the existing code base produces the published results.
- I added an additional metric to validate that the results were distinguishable from random sampling.
- I compared the effectiveness of random search on the Pascal VOC dataset to the method set forth by Alexe et al.
- I ran the experiment multiple times to corroborate the results statistically.
- I applied the objectness code to a new dataset to test the generality of the code.

In summary my results show:

- The objectness algorithm is at least as effective as claimed in the published results.
- The effectiveness of random search on the Pascal VOC dataset is quite high.
- The objectness algorithm appears to generalize well.

The methods and results are discussed in detail in the remainder of this thesis.

3 Previous Work on Objectness

"Windowed search" is a common technique in the literature on object localization. The basic idea is to generate bounding boxes containing sub-images of the main image, and evaluate each of these bounding boxes. These bounding boxes are called "windows," and there are a variety of techniques for finding them.

The naïve technique would apply an object classifier to every window, in every aspect ratio, in every size. For most images this would become computationally expensive very quickly. Typically in practice, a few aspect ratios and proportional sizes of windows are selected, and these windows are "slid along" the image.

A computationally cheap measure is applied to this set to select a promising subset of windows, which a more expensive classifier then examines. Alexe et al. [1, 2] used this technique in their "objectness" measure.

3.1 Overview of Algorithm

A high-level overview of how the "objectness" algorithm works is as follows. The "objectness" algorithm is composed of several so-called "cues." These cues are in and of themselves algorithms that, when given a window of an image, return a value or score representing the likelihood that an object is in that window. First, a window selection algorithm is run, which selects a large number (100,000) of windows, sampled using a cheap-to-compute cue (number chosen arbitrarily). Then four separate cues are run on each of these windows. These cues are then integrated using a naïve Bayesian approach, combining all cues to produce a composite scorer. This scorer is run on each of the 100,000 windows selected by the window selection algorithm, and these windows are then ranked by their respective scores. The top k windows are then returned, where k is specified by the user.

Each of these cues and the naïve Bayesian classifier have learned parameters. These parameters are learned before the algorithm is applied to score windows, by being trained using a set of images with ground-truth object windows. An additional cue, known as "location and size", was initially evaluated but ultimately dropped from the objectness model.

3.2 Definition of an Object

For the purposes of the development of the objectness algorithm, Alexe et al.[1] contend that any object has at least one of three characteristics. These characteristics are used as design parameters for developing the objectness algorithm[2]. These characteristics are:

- A well-defined *closed boundary* in space.
- A *different appearance* from its surroundings.
- Being *unique* in an image, and standing out therefore as "salient."

These guiding definitions are how the objectness algorithm is meant to function. The objectness algorithm has several components that each contribute towards the overall goal of object localization. In general these components correspond to one or more of the above items.

3.3 Object Localization through Multi-Cue Identification

Alexe et al.'s algorithm has two parts. The first part identifies windows to evaluate and the second part evaluates these windows. I will first focus on the second part of this algorithm, which calculates the likelihood that an object is present in a given window. The evaluation applies several sub-algorithms that generate features, or

measurable properties, called "cues". These cues map a given window to a numerical value, which allows the windows to be sorted in terms of how strongly the objectness algorithm believes there is an object in the window. These cues are called "saliency cues", and each generate a resulting set of numerical values (features), mapped across an image.

The objectness algorithm calculates four saliency cues:

- multi-scale saliency
- superpixel straddling
- color contrast
- edge density

Each of these cues have parameters that have to be trained before the cues can be used effectively. These parameters are learned using a set of training images with known ground truth values. The parameter-learning method is described in [2]. Given a window in a test image the objectness algorithm computes these cues in the window.

A naïve Bayesian scorer is then used to aggregate each of these cues into a composite score for the window. This scorer is itself a supervised machine learning algorithm that is trained to aggregate the results of these four cues into a single composite score for a given window. This is referred to as an "objectness composite scoring model" throughout this thesis. It is worth noting that this objectness composite scoring model itself does not generate windows. It merely gives a numerical value to windows representing the likelihood of an object being in that window.

3.4 Multi-Scale Saliency

Multi-scale saliency is a feature proposed in [8]. This feature is based on Fourier transform residuals, and is designed to locate regions that are "unique" in the frequency domain. The output forms a "saliency map", a mapping of pixels to a numerical value corresponding to that pixel's position in the frequency domain. This map is then used to score windows of the image. This salience map I is determined at each pixel p according to the following formula

$$I(p) = g(p) * \mathcal{F}^{-1}[\exp(\mathcal{R}(f) + \mathcal{P}(f))]^2 \tag{3.1}$$

where g is a Gaussian smoothing filter (Gaussian blur), f is the input image, $\mathcal{F}$ is the Fourier transform operation, and $\mathcal{R}$ and $\mathcal{P}$ are the spectral residual and phase spectrum. This forms the simple saliency map. However, this map promotes certain sized objects with respect to the FFT, particularly very small windows with only a few pixels that are very salient. In order to alleviate these effects, this saliency technique is extended to multiple scales. Additionally, the color channels are processed independently and then summed, as per [8]. This enhanced technique builds a saliency map $I^s_{MS}(p)$ for each scale s. The definitions of the saliency of a window w at scale s is

$$\mathrm{MS}(w, \theta^s_{\mathrm{MS}}) = \sum_{\{p \in w | I^s_{\mathrm{MS}}(p) \geq \theta^s_{\mathrm{MS}}\}} I^s_{\mathrm{MS}}(p) \times \frac{|\{p \in w | I^s_{\mathrm{MS}}(p) \geq \theta^s_{\mathrm{MS}}\}|}{|w|} \tag{3.2}$$

where θ^s_{MS} are the scale specific thresholds at s, a parameter to be learned, and the number of pixels is represented by the operator $|\cdot|$.

The scoring function is designed to promote larger windows rather than just windows with a high density of salient pixels, as using raw density of salient pixels would

favor very small windows with a large number of highly salient pixels. A threshold value is learned which defines the point at which a pixel becomes "salient." The multi scale saliency cue measures the *uniqueness* trait of objects.

To learn the parameters θ^s_{MS} of multi-scale saliency, each threshold is learned separately for each scale s, by optimizing the accuracy against a training set of windows $\mathcal{O}$. To this end, the saliency map I^s_{MS} is computed and the MS score of all windows is evaluated. Non-maximum suppression is then run on this score space using an efficient technique presented in [10]. The results of this non-maximum suppression is $\mathcal{W}^s_{\max}$, a set of local maxima windows. The optimal θ^{s*}_{MS} is calculated by maximizing

$$\theta^{s*}_{\text{MS}} = \arg\max_{\theta^s_{\text{MS}}} \sum_{o \in \mathcal{O}} \max_{w \in \mathcal{W}^s_{\max}} \frac{|w \cap o|}{|w \cup o|} \tag{3.3}$$

which is tantamount to seeking the threshold θ^{s*}_{MS} such that the local maxima of multi-scale saliency across the images covers the object $\mathcal{O}$ most accurately. Observe that maximizing this formula not only maximizes the windows covering annotated objects, or ground truths, but also minimizes the score of windows not covering any annotated objects.

3.5 Color Contrast

The color contrast cue measures the dissimilarity of a window and a window that is θ units larger. The function $\text{Surr}(w, \theta_{CC})$ takes a window w and returns a rectangular ring by enlarging the window w in all directions by a factor of θ_{CC} as given by

$$\frac{|\text{Surr}(w, \theta_{CC})|}{|w|} = \theta^2_{CC} - 1 \tag{3.4}$$

Then the color contrast between a window and its surrounding area is computed by calculating the distance between the window and its surrounding area, represented

as histograms in the LAB color space, denoted h, using the Chi-square distance as given by

$$\mathrm{CC}(w, \theta_{CC}) = \chi^2(h(w), h(\mathrm{Surr}(w, \theta_{CC}))) \tag{3.5}$$

Note the value θ_{CC} is a learned coefficient.

This cue is designed to detect the *different appearance* of an object from its background, in terms of the definition of an object used by the objectness algorithm. The surround aspect of this algorithm, that is, the comparison of a window to a window θ_{CC} units larger, also helps find the minimal window surrounding an object, something that is desirable.

A related concept to color contrast is the *center-surround histogram* given by [9], however color contrast is focused on scoring an entire window, while the center-surround histogram operates on a single pixel.

Note it is probable that color contrast and multi-scale saliency likely are not statistically independant. For the purposes of cue integration, this assumption is made.

3.6 Edge Density

Similar to the color contrast cue, the edge density cue difference between a window and a slightly smaller contained window, noted as $\mathrm{Inn}(w, \theta_{ED})$, where w is a window and θ_{ED} is the factor by which w is shrunk. This shrinking occurs such that

$$\frac{|\mathrm{Inn}(w, \theta_{ED})|}{|w|} = \frac{1}{\theta_{ED}^2} \tag{3.6}$$

The Canny edge detector [3] builds an edgemap $I_{ED}(p) \in \{0, 1\}$. This is used to calculate the density of edges, where a pixel p is defined as being a part of an edge if it is mapped to a 1 in the edgemap $I_{ED}(p)$, and $Len(\cdot)$ is defined as the length of the

perimeter of the inner ring. The edge density is defined as

$$\mathrm{ED}(w, \theta_{ED}) = \frac{\sum_{p \in \mathrm{Inn}(w, \theta_{ed}} I_{ED}(p)}{\mathrm{Len}(\mathrm{Inn}(w, \theta_{ED}))} \tag{3.7}$$

Edges have been normalized to be a single pixel in width, in order to prevent bias towards very small windows. The Canny edge detector is chosen for having very good runtime performance[12].

Most objects have edges in their interior and on their boundary, but not outside. To this end, the edge detection cue captures the closed boundary aspect of a region being an object.

3.7 Superpixel Straddling

An additional technique for detecting the closed boundary aspect of an object is known as superpixels[6]. This technique breaks an image into small regions or segments, where each segment has approximately uniform texture or color. The overarching goal of this technique is to preserve object boundaries, the idea being that all pixels in a superpixel belong to the same object [11]. In practice an object is usually broken up into several superpixels, but typically no superpixel straddles a boundary between two objects.

A superpixel s is said to straddle a window w if s contains pixels both inside and outside w. Ideally a window containing an object will contain superpixels that are entirely inside this window, i.e., not straddling this window. The superpixel straddling cue is described by the formula

$$\mathrm{SS}(w, \theta_{SS}) = 1 - \sum_{s \in \mathcal{S}(\theta_{SS})} \frac{\min(|sw|, |s \cap w|)}{|w|} \tag{3.8}$$

where $\mathcal{S}(\theta_{SS})$ is the set of superpixels obtained using [6], with a segmentation scale θ_{SS}. Note that θ_{SS} is a learned parameter. The area $|s \cap w|$ is calculated for each superpixel inside of w, and the area $|s\ w|$ is calculated for each superpixel outside of w. The lesser of the two values is summed. This causes $\mathrm{SS}(w, \theta_{SS})$ to be highest for tightly fitting windows, that is, minimal windows surrounding an object.

3.8 Learning Parameters of Color Contrast, Edge Density and Superpixel Straddling

The parameters θ_{CC}, θ_{SS}, θ_{ED} are all learned using the same mechanism. As a result of this, consider $\theta = \theta_{ED}$, without loss of generality. For the purposes of training, a window w is considered *positive* if it has an intersection over union of at least 0.5 with a ground truth window, and *negative* otherwise.

For every image i in the training set, generate 100,000 windows uniformly distributed across the image i. This training set is known as T. Note that the likelihoods for positive $p_\theta(\mathrm{ED}(w,\theta)|positive)$ and for negative $p_\theta(\mathrm{ED}(w,\theta)|negative)$ can be built for any value of θ. To find the optimal value of θ, noted θ^*, the following equation must be solved

$$\theta^* = \arg\max_{\theta} \prod_{w \in PositiveWindows} \frac{p_\theta(\mathrm{ED}(w,\theta)|positive)\dot{p}(positive)}{\sum_{c \in \{positive,negative\}} p_\theta(\mathrm{ED}(w,\theta)|c)\dot{p}(c)} \tag{3.9}$$

where the relative frequency is used to set priors. That is, given $T_{positive}$ are a set of selected windows with an intersection over union of at least 0.5 with a ground truth window, $p(positive) = T_{positive}/T$ and $p(negative) = 1 - p(positive)$.

3.9 Cue Integration

As these cues are not entirely related, creating a composite model combining several of them improves the performance of the objectness algorithm. Each of the different cues can refine the search and hopefully stack on top of each other in a constructive way. To this end, multi-scale saliency is fairly effective at finding "blobs" of pixels that are likely objects, but tends not to be able to refine that down more narrowly. Edge detection fails under specific circumstances, notably certain forms of textured regions. Color contrast is good at refining a candidate area, and location and size provides an additional prior with nearly no additional computation.

It would be natural to model the cues jointly, but this would require an excessive number of samples to accurately estimate the joint likelihood. Because of this a naïve Bayesian model was chosen. This Bayesian classifier was trained to distinguish between $n-$tuples (one element per cue) with a positive or negative annotation. For each training image 100,000 windows are sampled from the distribution given by the multi-scale saliency cue, which biases the distribution towards better location, and then all other cues are computed for these samples. If a sample window has an intersection over union of 0.5 with a ground truth window on a training image it is assigned a positive annotation, otherwise the annotation is negative.

The model has each cue trained independently. That is, the probabilities $p(\text{positive})$ and $p(\text{negative})$ are the priors set by frequency, and the individual cue likelihoods are given by the conditional probability $p(cue|c)$ for each cue $cue \in Cue$ and each c, $c \in \{positive, negative\}$. Once this process is done an objectness composite scoring model is formed, which is a model that, given a window w, returns a

posterior probability according to

$$p(\text{positive}|\mathcal{C}) = \frac{p(\text{positive}) \prod_{\text{cue}} p(\text{cue}|\text{obj})}{\sum_{a \in \{\text{positive,negative}\}} p(\text{a}) \prod_{\text{cue}} p(\text{cue}|\text{a})} \tag{3.10}$$

This equation gives a value representing the posterior probability score of a window w. Note that not all the cues have to be used in this objectness composite scoring model.

3.10 Window Selection

A component of the objectness algorithm is window selection. This is an algorithm that selects windows to be scored by the composite objectness scorer. There are two competing technologies, multinomial and NMS-enhanced window selection.

Multinomial window selection samples windows solely using the window scores. A multinomial distribution D is constructed from the given T window scores. From this distribution D, F windows are selected at random. Each window is given a normalized probability of the value assigned to a particular window by the objectness composite scoring model. This normalization process also helps improve the performance of the window selection.

The NMS-enhanced window selection is not only based on the score assigned to a window by the objectness composite scoring model, but also the intersection over union of a candidate window with all previously selected windows. To perform this sample selection of windows, first the highest scored window is selected. Then the next highest scored window is selected that does not have an intersection over union of at least 0.5 with a previously selected window. This pushes windows to cover more of a given image, and reduces the amount of times a particular object can be discovered. NMS-enhanced window selection is used for the remainder of this thesis.

4 Results of Alexe et al.

4.1 Pascal Visual Object Classes Dataset

The *Pascal VOC* dataset (Visual Object Classes) is a dataset that contains objects in an image which have had annotations created which note the position, or window, containing a variety of objects [5]. These annotations form the ground truth values for where an object is in a given image, and is used in the training process of Alexe et al.'s objectness algorithm, specifically the objectness composite scoring model. The objects in the images come from 20 different classes, and make no guarantees on orientation, count, overlap, etc. The dataset is split into a training set and a test set. The dataset also has additional markers for certain objects as being "difficult," which is usually indicative of an obscured, partial, or otherwise non-standard object. The images are very small, each approximately one thousand pixels. This has ramifications for the efficacy of random and brute force search, in that there are fewer possible total windows that can be found on the image than on a larger image. Random search seems to be more effective on these smaller images than on larger ones.

The disparity in the shape and position of the objects in this dataset make it ideal for testing object localization. In particular, the diverse nature of objects in this dataset make it very difficult for a localizer to be trained too narrowly on a particular category or shape of object. As one of the goals of the "objectness," algorithm is to not rely on any prior information about the objects being searched for, this property in a dataset is highly desirable.

4.2 Experiment Performed by Alexe et al.

Alexe et al. performed an experiment on the Pascal VOC dataset [1, 2]. I replicated this experiment and validated these results. I extended these results by comparing

the results of the experiment performed with random search and showed the degree to which the "objectness," algorithm performs better than random window selection.

The experiment performed by Alexe et al. first split the Pascal VOC dataset. The dataset was split into 6 object classes for training and 14 for testing. Images that have objects from both sets are not used, neither are objects marked as difficult or experimental. The reason for splitting up the training set and the test set is to validate whether or not the algorithm is generic across object classes, regardless of training objects. In addition, in a subsequent experiment, the algorithm is retrained on a dataset completely orthogonal to the Pascal VOC data set, having no images in common with Pascal VOC 07 at all, to further verify the claim that the objectness algorithm is generic across object classes. The results of the objectness algorithm on the images of the 14 test object classes are then analyzed to determine how effective the objectness algorithm is overall.

The definition of success comes from the definitions of detection rate and signal-to-noise ratios I described in earlier. In general a high detection rate and a low signal-to-noise ratio is the goal of an algorithm trying to perform object localization. The objectness algorithm has a high detection rate, but the signal-to-noise ratio is generally not very high. The objectness algorithm takes a parameter of a number between 1 and 1,000, and returns up to that number of windows, potentially fewer. The graphs of signal-to-noise ratio against detection rate, and detection rate against number of windows requested are presented.

Overall the objectness algorithm found objects effectively, given 1,000 windows per image it found over 90% of the objects. The different cues were shown to have differing levels of effectiveness in localizing objects, and different accuracy and detection rate curves, which could show applications in different use cases. For example, at lower numbers of windows the color contrast cue help substantially. Overall the most

effective cue, the one that had the highest detection rate at 1,000 windows selected, was multi-scale saliency. The greatest detection rate at 1,000 windows of any two cues combined was multi-scale saliency and super pixel straddling. Location and size was removed as a cue, as once at least 3 cues are applied location and size as a cue provides no advantage. This indicates whatever advantage location and size provides to localizing objects is subsumed by the other cues.

5 Validation of Previously Published Results

5.1 Experiment performed

I reproduced the experiment performed by Alexe et al. once I determined the exact details of the sets of images used. [2] only shows the results of a single run of the process, which being stochastic can change slightly. However when I ran the experiment 10 times, the variation was seen to be minor overall.

The results presented by Alexe et al. have different graphs across the two papers [1, 2], I present all the graphs using the same techniques as[2], which is specifically using the NMS window selection. These graphs are number of windows against detection rate and number of windows against signal-to-noise ratio. I demonstrate that the results in Alexe et al.'s paper are reproducible, and then extend their results to show the advantage provided by the composite objectness scoring model over simple random selection is significant.

5.2 Reproduction of Original Experiment

Figure 5.2 shows the original results that Alexe et al. produced from the experiment they performed, as described previously. This graph shows the effectiveness of different combinations of cues of the objectness algorithm at localizing objects in the Pascal VOC 07 dataset. Overall the combination of multi-scale saliency, color contrast and superpixel straddling combined together to form the most effective object localizer discussed in the paper [2].

Figure 2 is the plot of the results of my reproduction of the experiment performed by Alexe et al. The results are exactly the same assuming the exact same data is put in (this includes fixing random number generator values to known presets). The conclusion is that the experiment performed is reproducible, and by extension the

results presented in [1, 2] are correct.

Figure 3 is an extension I performed of the original results of the paper, and shows the signal-to-noise ratio, that is the number of true positives divided by the total number of proposed windows. The objectness algorithm with NMS-enhanced window selection has a worse signal-to-noise ratio than is presented in [1]. This is expected since the NMS-enhancement to window selection does not allow objectness to repeatedly find the same objects. This implies more windows returned are not going to contain objects. It is worth noting there are never more than a few labeled objects in each image in the Pascal VOC 07 dataset.

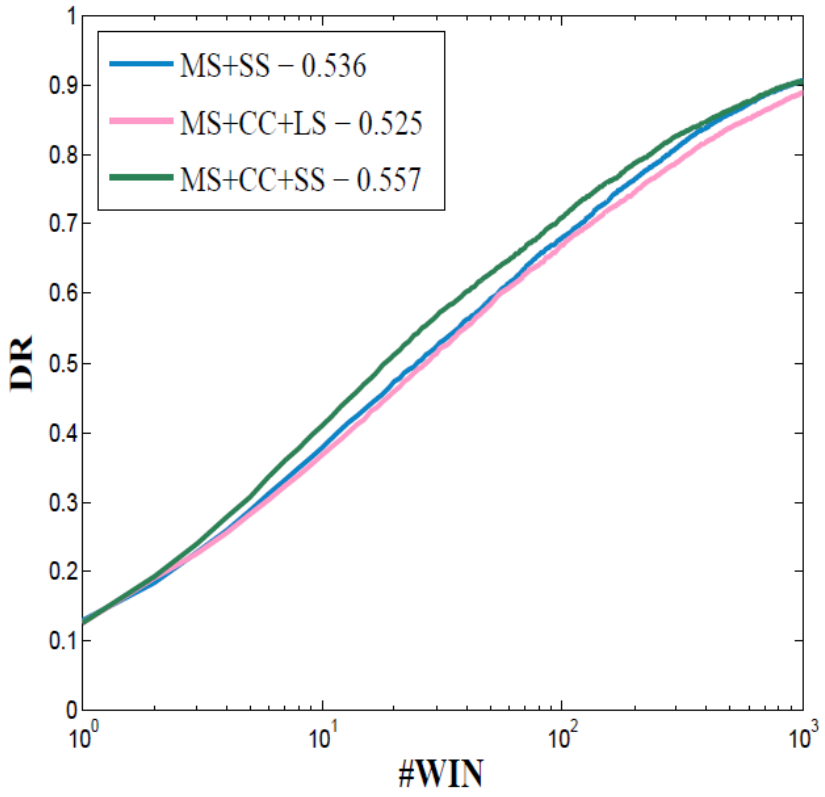

Figure 1: Reproduced from [2]. MS refers to multi-scale saliency, SS refers to superpixel straddling, CC refers to color contrast, LS refers to location and size (a cue not used in the final experiment). The number of windows requested plotted against the number of objects found across all images. Equivalently detection rate of the objectness algorithm on the Pascal VOC 07 dataset. The definition of detect is to have an intersection over union of at least 0.5, the detection rate is the percent of all objects that have at least one window detecting that object. The number of windows is the number of windows requested from the objectness algorithm. This graph represents the primary result I was attempting to replicate. Image best viewed in color.

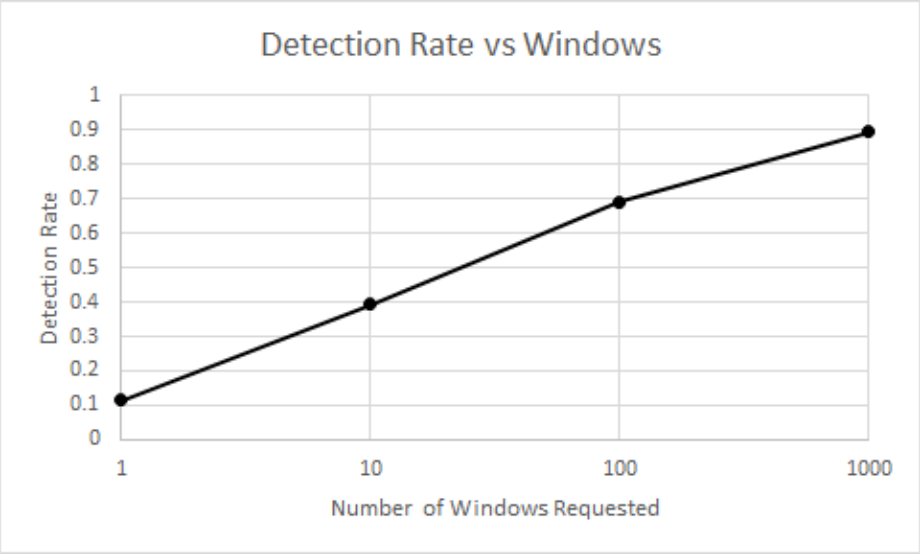

Figure 2: This is the results of my efforts at reproducing the experiment presented in [2]. The number of windows requested plotted against the number of objects found, or detection rate of the objectness algorithm on the Pascal VOC 07 dataset. The definition of detect is to have an intersection over union of at least 0.5, the detection rate is the percent of all objects that have at least one window detecting that object. The number of windows is the number of windows requested from the objectness algorithm, the objectness algorithm can choose to return fewer.

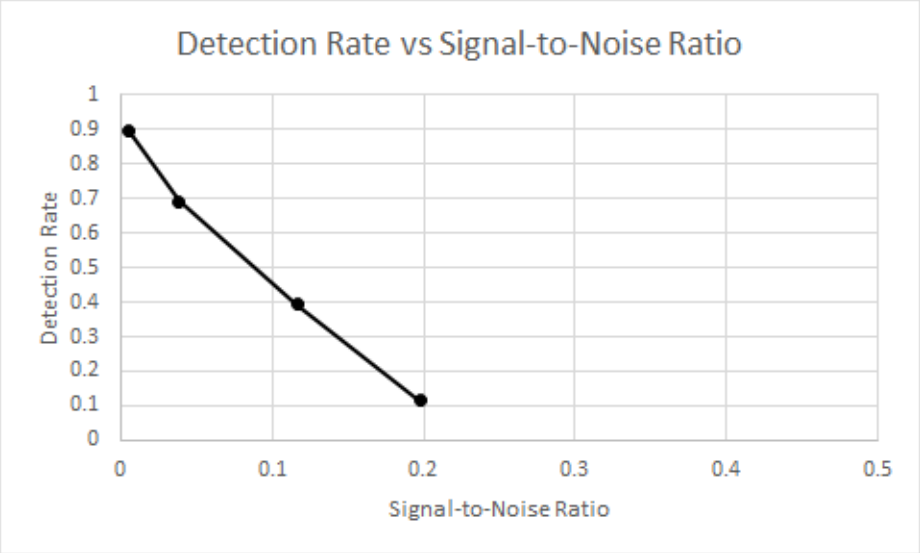

Figure 3: The detection rate of the objectness algorithm on the Pascal VOC 07 dataset, plotted against the signal-to-noise ratio.

6 Additional Experiments on Pascal VOC 07

I extended the results of Alexe et al.'s original experiment by comparing with a random window selection baseline. To perform this experiment I modified the code to use the window selection algorithm, but instead of scoring using the objectness composite scoring model, instead all windows were scored with a random value. This isolates the advantage provided by the objectness composite scoring model, and shows how much value the window selection algorithm provides. The driving goal of this experiment is to establish a baseline to see the enhancements of the objectness algorithm.

The results of this experiment were somewhat surprising. Overall the random window selection algorithm did very well at detecting objects in the Pascal VOC 07 dataset as can be seen in figure 5. The objectness composite scoring model does provide a noticeable improvement as well, however. On the side of signal-to-noise ratio the random window selector does not do nearly as well as the results provided by the objectness algorithm. It is also worth noting that the shape of this signal-to-noise ratio curve is different between the random selection and the objectness algorithm.

The images being very small caused the random search to be more effective. The small image size also caused a few other issues. Notably, the method has scaling problems when run on larger images. Scaling to larger images also showed objectness having a bigger advantage over random selection than was seen on smaller images. This suggests the small advantage provided by the objectness composite scoring model on the Pascal VOC 07 dataset to not be an indicator of weakness with the objectness composite scoring model. Finally window selection algorithm has the ability to fail to find windows, but this doesn't occur on larger images, which causes the results to be more consistent

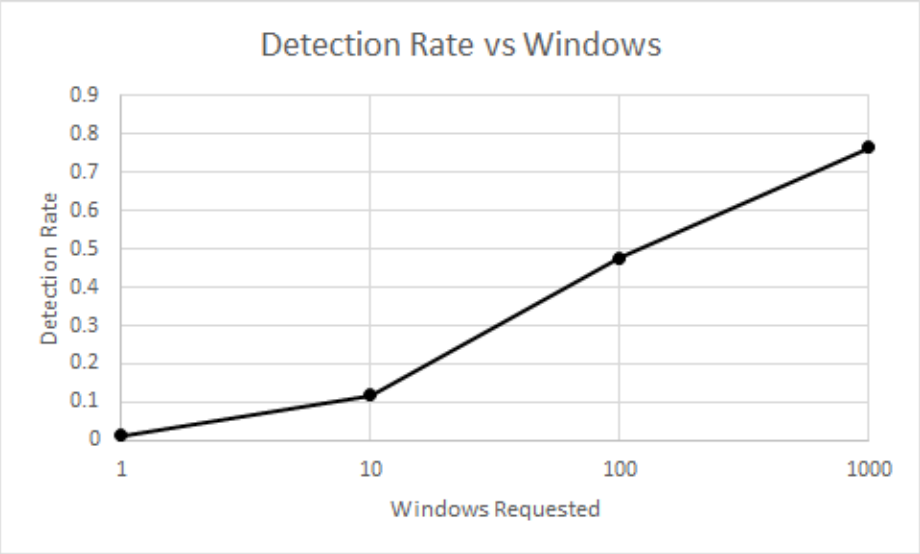

Figure 4: The detection rate of random window selection versus windows selected on the Pascal VOC dataset

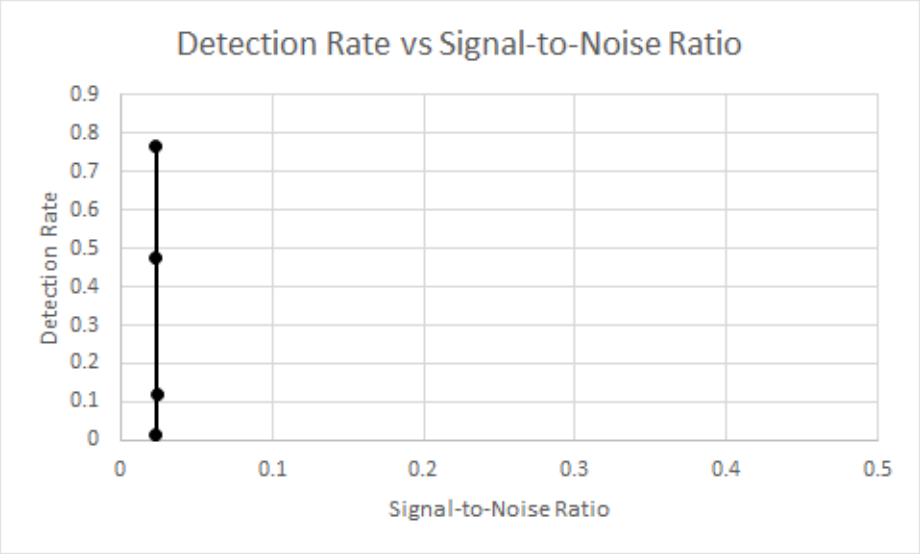

Figure 5: The detection rate versus signal-to-noise ratio of random window selection on the Pascal VOC 07 dataset. The vertical line is to be expected, since with random window selection, the probability of any window finding an object is the same.

6.1 Dataset Specific Training

Alexe et al.'s code is distributed with a set of learned values for the cues and composite model. These values are learned from a variety of images from a variety of sources. The thought is that since the objectness algorithm is theoretically agnostic to the class of object being searched for, an existing set of training values should be sufficient for using the objectness algorithm on any dataset. In practice these previously learned values are shown in the previous section to be quite successful at localizing objects.

The evaluator was retrained against the original test set, as presented in [1, 2], and a small improvement was shown in recall and signal-to-noise ratio. The improvement validated the claim of object agnosticism due to the improvement being, for all practical purposes, insignificant. It is worth noting that in some situations training the objectness algorithm against a specific dataset may improve results.

7 Experiment on Portland "Dog-walking" Dataset

7.1 Experiment Overview

I performed a similar experiment as was performed on the Pascal VOC 07 dataset, but with a different dataset, in order to validate the utility of the objectness algorithm in different domains. I used the code for the objectness algorithm and ran this code on the Portland Dog Walking dataset[4]. I compared the windows returned by the objectness algorithm on this dataset to the baseline values determined by hand by humans, and determined the detection rate and the signal-to-noise ratio. I then ran just the window selection algorithm as a modified random window generator, and determined what the advantage provided by the objectness composite scoring model in finding objects in the dataset, the details of the dataset and experiments are given below.

7.2 Data and Sources

The Portland Dog Walking dataset is a set of images taken by Melanie Mitchell's group for the purpose of semantic analysis. The images feature at least one dog and at least one human designated a dog walker. The images also feature some number of other people or objects. These other objects are not considered when calculating detection rate or signal-to-noise ratio. This implies the signal-to-noise ratio is likely much lower than what we would see in the Pascal VOC dataset, as most that would be considered "objects" are marked as such.

Each image has a set of ground truth bounding boxes labeled by hand, and the images are taken by multiple people in multiple environments with multiple cameras in order to fight bias. The fact that objectness seems mostly effective without application

specific training also helps prevent training bias that might be evident in the data set. There are two modes of use of the objectness algorithm, the first is to use the provided learned values and evaluate how effectively these "defaults" operate on a dataset. The other method is to train against a particular dataset and use more customized learned values. Training against the Portland Dog Walking dataset proved to be impossible, so only the first mode of using provided learned values is evaluated. Training was executed for over a week on a subset of the Portland Dog Walking dataset and no indication of convergence was observed. Further investigation into the performance of the training code and possibly errors in scaling is warranted.

7.3 Testing Generality of Objectness Algorithm

The Portland Dog Walking dataset differs from the Pascal VOC 07 dataset in several ways. The Portland Dog Walking dataset has the overwhelming majority of its ground truth bounding boxes focused onto dogs and humans, while Pascal VOC has 20 separate object classes, some of which have subclasses. This could make locating objects easier, but will likely also increase the number of false positives as the object detector will find windows in the image that reasonably could be considered "objects," such as cars, benches, tables etc. Since only detection rate is being emphasized, this ends up not being a major issue, but does imply that this is not necessarily the most ideal use case for this detection mechanism.

Secondly, and somewhat incidentally, many of the Dog-walking images are much larger than the Pascal VOC 07 images. This is mostly a practical consideration, but it ends up that the detection rate doesn't get reduced as much as one might expect. This is a strong indicator that the algorithm is working as intended and there is no need to reduce the image size. The one concern is the performance ramifications of these much larger images is significant, running in extremely large memory spaces

that might be impractical in some cases.

An evaluation of the objectness code on the dataset provided by Melanie Mitchell's group shows promise in the detection rate. Overall detection of dogs and dog walkers is high, which indicates the potential to apply this algorithm as a first filter in a chain of algorithms to try and locate and identify the dogs and their walkers.

It is worth noting that only humans with dogs have annotations. Additional humans in images will not be recorded as a part of this experiment, either the failure to detect them counting negatively against the detection rate, or successfully detecting them increasing the signal-to-noise ratio.

7.4 Results on Portland Dog Walking Dataset

Figures 6 and 8 show that the objectness algorithm is very effective at finding both dogs and dog walkers (humans). It nearly perfectly localized the humans in the image and had a very high (nearly 70%) detection rate for dogs at the 1,000-windows level. This shows that the objectness algorithm is a strong candidate for localizing objects. It also validates and strengthens the claim that the objectness algorithm performs similarly regardless of the class of object. While the detection rate is different on different objects, the objectness algorithm is able to effectively localize the majority of the objects being searched for.

The signal-to-noise ratio on the Dog-walking dataset, as seen in figures 7 and 9 is similar to what it was on the Pascal VOC 07 dataset. This further validates the claim of the objectness algorithm not needing class specific training to be effective.

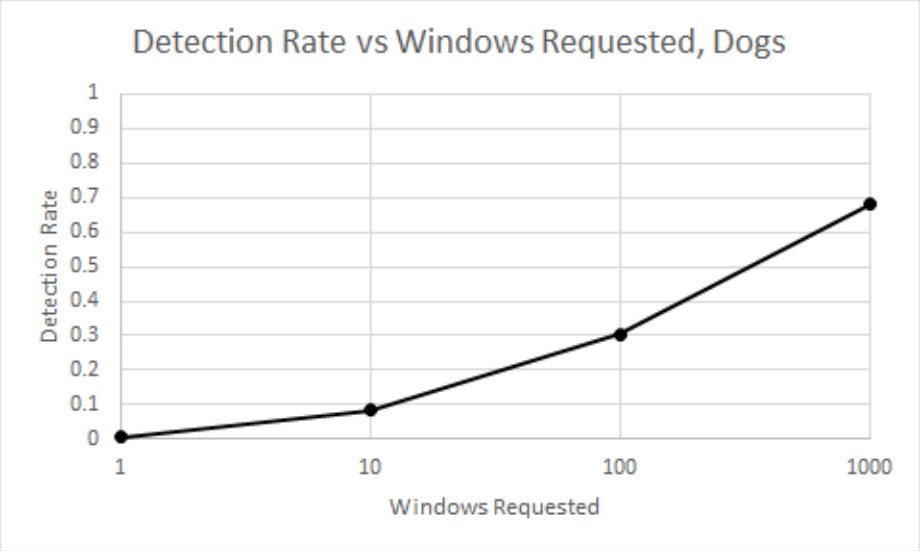

Figure 6: The number of windows requested plotted against detection rate of the objectness algorithm on the Portland Dog-walking dataset. This is considering the bounding boxes of dogs only.

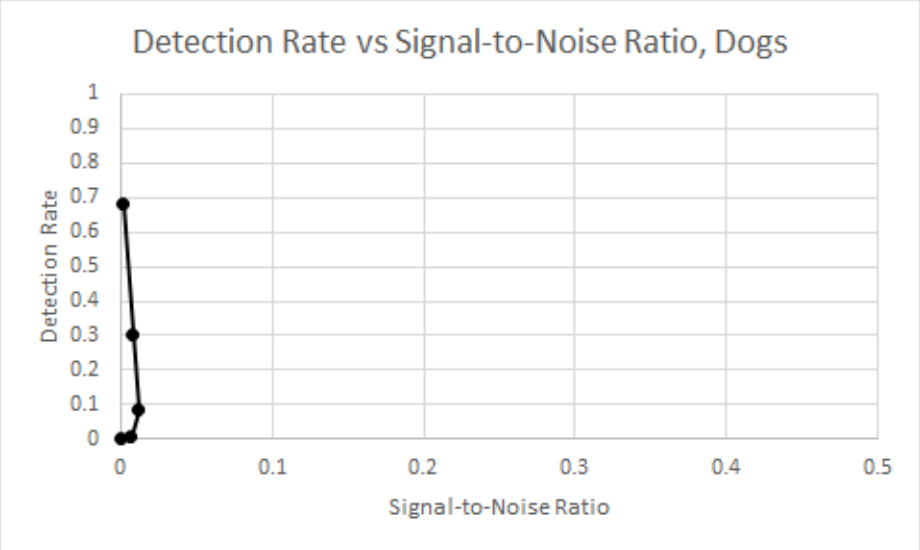

Figure 7: The detection rate of the objectness algorithm on the Portland Dog Walking dataset, plotted against the signal-to-noise ratio. This is considering all the bounding boxes on dogs only. This graph is parametrized by the number of windows requested.

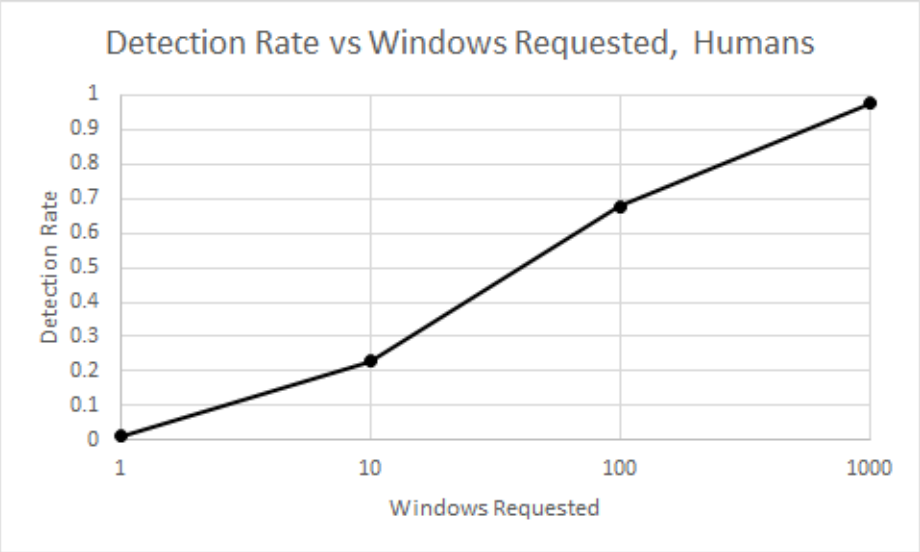

Figure 8: The number of windows requested plotted against detection rate of the objectness algorithm on the Portland Dog Walking dataset. This is considering only dog walkers (humans) bounding boxes.

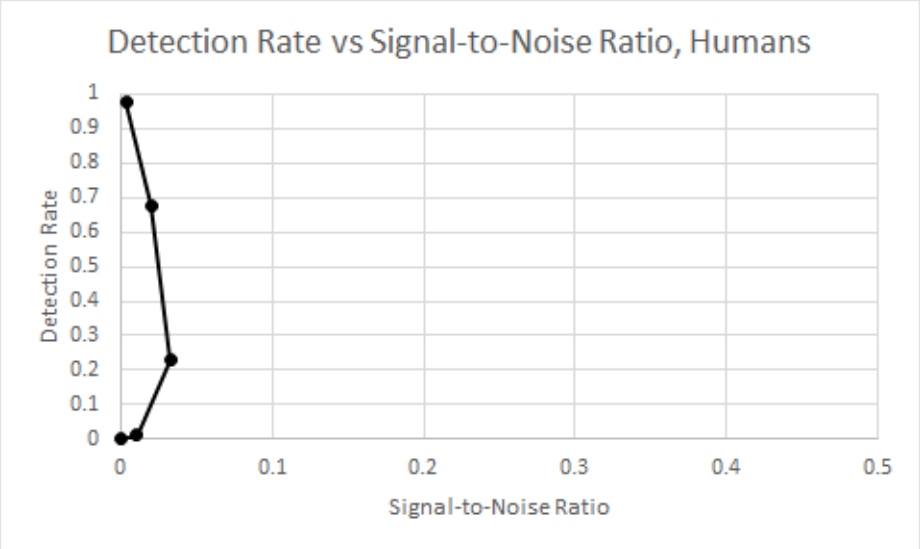

Figure 9: The detection rate of the objectness algorithm on the Portland Dog Walking dataset plotted against the signal-to-noise ratio. Only the bounding boxes around dog walkers is considered in this plot. This graph is parameterized by the number of windows requested.

7.5 Random Sampling on Portland Dog-walking dataset

The same experiment was applied to the dataset provided by Melanie Mitchell's group as was applied to the Pascal VOC 07 dataset of using modified random window selection. This experiment was performed to find a baseline to compare the improvement provided by the objectness algorithm over randomly selecting windows. As before, the window selection algorithm was kept the same, in order to isolate exactly how much the objectness composite scoring model improves detection rate and signal-to-noise ratio. That is to say, the objectness composite scoring model was replaced with a random scoring model, and then the window selection algorithm was run on the Portland Dog-walking dataset.

Figure 12 shows the effectiveness of the window selection algorithm in finding dog walkers (humans) on the dataset. This is used as a baseline to show the advantage provided by the more advanced composite scoring model provided by the objectness cues. It is worth noting that this window selection algorithm is actually very effective at finding humans in this dataset. This is likely due to the choice of aspect ratios and scales of windows being well chosen for the application of finding humans.

Figure 10 shows how well the window selection worked at detecting dogs in the dataset. This seemed to be a much more challenging task to the window selection algorithm. It is not clear why random detection worked so much worse on dogs as compared to humans. Additional analysis would have to be applied in order to isolate this behavior. I suspect the fact the dogs are smaller, or not of a specific aspect ratio, may have an impact on this detection rate.

Figures 11 7.5 shows how well window selection works at object detection compared to the signal-to-noise ratio. Note that the signal-to-noise ratio stays very consistent across the detection rate. This is because the probability of any given window

finding an object is uniform across all selected windows, so as more windows are selected, while the total detection rate of objects will go up the signal-to-noise ratio will stay relatively the same.

Overall, disabling the objectness composite scoring model significantly impairs object detection and the signal-to-noise ratio. This is further validation of the results in [1, 2]. In particular, the claim was that the objectness composite scoring model helps improve the detection of objects in images without prior knowledge either of the class of images being searched for, or of information about the particular images being searched.

Overall, the detection rate was quite high for random sampling. This is similar to the results obtained when random sampling was applied to Pascal VOC, although random selection did perform worse overall by a noticeable amount. This discrepancy is attributable to the difference in sizes of the images, and the shape of the ground truth objects. While the detection rate for random sampling was high, it was significantly lower than when the objectness scorer was used.

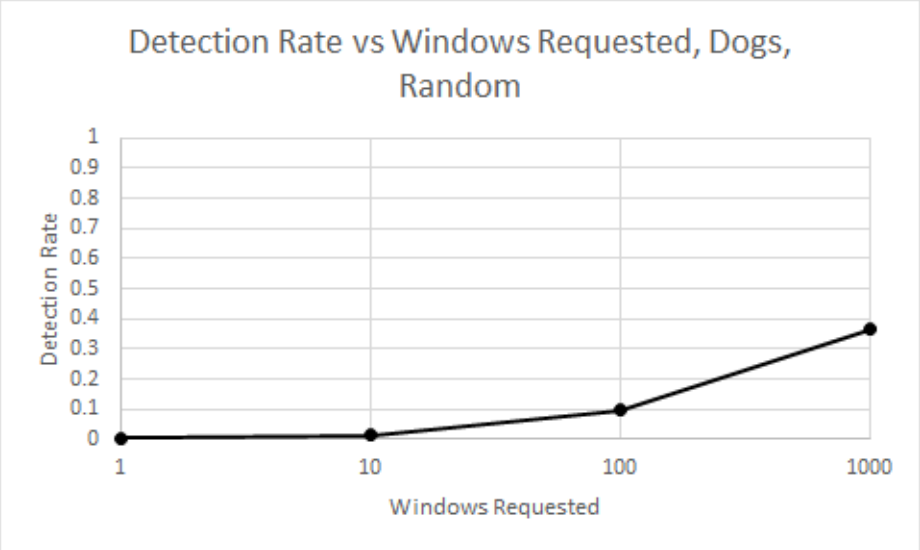

Figure 10: Detection rate of dogs in the Portland Dog Walking dataset using random window selection. Only bounding boxes of dogs are used as ground truth windows.

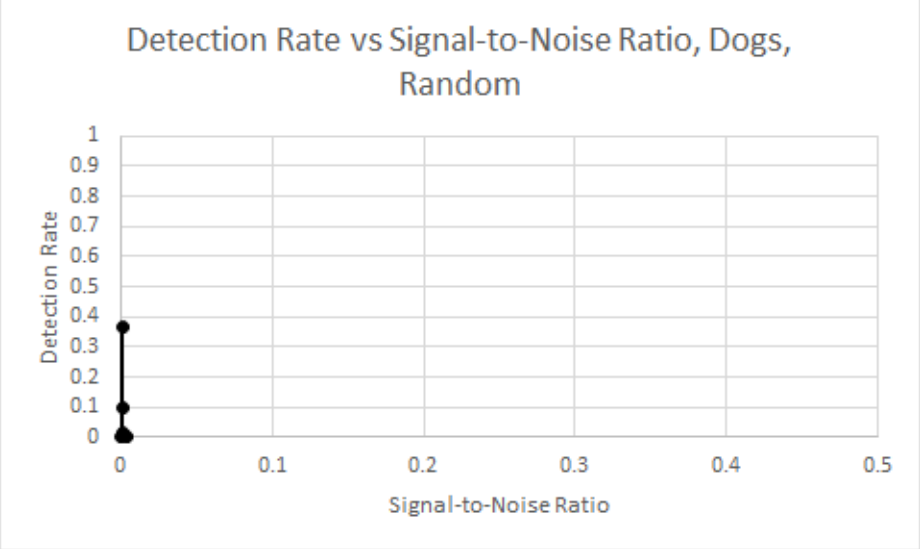

Figure 11: Detection rate of detecting dogs using random window on the Portland Dog-walking dataset plotted against signal-to-noise ratio. Only dog bounding boxes are used as ground truth windows.

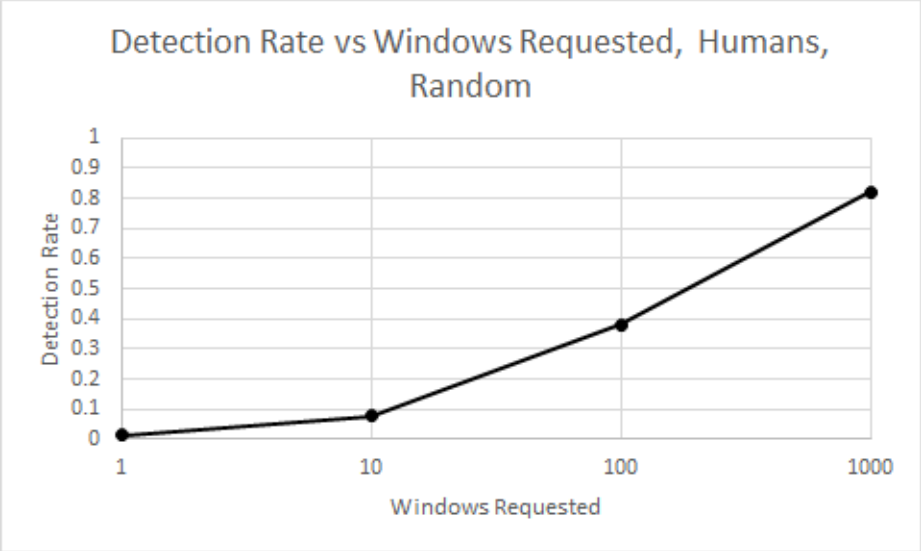

Figure 12: The detection rate vs windows of dog walkers (humans), on the Portland Dog-walking dataset. Only human bounding boxes are taken as ground truth windows.

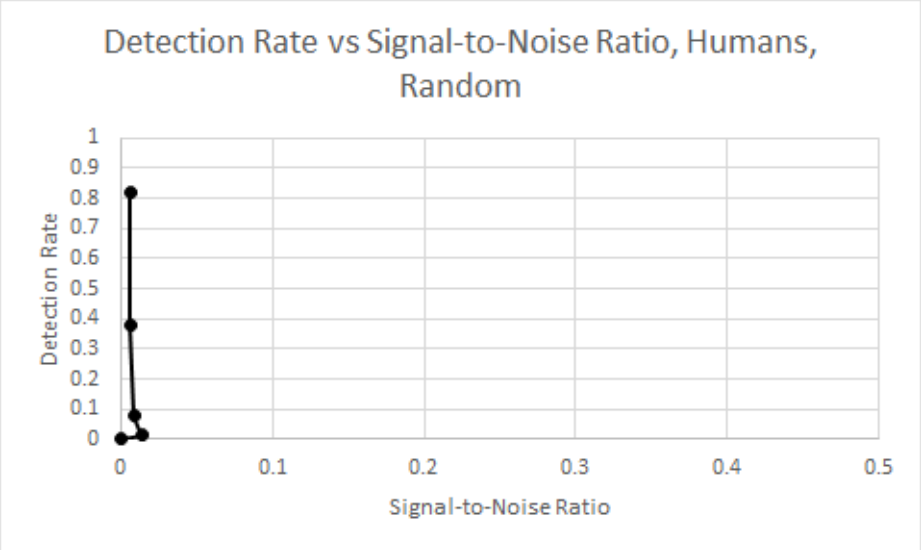

Figure 13: Detection rate vs signal-to-noise ratio of detecting dog walkers (humans) on the Portland Dog-walking dataset. Only dog-walker bounding boxes are used as ground truth windows.

8 Conclusions and Future Work

I reproduced the experiment run by Alexe et al. on the objectness algorithm. I extended their experiment to include a baseline that is generated using a modified random window selection algorithm. I then applied the objectness code and the random window selection code to a new dataset. The results of the objectness algorithm on the new dataset were exceptional; the objectness algorithm was nearly perfect in finding all dog-walkers (specifically humans) in the images in the dataset. Overall, the objectness algorithm was found to perform well at the task of object localization.

8.1 Improvements on Baselines

The effectiveness of random search, particularly on smaller images is worth focusing on as random search is a strong baseline against which other localization strategies need to be compared against. While having a high detection rate, random selection had a low signal-to-noise ratio, so random selection is not ideal. The definition of detection being an intersection over union of 0.5 is possibly not optimal as well. While this definition is used extensively throughout the relevant literature on the topic of object localization, I can find no actual justification for it.

It is likely that other metrics could be developed to define the definition of accuracy within the domain of object localization more correctly. A definition of accuracy as the average intersection over union is proposed in[2] where the objectness algorithm is seen to have a detection rate of over 70% by this definition. Additional definitions of correctness could be developed and applied to the objectness code. Further reading on this can be seen in[7]

8.2 Performance Improvements

The code base has several practical scaling flaws. On larger images there is a large memory requirement (tens of gigabytes), which is likely simply an implementation limitation. This, and other implementation issues, need to be addressed before the code is put into production or even applied to more extensive tests.

The code is written predominately in a combination of Matlab and C. There appears to be a resource issue with the Matlab-to-C interprocess communication, which causes extensive duplication of data structures when Matlab code calls C functions. This causes a very artificial scaling problem that can likely be fixed by either learning how to do more efficient resource management when transferring data from Matlab to C, or by rewriting the algorithm entirely in a single language.

Bibliography

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80. IEEE, 2010.

[2] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2189–2202, 2012.

[3] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, Nov 1986.

[4] Portland Dog Walking Dataset, courtesy of Melanie Mitchell.

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[6] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, September 2004.

[7] Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? 2015.

[8] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[9] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang,

and Heung-Yeung Shum. Learning to detect a salient object. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(2):353–367, 2011.

[10] A. Neubeck and L. Van Gool. Efficient non-maximum suppression. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 850–855, 2006.

[11] Bryan C Russell, William T Freeman, Alexei A Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1605–1614. IEEE, 2006.

[12] S Vijayarani and M Vinupriya. Performance analysis of canny and sobel edge detection algorithms in image mining. *Int. J. Innovative Res. Comp. Commun. Eng*, 1(8), 2013.