19th International Conference on Information Fusion
Heidelberg, Germany - July 5-8, 2016

Active Sampling Exploiting Detector Response Pattern for Efficient Target Detection

Wentao Luan
Institute for System Research
University of Maryland College Park
Email: wluan@umd.edu

Ren Mao
Institute for System Research
University of Maryland College Park
Email: neroam@umd.edu

John S. Baras
Institute for System Research
University of Maryland College Park
Email: baras@umd.edu

***Abstract*—Detecting targets in an image is a fundamental task in computer vision and robotic system. When only a trained detector (binary classifier) is at hand, the target detection problem becomes localizing the correct windows containing targets in the image and can be considered as a sampling problem. Exhaustive sliding window method is a common approach, but it is usually computationally expensive especially when the detection algorithm is time-consuming. In this work, we observe that detector's response scores of sampling windows fade gradually from the peak response window in the detection area and we approximate this scoring pattern with an exponential decay function. By exploiting this property, we propose an active sampling method for efficient target detection to avoid exhaustively searching all the window space. The method estimates the probability of windows containing the target by fusing information from sampled windows and their detector's scores and then decides the next window to be observed. Experiments have shown that our proposed method improves efficiency in human detection applications as it requires fewer windows to achieve similar performance compared to sliding windows and multi-stage particle window (MS-PW) method.**

I. Introduction

With the emerging social demand of robotics automation in both industry and daily life, robotics and computer vision systems have become an important and popular research area. Efficient object detection and recognition are among the most fundamental robot's tasks, on which many subsequent actions such as assembly, fetching, obstacle avoidance rely. Here, the task of object detection can be understood as segmenting the target out from the input image or video and the result can be in the form of a window (i.e. bounding box) or a contour enclosing the target.

The cardinality of search space could be extremely large considering windows of different locations and sizes. Therefore an exhaustive search would be very expensive. Many works in computer vision try to reduce the search space utilizing additional features. For example, segmentation techniques [1], [2] use information such as color, edges and texture similarity to cluster image pixels into super-pixels to avoid a brute-force searching.

In our work, we focus on a typical situation where only a trained detector (binary classifier) is available and we desire to detect the target efficiently from the given input image. The general pipeline of detection includes three steps: window selection, feature abstraction and classification, where the provided detector implements the last two stages. The window selection scheme will determine the detection system's efficiency and quality.

A traditional manner is to slide a window of various sizes over the input image, from left to right, top to bottom and feed image patches to a binary target detector indicating whether the target exists. However, this sliding window method would run the detector a lot of times considering the potentially large number of image windows and it gets even worse when the feature abstraction and classification are complicated. The scanning step size can be increased to speed up but the accuracy will be traded off because the target may be skipped or the windows may not be aligned with the target very well.

To improve this static scanning scheme, one practical way is to regard window selection as a sampling problem, which is to treat the provided vision detector as a black box and sample windows based on the detector's response characteristics. For example, assuming the detector's response score on adjacent windows are similar, multi-stage particle window method (MS-PW) [3] samples windows in stages and follows a "coarse-to-fine" principle.

With the same insight of using the detector's property but going deeper, in this work we propose an active sampling method considering response pattern for efficient target detection. The main contributions of this paper are: 1) We observe that the detector's response pattern of sampling windows in the image follows a "half-ellipsoid" shape in the detection area (i.e. positive classification area). Then an exponential decay function is used to model the response pattern in the positive area. 2) We propose an active sampling approach by exploiting such pattern, which estimates the probability of windows containing the target based on responses of observed windows and then chooses the next window according to posterior sampling. 3) The proposed method is implemented in the application of human detection and experimental results show that our method achieves higher detection rate with the same sampling windows budget and also requires fewer windows with comparable performance when compared with the sliding windows and MS-PW method.

II. Related Work

Efficient target detection has gained much attention and there are many directions of the trial to cut the detection time

978-0-9964527-4-8

while maintaining good detection performance. In general, the attempts in speeding up classification procedure tend to find an early rejection strategy on negative samples, while the work on candidate generation procedures can be summarized as reducing the search space using different sources of information. Also, there is not a clear boundary between classification and candidate proposal. Therefore, these methods can be combined.

An attentional cascade is a classical approach to boost average classification speed, in which the fundamental idea is that background and irrelevant image patches usually occupy the largest portion of all window space and they can be rejected early in the designed cascade classification pipeline. This mechanism achieves good results in applications such as face detection [4] and car detection [5]. Applying a similar idea to reduce the cost of the recognition pipeline, a deformable part model [6] firstly runs a root filter over a downsampled image to filter negative windows out. Andrea *et al*. [7] run an object detector with a linear kernel before using more discriminative but also more time-consuming non-linear kernel ones.

On the other hand, reducing the search space is an approach aiming to reduce the total times of running a target detector, instead of cutting the classification time for each time. Image segmentation is a classical method exploiting low-level information. A common process is to over-segment the image into small boxes, or superpixels, then use graph algorithms, such as minimal spanning tree and graph cut minimization, to build meaningful candidate regions [1], [8], [9]. Selective search [2] generates candidates by hierarchically grouping small regions in a bottom-up manner. Making use of the close contour property of daily objects, the torque operator [10] can provide a reliable source of object candidates and even in high clutter environments [11].

Similar with image segmentation, though more bio-inspired, saliency can be another scheme to speed up detection by imitating human recognition behavior, which always focuses objects standing out of their neighbors pre-attentively. Koch and Ullman [12] firstly put forward a computational attention architecture consisting of the Winner-Take-All network to determine the most salient region, and one of its most well-known derivatives is the Neuromorphic Vision Toolkit [13] proposed by Itti, which is a bottom-up computational attention framework based on the center-surround mechanism of color, intensity and orientations.

Another approach to reduce searching workload is to take advantage of context information. It has attracted more attention recently when incorporated with a sequential decision strategy to optimize the observation path. Gonzalez-Garcia *et al.* [14] adopts context knowledge (a spatial distribution of target) into the windows selection procedure achieving the same detection accuracy with the original region feature convolutional neural network pipeline [15], while using a reduced number of sampling windows. In the indoor environment Nagaraja *et al.* [16] studies structure information such as objects' relative positions to choose the next candidate to observe for target detection. Mnih *et al.* [17] presents a recurrent neural network framework that can decide the next observation region and recognize a target with the same state configuration.

The last category mentioned is window sampling, which our work falls into. It seeks to learn the distribution of the target via sampling the input image. One advantage is avoiding the preprocessing such as edge detection, context analysis, which makes its application general. Multi-stage particle window (MS-PW) [3] samples images iteratively and updates the distribution of the target by a mixture of Gaussians. Pang *et al.* [18] advances MS-PW by classifying observed regions as rejection, ambiguity and acceptance regions based on classifier's response scores. Compared to those attempts, our work applies a distinct way in learning the target distribution which focuses on the detector's response pattern on the positive classification area.

III. Problem Definitions

Given a sensor image I as input, the goal is to find out sampling windows within the image that contain target objects. We describe the center point of sampling window w_i as pixel coordinate (x_i, y_i). As we fix the ratio between length and width of the sampling window according to the property of target detectors, the size of window w_i could be represented as an integer scale level $(s_i = 1, 2, \dots)$ given base size and scale factor. Examples are shown in Fig. 1. Therefore, the complete set of possible sampling windows is defined as $\mathbb{W} = \{w_i | w_i = (x_i, y_i, s_i)\}$.

Fig. 1. Windows of different sizes with the same center point. Base size is 64×128 and scale factor between two scale levels is 1.05. From left to right, the scale levels are 1, 5, 9 and 13.

After selecting the sampling window w_i for the current iteration, the target detector, a binary classifier, takes the corresponding patch from the image I as input and returns a detection score $f(w_i)$ as output, where $f(\cdot)$ depends on the classification algorithms in the detector. The range of such response scores may be different in different detection applications. For instance, the human detection system [19] uses the histogram of oriented gradients (HOG) feature and the SVM classifier, while the detector's response score is a real value which mostly falls into $(-10, 10)$. Score higher than a specified threshold indicates a detection of a target. While

in the case of face detection using Haar-like features and the cascade AdaBoost classifier [4], the detector's response score can be defined as $f(w_i) = l_{w_i}/L$ where l_{w_i} is the largest index of stage returning positive results for input window patch w_i and L is the total number of stages in the cascade classifier. Then, the range of such response is $[0, 1]$.

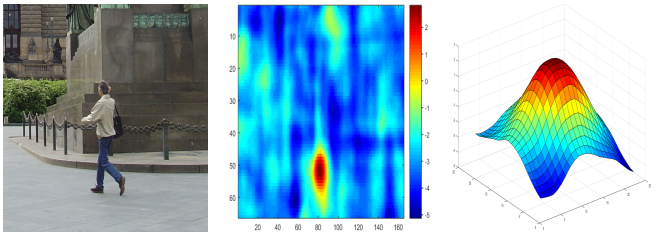
To efficiently detect the target, we aim to sample as small number of windows as possible to reduce the usage of the target detector while maintaining good detection performance, especially when the feature abstraction and classification processes are time-consuming.

IV. Active Sampling with Response Pattern

A. Detector's Response Pattern

To start with, let us look at detector's response pattern with an example in Fig. 2, where (a) is an input image and (b) is the heat map for the response score of human detector [19]. Each point in the heat map represents the center point of a sampling window and all sampling windows are of the same size. Fig. 2(c) explains the pattern of regions that can return positive classification results in 3D. From the figure, we can observe:

- **"Continuity" of Detector's Response Score** The response score of the detector on two nearby windows (same size and close center points) will not change too significantly.
- **Half-ellipsoid Pattern of Detection Area** The red region in the heat map is the detection area that returns positive results if detection threshold is set as 0. By looking at it in 3D, we recognize the overall shape of the detection area is like a half-ellipsoid, which tells that the detector's response score decays gradually with the increment of a window's distance to the peak response window in the detection area.



(a) (b) (c)

Fig. 2. Illustration of response pattern. (a) Input image. (b) Heatmap of human detector's response score. (c) Positive classification region (red area in (b)) in 3D.

Although different detectors (binary classifiers) may have diverse ranges of response score, many of them may still have similar response patterns when the target is not occluded severely. Also, this reaction pattern could be observed in some other target applications though we are focusing on visual object detectors here, thereby the sampling strategy exploiting such pattern can also be applied. Next, we approximate the response decay using an exponential function and utilize this pattern to estimate the probability of an unobserved window containing the target given observed results. Therefore, we can sample windows more efficiently.

B. Formulation

In general, we formulate this process of window sampling for target detection as a Markov Decision Process (MDP).

At iteration t, the fully-observable state consists of all sampled windows and their corresponding detector's response scores $s_t = \{(w_i, f(w_i)), w_i \in \mathbb{W}_e^t\}$, where $\mathbb{W}_e^t$ represents the set of all sampled windows at iteration t. Action a_{t+1}, which is the window to observe at time $t+1$, is selected among all the unexplored windows $\mathbb{W}/\mathbb{W}_e^t$. A binary reward is defined such that the reward is 1 for sampling a window that can return highest local response score in positive classification regions (i.e. $h(w) = 1$ defined in equation (2)) and 0 otherwise.

Our goal of efficient target detection is to minimize the number of total sampling windows while still achieving a certain number of windows containing the target. This could also be considered as maximizing the number of sampled windows that provide a local peak (highest) response in detection area given a constraint on the total number of windows to be checked.

Formally, our objective function is:

$$\begin{aligned} &\text{maximize} \quad |\{w \in \mathbb{W}_e^t | h(w) = 1\}| \\ &\text{subject to} \quad t \leq M. \end{aligned} \tag{1}$$

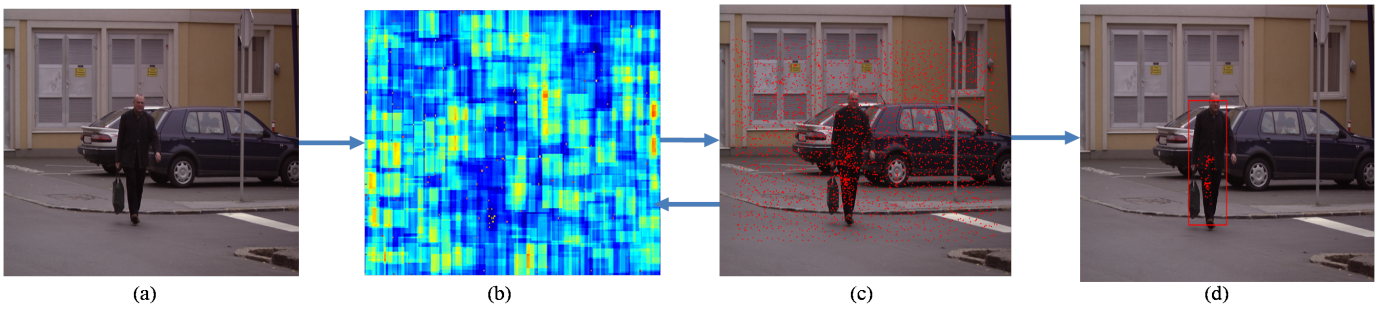
where M is the bound on total iterations and also is the total number of windows to be sampled since only one window would be sampled in each iteration, $|\cdot|$ denotes the cardinality of the set and the function $h(\cdot)$ is an indicator of whether a window has a local maximum response in the detection area:

$$h(w) \triangleq \begin{cases} 1 & \text{If } f(w') \leq f(w) \text{ and } f(w) > \tau \\ & \text{for } \forall w', d(w', w) < \delta \\ 0 & \text{o.w} \end{cases} \tag{2}$$

In (2), τ is a threshold related to the detector that is used to determine positive results. $d(\cdot)$ measures the distance between two windows and $\delta > 0$ is a threshold to determine local neighbors.

Since it is difficult to estimate directly the detector's response score of a selected window patch in each iteration based on observed windows and their scores, i.e., the transition probabilities are unknown, traditional MDP solutions cannot be adopted here. However, through sampling interaction between the input image and the detector's response, it is achievable to learn the distribution of the defined binary reward among unexplored windows. Accordingly, we could maximize our objective rewards according to that estimated distribution.

The overall procedure is demonstrated with an example in Fig. 3. Given an input image Fig. 3(a), we calculate an estimation error (Fig. 3(b)) of each window having local peak response in the detection area based on all the sampled windows (Fig. 3(c)) and their corresponding detector's response score s_t. Then the next window to be tested is chosen according to the posterior sampling on the distribution of



(a) (b) (c) (d)

Fig. 3. System procedure example. (a) Input image. (b) Estimation error heat map of all windows with same scale. (c) The center points of observed windows. (d) Output the positive classified windows.

the binary reward derived from the estimation error obtained above. There is a loop between (b) and (c) because with the newly sampled window and its detector's response added, the reward distribution is reevaluated, and a new window will be selected to be sampled until it achieves the limited total number. Finally, outputs are the positive classified windows (Fig. 3(d)).

In the following sections, we detail on how to evaluate the distribution of the binary reward and how to choose the next action given current observations.

C. Reward Distribution Evaluation

In this section we will elaborate our reward distribution evaluation method. Based on the definition of the binary reward above, the probability of getting a reward 1 is the same as the probability of the selected window returning locally highest response score in the detection area given current observations, i.e. $P(h(w) = 1|s_t)$.

The procedures to calculate the probability $P(h(w) = 1|s_t)$ at window w are as follows. 1) We predict the detector's response score $\hat{f}(w')$ of windows w' that locally surround window w, assuming window w was the peak window in the detection area. This step applies the response pattern that the detector's score exponentially decayed with the increment of distance between a surrounding window w' and the peak response window w. 2) After we observe the response score $f(w')$ for each iteration, we compare it with the predicted one and obtain the prediction error. 3) The prediction errors of all surrounding windows of window w are entered in an energy function, and we evaluate the probability of window w being the local peak window in the detection area.

Formally, given current observation s_t, the probability of a window w being the peak window in the detection area is evaluated as:

$$P(h(w) = 1|s_t) = \frac{1}{Z} \exp(-\sum_{i=1}^{t} E(w_i, f(w_i)|w)) \quad (3)$$

where Z is the normalization factor and the energy function $E(\cdot)$ is defined regarding the error between the observed and predicted detector's response score. The error function is defined as:

$$E(w_i, f(w_i)|w) = \begin{cases} ||f(w_i) - \hat{f}(w_i|w)||^2 & \text{if } w_i \in R(w) \\ 0 & \text{o.w} \end{cases} \quad (4)$$

Here $\hat{f}(\cdot|w)$ is the predicted detector's response function assuming w was the peak response window, and $R(w)$ denotes the influence (cutoff) area for window w.

According to the detector's response pattern observed above, the predicted detector's response could be written as:

$$\hat{f}(w'|w) = C \exp(-(w' - w)^T \Sigma^{-1} (w' - w)) \quad (5)$$

and $\theta = (C, \Sigma^{-1})$ are parameters determining the peak response score and the decaying speed of scores surrounding the peak window.

Given a range for parameter θ, we need to estimate a value best fitting the current observation s_t. The estimation is done by minimizing the prediction error of all observed window patches:

$$\theta^* = \arg \min_{\{\theta: C > \tau\}} \sum_{i=1}^{t} E(w_i, f(w_i)|w, \theta) \quad (6)$$

Finally the predicted detector's score is determined as $\hat{f}(w'|w, \theta^*)$ and the energy function will compare the truly observed response score $f(w_i)$ with the predicted score $\hat{f}(w_i|w, \theta^*)$ to update the probability of window w being the peak window in the detection area.

Even though we determine the maximum likelihood (minimum prediction error) parameter θ^* for all the unexplored windows, the update process can be fast if we restrict to a finite set of values for θ and use the kernel trick. A kernel function based on the observed windows can be defined: $q(w|w_i, f(w_i), \theta) \triangleq E(w_i, f(w_i)|w, \theta)$. The function's value under different θ and $f(w_i)$ can be pre-computed, where we can discretize $f(w_i)$ by binning if it takes a continuous value.

As a result, the probability can be simply estimated through kernel functions:

$$\begin{aligned} -\log P(h(w)=1|s_t) &\propto \min_{\theta} \sum_{i=1}^{t} q(w|w_i, f(w_i), \theta) \\ &= \min_{\theta} \sum_{w_i \in R(w)} q(w|w_i, f(w_i), \theta) \end{aligned} \tag{7}$$

When a new observation $(w_t, f(w_t))$ is made, only the probability of windows within the influence area of w_t: $w \in R(w_t)$ needs to be updated.

D. Active Sampling Action Policy

Given the reward distribution estimated based on the current observed state, we select an unexplored window to be sampled at the next iteration. In order to better balance exploration and exploitation during iterations, *Posterior Sampling* [20] is employed here as our action policy. The key idea of posterior sampling is to instantiate beliefs based on the posterior distribution given current observations in each iteration, then choose an action that can maximize the expected reward.

As the binary reward is gained only when the sampling window w is a peak response window in the detection area and the reward posterior distribution is estimated as described in the previous section, our action policy to select the next sampling window simply becomes:

$$P(A_{t+1} = w|s_t) \propto P(h(w) = 1|s_t) \tag{8}$$

where A_{t+1} denotes the action variable for iteration $t+1$.

Algorithm 1 shows the overall active sampling algorithm.

Algorithm 1: Active Sampling with Response Pattern

Parameters:
Total number of windows to be sampled: M;
Parameters set for prediction functions $\{\hat{f}_i\}$: $\{\theta_i\}$;
Influence region function $R(\cdot)$;
Detection threshold τ.

Input:
Image to be detected: I;
Target detector returning response $f(w)$ with input w.

Output:
Set of sampled windows with postive results: $\mathbb{W}_p$.

Pre-compute / load kernel functions $\{q_i(\cdot)\}$ for all $\{\theta_i\}$
Initialize the prediction error w.r.t each kernel function and minimum prediction error for all the window: $\{E_i(w) = 0\}$, $E^*(w) = 0$
Initialize the probability of each window being locally peak window in detection area: $p(w) = P(h(w) = 1|s_0) = \frac{1}{Z}\exp(-E(w)) = \frac{1}{Z}$
Set: $\mathbb{W}_p = \emptyset$
for $t = 1$ to M **do**
  Sample a window w_t proportionally to $p(w)$
  Observe detector's response $f(w_t)$
  for $\forall w \in R(w_t)$ **do**
    for each kernel function q_i **do**
      $E_i(w) = E_i(w) + q(w|w_t, f(w_t), \theta_i)$
    end for
    Update $E(w)$: $E(w) = \min_{i \in \{1,\ldots,M\}} E_i(w)$
    Update $p(w)$: $p(w) = \frac{1}{Z}\exp(E(w))$
  end for
  if $f(w_t) > \tau$ **then**
    $\mathbb{W}_p = \mathbb{W}_p \cup \{w_t\}$
  end if
end for

V. Experiments

In this section, we evaluate our sampling method with Multi-Stage Particle Windows sampling (MS-PW) [3] to demonstrate that our proposed method obtains better efficiency while maintaining good detection performance through exploiting the detector's response pattern.

MS-PW is chosen as a comparison method because both methods detect targets only by sampling and using the detector's response without adopting other pre-processing techniques such as segmentation [2], [9].

We test the algorithms' performance via several evaluation metrics including detection rate, window usage efficiency, the average precision rate given the same budget and overall system detection performance using different sampling window budget.

A. Dataset and Settings

We assess our sampling method on the INRIA person dataset [19]. The training set contains 1208 cropped person patches for positive examples and 1218 non-person images where negative example patches can be sampled from. In the testing set, there are 453 images of scenery and buildings without people and 288 images containing one or more persons. Most people in testing images are standing, but they appear in different orientations and various backgrounds such as shops, statues and pillars. In this work we are addressing a detection problem, so that full images in the testing dataset are used to evaluate our algorithm's performance. A SVM classifier trained with HOG features is employed as the human detector, which takes input images of 64×128 pixels.

In all experiments, we set our influential region $R(w)$ as a cube of size $21 \times 31 \times 5$ pixels (width, height, scale) centered at observed window w. According to the observed detector's response pattern, we restrict the prediction function $\hat{f}(\cdot)$ to the set of parameters $\{\theta_1, \theta_2\}$: $(C_1, \Sigma_1^{-1}) = (1.2, diag(10, 20, 5))$ and $(C_2, \Sigma_2^{-1}) = (2.2, diag(25, 35, 5))$. Fig. 4 illustrates our prediction function using parameters θ_1, θ_2.

The ratio between width and height of each window is fixed as $\frac{1}{2}$ according to the detector's input requirements. And the scaling factor for the window size of two adjacent levels is set as 1.05. Meanwhile, the total number of possible sampling windows varies with different sizes of input images. We denote

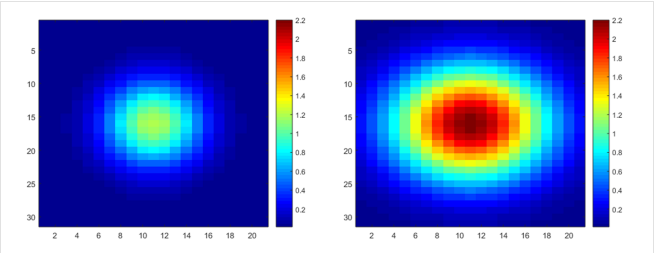


Fig. 4. The heat map of predicted score with our two settings of parameters. Each point corresponds to a window in the same size with the selected observed window.

N_{sw} as the total number of sliding windows when we scan images both vertically and horizontally with a stride of 8. Then we limit the total number of windows to be sampled in experiments proportional to N_{sw}.

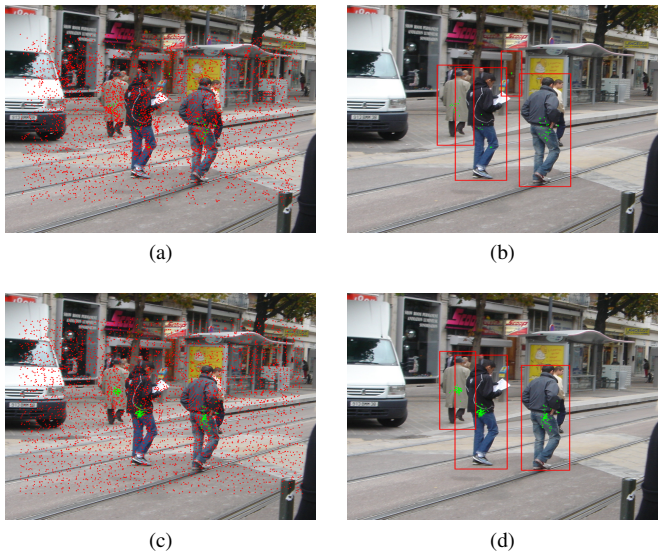
B. Experimental Results

The first experiment compares the detection rate under the same false positive rate per image (FPPI = 1) between MS-PW and our method. Here the false positive rate is measured per image instead of per window because we allow multiple targets detected in one picture, even though the latter one is the standard metric for traditional classification problems. The outcome is shown in Table I top, from where we can notice that with the same budget number of windows to be sampled, our method has higher detection rate and hits more windows with positive results than MS-PW.

	# of win.	Method	Detection Rate	# of Positive Windows
FPPI = 1	$1/7N_{sw}$	MS-PW	0.624	10.4
		Our	0.716	18.3
	$1/6N_{sw}$	MS-PW	0.650	11.4
		Our	0.718	23.7
	$1/5N_{sw}$	MS-PW	0.652	12.0
		Our	0.721	31.0
	$1/4N_{sw}$	MS-PW	0.667	14.1
		Our	0.725	42.3
	$1/3N_{sw}$	MS-PW	0.691	17.5
		Our	0.726	57.6
	$1/2N_{sw}$	MS-PW	0.708	24.0
		Our	0.728	75.2
$\tau = 0$	$1/7N_{sw}$	MS-PW	0.587	7.0
		Our	0.677	15.4
	$1/6N_{sw}$	MS-PW	0.596	8.1
		Our	0.681	20.6
	$1/5N_{sw}$	MS-PW	0.604	9.1
		Our	0.688	28.0
	$1/4N_{sw}$	MS-PW	0.652	11.8
		Our	0.713	38.6
	$1/3N_{sw}$	MS-PW	0.684	15.6
		Our	0.714	53.5
	$1/2N_{sw}$	MS-PW	0.708	23.7
		Our	0.719	72.5

TABLE I
DETECTION PERFORMANCE WITH SAME SAMPLING WINDOW BUDGET. TOP: CLASSIFICATION THRESHOLD CUSTOMIZED TO FPPI = 1. BOTTOM: CLASSIFICATION THRESHOLD $\tau = 0$

The second experiment contrasts sampling efficiency between methods, i.e., the number of sampled windows with positive detection results per image using the same number of total sampling windows. The detector (binary classifier)'s threshold is identical ($\tau = 0$) for fair comparison. The consequence is displayed in Table I bottom. It is evident that our method can discover more positive windows and achieve higher detection rate than MS-PW.



(a) (b) (c) (d)

Fig. 5. The qualitative result of sampling the same number of windows. Top row: MS-PW. Bottom row: Our method. Left column: The center points of the windows selected by each method (both red and green dots). Right column: The center points of positive classified windows sampled (green dots).

An intuitive explanation of these results would come from the qualitative comparison in Fig. 5, where our method exhibits better performance in locating windows containing targets when sampling the same number of windows and classifying with the same threshold.

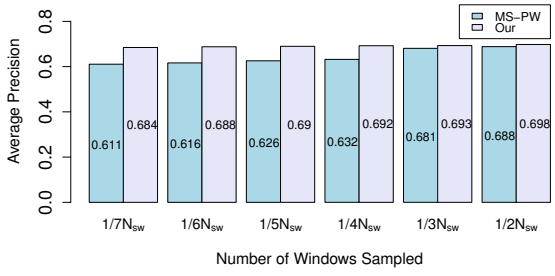


Fig. 6. Average precision rate of two methods with same window budget

Meanwhile, Fig. 6 demonstrates the average precision rate of system's performance in retrieving targets from images under different sampling budget. Although the average precision of MS-PW method increases along with the sampling budget, our method remains favorable because of better performance for all budgets. More interestingly, our method could hold a relatively high average precision rate when the budget number

is small. This suggests our approach properly exploits the detector's response pattern and facilitates sample efficiency.

In the last experiment, we examine system's detection performance using *Detection Error Tradeoff (DET)* curves, which represent how missing rate (1 - detection rate) changes with the false positive rate per image (FPPI). Performance using sliding window method with N_{sw} budget windows (scanning step = 8) is also shown as a baseline. Results in Fig. 7 reveal similar DET curves when we set windows budgets for our method and MS-PW as $1/7N_{sw}$ and $1/3N_{sw}$. The results mean that to achieve the same detection performance with the sliding windows method, our method only uses $1/7$ of the total windows which outperforms MS-PW that needs $1/3$.

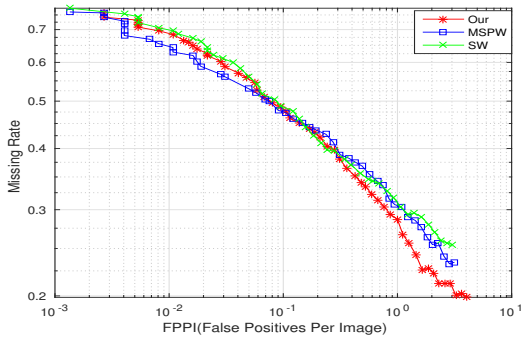


Fig. 7. DET curve of MSPW, SW and our method

VI. Conclusions And Future Directions

In this work, we present a method of active sampling with response pattern to detect targets efficiently in a visual image. The proposed method exploits the detector's response pattern to avoid an expensive, exhaustive searching for targets. An exponential decay function is used to model the pattern of detection score in the positive classification region. By comparing the predicted response score and the observed one, we estimate the probability of an unobserved window containing targets and having locally maximum response. Based on that, posterior sampling is applied to decide the next window to observe. Experimental results on human detection show that our approach can achieve higher detection rate than the MS-PW method using the same total windows budget, and also requires less number of windows to achieve similar detection performance compared to the sliding window and MS-PW methods.

In the future, we will consider integrating this sampling method with other search space reduction algorithms such as segmentation or saliency-based image processing techniques to achieve better target detection performance. Also, we may investigate other action policy strategies such as information-directed sampling [21], so that we can further incorporate the potential information gain of sampling each window into our reward evaluation to improve the balance between exploitation and exploration during detection iterations.

Acknowledgment

Research supported in part by DARPA (through ARO) grant W911NF1410384 and by NSF grant CNS-1544787.

References

[1] B. Peng, L. Zhang, and D. Zhang, "A survey of graph theoretical approaches to image segmentation," 2012.

[2] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[3] G. Gualdi, A. Prati, and R. Cucchiara, "Multistage particle windows for fast and accurate object detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 8, pp. 1589–1604, Aug 2012.

[4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–511–I–518 vol.1.

[5] P. Negri, X. Clady, S. M. Hanif, and L. Prevost, "A cascade of boosted generative and discriminative classifiers for vehicle detection," *EURASIP J. Adv. Sig. Proc.*, vol. 2008.

[6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, Sept 2010.

[7] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.

[8] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[9] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Computer Vision and Pattern Recognition*, 2014.

[10] M. Nishigaki, C. Fermuller, and D. DeMenthon, "The image torque operator: A new tool for mid-level vision," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 502–509.

[11] C. Teo, A. Myers, C. Fermuller, and Y. Aloimonos, "Embedding high-level information into low level vision: Efficient object search in clutter," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 126–132.

[12] C. Koch and S. Ullman, "Shifts in selective visual attention: Towards the underlying neural circuitry," in *Matters of Intelligence*, ser. Synthese Library, L. Vaina, Ed. Springer Netherlands, 1987, vol. 188, pp. 115–141.

[13] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 11, pp. 1254–1259, Nov 1998.

[14] A. Gonzalez-Garcia, A. Vezhnevets, and V. Ferrari, "An active search strategy for efficient object detection," *CoRR*, vol. abs/1412.3709, 2014.

[15] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013.

[16] V. K. Nagaraja, V. I. Morariu, and L. S. Davis, "Searching for objects using structure in indoor scenes," in *British Machine Vision Conference (BMVC)*, 2015.

[17] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," *CoRR*, vol. abs/1406.6247, 2014.

[18] Y. Pang, J. Cao, and X. Li, "Learning sampling functions for efficient object detection," *CoRR*, vol. abs/1508.05581, 2015.

[19] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision & Pattern Recognition*, C. Schmid, S. Soatto, and C. Tomasi, Eds., vol. 2, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005, pp. 886–893.

[20] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 2249–2257.

[21] D. Russo and B. V. Roy, "Learning to optimize via information directed sampling," *CoRR*, vol. abs/1403.5556, 2014.