Efficient Bayes Saliency-Based Object Detection on Images Using Deep Belief Networks

Ratnababu Mamidi [a], Merchant S.N[b]

[a] Professor, Department of ECE, Rise Krishna Sai prakasam group of institutions, Valluru, Ongole, Andhra Pradesh, India
[b]Professor, Electrical Engineering Department, Indian Institute of Technology Bombay, Powai, Mumbai, Maharashtra, India

Abstract: Object detection has been one of the hottest issues in the field of remote sensing image analysis. The purpose of object detection is to find precise locations of the objects, one location at a time or all locations of all objects in an image. However, most current object detection methods developed earlier demonstrate unsatisfactory results. Therefore, this paper presents efficient Bayes saliency-based object detection on images using deep belief networks. First, a new Bayes saliency detection approach is presented in which prior estimation, feature extraction, weight estimation, and Bayes rule are used to compute saliency maps. In particular, an efficient coarse object locating method is used based on a saliency mechanism. Then, an efficient object detection framework is implemented which combines the unsupervised feature learning of Deep Belief Networks (DBNs) and visual saliency. After that, the trained DBN is used for feature extraction and classification on sub images. The method could avoid an exhaustive search across the image and generate a small number of bounding boxes, which can locate the object quickly and precisely. Comparative experiments are conducted on the dataset and result analysis demonstrate that the accuracy and efficiency of our method than state of-the-art methods in terms of various evaluation metrics. Furthermore, this object proposal approach can improve the detection performance and the speed of several detection approaches.

Keywords: Object detection, Bayes saliency detection, coarse object locating, DBNs

1. Introduction

Object detection is a method used to find precise locations of the objects. It can be applied in commercial applications, for examples, real-time video analytics, customer behavior analysis, face detection, traffic analysis and autonomous vehicle system [1]. Fast and accurate are major subjects of object detection which is hard to achieve. Due to variations in poses, lighting, occlusion, size of the objects, it is difficult to accomplish real-time multiple object detection. Salient object detection aimed at stimulating the visual characteristics of human beings and extracts the most attractive regions from images or videos [2]. The content in these saliency areas is what called as salient objects. Since saliency detection is a relatively basic task that can increase computational efficiency, it has played an important role in many fields of computer vision, such as foreground extraction, visual tracking, scene classification, semantic segmentation, video summarization and image retrieval [3].

Over the years, there have been various methods of salient object detection. These methods can be divided into two categories: conventional methods and deep learning based methods. Conventional methods typically use low-level visual information of an image to predict saliency maps, such as varies heuristic priors, color and contrast information, and so on [4]. However, for images with low color contrast and complex background, conventional methods have difficulty in accurately identifying these obvious objects. Because conventional methods lack the ability to capture high level and global semantic features of the object, they cannot accurately detect these objects that attract people but are not obvious in visual characteristics. Traditional saliency methods aim to generate a heat map which gives each pixel a relative value of its level of saliency. In recent years, the fashion moves to salient object detection which generates pixel-wise binary label for salient and non-salient object. In comparing with the heat map, the binary label would further benefit segmentation based applications such as semantic segmentation, and thus attracts more attention [5].

A novel and fast Bayes saliency detection which detects salient pixels within regions of vehicles is presented. In this letter, an efficient coarse object locating method based on a saliency mechanism is introduced, which can generate a small number of bounding boxes as object candidates. An effective object detection approach is present to relate the saliency map with object detection and generate models with high detection rate. A block wise training strategy is implemented to pertain the restricted Boltzmann machines (RBMs), which combines the raw pixels with a saliency map as input. The strategy leads the RBM to generate local and edge filters. Thus, the higher layer of RBMs could get more opportunities to extract the spatial and configuration information of the object.

2. Literature Survey

Semantic relationships between different objects or regions of an image can help detect occluded and small objects. Bae et al. [6] utilize the combined and high-level semantic features for object classification and localization which combine the multi-region features stage by stage. Zhang et al. [7] combine a semantic segmentation branch and a global activation module to enrich the semantics of object detection features within a typical deep detector. Scene contextual relations can provide some useful information for accurate visual recognition. Liu et al. [8] adopt scene contextual information to further improve accuracy. Modeling relations between objects can help object detection. Singh et al. [9] process context regions around the ground-truth object on an appropriate scale. Hu et al. [10] propose a relation module that processes a set of objects simultaneously considering both appearance and geometry features through interaction. Mid-level semantic properties of objects can benefit object detection containing visual attributes.

In recent decades, deep learning has achieved excellent performance in almost all areas of computer vision. Since this method can fully explore the high-level semantic information of an image, it is very popular in salient object detection. Many CNN-based models attempt to learn the deep semantic properties of salient objects, which can further improve the prediction performance. Ishikura et al. [11] detected potentially salient regions based on multiscale extreme of local and global perceived color differences measured in the CIELAB color space. Hou et al. proposed a short connection [12] based on HED to solve the scale-space problem. However, they simply integrated high-level features with low-level features, which cannot extract features effectively. They combined the CNN method with traditional methods, and learned multi-scale depth features through CNN to obtain high-quality visual saliency maps.

As convolutional neural networks become deeper and deeper, Kaiming He et al. in [13] showed how their accuracy first saturates and then degrades rapidly. They proposed the use of residual learning to the stacked layers to mitigate the performance decay. It is realized by addition of a skip connection between the layers. This connection is an element wise addition between input and output of the block and does not add extra parameter or computational complexity to the network. A typical 34 layer ResNet is basically a large (7x7) convolution filter followed by 16 bottleneck modules (pair of small 3x3 filters with identity shortcut across them) and ultimately a fully connected layer. The bottleneck architecture can be adapted for deeper networks by stacking 3 convolutional layers (1x1,3x3,1x3) instead of 2. Long et al. [14] proposed a Fully Convolutional Neural Network (FCN) to predict the semantic label for each pixel which was the first end-to-end network that predicted pixels and used supervised pretraining. With the implementation of FCN, more and more pixel wise saliency detection methods have appeared. Olaf et al. [15] presented the U-Net by adding skip connections between the corresponding layers of the decoder and encoder on the FCN.

3. Bayes Saliency-based Object Detection Using DBNS

In this section, the ideas of the presented method and implementation details will be described. The Bayes saliency-based object detection framework using DBNS is shown in Fig. 1. The framework can be divided into two stages, i.e., the training stage and the Bayes saliency detection stage. The training contains the unsupervised feature learning and the discriminative fine-tuning. A disjoint image collection is used as the training set to train a DBN. At the detection stage i.e. testing stage initially background prior, which assumes that the area in the narrow border of the image belongs to the background, performs very robust in many detection tasks. So we use background prior to determine the background location information in images. Meantime, features are extracted and map which robustly represent the attributes of objects within the image. To compute the condition likelihood functions of the observed background and potential salient pixels, following steps are utilized. To divide all pixels into background set and salient pixel set, the prior map is binarized by searching the optimal threshold which separates the background from the possible salient pixels in each feature map. Weights of each feature for each class (background and salient pixels) are estimated using variance of feature of each class. The conditional probabilities are computed by multiple probabilities of each feature with weights. Using Bayes rule, the posterior probability of a pixel belong to the salient pixels (i.e., the saliency map) is obtained. Then a coarse object locating method is operated on the test images to generate a small number of bounding boxes as object candidates. Finally, the sub images are classified by the DBN. The feature learning of the DBN is operated by pretraining each layer of restricted Boltzmann machines (RBMs) using the general layerwise training algorithm. An unsupervised blockwise pretraining strategy is introduced to train the first layer of RBMs, which combines the raw pixels with a saliency map as inputs. This makes an RBM generate local and edge filters.

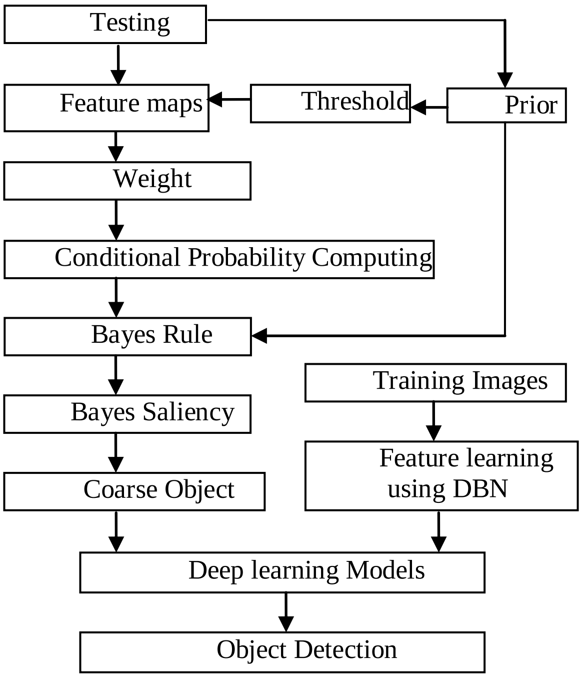


Fig. 1: BAYES SALIENCY-BASED OBJECT DETECTION FRAMEWORK USING DBNS

3.1 Bayes Saliency Detection

1) Prior Estimation: It is known that the low brightness traffic images have low lightness and low contrast, edges are noticeable in salient objects. The surrounding pixels of edges are more likely to belong to salient objects. To estimate priors, the edge map of the input traffic image is first computed which can extract accurate edge maps. Considering the surrounding pixels of edges, the edge map is calibrated by convoluting with an average filter. In this paper Gaussian filter design is used. The calibrated edge map P(x, y) is normalized to the range [0, 1] and vectorized to final estimated prior probability of a pixel belonging to the salient pixels (p(s)). According to the probability theory, the probability of a pixel being background pixel (p(b)) is equal to $1 - p(s)$. It can find that the prior estimation is effective because the pixels in the two objects are white (close to 1) while most of background is black. Note that the prior map in this method does not have to be very accurate. It is enough for the prior map to include the most edges of vehicles and provide information on the rough locations of the vehicles, since the inaccuracy of the prior map, for example, of a relatively large vehicle, will be corrected as much as possible by our other strategies described below, for example, the three types of features, the Bayes rule and multi-scale sliding window based object detection method.

2) Feature Extraction: The salient objects in low brightness often have bright regions and the contrast of object regions is often higher than the contrast of background regions. Besides, at night vehicles often turn on the back light which is very salient and useful for distinguishing vehicles from the background. Thus, to represent the differences between the salient vehicles and the background pixels, three types of features: luminance, local contrast and vehicle taillight map are extracted. The luminance image is computed as the average of the R, G, B channels of the input image. The local contrast of each pixel is computed as the normalized variance of gray values within a $S_w = 7 \times 7$ window centered at this pixel. The luminance and local contrast features are smoothed using an Gaussian average filter and they are also normalized to [0,1]. The computation process of the vehicle back light map is first convert the RGB low brightness images to intensity images and reduce noise using an empirical threshold. Second, the Nakagami images are estimated by utilizing a sliding window mechanism. Subsequently, possible vehicle back light is detected by the optimal thresholds.

3) Thresholding: After computing the Nakagami images of all positive samples in the dataset, low-level and high-level thresholds close to the maximum and minimum values respectively are tried to segment the Nakagami images. The values between the low- and high-level thresholds are set to 0 and other values are set to 1. If a pair of thresholds can detect all back light (most of pixels within taillights are retained) in all positive samples and detect minimum non- back light regions, choose them as the optimal thresholds. Then, the red pixel rate of each pixel is computed: the rate of the number of red pixels in a $Sw = 7 \times 7$ window centered at this pixel and the number of all pixels in the window, and the thresholding result is multiplied by the red pixel rate to obtain the coarse vehicle back light map. Because the pixels which are close to the detected taillight regions are more likely to be salient pixels than the pixels which are far away from the detected taillight regions, the origin vehicle taillight map is convoluted with a Gaussian filter ($S_f = 21 \times 21$) to compute the final vehicle taillight map flight which is normalized to [0,1]. During thresholding the weights of each feature are equal and in the following step the weights of each feature are updated.

4) Weight Computing: Different features demonstrate various abilities to distinguish salient pixels from background pixels. Therefore, it is required to consider the importance of each feature of each class when computing the likelihood of the observed salient pixels and background pixels. Robust features of the salient pixels should contain stable feature values (i.e., small variation). Then the weights of each feature are computed for the salient or background pixels as

$$w_i^k = \frac{1}{\mu}\sqrt{\frac{1-e^{1-var_i^k}}{1-e}},\ k\epsilon\{-1,1\} \quad \text{----- (1)}$$

where -1 and 1 denote background pixels and salient pixels respectively, $\mu = \sum_i w_i^k, i\epsilon\{f_{\text{lum}}, f_{\text{con}}, f_{\text{light}}\}, var_i^k$ is the normalized variance ($\in$ [0, 1]) of feature values of the k class pixels obtained after thresholding in the i feature channel. The weights are self-adapting.

5) Conditional Likelihood Computation: After estimating the possible salient pixels and background pixels of each feature map and computing the weight of each feature for each class, the conditional likelihood of observed salient and background pixel sets can be computed. The joint probability distribution of various features can be expressed as the product of all individual probability distributions with the conditional independence assumption between any two features. Thus, the observation likelihood at pixel x can be computed as

$$p(x|s) = \prod_i p\left(x_i \middle| s_{T_{opt}}\right)^{w_i^1} \quad \text{--------- (2)}$$

$$p(x|b) = \prod_i p\left(x_i \middle| B_{T_{opt}}\right)^{w_i^{-1}} \quad \text{--------- (3)}$$

where $p\left(x_i \middle| s_{T_{opt}}\right)$ and $p\left(x_i \middle| B_{T_{opt}}\right)$ are respectively the distribution functions of each feature in the salient object and background regions. In detail, observation likelihood of each feature is computed as the normalized histogram distribution of pixels in salient or background regions in each features map.

6) Saliency Map Based on Bayes Rule: With Bayes rule, the possibility of a pixel belonging to salient pixels (i.e., posterior probability and saliency map) can be computed as

$$p(s|x)\frac{p(s)p(x|s)}{p(s)p(x|s)+p(b)p(x|b)} \quad \text{----- (4)}$$

Where $p(s)$ and $p(b)$ are computed during prior estimation step and the prior probabilities of a pixel belonging to salient pixels and the background, $p(x|s)$ and $p(x|b)$ are conditional likelihood functions based on the observed salient pixels and the background obtained using (2) and (3) respectively. Bayes rule has been applied to the saliency detection before where multiple simple features are combined using Bayes rule for saliency detection on daytime images. Afterwards, we obtain the Bayes saliency map of each image.

7) Coarse Object Locating: Object locating is started by generating a saliency map of the test image. It should be emphasized that the norm gradient map will be used as the saliency map. After that, the initial windows are set non-overlapped across the saliency map. As a consequence, at least one initial window will contain most part of the object. It is quite possible that the window that contains part of the object will move to the precise location of the object using this method. Finally, the overlapped windows that contain the same object will be fused into one window. In this manner, the number of windows can be reduced significantly, and the object locating precision is improved, which will benefit the process of feature extraction. The method can not only increase the accuracy of detection but also enhance the robustness of different detection methods.

3.2 Training stage

1) Feature Learning using DBN: The training samples of images are considered here. All the training images are collected from a disjoint QuickBird image collection with a 0.6-m resolution. The images are normalized to a size of 36 × 36 before training. Our model contains DBN with layers. The visible layer contains the raw image and the saliency map units. In experiments, the visible units of every RBM are set to real values, which are in the range of [0, 1] for logistic units. While training higher level RBMs, the visible units are set to the output values of the hidden units in the previous RBM.

In general, only one single RBM is trained for each layer of the DBN, and the pretraining is operated by setting the visible units of RBMs to each pixel value of the image directly. This always leads the extracted features to be global features. In this part, a block wise strategy is introduced for pretraining the RBM. The training is started by segmenting the two images pixel patches. Independent RBMs with hidden units are trained on each of these tiny image patches. After that, they are combined in a very straightforward way by training a new big RBM with hidden units. The weights of the first hidden units are initialized with the weights from the small RBM trained on patch 1, the weights of the next hidden units are initialized with the weights trained on patch 2, and so forth. However, each of the hidden units in the big RBM is connected to each pixel in image; thus, the weights that did not exist in the RBMs trained on patches are initialized to 0. The bias of the visible unit of the big RBM is initialized to the average bias that the unit received among the small RBMs.

3.3 Object Detection: After the unsupervised pretraining, a supervised layer is added to the top of the DBN to build a classifier. The probability distribution of the layer is defined as follows:

$$P(class = j) = \frac{e^{f_j(x)}}{\sum_k e^{f_k(x)}} \quad \text{------ (5)}$$

This is also known as softmax regression. $P(class = j)$ is the probability that the data are assigned to class j. $f_k(x) = w_k x + b_k$ is the function of the model. At the fine-tuning stage, the backpropagation algorithm is used to fine-tune the whole network until convergence. After training the deep model, detection is conducted for test images. In the coarse object locating stage, the size of initial windows is set to 80 × 80 pixels. After object locating, the subimages of the candidate windows are normalized to a size of 36 pixels × 36 pixels. Then, raw pixels and saliency maps are classified by the deep model. Finally, the predicted labels are generated from the supervised layer.

4. Results

The method presented in this work is termed as BGM-DBN (a norm gradient map and raw pixels are fused by the DBN using the blockwise pretraining strategy), which is compared with classification by a Support Vector Machine (SVM) using raw pixels as features (denoted as RP-SVM), classification by a DBN using raw pixels as features (denoted as RP-DBN), and the method where a norm gradient map and raw pixels are fused by a DBN (denoted as GM-DBN).

Detection is marked as true positive (TP) if more than 75% of the ground truth is covered. In order to quantify the performance, two commonly used criteria were computed, i.e. the TP rate (TPR) and the false positive rate (FPR). They are defined as follows:

$$TPR = TP/(TP + FN)$$
$$FPR = FP/(FP + TN)$$

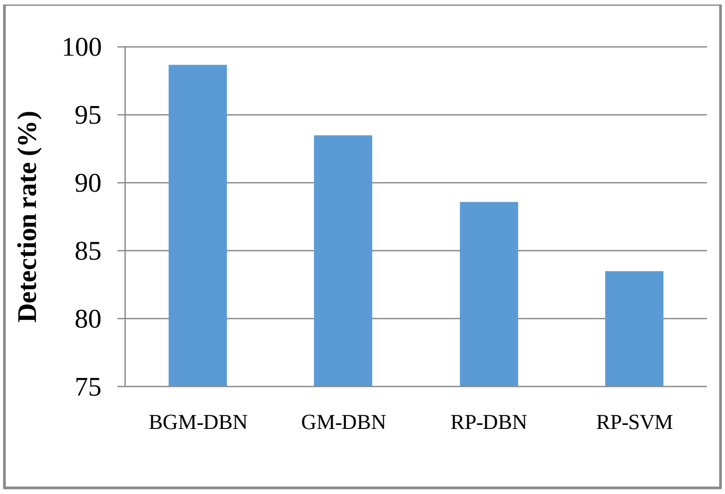
The object detection results are shown in Table 1 and demonstrate the performance of different methods. It shows the detection performance using the sliding window method and proposed Bayes Saliency object detection methods with different DBN feature learning approaches. One can see that the proposed method achieves the best performance both in the TPR and the FPR. All the results are significantly improved compared with the results of the corresponding methods, which demonstrate that the detection performance really benefits from the advantages of the proposed locating method.

Table. 1. Object Detection Results

Methods	**Sliding window method**		**Bayes Saliency method**	
	TRP	FPR	TRP	FPR
RP-SVM	0.79	0.54	0.83	0.46
RP-DBN	0.85	0.39	0.87	0.35
GM-DBN	0.89	0.32	0.94	0.26
BGM-DBN	0.91	0.18	0.97	0.12

The detection rate is the ratio of the number of objects detected accurately and the number of all objects. The comparison of detection rate is shown in figure 3. The presented method in this paper obtains the highest detection rate, which means our method is better than these state-of- the-art methods.

Fig. 2. Comparative Analysis On Object Detection Rate



The results demonstrate that our method is more robust to the variance both in the object and the background.

5. Conclusion

This paper presents a novel object detection framework for low brightness traffic images which combine the Bayes saliency map with DBN feature learning based object detection. In the novel Bayes saliency detection approach, Bayes rule is used to integrate multiple features and compute the probability of a pixel belonging to salient pixels, and weights of each feature for each class are carefully designed. In addition the coarse object locating method is introduced by using a visual saliency prior, which locates the object more precisely and more quickly. Then object detection framework has been implemented using a Bayes saliency map and DBNs. The DBN was trained using the general unsupervised layer wise pretraining. The norm gradient map and the raw pixels of the image were utilized as training samples. The proposed blockwise pretraining strategy improves the generalization ability of the features. The sub images were then classified by the deep model. The experimental results showed that the proposed Bayes saliency detection and BGM-DBN object detection achieved better detection performance.

References (APA)

[1] B. Zoph, E. D. Cubuk, G. Ghiasi, T. Lin, J. Shlens, and Q. V. Le, "Learning data augmentation strategies for object detection," CoRR, vol. abs/1906.11172, 2019

[2] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," arXiv preprint arXiv:1904.08739, 2019.

[3] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," arXiv preprint arXiv:1904.09569, 2019

[4] Wang Fan, Peng Guohua, "Salient Object Detection via Quaternionic Local Ranking Binary Pattern and High-Level Priors", 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC), 2019

[5] Wenji Wang, Haogang Zhu, "Learning Adversarially Enhanced Heatmaps for Aorta Segmentation in CTA", 2019 IEEE International Conference on Imaging Systems and Techniques (IST), 2019

[6] S.-H. Bae, "Object detection based on region decomposition and assembly," arXiv preprint arXiv:1901.08225, 2019.

[7] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4203–4212, 2018.

[8] Y. Liu, R. Wang, S. Shan, and X. Chen, "Structure inference net: Object detection using scene-level context and instance-level relationships," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6985–6994, 2018.

[9] B. Singh, M. Najibi, and L. S. Davis, "Sniper: Efficient multi-scale training," in Advances in Neural Information Processing Systems, pp. 9310–9320, 2018.

[10] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3588–3597, 2018.

[11] K. Ishikura, N. Kurita, D. M. Chandler, and G. Ohashi, "Saliency detection based on multiscale extrema of local perceptual color differences," IEEE Trans. Image Process., vol. 27, no. 2, pp. 703–717, Feb. 2018.

[12] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, “Deeply supervised salient object detection with short connections,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 3203–3212.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778

[14] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 3431–3440.

[15] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent., 2015, pp. 234–241.