Leveraging Contextual Relationships Between Objects for Localization

by

Clinton Leif Olson

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
Tim Sheard

Portland State University
2014

Abstract

Object localization is currently an active area of research in computer vision. The object localization task is to identify all locations of an object class within an image by drawing a bounding box around objects that are instances of that class. Object locations are typically found by computing a classification score over a small window at multiple locations in the image, based on some chosen criteria, and choosing the highest scoring windows as the object bounding-boxes. Localization methods vary widely, but there is a growing trend towards methods that are able to make localization more accurate and efficient through the use of context.

In this thesis, I investigate whether contextual relationships between related objects can be leveraged to improve localization efficiency through a reduction in the number of windows considered for each localization task. I implement a context-driven localization model and evaluate it against two models that do not use context between objects for comparison. My model constrains the search spaces for the target object location and window size. I show that context-driven methods substantially reduce the mean number of windows necessary for localizing a target object versus the two models not using context. The results presented here suggest that contextual relationships between objects in an image can be leveraged to significantly improve localization efficiency by reducing the number of windows required to find the target object.

Dedication

This thesis is dedicated to the memory of my mother, Cynthia Olson. She was my greatest champion and support throughout my time at Portland State University.

Acknowledgments

First, I would like to thank the faculty and staff of Portland State University for providing a quality educational experience and for encouraging me throughout my studies.

Special thanks to my advisor, Melanie Mitchell, for all the patience and mentoring that she has provided me during my time at Portland State University. I would also like to thank all the members of the Mitchell Research Group: Will Landecker, Max Quinn, Vladimir Solmon, and Jordan Witte. Each member has been instrumental in my research and has, on more than one occasion, provided me valuable insights into my work.

Thank you to those closest to me, my family. My son, Andrew, has been a good sport throughout my education even though it has so limited the time I have been able to give him. Finally, I would like to thank my fiance, Nicole. She has been the definition of patience and support during my studies at PSU.

This material is based upon work supported by the National Science Foundation under Grant Numbers (IIS-1018967 and IIS-1423651) Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Contents

List of Tables

List of Figures

Chapter 1 Introduction

An important goal in computer vision is to be able to locate all instances of a target object class in an image. For instance, the military may want to automatically locate all instances of missile silos in a few million satellite images. Alternatively, a self-driving car may need to quickly locate all pedestrians in a series of video images. Regardless of the application, the goal is to find all the target objects in an image.

Each object instance in an image is *localized* by drawing a box that tightly encloses the borders of the object. This process of finding all instances of an object class in an image is known as *object localization*, or simply, *localization*. Most state-of-the-art computer vision systems use an exhaustive search approach to localization, called *sliding-windows* [3, 4, 6–8, 16, 18]. A downside of the sliding-window approach is that, in a typical task, it can require a search through tens of thousands of windows to localize each object instance, making it a rather inefficient [11]. Recently, a number of methods propose using the context present within an image (e.g. horizon lines, sky and ground locations, etc.) to improve localization efficiency by reducing the number of windows considered. [2, 9, 10, 17].

In this thesis, I investigate whether the context between object classes can be learned and then leveraged to make the localization process more efficient. In particular, if the location and size of one object class instance is known, I hypothesize that a previously learned context model can be used to reduce the number of windows required to localize an instance of a related object class by constraining the location and size search space for that object.

1.1 Background

Over the past decade, computer vision systems have become increasingly adept at object localization. However, current state-of-the-art computer vision systems still perform well below that of human ability [15]. For example, if shown Figure 1.1, most humans would easily interpret this image as an instance of the concept *dog-walking* and very quickly locate the human dog-walker, the dog, and even the hard-to-see leash.

Figure 1.1: Image demonstrating the concept of *dog-walking*.

In contrast, most current computer vision systems would take an exhaustive approach to locating the objects in this image. Such a system might start with a small window in the upper left corner of the image and assign a score indicating the confidence of the system that a dog is present in that window. The system would then continue this process repeatedly over the whole image with different size windows and pick the highest scoring windows for dog locations. If the system then wanted to find the dog-walker, this exhaustive search process would begin anew. In the end, this system might be able to locate the dog or dog-walker but it would do so in an exhaustive, brute-force, kind of way. As is typical, brute-force methods are very inefficient. Such an approach makes no effort to use the situation (i.e., dog-walking) represented in this image to improve its search strategy. While

there are rich contextual interactions between the dog, dog-walker, and leash in the dog-walking situation, most current approaches would not attempt to model these relationships, instead opting to try window after window in a sequential, linear fashion.

The brute-force method I just described is used by most state-of-the-art systems and is more widely known as the sliding-window approach. More formally, in the sliding-window approach, a confidence score is computed for a fixed aspect-ratio window over many image locations and scales. The confidence score is computed by extracting image *features* (e.g., pixel values, object edges, textures, etc.) from the window and feeding them to a trained object classifier that computes the score. The highest scoring window is then returned as the likely object location within the image [11]. Since it is generally intractable to consider all locations and scales within the image, only a small subset of windows are actually evaluated for any particular image. Even so, the number of window samples required to localize an object can run into the tens to hundreds of thousands [11]. If a system needs to locate instances of many different object classes in an image, exhaustive search techniques for localization will be unacceptably expensive.

To address this kind of problem, some promising methods employ a *context-driven localization* approach where image context is used in order to improve the efficiency of localizing objects in the image [2,9,10,17]. Global image context, such as 3D structure [10] or general scene shape [17], is used by some, while others [1] use the local context directly surrounding an object (e.g., color contrast, object edges, etc.) to constrain the search space. However, few approaches directly use the situation-specific contextual relationships that exist between object classes to improve localization.

When contextual relationships, spatial or otherwise, exist between object classes in a particular situation (e.g., "dog-walking"), it may be possible to use these relationships for more efficient localization. For instance, knowing the location and size of the person in Figure 1.1 may tell us something about where to look for

an instance of the dog object class. If the context provided by the person is able to reduce the search space for both dog location and size, it is possible that the localization task for the dog instance can be completed using fewer windows. This is precisely the idea that this thesis explores.

Motivating this work is Petacat [14], a system in development by the Mitchell Research Group that focuses on the problem of image interpretation. As one of its initial computer vision goals, Petacat seeks to recognize image situations, with dog-walking being the first such situation to recognize. Given a new image, Petacat's task is to determine if the image is an instance of a situation category it has previously learned. In this sense, Petacat already knows what situation it is currently trying to find and can use this knowledge to drive which objects and contextual relationships to look for in the image. A future goal is to integrate the work presented in this thesis with the Petacat system to help make localization efficient and scalable.

In the next sections, I discuss related work and give a high-level description of my approach to context-driven localization.

1.2 Related Work

In this section, I describe some of the more prominent approaches to object localization in recent literature.

Torralba et al. [17] use the global context of the image by computing the "gist" of an image. This is done by pre-processing the image with a series of Gabor filters to obtain a gist feature vector that describes the spatial layout of the image. These gist features are then used with a weighted average of linear regression models to learn a distribution over vertical image locations for a particular object class [17]. This distribution can then be used to focus the localization task on a narrow horizontal band within the image through a process that Torralba et al. call *location priming*. However, this method is unable to restrict the horizontal object location search space. In contrast, the methods proposed in my thesis allow

for constraining both vertical and horizontal search space.

Elazary and Itti [5] use general saliency maps to restrict the search space prior to localization attempts. The saliency maps are constructed using the pixel intensities, color opponencies (e.g., green vs. red) and edge detectors at four different orientations. Using these features, each pixel in the image is given a saliency value. Higher values indicate likely locations of objects (of any type) in the image. The localization search space is then constrained to the most salient areas of the image. While Elazary and Itti's saliency approach uses local pixel and color context, it does not otherwise leverage spatial or size relationships in the image.

Alexe et al. [1] use what they call image "cues" to obtain a measure of "objectness" for segmented regions of the image. These cues include multi-scale saliency maps, color contrasts, edge densities, and superpixel straddling to combine oversegmented image patches into groups of superpixels likely to contain an object. This effectively reduces the search space prior to localization but only uses the local context immediately surrounding the object instead of the contextual relationships between objects in the image.

Hoiem et al. [10] attempt to estimate the 3D structure of an image by calculating the camera viewpoint and surface geometry. The surface geometry classifies each pixel as belonging to the sky, ground, or vertical class (a surface sticking up from the ground). The vertical class also has further subclasses: planar, left, center, right, non-planar solid or porous. These features are then used in a Bayesian network to estimate prior locations and scales for various objects in the image. However, the vertical class labeling involves considerable hand annotations for the image set. Again, no attempt is made to use contextual object interactions in Hoiem et al.'s approach.

Perhaps most similar to my work is another approach of Alexe et al., described in [2]. Here they use the spatial context between similar randomly selected windows and a target object class. During training, [2] samples a large number of windows from all training images and records their location and size within the

image they were sampled from. Features are extracted for each of these windows and a displacement vector from the window center to the ground truth target object is calculated. To perform localization in a new test image, a random window is chosen in the image and its features computed. These features are compared with the windows sampled during training to find the ten most similar training windows. The displacement vectors from the similar windows "vote" for the location of the target object. A probability distribution over possible image locations for the target object is then computed from the votes using kernel density estimation. A new window is then sampled at the highest probability location and the process repeats for a fixed number of iterations T. The distribution over possible locations is updated with each iteration and the window sampled on iteration T is used as the object location in the image. While this approach does use the spatial context between regions in the image, it does not directly use the interaction between target objects in the image. Instead, [2] uses the spatial context between clustered groups of windows and the object of interest.

While various types of image context are used in the above mentioned approaches, none make use of the direct contextual interactions between objects of interest to restrict the search during localization. As such, my focus is on learning situation-specific context models that allow the size and location of one object to constrain the search for another related object in the image. For example, consider Figure 1.2, which depicts a person skateboarding. Here the size and location of the person can heavily influence the size and location of the skateboard and vice versa.

Figure 1.2: Image representing a skateboarding situation.

1.3 Summary of Experimental Approach

In this section, I present a high-level overview of my approach to context-driven localization and define the evaluation metrics used during my experiments.

I chose the Portland Dog-Walking image corpus for the context-driven localization task. This dataset consists of photographs of people walking dogs. The photographs were taken by members of the Mitchell Research Group at Portland State University. A few examples of these images are shown in Figure 1.3 and the dataset is explained in detail in Section 2.1.

Figure 1.3: Example images from the Portland Dog-Walking corpus.

The essential characteristic of this dataset is that there are two main object classes, person and dog, that maintain similar contextual relationships throughout all the images (i.e., the person is walking the dog).

To examine the role of inter-object context in localization, I assume that one object has been located with a bounding box and the second object must be found (i.e., the system must draw a bounding box around it). The context of the first object will be used to localize the second object. In particular, a conditional probability distribution over object locations is estimated for one object given the other. This results in a distribution over all image pixels where each pixel has some probability of being the center of the target object given the localized object. Additionally, probability distributions are learned over the ratio of bounding-box heights and ratio of bounding-box areas between objects. For instance, a probability distribution for the ratio of dog bounding-box heights to dog-walker bounding-box heights is estimated. Given the bounding-box height of a localized dog-walker, the height ratio distribution can be sampled to estimate the likely height of the dog given the dog-walker height. The probability distributions over image pixels encode the spatial context while the height and area ratios encode the relative size context. Together, these learned conditional probability distributions are used by my system to localize objects in new ("test") images.

In my system, the above distributions are learned on a training set of images in which each image contains a single dog and dog-walker. On a test image, the location and bounding-box height and area of one object are given. The context encoded in the learned distributions is then used to constrain the search space and window size for the related object. Likely object locations are sampled from the learned spatial context distribution and a window is generated by sampling from the learned size context distribution. For each sampled window, the intersection over union (IOU) of the sampled window and ground truth bounding-box is computed. The IOU is a measure of the correct overlap between a predicted bounding-box and the ground truth bounding-box of the target object (see Equation 2.3). An IOU ≥ 0.5 is a standard measure of localization success in current literature [15]. If the IOU for the sampled window meets or exceeds this threshold, it is used as the bounding-box for that object instance.

To evaluate the efficiency gains of my system with respect to models that do not use context, I compare my system with a uniform model in which image locations and window sizes are selected uniformly. I also compare my system against saliency models similar to those proposed by Elazary and Itti [5]. The primary quantity of interest is the mean number of windows required to localize dogs and dog-walkers in the Portland Dog-Walking corpus. In this thesis, I show that my context-driven system is able to substantially reduce the mean number of windows required to localize dogs and dog-walkers. Furthermore, I show that it is possible to combine my system with saliency based models to make additional efficiency gains.

The rest of my thesis is organized as follows: Chapter 2 describes the Portland Dog-Walking corpus, the methods used to learn the conditional probability distributions for spatial and size context, and implementation details for each localization model. Chapter 3 presents the results of running each localization model on the Portland Dog-Walking corpus. Chapter 4 presents a method for combining my context-driven model with a saliency approach and Chapter 5 presents the performance results of the combined model. Finally, in Chapter 6, I discuss the localization performance of each model, discuss future work, and make my final conclusions.

Chapter 2 Methods

In this chapter, I describe the dataset and methods used by my system to build localization models. I also describe the localization process used for testing. Object localization can be broken into two essential parts: location selection and window generation. During location selection, the localization model probabilistically chooses image locations one at a time. This is done by sampling from a probability distribution over image locations. At each sampled location, a window is generated. The goal is for the target object instance to be contained within the generated window. Each model handles window generation differently, as will be described below. A target object is considered to be correctly localized if the window has significant overlap (IOU ≥ 0.5) with the ground truth bounding-box for that object instance.

There are three localization models under consideration: *Salience*, *Context*, and *Uniform*. The Salience model uses saliency maps like those in [5] for location selection and a learned distribution over relative size of object classes and images for window generation. The Salience model is used for comparison purposes to determine if the Context model provides additional benefits over a model that does not use contextual relationships between objects. The Context model I developed uses learned conditional probability distributions over image locations for choosing possible object locations. For window generation, probability distributions over relative object sizes (i.e., dog size vs. dog-walker size) are sampled to obtain the window size parameters. Finally, the Uniform model serves as a baseline for performance comparison purposes. As its name implies, the Uniform model takes a uniform approach to both location selection and window generation.

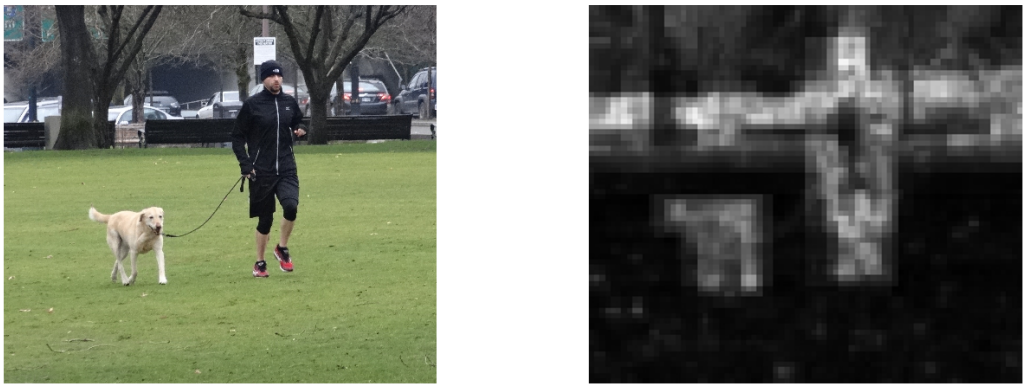
The rest of this chapter describes the dataset used for training and testing, each of the model implementations, and the localization process used during testing.

2.1 Portland Dog-Walking Image Corpus

I use the Portland Dog-Walking image corpus for training and testing all models in this work. The corpus consists of 562 dog-walking images of various aspect-ratios (421 training, 141 testing). Each image contains a single dog-walker and dog, but may have multiple occurrences of other objects (people, bicycles, etc.). The training images are used to learn object location and window size probability distributions. Test images are used only for object localization and kept strictly segregated from training data. Each image has a corresponding label file that specifies the ground truth bounding-box height and width for both dogs and dog-walkers. All photographs in this corpus were taken by members of the Mitchell Research Group from various viewpoints, times of day, and orientations, using multiple camera types. Image labels are manually annotated by members of the group using a web-tool that automatically generates the corresponding label file.

2.2 Salience Model

I use Elazary and Itti's salience model [5] as a non-context-driven baseline with which to compare context-driven models. According to this salience model, the most salient areas of the image are thought to be the most likely locations for objects. In particular, given an image, the salience model computes 42 types of features based on edges, color, and contrast, at several difference scales. These features are computed at each location in the image, producing 42 saliency maps. The maps are then combined into a single saliency map for the image, as described in [5]. The output is a probability distribution over locations in the image, where the most salient locations have the highest intensity (see Figure 2.1). Note that the model implemented in [5] does not perform any learning on training data; the salience of a given input image location depends only on simple features directly computed from the image itself.



(a) Test image (b) Saliency map

Figure 2.1: Original image (a) and saliency map (b), where the most salient (i.e., brightest) pixels in (b) represent regions of interest in (a) that are likely to belong to objects.

All saliency maps for this model are generated using Max Quinn's Matlab implementation of Elazary and Itti's algorithm [5]. In my system, I use the saliency maps to perform location selection and I extend the model by implementing a window generation method, as described below.

While an absolute location prior could have been used

2.2.1 Salience Model Location Selection

Location selection for the Salience model is straightforward. Given a test image, my system computes the saliency map from the image features as described in [5]. The resulting saliency map is interpreted as a probability distribution over image locations, where the most salient locations have the highest probability of belonging to an object of any class. During localization, likely object locations can be sampled directly from the saliency map distributions.

For example, suppose the task is to localize the dog in Figure 2.1a. To pick a new location in the image to look for the dog, an image location would be sampled directly from the saliency map in Figure 2.1b, where the highest intensity pixels have the highest probability of being chosen.

To be precise, let O denote any object class (e.g. dog-walker, dog, leash, etc.)

and let O_{xy} be the event that the pixel located at image coordinate (x, y) is the center of an instance of the object class O. Additionally, let θ_s represent the salience map used to generate saliency values at each image pixel. For an $N \times M$ input image, the probability distribution over all pixel coordinates (i, j) is:

$$\Pr(O_{ij}|\theta_s) = \mathbf{G}_{ij} \quad i = 1 \ldots N, j = 1 \ldots M \tag{2.1}$$

where $\mathbf{G}_{ij}$ is the saliency value at index (i, j) of the image saliency map.

2.2.2 Salience Model Window Generation

After sampling an image location from $\Pr(O_{xy}|\theta_s)$, a window must be generated. Recall, the goal is for the window to have significant overlap (IOU ≥ 0.5) with the ground truth bounding-box of the target object instance. Inherently, the approach used in [5] provides no method for window generation. To deal with this issue, I decided to have the Salience model learn a probability distribution for window sizes from the observed ratios of object bounding-box height and area with respect to image height and area.

Specifically, for a training image, let O_a and O_h represent the target object ground truth bounding-box area and height, respectively. Similarly, let I_a and I_h denote the image area and height, respectively. My system estimates two probability distributions, $\Pr(\beta)$ and $\Pr(\eta)$, from the training images, where $\beta = O_h/I_h$ and $\eta = O_a/I_a$. This is done by observing β and η for all training images and using the Matlab function `ksdensity` [13] to estimate both $\Pr(\beta)$ and $\Pr(\eta)$. To generate a new window, W, with a sampled location as its center, my system samples a new β and η from their respective distributions and computes the window height, W_h, and area, W_a, as follows:

$$W_h = \beta * I_h$$

$$W_a = \eta * I_a$$

Once the window has been generated, the IOU of the window and the ground truth object bounding-box can be computed to determine if the object has been localized.

2.3 Context Model

The Context model I developed leverages the known location and bounding-box of one object class instance to probabilistically choose a possible image location and size for a related target object class. In my system, the two object classes of interest are dog and dog-walker. As an example, suppose we have identified the location and have specified a bounding-box for the dog-walker as depicted in Figure 2.2. The task of the Context model is now to localize the dog in this image using the context of the dog-walker.

Similar to the Salience model, the Context model uses probability distributions over image locations and window size during localization, but the methods used to obtain them differ from the salience approach. Specifically, the context between dogs and dog-walkers observed in training data is leveraged to generate the distributions over locations and window size. The next sections explain how these distributions are learned from the training data. Note, throughout this chapter, I will assume that the dog-walker has been localized and the dog is the target object for localization. The reverse process is completely analogous.

2.3.1 Context Model Location Selection

In the Context model, location selection is conditional on the known location of one of the object classes. For example, in Figure 2.2, the dog-walker has been localized. Given the location of the dog-walker, the Context model will probabilistically choose possible dog locations by sampling from a learned image location distribution. Context location distributions differ from salience map distributions in that the context distributions are object-specific and conditional on the location of some other object class. The distributions are object-specific because a

different distribution is used for each object class (e.g. dog, dog-walker, etc.).

Figure 2.2: An image where the dog-walker has been localized.

My approach to learning these conditional probability distributions is straightforward. For each training image, the displacement from the center of the dog-walker bounding-box to the center of the dog bounding-box is calculated and represented as a point (x, y) where the center of the dog-walker is considered the origin. The point (x, y) is then normalized by the height[1] of the dog-walker bounding-box so that points from different images will be on the same scale. More formally, let N be the number of training images; (d_x,d_y) be the coordinate of the dog center; (w_x,w_y), be the coordinates of the dog-walker center, and w_h be the height of the dog-walker bounding-box. A collection of displacement points for

[1]Height is chosen because it more naturally captures the distance of an object from the camera.

the i^{th} image are calculated as follows:

$$(x^{(i)}, y^{(i)}) = \left(\frac{d_x^{(i)} - w_x^{(i)}}{w_h^{(i)}}, \frac{d_y^{(i)} - w_y^{(i)}}{w_h^{(i)}} \right) \quad i = 1 \ldots N \tag{2.2}$$

A plot of all the points $(x^{(i)}, y^{(i)})$ from the training images can be seen in Figure 2.3, where the origin represents the center of the dog-walker and the axes are the normalized distances in pixels from the dog-walker to the dog.

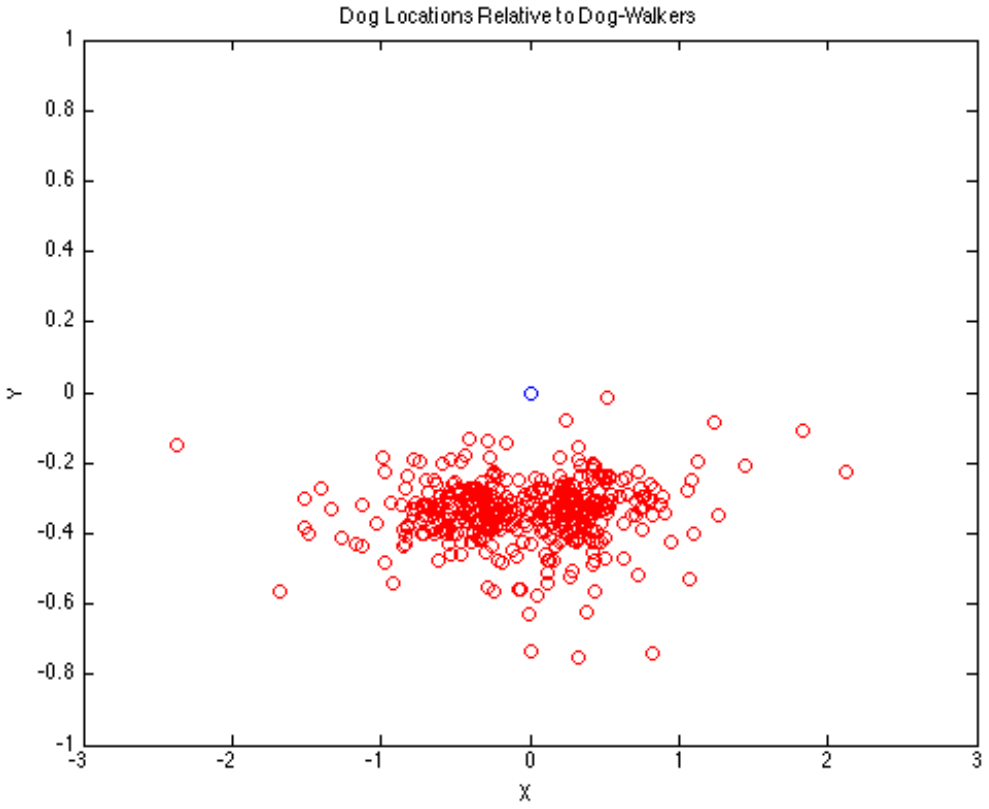


Figure 2.3: Plot of dog locations relative to dog-walkers from the training set. The origin represents the center of the dog-walker bounding-box. Figure best viewed in color.

The grouping of points in the plot make it clear that there is a non-random spatial relationship between dogs and dog-walkers.

Of course, a collection of points is not a probability distribution. To resolve this problem, a probability distribution must be estimated from the points computed in Equation 2.2. To do this, a distribution over image locations is estimated

with the Matlab function `p2kde`, written by Max Quinn. The `p2kde` function takes as input a collection of points and performs kernel density estimation to approximate the two-dimensional probability distribution from which the points were likely sampled. For example, Figure 2.4b visualizes the estimated probability distribution obtained from the points in Figure 2.4a.

(a) Point samples (b) Estimated distribution

Figure 2.4: Example of kernel density estimation using Matlab function `p2kde`. (a) is a collection of points represented as an image. (b) is the resulting probability distribution. High intensity (bright) locations correspond to high probability.

From the process described above, two conditional probability distributions over image locations are learned. The first, $\Pr(d_{xy}|w_{xy})$, is the conditional distribution of dog locations given the dog-walker location, where d_{xy} and w_{xy} denote the dog and dog-walker locations, respectively. The second learned conditional distribution, $\Pr(w_{xy}|d_{xy})$, is over dog-walker locations, given the dog location.

For a new image with the dog-walker localized, possible dog locations are selected by sampling points from the learned conditional probability distribution, $\Pr(d_{xy}|w_{xy})$. If the dog was the localized object, then possible dog-walker locations would be sampled from $\Pr(w_{xy}|d_{xy})$ instead. As in the Salience model, once a point is selected, a window must be generated in order to compute the IOU. The window generation method for the Context model is described next.

2.3.2 Context Model Window Generation

Window generation for the Context model is similar to that in the Salience model except I leverage the size relationships between the dog and dog-walker object classes rather than the size relationship between the object and image. Specifically, the ratio of ground truth bounding-box area and height between dogs and dog-walkers is used to estimate two probability distributions for window size.

The first distribution is over the height ratios between dogs and dog-walkers. Let d_h and w_h be the height of the ground truth bounding-box for the dog and dog-walker, respectively. Also, let $\gamma_d = d_h/w_h$ be the ratio of dog height to dog-walker height. A probability distribution, $\Pr(\gamma_d)$, over the height ratio is estimated by observing all height ratios in the training set and using Matlab's `ksdensity` function [13] to perform one-dimensional kernel density estimation on the observed ratios. The distribution, $\Pr(\gamma_w)$ for the ratio of dog-walker to dog heights is similarly estimated.

The second distribution learned is over the area ratios between dogs and dog-walkers. Let d_a and w_a be the ground truth bounding-box area for dogs and dog-walkers, respectively. Let $\alpha_d = d_a/w_a$ be the ratio of dog area to dog-walker area. The distribution over area ratios, $\Pr(\alpha_d)$ is also estimated by observing all α_d in the training set and obtaining an estimate of $\Pr(\alpha_d)$ from the `ksdensity` function. Again, the distribution for dog-walker to dog area ratios, $\Pr(\alpha_w)$, is estimated in a similar way.

The window size distributions just described can be used for localization on a new image to probabilistically generate window height and area based on the context of the localized object. For example, given a new image where the dog-walker has been localized, the Context model samples γ_d and α_d from their appropriate distributions and computes the window height, W_h, and area, W_a, as follows:

$$W_h = \gamma_d * w_h$$

$$W_a = \alpha_d * w_a$$

Generating candidate windows for dog-walkers is accomplished in the same way, with the appropriate adjustments to the sampling distributions.

2.4 Uniform Model

The Uniform model serves as a null comparison model for localization on the dog-walking dataset. For the Uniform model, the probability distribution for object locations is uniform (i.e., all object locations are equally likely). During location selection, an image location is simply selected uniformly from all image locations.

Window generation in the Uniform is very simple. A uniform distribution over the height ratios β between the target object class and the image is obtained by observing the minimum and maximum β in the training set and using those values as the extreme for the distribution. The uniform distribution for area ratios η is computed in a similar way. On a new image, a window W is generated by picking a β and η uniformly from their respective distributions and computing the window height W_h and area W_a as follows:

$$W_h = \beta * I_h$$

$$W_a = \eta * I_a$$

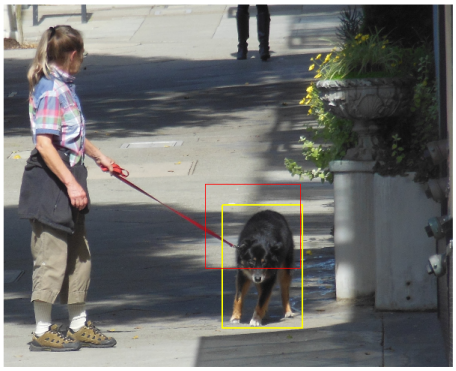
where I_h and I_a represent the image height and areas, respectively, as in the Salience model.

2.5 Localization Procedure

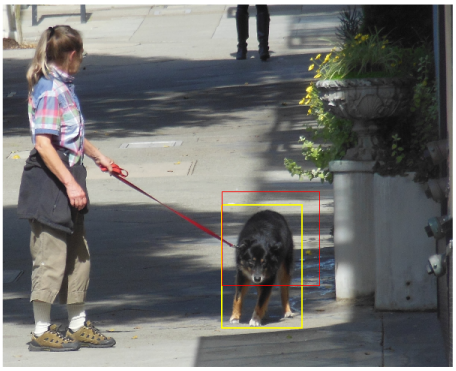
As I described above, the goal of object localization is to specify the location of the target object in an image by drawing a bounding-box around the object. The standard measure of an accurate localization is the intersection over union (IOU) metric [15]. Specifically, an object is considered successfully localized if the IOU is greater or equal to 0.5. IOU is calculated as:

$$IOU(B_{gt}, B_p) = \frac{area(B_{gt} \cap B_p)}{area(B_{gt} \cup B_p)} \tag{2.3}$$

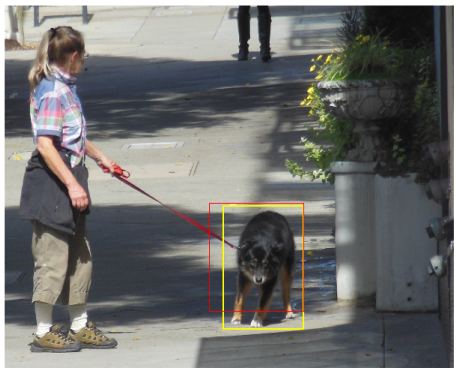
where B_{gt} and B_p are respectively the ground truth and predicted bounding boxes. As an example, Figure 2.5 shows some predicted bounding-boxes and the corresponding IOU values.



(a) IOU = 0.38



(b) IOU = 0.51



(c) IOU = 0.72

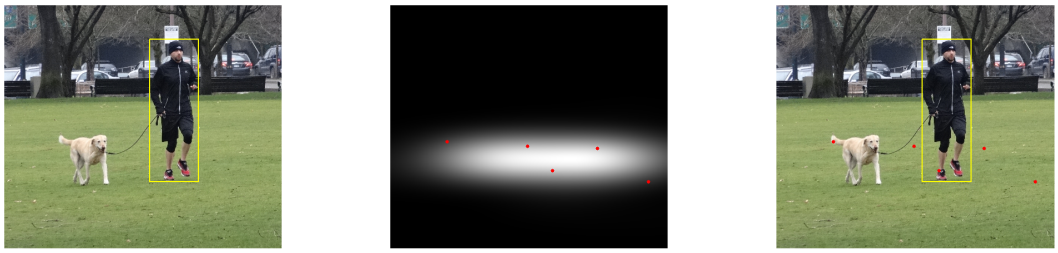
Figure 2.5: Example bounding-box predictions with IOU values. Yellow boxes are ground truth and red are window predictions. Figure best viewed in color.

Throughout this section I assume the target object is the dog. The process for localizing the dog-walker is essentially the same. Additionally, for the Context model, it is assumed that the dog-walker has been localized and we know its bounding-box size.

The localization process has two steps: location generation and window generation. The first step is to sample a location from one of the model probability distributions over image locations. Figure 2.6 illustrates sampling from the Context model distribution. By sampling a point from the distribution in Figure 2.6b, we obtain a point in the test image, shown in Figure 2.6a, that is thought likely to locate the center of the dog.

This location selection procedure is identical for all of the models. That is, points are sampled from the respective model probability distributions to generate

points of interest that may locate the center of the target object in the test image.



(a) Test image (b) Sampled points (c) Plotted points

Figure 2.6: An example of location sampling. (a) is the image under consideration, (b) shows five points sampled from $\Pr(d_{xy}|w_{xy})$, and (c) shows the points plotted on the test image. Figure best viewed in color.

Once a location has been sampled for a possible dog location, a window is generated. The sampled locations serve as the center of the window. Window size is determined by sampling from the appropriate size distributions as described above for each model.

Figure 2.7 illustrates an example of window generation for the Context model. The red point is the sampled location for a possible dog location relative to the dog-walker. The three windows centered on the sampled location were generated by sampling height and area ratios from $\Pr(\gamma_d)$ and $\Pr(\alpha_d)$ distributions, respectively. Finally, the actual height and area for each window is computed as described in Section 2.3.2 using the localized dog-walker height.

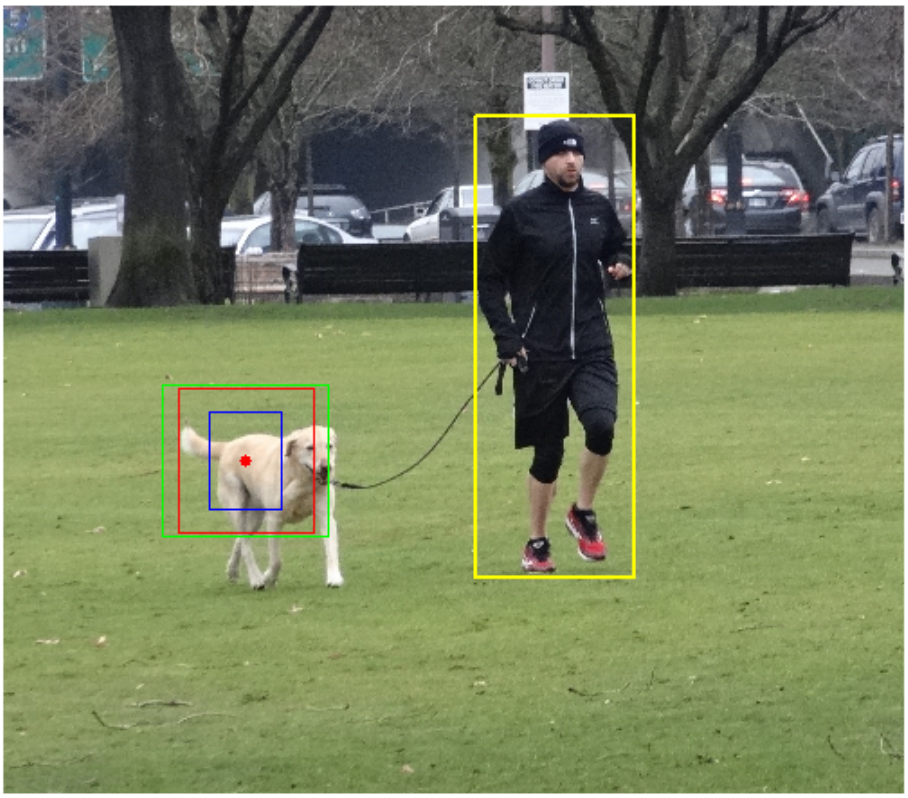


Figure 2.7: An example of window generation for Context model. Here the red point has been chosen as a possible location for the dog. Three windows are shown where the area and height parameters are calculated by sampling from $\Pr(\alpha_d)$ and $\Pr(\gamma_d)$. Figure best viewed in color.

Finally, for each point and window sampled, I calculate the IOU for the ground truth dog bounding-box and the generated window. If IOU ≥ 0.5, the dog has been successfully localized and, if not, the process can begin anew by sampling a new point and generating a new window from the appropriate distributions.

It is worth noting that no classifiers for dogs or dog-walkers are used in my localization process. I chose to isolate and evaluate the merits of these localization models directly. Using window classifiers would introduce errors into my results, making it difficult to determine the true contribution of context for object localization. In practice, any object window classifier could be used in the process explained above to assign a score for each window.

Chapter 3 Results

In this chapter, I report the dog and dog-walker localization performance results for the Uniform, Salience, and Context models on 141 test images from the Portland Dog-Walking corpus. All models were trained on 421 images from the same corpus. Localization results are obtained by sampling locations from the specified model probability distribution and generating a window for the target object at each location until an IOU ≥ 0.5 is reached or 5000 locations have been sampled. If the target object has not been localized after 5000 window samples, the model is considered to have failed the localization task for that image. The window generation method is dependent upon the type of localization model under consideration. The entire process described above is repeated ten times for each model to yield an average number of windows required to localize the target object in each test image.

The graphs in this chapter plot the percentage of the target object class that was successfully localized against the mean number of windows sampled to localize that percentage of targets. For example, Figure 3.1 plots the model results for the dog localization task. If we choose a window threshold along the x-axis, say 1000, the y-axis tells us what percentage of dogs, across all test images, were successfully localized within an average of 1000 window samples. Each plot also contains error bars that extend one standard deviation above and below the graph line at periodic intervals.

3.1 Dog Localization Task

Figure 3.1 plots the performance results of the Uniform, Salience, and Context models on the dog localization task. The Context model clearly localizes a greater percentage of dogs than either the Uniform or Salience models for all window

sample thresholds. In particular, the Context model successfully localizes at least 80% of the dogs within 500 box evaluations while the next best model, Salience, has only localized around 20% of dogs in the test images.

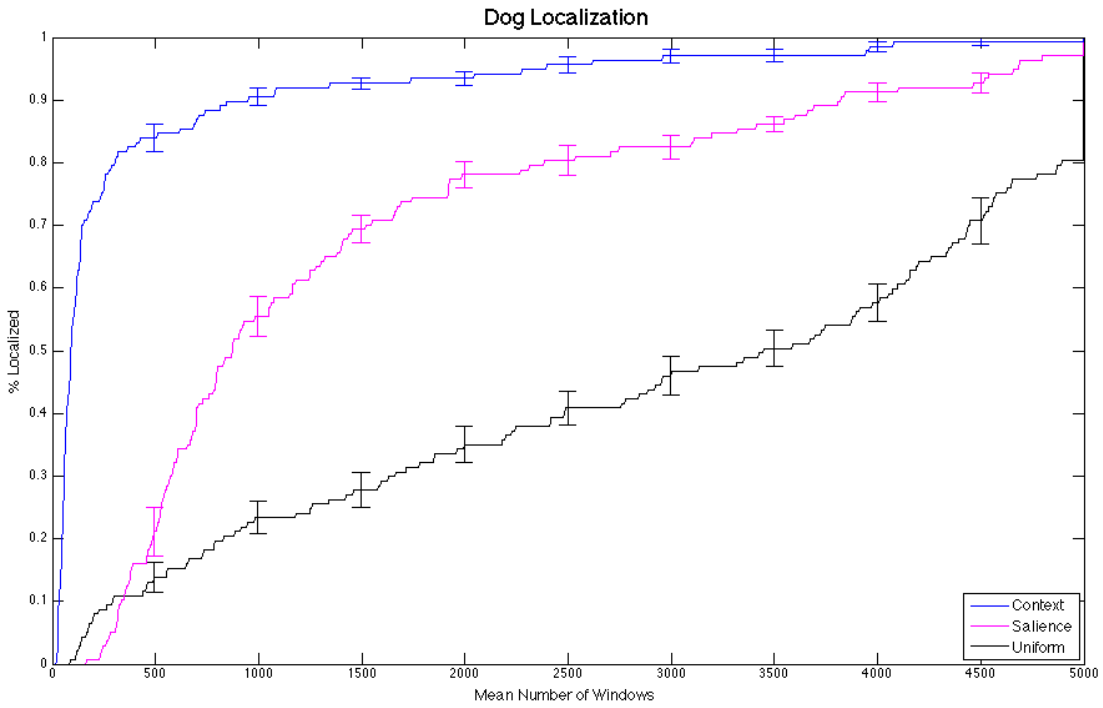


Figure 3.1: Localization performance plot for Uniform, Salience, and Context models. The y-axis plots the percentage of dogs in all test images that are successfully localized within the mean number of sampled windows plotted along the x-axis. Figure best viewed in color.

The Salience model also significantly outperforms the Uniform model for nearly all window sample thresholds. However, it appears that the Uniform model is able to localize a greater percentage of dogs than the Salience model for window sample thresholds less than 300.

As an overall quantitative measure of model performance on the dog localization task, I calculate the mean and standard deviation of the number of window samples required to successfully localize a single dog instance for each model. Table 3.1 reports these results.

The best performing model for dog localization across all metrics is the Context model, with a mean of 352.7 window samples required for a successful dog

localization. This is a reduction of 85.8% over the Uniform model and 74.1% over the Salience model for mean window samples. The Salience model is also able to substantially reduce the mean window samples versus the Uniform model, requiring only 1363.7 samples on average, a reduction of 45.1% over the Uniform model.

Dog Localization		
Model	**Mean Windows**	**StdDev**
Uniform	2485.6	1628.5
Salience	1363.7	1230.9
Context	352.7	748.9

Table 3.1: Performance of the Uniform, Salience, and Context models on dog localization task. The mean windows column represents the mean number of window samples required to successfully localize a single dog instance, averaged over all test images, for ten independent trials.

3.2 Dog-Walker Localization Task

The results for the dog-walker localization task exhibit some substantial differences from those for dog localization. Figure 3.2 plots the results over 5000 window samples. Here, the Context model is able to localize a greater percentage of dog-walkers for all window thresholds, however, the improvement over the Salience model is less pronounced. For the 500 window threshold, the Context model successfully localizes approximately 90% of the dog-walkers in all test images, while the Salience model localizes roughly 82%. Both the Context and Salience models localize a significantly greater percentage of dog-walkers than the Uniform model across nearly all window sample thresholds.

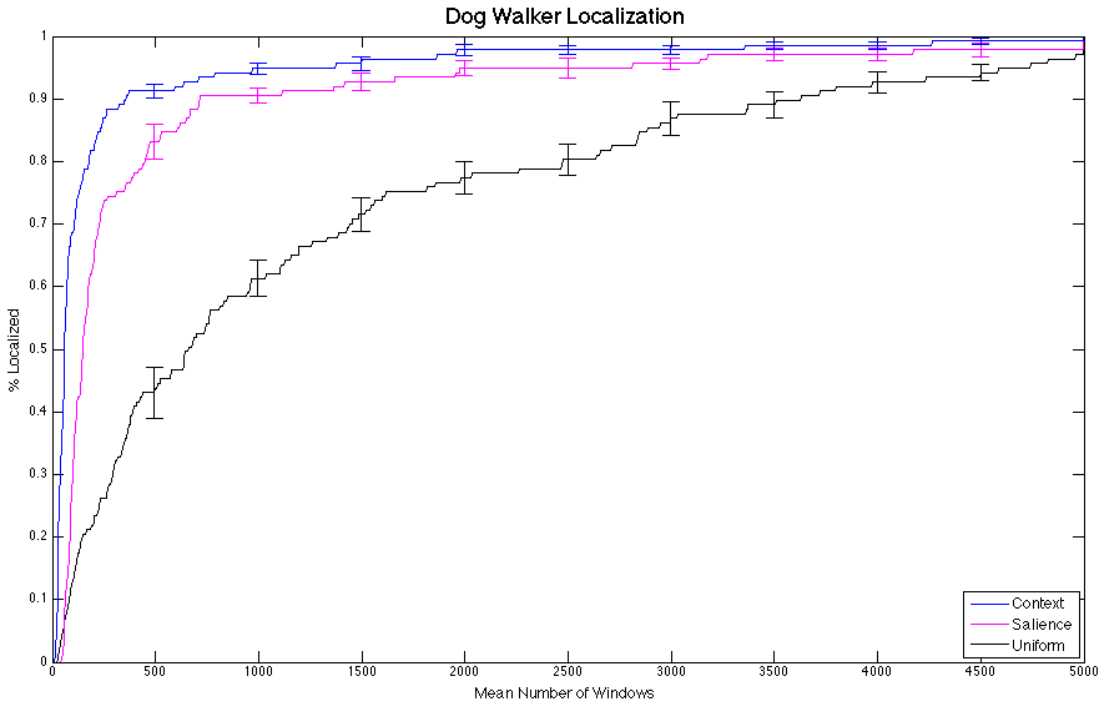


Figure 3.2: Dog-walker localization performance up to 5000 window samples. Figure best viewed in color.

Table 3.2 reports the mean and standard deviation of the number of window samples required to successfully localize a single dog-walker instance. All models require significantly fewer window samples on the dog-walker localization task in comparison to dog localization. The Context model requires the fewest windows at 207.7, a reduction of 81.8% over the Uniform model and 40.7% over the Salience model. The Salience model again requires fewer window samples than the Uniform model at 350.5, a reduction of 69.3% over the Uniform model.

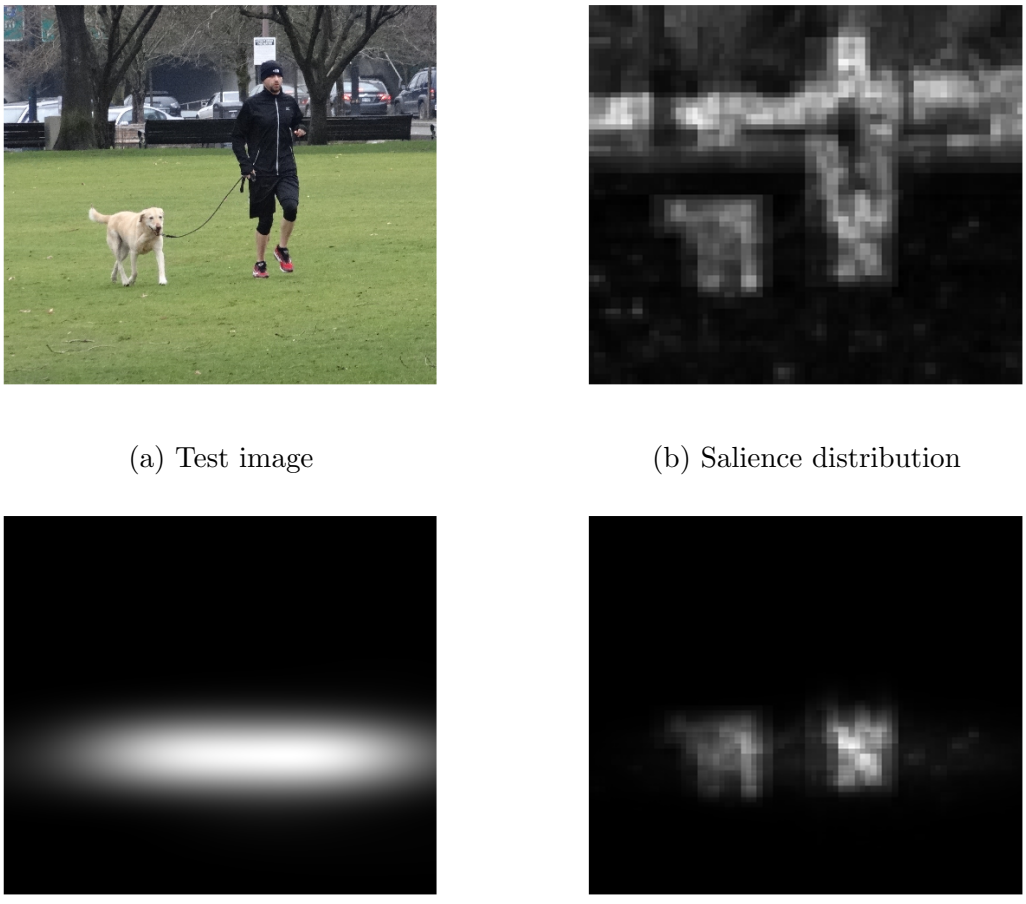
Dog-Walker Localization		
Model	**Mean Boxes**	**StdDev**
Uniform	1141.4	1269.5
Salience	350.5	628.1
Context	207.7	540.0

Table 3.2: Performance of Uniform, Salience, and Context models on dog-walker localization.

In summary, for both localization tasks, the Context model requires significantly fewer window samples on average to successfully localize the target object than the Uniform or Salience models. Additionally, all three models require fewer window samples on average for dog-walker localization than for the dog-localization task.

Chapter 4 Combining Context and Salience

In this chapter I present a methodology for combining the Context and Salience models. Because the Context and Salience models both use a probability distribution over image locations to select possible object locations, I was interested in seeing if the performance of Context model improves by combining it with the Salience model approach. To do this, I combine the probability distributions over image locations from both models as illustrated in Figure 4.1.



(a) Test image

(b) Salience distribution

(c) Context distribution

(d) Combined distributions

Figure 4.1: An illustration of the result (d) from combining the Context and Salience model location distributions. (b) and (c) are the distributions obtained from the Salience and Context models, respectively, for the image in (a).

Combining the Context and Salience model location distributions is done in a simple way. The probability distribution, $\Pr(d_{xy}|w_{xy})$, computed as described in Section 2.3, is simply point-wise multiplied with the salience distribution $\Pr(O_{xy}|\theta_s)$. Specifically, since the discrete Context and Salience location distributions are both the same dimensions (i.e., the dimensions of the image under consideration) and are represented as a probability matrix, the combined distribution for dog locations, $\Pr(d_{xy}|w_{xy}, \theta_s)$, is calculated as follows:

$$\Pr(d_{ij}|w_{xy}, \theta_s) = \frac{\Pr(d_{ij}|w_{xy}) \Pr(O_{ij}|\theta_s)}{\sum_{ij} \Pr(d_{i,j}|w_{xy}) \Pr(O_{ij}|\theta_s)} \quad i = 1 \ldots N, j = 1 \ldots M \tag{4.1}$$

where the denominator is simply a normalization term to make the distribution sum to 1. The result of Equation 4.1 is visualized in Figure 4.1d for the test image in Figure 4.1a.

During localization, $\Pr(d_{xy}|w_{xy}, \theta_s)$ can be sampled in the same way as the Context model to generate possible object locations. Window generation also is performed the same way as that described for the Context model.

Next, I present the results for the combined model in comparison to the Context model.

Chapter 5 Combined Model Results

In this chapter, I present the localization performance results for the combined Context and Salience model. For clarity, I will refer to the combined model by the title *Combined*. I also include the results for the Context model alone for comparison purposes.

5.1 Dog Localization Task

Figure 5.1 plots the performance results of the Combined and Context models on the dog localization task. The Combined model slightly outperforms the Context model for most window threshold values. For instance, at a threshold of 500 windows, the Combined model has localized around 90% of the dogs while the Context model has localized a little over 80% of the dogs in the test set.

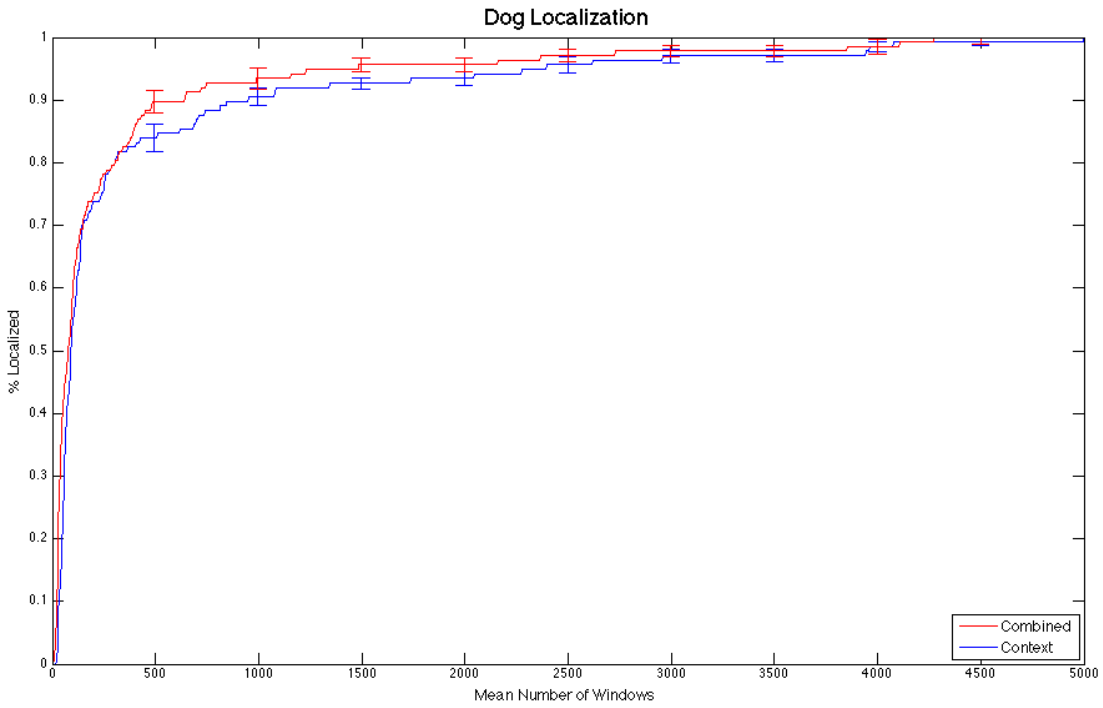


Figure 5.1: Localization performance plot for the Context and Combined models on dog localization task. Figure best viewed in color.

As was done for the previous model results, I calculate the mean and standard deviation of the number of window samples required to successfully localize a single dog instance for the Combined model and present them in Table 5.1.

The Combined model makes slight improvements in the mean number of windows required to successfully localize a dog. In particular, the Combined model requires only 266.7 windows on average, for an improvement over the Context model of 24.4%.

Dog Localization		
Model	**Mean Windows**	**StdDev**
Context	352.7	748.9
Combined	266.7	613.2

Table 5.1: Performance of the Context and Combined models on the dog localization task. The mean windows column represents the mean number of window samples required to successfully localize a single dog instance, averaged over all test images, for ten independent trials.

5.2 Dog-Walker Localization Task

Figure 5.2 plots the performance results of the Combined and Context models on the dog-walker localization task. For dog-walker localization, the Combined and Context models have very similar results. In fact, it is difficult to say which model performs best for various window thresholds. For instance, at a threshold of 500 window samples, both models are able to localize around 90% of the dog-walkers successfully.

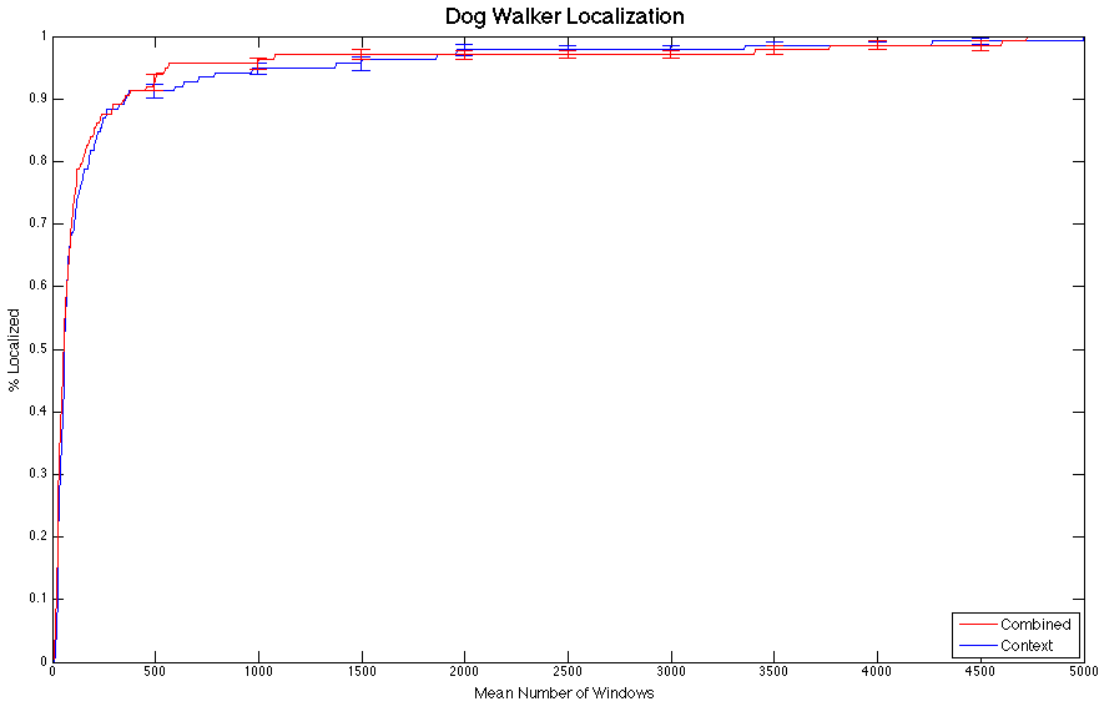


Figure 5.2: Localization performance plot for the Context and Combined models on dog-walker localization task. Figure best viewed in color.

Table 5.2 reports the mean window samples required to successfully localize a dog-walker on the test image set for both the Combined and Context models. Again, the performance between the two models is very close, with the Combined model requiring slightly fewer window samples, 192.2, versus the Context model. This is a small improvement approximately 7.5% over the Context model for this task.

Dog-Walker Localization		
Model	**Mean Windows**	**StdDev**
Context	207.7	540.0
Combined	192.2	589.9

Table 5.2: Performance of the Context and combined Context and Salience models on the dog-walker localization task.

In summary, the Combined model performs slightly better than the Context model alone for the dog localization task. However, the results for the dog-walker localization task are nearly equivalent for both models. As was seen with the Uniform, Salience, and Context models, the Combined model required fewer window samples on average for the dog-walker localization task than for the dog localization task.

Chapter 6 Discussion

In this thesis, I presented the Context object localization model that leverages the contextual relationships between dogs and dog-walkers in the "dog-walking" situation to constrain the search space for object location and size during localization. There are two ways in which the contextual relationships between dogs and dog-walkers are used in the Context localization model I developed. First, the location context of one object is used to restrict the search space over possible target object locations via learned conditional probability distributions. Second, the size context of one object is being leveraged to constrain the search space for target object window sizes. I also presented a method for combining Salience models with the Context model to further constrain the search space over object locations.

In the next sections, I discuss the results of the Context and Combined models, highlight a few potential drawbacks of the Context model, mention future work, and finish with my conclusions.

6.1 Context-Driven Localization

The Context model I created in this work directly leverages the location and size context of dogs and dog-walkers to substantially reduce the number of window samples required for a successful object localization. As we can see in Figures 3.1 and 3.2, the Context model is able to successfully localize a much greater percentage of target objects with substantially fewer window samples than either the Salience or Uniform models. For example, on the dog localization task, the Context model successfully localizes at least 80% of all dogs in test images in as few as 500 window samples. The Salience model would require at least 2000 window samples to achieve the same localization percentage and the Uniform

model would require 5000 or more windows. A similar pattern is seen for the dog-walker localization results.

From Tables 3.1 and 3.2, we can also see that the average number of windows required in the Context model to successfully localize either a dog or dog-walker in a test image is significantly lower than the Uniform and Salience models, suggesting that the use of context in localization is able to substantially reduce the number of windows required for a successful localization. These reductions in the number of windows required translate to substantial efficiency gains. Recall that a sliding-window localization method can require a search over tens of thousands or more window samples to localize a single object class. These results also suggest that using object specific context can yield greater localization performance benefits than an object-neutral method like that of the Salience model.

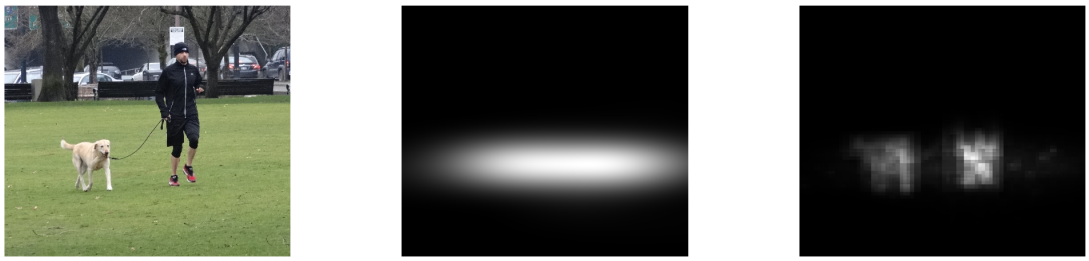
The significant improvement of the Context model over the Uniform model is largely the result of how the Context model constrains the search space over image locations during localization. For instance, in Figure 6.1, the context-driven probability distributions are able to constrain the possible dog locations to a fraction of the total image area.

Figure 6.1: Illustration of Context probability distributions overlaid on three test images. Figure best viewed in color.

The constrained search space and the fact that the distributions typically have high density over actual target object locations, results in the substantial reductions in window samples.

It is for similar reasons that the Combined model is able to further reduce

the mean number of window samples required during localization over that of the Context model alone. For example, Figure 6.2 shows the Context and Combined model probability distributions for the same test image. The Context location distribution in Figure 6.2b has been suppressed in the least salient areas so that the dog location search space has been even further constrained to the most relevant parts of the image.



(a) Test image (b) Context distribution (c) Combined distribution

Figure 6.2: A comparison of the Context and Combined probability distributions for the same test image.

However, the reduction in the object location search space from the Context distribution to the Combined distribution is not nearly as significant as the reduction in going from a uniform distribution to the Context distributions. This explains why the percentage window sample reduction (24.4% and 7.5%) is not nearly as large as that seen for the Context versus the Uniform model (85.8% and 81.8%). While the reduction in window samples for the Combined model over the Context model is relatively small, it is still an interesting result as it suggests that localization models that use probability distributions over image locations can be combined for better localization results.

Even though the Context model presented here performs quite well, it is not without its drawbacks. First, this method depends upon the successful localization of one of the two object classes from which the probability distributions were learned (e.g. dog or dog-walker). This creates a sort of chicken-and-egg dilemma because the localization model can not be applied until one of the objects has been localized. However, it may be possible to first apply the Salience model to initially

localize one of the objects and then use the Context model to substantially reduce the number of window samples required to localize related objects. The end result would still likely be a net-savings in window samples. Exploring this idea is left to future work.

A second drawback of the Context model is that it does a poor job of dealing with outliers. In particular, if the object we are trying to localize lies far outside of the high-density areas of the probability distribution, it is extremely unlikely that we would ever sample that object location from the distribution. For example, the dog in the test image shown in Figure 6.3 was never successfully localized by the Context model during my tests. However, the Uniform model was able to localize this dog successfully within the 5000 window sample threshold. A possible remedy to this issue is to suppress locations that have already been sampled. As the high density areas become suppressed, choosing a location at a low probability location becomes more likely. This type of approach may allow the Context model to more easily localize objects that do not adhere closely to the learned location distributions.

Figure 6.3: Dog outlier. Figure best viewed in color.

Finally, the Context model is designed to leverage the contextual relationships between object classes involved in some type of image situation (e.g., dog-walking,

skateboarding, etc.). If the objects of interest do not have any such relationships, there would likely be little value in using the Context model for that localization task. However, in situations where the objects of interest do have location and size relationships, the results here indicate the Context localization model would perform reasonably well.

Despite the drawbacks present in the Context model, the substantial reduction in window samples for localization over the Uniform and Salience models indicates that there is value in using context-driven probability distributions to improve localization efficiency.

6.2 Future Work

A primary future goal for the work presented in this thesis is the integration of the Context model into the Petacat computer vision system. When presented with a new image, Petacat will attempt to determine if the image is in various situation categories. Assuming the situation category currently being considered is dog-walking, Petacat will first try to localize one of the relevant objects for this situation (e.g., dog, dog-walker, leash, etc.). Once that object is found, other objects belonging to that the dog-walking situation will need to be localized. This is the point of integration for my Context model. At this stage of analysis, the Context model could be applied to efficiently localize the remaining objects belonging to the dog-walking situation, if they are present in the image.

While Petacat is the primary motivation for this work, it can be extended in a number of other ways. For instance, the current Context model requires that the target objects that share a contextual relationship be defined in advance. Ideally, the model would be able to learn these relationships in an unsupervised manner. In doing so, the model could automatically learn pairs, or groups, of objects that share a contextual relationship in a training set and leverage this information for faster localization on test images.

It would be interesting to explore the possibility of using the Context model to

resolve class labelings of candidate windows within an image. For example, if two candidate windows in a test image were classified as a dog and dog-walker, the Context probability distribution over object locations could be applied to determine if windows adhere to typical locations encoded in the distribution. This would make it possible to decrease or increase the confidence in a candidate window proportional to the adherence of the windows to the learned contextual relationships.

Another future step would be to incorporate real window classifiers into the Context model. Currently, the Context model assumes an oracle window classifier. In theory, any window classifier could be used with this model. It is possible that using real window classifiers could further constrain the image search space during localization by updating the probability distributions over image locations based on the window score for each sampled location. Such an implementation may reduce repeated evaluations of locations that are unlikely to contain the object.

Of course, an obvious drawback of the object context models presented in this work is that one of the objects must be localized prior to applying the model. The downside of this is that other methods must be used to localize the first object. Because of this drawback, further investigation on how such context driven models can be incorporated with other localization models that constrain search without the object context (e.g., the salience models) is necessary.

6.3 Conclusions

Using context for object localization has been shown to improve both localization accuracy and efficiency through various approaches [2, 5, 9, 10, 17]. Here, I investigated how the contextual interactions between two objects, dogs and dog-walkers, in a situational relationship can be leveraged to constrain the search space over object location and window size during the localization task. I presented the Context localization model and evaluated it against the Uniform and Salience models. For both dog and dog-walker localization tasks, I have shown that the Context localization model is able to make sharp reductions in the mean number

of windows sampled during localization versus the other models. Additionally, the combination of models using probability distributions over image locations can be combined to yield greater performance benefits than any individual model alone. The ability of the Context model to constrain the object location and size search space to reduce window samples suggests that the contextual interactions between objects can be a valuable tool for boosting localization efficiency.

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, Nicolas Heess, Yee W Teh, and Vittorio Ferrari. Searching for objects driven by context. In *Advances in Neural Information Processing Systems*, pages 881–889, 2012.

[3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[4] Chaitanya Desai, Deva Ramanan, and Charless C Fowlkes. Discriminative models for multi-class object layout. *International Journal of Computer Vision*, 95(1):1–12, 2011.

[5] Lior Elazary and Laurent Itti. A bayesian model for efficient visual search and recognition. *Vision research*, 50(14):1338–1352, 2010.

[6] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[7] Hedi Harzallah, Frédéric Jurie, and Cordelia Schmid. Combining efficient object localization and image classification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 237–244. IEEE, 2009.

[8] Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *Computer Vision–ECCV 2008*, pages 30–43. Springer, 2008.

[9] Derek Hoiem, Alexei A Efros, and Martial Hebert. Geometric context from a single image. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 654–661. IEEE, 2005.

[10] Derek Hoiem, Alexei A Efros, and Martial Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80(1):3–15, 2008.

[11] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[12] MathWorks. imresize: Resize image, mathworks documentation center, 2014. [Online; accessed 26-September-2014].

[13] MathWorks. ksdensity: Kernel smoothing function estimate, mathworks documentation center, 2014. [Online; accessed 7-September-2014].

[14] Melanie Mitchell. A scalable architecture for image interpretation: The petacat project. http://web.cecs.pdx.edu/∼mm/Petacat.html.

[15] Olga Russakovsky, Jia Deng, Zhiheng Huang, Alexander C Berg, and Li Fei-Fei. Detecting avocados to zucchinis: what have we done, and where are we going? In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2064–2071. IEEE, 2013.

[16] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[17] Antonio Torralba, Kevin P Murphy, and William T Freeman. Using the forest to see the trees: exploiting context for visual object detection and localization. *Communications of the ACM*, 53(3):107–114, 2010.

[18] Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman. Multiple kernels for object detection. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 606–613. IEEE, 2009.