Salient Object Recognition Using Label Propagation and Saliency Maps

Arya Jayakumar, Priyadarsini .S, Leena Chacko

Department of Computer Engineering, College of Engineering Chengannur, aryajayakumar92@gmail.com
9526224461

Abstract— In this paper an object recognition method using saliency maps is proposed. Saliency maps are used to find the most informative part in an image. Label Propagation Saliency (LPS) method is used to generate saliency maps on the basis of background and objectness labels extracted from the image. Object recognition can be performed on saliency maps using Principal Component Analysis (PCA) and k-Nearest Neighbor search (k-NN). During training phase 357 images are trained by applying PCA to retrieve the key features and they are stored in database. During testing phase PCA is applied on the input image to retrieve features. The stored values are loaded and a kd-Tree is constructed and searching is done on it which trains the features into different categories based on Euclidean distance. kd-Tree search values and PCA values of input image are given as input to k-NN algorithm to find the category of input image.

Keywords— Saliency Maps, Background labels, Objectness labels, Label Propagation Saliency(LPS), Principal Component analysis(PCA), kd-Tree, k-Nearest Neighbor(k-NN)

INTRODUCTION

Object recognition is considered as an important application of image processing. Object recognition is a process of identifying specific objects in an image. Object recognition can be effectively computed using saliency maps. Image saliency detection aims to effectively identify important and informative regions in images. Salient object detection is typically accomplished by image contrast computation, either on a local or a global scale estimates the saliency degree of an image region by computing the contrast against its local neighborhood. Various contrast measures have been proposed, including mutual information, incremental coding length and center- versus-surround feature discrepancy.

It has been verified that color contrast is a primary cue for satisfying results .Other representations based on the low-level features try to exploit the intrinsic textural difference between the foreground and background, including focusness ,textual distinctiveness, and structure descriptor . They perform well in many cases, but can still struggle in complex images.

In the work done by [1], viewed the problem from a different viewpoint: It focused more on the background instead of the object. They exploited two common priors about backgrounds in natural images, namely boundary and connectivity priors, to provide more clues for the problem. Accordingly a novel saliency measure called geodesic saliencies proposed. It is intuitive, easy to interpret and allows fast implementation. Furthermore, it is complementary to previous approaches, because it benefits more from background priors while previous approaches do not.

In [2] a unified approach to incorporate low-level features and the objectness measure for saliency detection via label propagation is proposed. Since the border regions of the image are good indicators to distinguish salient objects from the background ,we observe that the boundary cues can be used to estimate the appearance of the background while the objectness cues focus on the characteristics of the salient object. Therefore, a refined co-transduction [3] based method, namely label propagation saliency (LPS), is proposed. In this framework, the most certain boundary and object regions are able to propagate saliency information in order to best leverage their complementary influence. As the boundary cue can be quite effective in some cases and the objectness measure requires additional computation, a compactness criterion is further devised to determine whether the results propagated by boundary labels are sufficient.

Object recognition is performed on saliency maps using Principal Component Analysis (PCA) and k-Nearest Neighbor algorithm (k-NN). During the training phase image features are extracted using PCA. During the testing phase, a kd-Tree is constructed from the stored values during the training phase. PCA is applied on test image to extract the key features .This kd-Tree along with the key features of test image are given as input to k-NN algorithm. It gives exact classification and class of test image.

RELATED WORK

Due to the absence of high level knowledge, all bottom up saliency methods rely on assumptions or priors1 on the properties of objects and backgrounds. Arguably, the most fundamental assumption is that "appearance contrast between object and background is high". This assumption states that a salient image pixel/patch presents high contrast within a certain context. It is intuitive and used in all saliency methods, explicitly or implicitly. In geodesic saliency it is called as contrast prior for conciseness.

Contrast prior based methods have achieved success in their own aspects, but still have certain limitations. Typically, the boundaries of the salient object can be found well, but the object interior is attenuated. This "object attenuation" problem is observed in all local methods and some global methods. It is alleviated in global methods [4], [5], but these methods still have difficulties of highlighting the entire object uniformly.

In essence, the saliency object detection problem on general objects and background is highly ill-posed. There still lacks a common definition of "what saliency is" in the community and simply using contrast prior alone is unlikely to succeed. While previous approaches are mostly based on their own understanding of "how the contrast prior should be implemented" huge behavioral discrepancies between previous methods can be observed, and such discrepancies are sometimes hard to understand.

Jiang et al.[6] introduce an absorbing Markov chain method where the appearance divergence and spatial distribution between salient objects and the background are considered. Cheng et al. [7] formulate a regional contrast based saliency algorithm which simultaneously evaluates global and local contrast differences. Inspired by these works, in LPS an affinity matrix is constructed based on the color feature of superpixels with two adjustments to involve spatial relations.

A novel label propagation method is proposed in [8] to rank the similarity of data points to the query labels for shape retrieval. We apply and refine the theory to make full use of the background and foreground superpixels, which has been rarely studied in saliency detection. Distinct from the work of Yang et al. [9] where a manifold ranking algorithm assigns saliency based on priors of all boundary nodes, in this work,(a) we only take some boundary nodes to eliminate salient regions that appear at the image border; (b) both boundary and foreground nodes are selected as complementary labels in a co-transduction framework to fully distinguish salient areas from the background; and (c) the revised label propagation algorithm has zero parameter whereas in [9] the sensitive α has a vital effect on results in different datasets.

PCA- Based Feature Extraction and k-NN algorithm for detection of jaundice in children was proposed in[12].The proposed jaundice detection algorithm localizes the face from the given input image using the Haar Classifer method were employed. The detected face image is projected using Eigen face analysis and classified using the k nearest neighborhood (k-NN).

PROPOSED METHOD

In this object recognition method two phases are included Training phase and Testing phase to correctly classify the object into corresponding class.

1. TRAINING PHASE:

Images are trained to extract their key features in the training phase. 357 images of different category are used. At first saliency detection is performed on each of training images and then Principal Component Analysis(PCA) is used to extract the key features. PCA helps to extract those values which show maximum variance in the shape of saliency image and the data is projected to a new dimension of maximum variance that best discriminates the data. Saliency detection for each image is performed using Label Propagation Saliency (LPS) method and this saliency map is used for object recognition.

1.1 Label Propagation Saliency (LPS)

LPS mainly uses background and objectness labels to propagate their information and find saliency maps from the propagated information. First an affinity matrix is constructed among superpixels to be used in the propagation algorithm. L0 gradient minimization [10] is implemented to obtain a soft abstraction layer while keeping vital details of the image. Superpixels are generated to segment the smoothed image into N regions by the SLIC algorithm [11], where regions at the image border form a set of boundary nodes, denoted as B. In this work, we refer the superpixel as a node or a region.

The similarity of two nodes is measured by a defined distance of the mean features in each region. Based on the intuition that neighboring regions are likely to share similar appearances and that remote ones do not bother to have similar saliency values even if the appearance of them are highly identical, we define the affinity entry w_{ij} of superpixel i to a certain node j as the difference between mean of features between two nodes.

Some background nodes are selected as boundary nodes and they are set as query labels to propagate saliency information on the basis of affinity matrix. This method is called inner propagation.

Given an affinity matrix, we endeavor to propagate the information of the background labels to estimate saliency measure of other superpixels. The similarity measure V (r_i) satisfies.

$$V_{t+1}(r_i) = \sum_{j=1}^{N} a_{ij}\ V_t(r_j) \qquad (1)$$

Algorithm 1 Inner Label Propagation via Boundary Nodes

Input: The N× N row –wise normalized color affinity matrix A_c.The set of selected boundary labels B' and the set of unlabelled *U={R\B'}.*

1. t = 0
2. Initialize, set $V_t(r_i)$=1 for $r_i \in B'$ and $V_t(r_i)$= 0 for $r_i \in U$
3. **while** *check* > *thres* do
4. **for** $r_i \in U$ **do**
5. $V_{t+1}(r_i) = \sum_{j=1}^{N} a_{ij}\ V_t(r_j)$
6. **end for**
7. *t = t+1*
8. *check* = *var* $(\boldsymbol{V_t}, \boldsymbol{V_{t-1}} \ldots\ldots\ldots\ldots, \boldsymbol{V_{t-const}})$
9. **end while**
10. $\boldsymbol{S}^B$ = ones (*N*) – normalize ($\boldsymbol{V_t}$)
11. $\boldsymbol{S}^B(r_i)$ = sp2map (S^B)

Output: The regional map .$S^B(r_i)$ from background labels.

Algorithm 1 summarizes the inner label propagation via boundary nodes. The convergence of the similarity measure V is ensured by checking whether its average variance in the last 50 iterations (i.e., const = 49) is below a threshold. sp2map(.) means mapping the saliency measures of N regions into an image-size map.

In most cases, the inner propagation with help of the boundary labels works well whereas in some complex scenes, depending on the boundary prior alone might lead to high saliency assignment to the background regions. It naturally suggests us to use some foreground prior to improve the results further.

In some cases, the inner propagation via boundary labels alone has better saliency maps than a combination of boundary and objectness labels, which results from the slight disturbance of objectness measures near the salient object. A compactness score is used to evaluate the quality of regional saliency map $\boldsymbol{S}^B(r_i)$ generated by Algorithm 1.

$$C(S) = \sum_{b=1}^{10} w(b).\, h^s(b) \qquad (2)$$

b denotes each quantization of the resultant saliency map, $h^s(b)$indicates a 10 bin histogram distribution of the map and w(b) indicates the weight upon each bin. Based on the aforementioned characteristic of the failure saliency maps in the inner boundary propagation, we take a triangle form of the weight term, ie. w(b) = min(b,(11-b)).Only the saliency maps with score lower than a compactness score(1.6) will be updated by the inter propagation via a co-transduction algorithm. Such a scheme not only ensures high quality of the saliency maps, but also improves the computational efficiency.

Due to shortcomings of background priors in case of complex images objectness priors are introduced. Several useful priors are exploited and combined in a Bayesian framework, including multi-scale saliency (MS), color-contrast (CC), edge density (ED). MS measures the uniqueness of objects according to the spectral residual of the image's FFT. CC considers the distinct appearance of objects via a center-surround histogram of color distribution. ED capture the closed boundary of objects. It computes the density of edges near window borders. These cues are combined independently in a naive Bayes model.

Let $p_m$be a probability score of the m-th sampling window, the pixel-level objectness map O (p) is obtained through overlapping scores multiplied by the Gaussian smoothing kernel of all sampling windows.

$$\mathrm{O}\,(\mathrm{p}) = \sum_{m=1}^{M} p_m .\ \exp\left[-\left(\frac{(\mathrm{x_p}-\mathrm{x_m^c})^2}{2\sigma_\mathrm{x}^2} + \frac{(\mathrm{y_p}-\mathrm{y_m^c})^2}{2\sigma_\mathrm{y}^2} \right)\right] \quad (3)$$

where M = 1000 is the number of sampling windows x_p ,y_p,x_m^c,y_m^cdenote the coordinates of pixel p and the center coordinates of window m respectively. σ_x= .25W and σ_y= .25 H, where W is the width and H the height of an image. The region-level objectness map O (r_i) is the average of pixels objectness values within a region:

$$\mathrm{O}\,(r_i) = \frac{1}{n_i} \sum_{p \in r_i} O_p \quad (4)$$

Where n_i indicates number of pixels in region r_i .A simple average of pixels scores within a region leads to mid-value saliency in vast background areas since the pixel-level map from which the region-level map is generated is ambiguous around the salient object in the first place.

Based on the fact that high values of region-level objectness score calculated by Eqn.4 can better indicate foreground areas, the set of objectness labels O is created from superpixels whose region-level objectness O (r_i) is no less than the objectness criterion. Thus a complementary combination of the boundary and objectness labels could be a better choice.

A new co-transduction algorithm for saliency detection is implemented which uses one label set to pull out confident data and add additional labels as new hints to the other label set. The inter label propagation algorithm is summarized in Algorithm 2. During each iteration in Algorithm 2 (through line 11 to 16), $p_1$superpixels which are most different from the boundary labels are picked out and added to the objectness set and the update of the boundary set is similarly achieved with a different superpixel number p_2.We set $p_1, p_2$to be $p_1 << p_2$ because the background regions often significantly outnumber the foreground ones. Final saliency measure is computed as a linear combination of the resultant $S^B$and S^Oin the last iteration from boundary and objectness labels respectively.

The inter propagation algorithm strengthens the connection of salient regions by employing objectness labels and distinguishes the foreground better from the background by enlarging the set of boundary labels from objectness cues, thus best leveraging the complimentary information of both label sets.

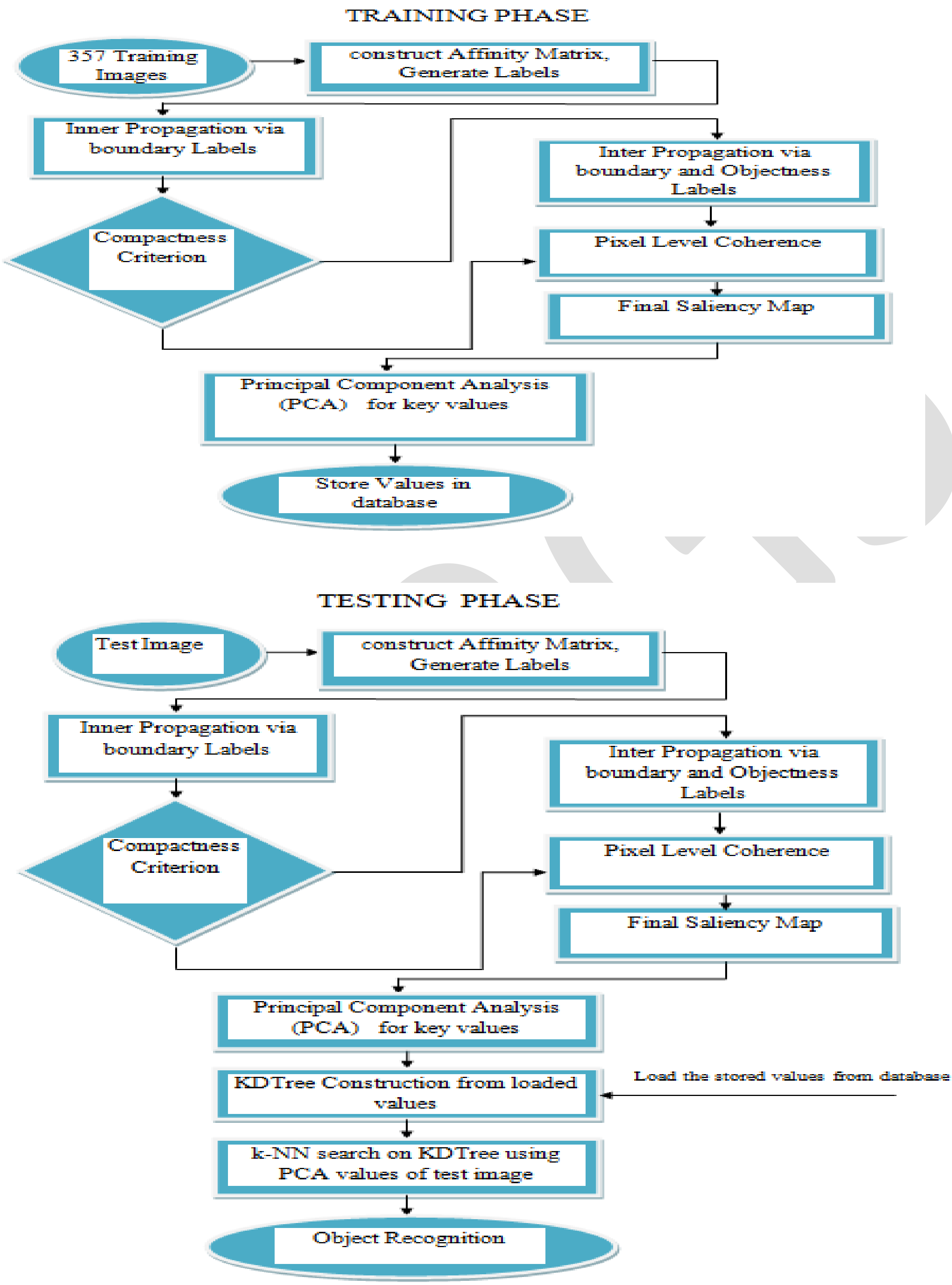


Fig:1 Block Diagram of Proposed System

Algorithm 2 Inter Label Propagation via Boundary and Objectness Nodes

Input: The N ×N row-wise normalized color affinity matrix Ac. The set of selected boundary labels B0 and the set of objectness labels O.

1. t = 0
2. Initialize V_t^B= 0, V_t^O= 0
3. **while** $check^B$,$check^O$> thres do
4. set $V_t^B(r_i)$=1 for $r_i \in B^{'}$ and $V_t^O(r_i)$=1for $r_i \in O$
5. Create unlabelled sets $U_1$and U_2such that U_1= {R\\$B^{'}$},$U_2$= {R\O}
6. **for** $r_i \in U_1, r_i \in U_2$ **do**
7. $V_{t+1}^B(r_i) = \sum_{j=1}^{N} a_{ij}\ \ V_t^B(r_j)$
8. $V_{t+1}^O(r_i) = \sum_{j=1}^{N} a_{ij}\ \ V_t^O(r_j)$
9. end **for**
10. t = t + 1
11. $check^B\ = var(V_t^B$,……. $V_{t-const}^B)$
12. $check^O\ = var(V_t^O$,……. $V_{t-const}^O)$
13. temp1= sort (V_t^B ,'ascent')
14. temp2= sort (V_t^O ,'ascent')
15. L^B= temp1(1: p_1) , L^O= temp2(1: p_2)
16. $B^{'}$ = $B^{'} \cap L^O$, O = $O \cap L^B$
17. end **while**
18. S^B = ones(N) – normalize(V_t^B)
19. S^O = normalize(V_t^O)
20. S^c = normalize (αS^B +βS^O)
21. $S^c(r_i)$ = sp2map (S^c)

Output: The combined regional saliency map $S^c(r_i)$.

In order to eliminate the segmentation errors of the SLIC algorithm, pixel-level saliency is defined as a weighted linear combination of the regional saliency S^B (r_i) or S^c (r_i) of its surrounding superpixels.

$$S(p) = \sum_{i=1}^{G} \exp(-(k_1 \| c_p - c_i \| + k_2 \| z_p - z_i \|))\ S^{B/C}(r_i) \quad (5)$$

Where c_p, c_i, z_p, z_i are the color and coordinate vectors of a region or a pixel, G denotes the number of direct neighbors of region $r_i$and S^B, S^c indicates the straightforward region level result descending from Algorithm 1 or Algorithm 2. By choosing a Gaussian weight it is ensured that the up sampling process is both local and color sensitive.k_1 and $k_2$are parameters controlling the sensitivity to color and position, where k_1= 0.2 and k_2 =0.01.

1.2 Principal Component Analysis (PCA)

PCA is applied on all the 357 saliency maps produced from the corresponding training images to extract the key values. Key values produced as a result of application of PCA along 50 dimensions are stored on database along with category labels.PCA can be used to find the maximum variance in the original space.PCA is used to extract feature vector and reduce the dimensions of process data.PCA is a technique that can be used to simplify a dataset. It is a linear transformation that chooses a new coordinate system for the data set such that variance by any projection of the data set lie on the first axis (first principal component), the second variance on the second axis, and so on. PCA can be used for reducing dimensionality by eliminating the later principal components. Principal component analysis provides a road map for how to reduce a complex data set to a lower dimension. Thus dimensionality reduction is also performed which helps classification works faster. The important steps of PCA are:

1. Take the whole training images consisting of d-dimensional samples ignoring the class labels.
2. Compute the d-dimensional mean vector (i.e., the means for every dimension of the whole training images.
3. Compute eigenvectors ($e_1, e_2 \ldots. e_d$) and corresponding eigenvalues ($\lambda_{1,}\lambda_2 \ldots \lambda_d$). Eigen values are a product of multiplying matrices however they are as special case. Eigen values are found by multiples of the covariance matrix by a vector in 2 dimensional space.
4. Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a d × k dimensional matrix.
5. Use this d × k eigenvector matrix to transform the samples onto the new subspace.

2. TESTING PHASE

When an test image is given as input saliency map is constructed from that image using the Label Propagation Saliency (LPS) methods used in training phase. Principal Component Analysis (PCA) is applied on test image to extract key features of input image. Object values extracted from training images that are stored in database are loaded in this phase. These loaded features are given as input to kd-Tree which trains the features into different categories based on Euclidean distance.

2.1 kd-Tree Construction

The kd-Tree is a generalization of the simple binary tree used for sorting and searching. The kd-Tree is a binary tree in which each node represents a subfile of the records in the file and a partitioning of that subfile. The root of the tree represents the entire file. Each non terminal node has two sons or successor nodes. These successor nodes represent the two sub files defined by the partitioning. The terminal nodes represent mutually exclusive small subsets of the data records, which collectively form a partition of the record space.

In the case of one-dimensional searching, a record is represented by a single key and a partition is defined by some value of that key. All records in a subfile with key values less than or equal to the partition value belong to the left son, while those with a larger value belong to the right son. The keg variable thus becomes a discriminator for assigning records to the two sub files.

In k dimensions, a record is represented by k keys. Any one of these can serve as the discriminator for partitioning the subfile represented by a particular node in the tree; that is, the discriminating key number can range from 1 to k. The discriminator for each level is obtained by cycling through the keys in order. That is, d= L mod k + l where d is the discriminating key number for level L and the root node is defined to be at level zero. The partition values are chosen to be random key values in each particular subfile.

The k-d tree data structure provides an efficient mechanism for examining only those records closest to the query record, thereby greatly reducing the computation required to find the best matches.

The search algorithm is most easily described as a recursive procedure. The argument to the procedure is the node under investigation. The first invocation passes the root of the tree as this argument. Available as a global array is the domain of that node; that is, the geometric boundaries delimiting the subfile represented by the node. The domain of the root node is defined to be plus and minus infinity on all keys. These geometric boundaries are determined by the partitions defined at the nodes above it in the tree. At each node, the partition not only divides the current subfile, but it also defines a lower or upper limit on the value of the discriminator key for each record in the two new subfiles. The accrual of these limits in the ancestors of any node defines a cell in the multidimensional record-key space containing its subfile. The volume of this cell is smaller for subfiles defined by nodes deeper in the tree. If the node under investigation is terminal, then all the records in the bucket are examined. A list of the m closest records so far encountered and their dissimilarity to the query record is always maintained as a priority queue during the search. Whenever a record is examined and found to be closer than the most distant member of this list, the list is updated. If the node under investigation is not terminal, the recursive procedure is called for the node representing the subfile on the same side of the partition as the query record. When control returns, a test is made to determine if it is necessary to consider the records on the side of the partition opposite the query record. It is necessary to consider that subfile only if the geometric boundaries limiting those records overlap the ball centered at the query record with radius equal to the dissimilarity to the m^{th}closest record so far encountered. This is referred to as the "bounds-overlap-ball" test. If the bounds-overlap ball test fails, then none of the records on the opposite side of the partition can be among the m closest records to the query record. If the bounds do overlap the ball, then the records of that subtree must be considered and the procedure is called recursively for the node representing that subfile. A "ball-within-bounds" test is made before returning to determine if it is necessary to continue the search. This test determines whether the ball is entirely within the geometric domain of the node. If so, the current list of m best matches is correct for the entire file and no more records need be examined.

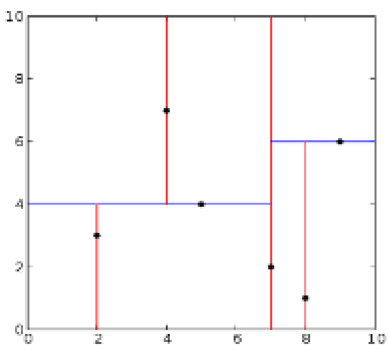


Fig:2 kd-Tree decomposition for the point set

(2,3), (5,4), (9,6), (4,7), (8,1), (7,2)

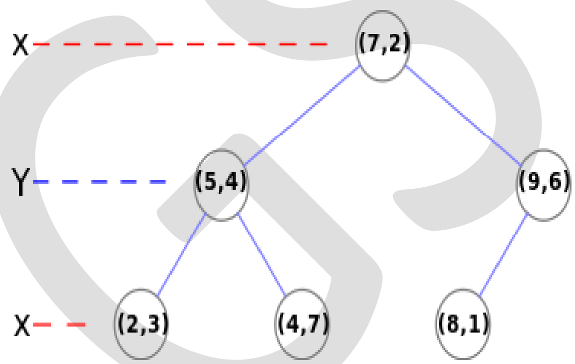


Fig: 3 The resulting k-d tree

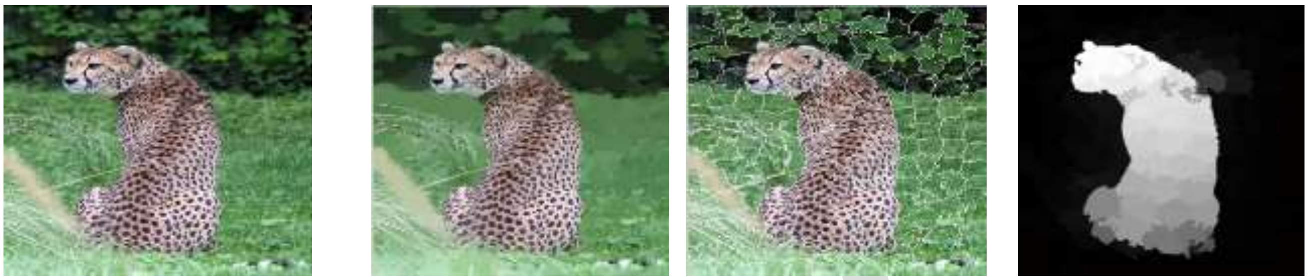
2.2 k-Nearest Neighbor Search(k-NN)

Once a kd- Tree searcher model object is created, the stored tree can be searched to find all neighboring points to the query data by performing a nearest neighbors search using k-NN search. The nearest neighbor search (NN) algorithm aims to find the point in the tree that is nearest to a given input point. This search can be done efficiently by using the tree properties to quickly eliminate large portions of the search space. Searching for a nearest neighbor in a kd-Tree proceeds as follows:

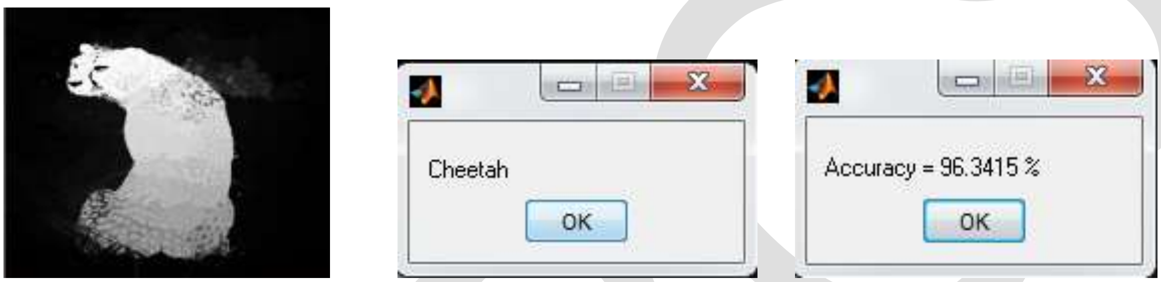
1. A positive integer k is specified, along with a new sample. It can provide the k nearest neighbors to a sample by maintaining k current bests instead of just one. A branch is only eliminated when k points have been found and the branch cannot have points closer than any of the k current bests.
2. Starting with the root node, the algorithm moves down the tree recursively, in the same way that it would if the search point were being inserted (i.e. it goes left or right depending on whether the sample value of test image is lesser than or greater than the current node in the split dimension).
3. Select the k entries in our database (nodes of kd-Tree) which are closest to the new sample.
4. Find the most common classification of these entries. This is the classification we give to the new sample

EXPERIMENTAL RESULTS

The proposed method is evaluated on cmu-cornell icoseg dataset. 357 images are used for training and more than 150 images are tested. The results produced shows that the saliency map produced from the method is of high quality. PCA gives the exact key features which helps to discriminate classes. kd-Tree helps to reduce the execution time. The execution of the entire process is estimated to be 30 seconds which is possible with help of fast searching using kd-Tree. k-NN correctly classifies the test image. The entire process is evaluated and accuracy of overall process is estimated to be 96.3415%. The results of the entire process is shown in Fig:3.



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



(e) **(f)** **(g)**

Fig 4: Results of object recognition method (a) Input Image (b) L0 Smoothing (c) SLIC segmentation (d) LPS method of saliency detection (e) Pixel Level Coherence (f) Detection from saliency map (g) Accuracy of method

CONCLUSION

An object recognition method from saliency maps is proposed. Saliency maps are generated using Label Propagation saliency method (LPS). In this method background and objectness labels are extracted from the SLIC segmented image. These labels are used to propagate the similarity of other segments to these labels. On the basis of the LPS algorithm the salient portion in an image is extracted. This saliency detection is performed on all the test images during the testing phase and then PCA is used to extract key features and they are stored in database along with class labels. During testing phase the stored values are loaded and a kd-Tree is constructed from the values and then k-Nearest Neighbor(k-NN) algorithm is applied on kd-Tree to get the correct classification of test image and hence recognize the class of test image. Performance evaluation shows the method works well in the case of almost all images.

REFERENCES:

[1] Y. Wei, F. Wen, W. Zhu, and J. Sun, “Geodesic saliency using background priors,” in ECCV, 2012.

[2] Hongyang Li,Huchuan Lu, Zhe Lin, Xiaohui Shen, Brian Price ‘Inner and Inter Label Propagation :Salient Object Detection in the Wild", IEEE TRANSACTIONS ON IMAGE PROCESSING, August 2015.

[3] X. Bai, B. Wang, C. Yao, W. Liu, and Z. Tu, “Co-transduction for shape retrieval,” IEEE Trans. Image Proc., vol. 21, no. 5, pp. 2747–2757, 2012.

[4] Goferman, S.,Manor, L.,Tal,A.”Context-aware saliency detection.” in CVPR(2010).

[5] Cheng, M., Zhang, G., Mitra, N., Huang, X., Hu, S.: Global contrast based salient region detection. In: CVPR (2011).

[6] B. Jiang, L. Zhang, H. Lu, and M.-H. Yang, "Saliency detection via absorbing markov chain." in ICCV, 2013.

[7] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu,"Global contrast based salient region detection," in CVPR, 2011.

[8] X. Bai, X. Yang, L. J. Latecki, W. Liu, and Z. Tu, "Learning contextsensitive shape similarity by graph transduction," IEEE Trans. PAMI, vol. 32, no. 5, pp. 861–874, 2010.

[9] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in CVPR, 2013.

[10] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L0 gradient minimization," ACM Trans. Graph., Dec. 2011.

[11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. S¨usstrunk,"SLIC superpixels," EPFL, Tech. Rep. 149300, June 2010.

[12] Muhammad Naufal Mansor, Sazali Yaacob, Hariharan Muthusamy, "PCA- Based Feature Extraction and k-NN algorithm for Early Jaundice Detection", JSCSE, Vol.1,No.1,2011

[13] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in NIPS,2007

[14] K. Shi, K. Wang, J. Lu, and L. Lin, "PISA: Pixelwise image saliency by aggregating complementary appearance contrast measures with spatialpriors," in CVPR, 2013.