TITLE: An Approach for segmentation of Image using merge tree

Miss.Vrushali Vijay Jadhav
Prof.Swati A.Bhavsar

[1] *Student, Department of Computer EngineeringMatoshree college of engineering,Nashik. Maharashtra, India*
[2] *Professor, Department of Computer EngineeringMatoshree college of engineering,Nashik. Maharashtra, India*

ABSTRACT

The system investigates one amongst the foremost elementary pc vision problem that is image segmentation. It is a semi supervised hierarchal approach to object-independent image segmentation as a contribution. System begins with over segmenting super pixels, after over segmenting the super pixels tree structure is used to represent the hierarchy of region merging, according to that reduce back the problem of segmenting image regions to finding a group of label assignment to tree nodes and to show system anytime which can be long .Tree structure Formulation is as a constrained conditional model to associate region merging with likelihoods predicted using an assemble Boundary classifier .Final segmentations will then be achieved by finding globally best solutions to the model with efficiency, also it associate iterative work and testing formula that generates varied tree structures and combines them to correct boundaries by segmentation accumulation.

Keyword—*Image segmentation, hierarchal merge tree, constrained conditional model, supervised classification, object-independent, ensemble model.*

INTRODUCTION

Image segmentation is very essential midlevel computer vision drawback that has been studied for a prolonged time however stays difficult. Trendy Image segmentation is used as a pre-processing step for resolution excessive-level imaginative and prescient troubles, like visual perception and image classification. in any other-disciplinary regions, e.g., organic and medical imaging, picture segmentation additionally plays a large position in supporting scientists quantify and examine picture statistics. While lots of analysis has been achieved to realize excessive segmentation accuracy for particular styles of images, the usual of photograph segmentation for fashionable scenes remains work, introduce a semi-incredible wise getting to know Much less than Nice. At some point of this primarily based photograph segmentation framework, specifically, the hierarchical merge tree model starting with over-segmenting remarkable pixels, advice to represent the vicinity merging hierarchy with a tree like restrained conditional version. An ensemble boundary classifier is trained to reap every don't forget the graphical model. a globally most useful label task to the model Miscomputed by minimizing the complete strength below the place consistency constraint and a very last segmentation is recovered from the labeling. Additionally endorse an iterative technique that generates numerous vicinity merging hierarchies and combines them to boost the general overall performance through segmentation accumulation. By way of lowering the window of the pixel. Contribution of this previous work is as a way to develop semi supervised machine in an effort to reduce the value of the device and no mandatory schooling will constantly be required. Also window length can be decreased for extract boundary pixel on the way to deliver higher segmented output.

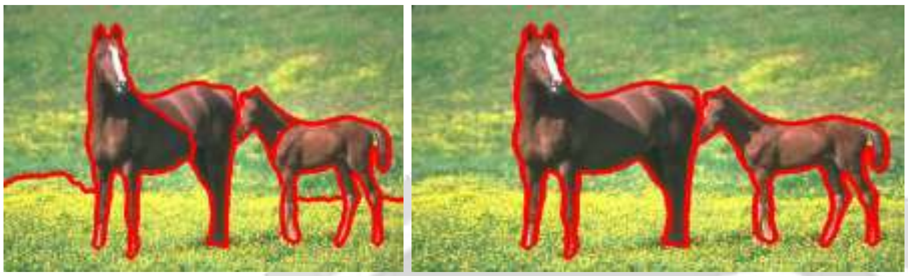
REVIEW OF LITERATURE

There are 2 totally different views of image segmentation [1]. One is edge detection that aims at finding edges between totally different perceptual pixel groups. The other one is region segmentation that partitions an image into disjoint regions. Usually, edge detection focuses on assigning a binary label to every pixel with certain confidence indicating if it belongs to an edge or not and doesn't guarantee closed object contours. Though closed contours and therefore regions they encircle are recovered from edges, such transformation with high accuracy is sometimes non-trivial. On the other hand, region segmentation seeks to search out the cluster membership of every pixel, and closed contours of an object are trivially generated because the outer points of an area. Several region segmentation ways also benefit of the edge detection outputs as boundary cues to help with the search for correct partitioning. Our technique belongs to the region segmentation category, and during this section emphasizes reviewing previous related works during this category. one amongst the most notable works, gPb [1], combines multi-scale native cues and globalized cues via spectral clustering and setup a benchmark for edge detection and region segmentation analysis. Taking advantage of supervised learning techniques has also become the recent trend in edge detection.Xiaofeng and Bo [6] train a classifier with sparse code son local neighborhood data and improve the edge detection performance. Dollár and Zitnick [7] propose structured learning framework using changed random call forest for efficient edge detection. Seyedhosseiniand Tasdizen [8] propose a hierarchical model to capture multi-scale discourse data and deliver the goods progressive edge detection performance. Early works on region segmentation get to directly cluster image pixels in an unsupervised manner. Belonged et al. [9] fit Gaussian mixture models to cluster pixels supported six-dimensional color and texture options. Mean shift [10] and its variant [11] take into account region segmentation as a density mode looking drawback. variety of works belong to graph partitioning category, that regards an image as a graph with pixels being nodes and edge weights indicating unsimilarity between neighbor pixels. Normalized cuts [12] take the image affinity matrix and partition an image by finding eigenvalue issues. Felzenszwalb and Huttenlocher [13] propose to greedily merge 2 connected elements if there exists anointer-component edge weight that's less than the biggest edge weights within the minimum spanning trees of each elements.Arbeláez et al. [1] propose a variant of watershed rework to get a hierarchy of closed contours. Refer readers to [14] for a comprehensive review of existing ways. As in edge detection, supervised learning based methods for region segmentation have gained enhanced quality indecent years. This trend results in and is additional promoted bay variety of in public accessible computer vision data sets with human-labeled ground truth [1], [15]–[19]. though unsupervised ways, like [20] and [21], are shown to generate perceptually coherent segmentations, learning segmentation models from supervised data enables much more capability and flexibility of incorporating preference from human observers and leads to many more interesting works. Following the classic foreground/background segmentation, object-independent segmentation methods seek to partition an image based only on its appearance and do not utilize underlying semantics about the scene or specific information about target objects. Kim et al. propose a hypergraph-basedcorrelation clustering framework [22] that uses structured SVM for learning the structural information from training data. Arbelaez et al. develop the multi-scale combinatorial grouping (MCG) framework [23] that exploits multi-scale information and uses a fast normalized cuts algorithm for region segmentation. Yu et al. [24] present a piecewise flat embedding learning algorithm and report the best published results so far on Berkeley Segmentation Data Set using the MCGframework. Two other recent super pixel-merging approaches are ISCRA [25] and GALA [26]. Starting with a finesuperpixel over-segmentation, ISCRA adaptively divides the whole region merging process into different cascaded stages and trains a respective logistic regression model at each stage to determine the greedy merging. Meanwhile, GALA improves the boundary classifier training by augmenting the training set via repeatedly iterating through the merging process. Moreover, impressive results in the extensive evaluations on six public segmentation data sets are reported in [25].Object-dependent or semantic segmentation is another branch of region segmentation. Object-dependent prior knowledge is exploited to guide or improve the segmentation process. Borenstein and Ullman [27] formulate object segmentation as a joint model that uses each low-level visual cues and high-level object category data. another object segmentation ways 1st generate object segmentation hypotheses using low-/mid-level options and so rank segments with high-level previous information [28], [29].A recent work, surgical knife [30], incorporates high-level data within the segmentation method and might generate object proposals additional with efficiency and accurately. There are also a group of ways, known as co-segmentation, that utilizes the homogeneity between totally different target objects and together segments multiple images at the same time [31]–[33].Our technique falls into the object-independent hirarchical segmentation class. A preliminary version of our technique with the merge tree model and a greedy illation algorithmic program appeared in [34] and [35] and was solely applied to segmenting microscopy pictures, except for that the contributions of this paper include:

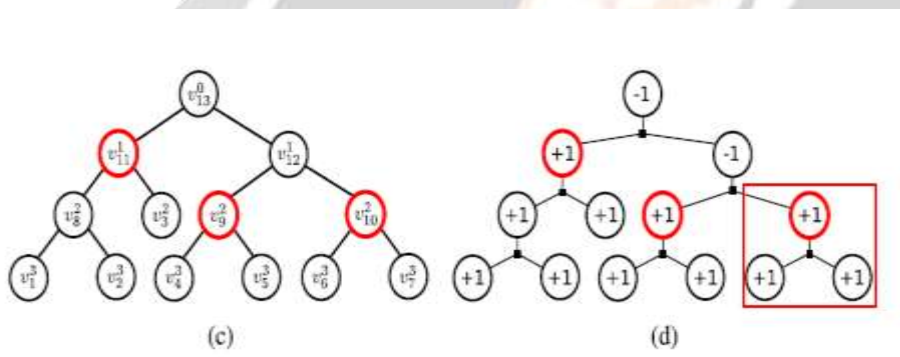
• Reformulation of the ranked merge tree as a constrained conditional model with globally optimum solutions outlined associate degreed an economical illation algorithmic program developed, rather than the greedy tree model in [34] and [35].
• Associate degree unvaried approach to diversify merge tree generation and improve results via segmentation accumulation.

Contribution of this previous work is that will develop semi supervised system which will reduced the cost of the system and no mandatory training will always be required. Also window size will be reduced for extract boundary pixel which will give better segmented output. This merge tree model has 2 major blessings. First, the tree structure allows the incorporation of upper order image into segmentation. The merge/split choices are created along during a globally optimum manner rather than by trying solely at native region pairs. Second, our technique doesn't need the parameter to work out once to prevent merging as in ISCRA and GALA, which can be thus vital to the results that wants parameter-free given the initial super constituent over-segmentation. The sole parameter is that the variety of iterations, which may be mounted as shown within the experiments on all the info sets.

SYSTEM OVERVIEW



(a) An Initial segmentation (b) Final segmentation



A merge tree conditional model graph with labeling.

Fig 1.Example of Image segmentation

A.Hirarchical Model

Consider a graph, in which every node corresponds to a super pixel, and an edge is defined between 2 nodes that share boundary pixels with one another. Starting with the initial over-segmentations, finding a final segmentation, which is basically the merging of initial super pixels, can be considered as combining nodes and removing edges between them. This super pixel merging will be done in an iterative fashion: every time a pair of neighboring nodes are combined within the graph, and corresponding edges are updated. To represent the order of such merging, use a full binary tree structure, which state as the hierarchical merge tree. In a merge tree $Tr = (V, E)$, a node $v^d_i \in V$ represents an image segment $S_i \in 2^p$, where denotes the depth in T_r at which this node occurs. Leaf nodes correspond to initial super pixels in So. A non leaf node corresponds to an image region formed by merging super pixels, and the root node corresponds to the total image as one single region. An edge $e_{ij} \in \mathcal{E}$ between node $v_d i$ and its child v^{jd+1} exists once $s_j \subset s_j$, and a local structure represents $s_i = s_j \cup s_k$. In this way, finding a final segmentation becomes finding a subset of nodes in T_r. Fig. 1. Example of (a) an initial segmentation, (b) a consistent final segmentation, (c) a merge tree, and (d) the corresponding conditional model factor graph with correct labeling. In (c), the leaf nodes have labels identical to those of the initial regions. The red nodes correspond to regions in the final segmentation.

The red box in (d) indicates a clique in the model. It is noteworthy that a merge tree defined here is seen as a dendrogram in hierarchical cluster [40] with every cluster being an image region. In order to work out the merging priority, define rising saliency function $fms: S_2 \rightarrow R$ that assigns a real number to every try of regions in S as a measurement of their merging probability. For any pair of regions s_i and s_j that are not neighbors, define $fms(s_i , s_j) = -\infty$. Then starting from a group of initial super pixels $S = S_0$ as leaf nodes, a merge tree is constructed by iteratively merging $(s_i^*, s_j^*) = arg\ maxs_i ,s_j \in S, i!=j\ fms(s_i , s_j)$ to a parent node, until only one region remains in S corresponding to the root node. Statistics over the strengths of boundary pixels between two merging regions from boundary detection chance maps could be used as fms. Following [35], use negated median

$$fms(s_i , s_j) = 1\ \ median(\{Pb(k) \mid k \in \mathcal{B}(s_i , s_j)\}) \tag{1}$$

Where pb(k) is the value of the k-th pixel on some boundary detection likelihood map Pb, and $\mathcal{B}(s_i , s_j)$ is the set of boundary pixels between s_i and s_j . $\mathcal{B}$ can be completely different on
implementation. In this work, define

$$\mathcal{B}(s_i , s_j) = \{s_i \cap N(s_j)\} \cup \{s_j \cap N(s_j)\}- \tag{2}$$

Where $N(s\cdot)$ is the set of neighbor pixels of s.

B. Constrained Conditional Model

For selecting a subset of nodes which forms an optimal Segmentation, create a merge tree which is as a constrained conditional model. It is primarily a factor graph for the merge tree, in which the node set arrenge ideally with V,and each merge in the merge tree that involves three nodes $(\{v^d{}_i , v^d{+}1_j , v^d{+}1_k \}, \{e_{ij} , e_{ik} \})$ is considered as a clique p_i in the graph. A label $y_i = +1$ or $y_i = -1$ is assigned to every node which indicate either its children merge or not. All leaf nodes is mandatory to be labeled +1. A complete label assignment $Y = \{y_i\}_i$ .if a node is labeled +1, all of its descendants must be labeled +1 as well. Then the nodes whose labels are +1 and parents are labeled −1 are selected as segments in the final segmentation. Fig. 1(d) is the factor graph for the constrained conditional model derived from the merge tree in Fig. 1(c). The red box shows a clique, and a set of consistent labeling is shown.In this system system train a classifier for predict the probability $P(y_i)$ for each merge $(\{v^d{}_i , v^{d+1}{}_j , v_k^{d+1} \}, \{e_{ij} , e_{ik} \})$. Then we score each clique p_i by associating it with energy with respect to its label

$$E_i (y_i) = -log\ P(y_i), y_i = \pm 1. \tag{3}$$

Under the Markov theory,formulation of labeling problem as a constrained optimization problem

$$\underset{Y}{MIN} \sum_{y_i \in Y} Ei\ (yi\ ), yi = \pm 1$$

$$,s.t.\ yi = +1,\ \forall i,\ v^d{}_i \text{ is a leaf},$$
$$Y_i \leq_j\ ,\ \forall_{i,j},\ v^d{}_i \text{ is parent to } v_j^{d+1}\ , \tag{4}$$

C. Boundary Classifier

To mark every clique, train a boundary classifier to tell previously the probability of every merge. For creating training labels that indicate if the boundary between two regions present or not, by comparing both the merge and the split possibly in oppose of the ground truth under specific error, such as the Rand error [41] and the variation of information [42], [43]. The case with smaller error contribut less from the ground truth and is adopted. Boundary features and region features are fetching for existence, and regional features measure actual geometric area and textural classification. Because of a pair of merging regions, boundary features provide direct clues about how it is the boundary truly similarities amongst the two regions, which may both be informative to boundary classification. For this choose features following [25] for comparison requirement. The boundary classifier is very wast topic it is not bounded to any specific supervised classification model. For this random forest [44] will use in experiments. The boundary classification problem is highly non-linear, and learning one universally good boundary classifier for all merging cases is essentially difficult. The size of merging regions affects the feature representativeness in classification. For instance, textural features in the form of averaged histograms among patches may not be informative when the merging regions are too small, because textural features can be extracted from only a very limited number of image patches and is thus noisy. On the other hand, when two regions are so big that they contain under-segmentation from different perceptual groups, the features again may not be meaningful, but for a different reason, that is, the histogram averaging is not able to represent the variation of textures. It is worth noting that for the same reason, different classifiers have to be learned at different merging stages in [25]. classification problem is categorize into sub-problems, train a separate sub-classifier for each sub-problem, and form the boundary classifier as an ensemble of sub-classifiers. The median size computation is as $|s|med$ of all regions will observed in the training set and assign a category label to training sample that involves regions s_i and s_j based on their sizes as in (5). Three sub-classifiers will be trained respectively using only samples with identical category labels.

$$c(si\ ,\ s\ j\ )= \begin{cases} 1, \text{ if } max(|s_i\ |,\ |s_j\ |) < |s|med, \\ 2, \text{ if } min(|s_i\ |,\ |s_j\ |) < |s|med \leq\ max(|s_i\ |s_{j)} \\ ,3, \text{ otherwise} \end{cases} \quad (5)$$

When testing is executed, a sample is categorized based on its region sizes and give it to the respective sub-classifier for prediction. Since all the sub-classifiers are always used ad jointly.

D. Inference

The tree structure, use a bottom-up/top-down algorithm to effectively find the perfect solution for the region consistency constraint. The idea of the bottom-up/top-down algorithm is dynamic programming: in the bottom-up step, the smallest energies for both decisions (merge/split) below the constraint are kept and Constructed from leaves to the root, based on which the best decisions will made from the root to leaves in theTop-down step. It is noticeable that bottom-up/top-down Algorithm is only for inference and concept is different from the top-down/bottom-up framework in [27], which have to combine high-level semantic information and low-level image features. On the other hand, the two-way message passing algorithm used in [27] and algorithm both belong to the Pearl's belief propagation [45], [46] category, except that inference algorithm explicitly incorporates the consistency constraint into the optimization procedure. In the bottom-up step, a pair of energy sums is kept track of for each node v^d_i with children $v_j^{\ d+1}$ and $v_k^{\ d+1}$: the merging energy $E_m^{\ i}$ of node v_i^d and its descendants all starts labeled +1 (merge), the splitting energy E^s_i of it that v^d_i is labeled −1 (split), and its descendants are labeled considerably . Then the energies can be computed bottom-up recursively as

$$E^m_{i=} E^m_i + E^{\ m}_{\ k} + E_i\ (y_i = +1), \quad (6)$$

$$E^s_{i=} min(E^{\ m}_{\ j}, E^{\ s}_{\ j}) + min(E^m_{\ k}\ , E^s_{\ k}) + E_i\ (y_i = -1). \quad (7)$$

For leaf nodes, we assign $E^m_i = 0$ and $E^s_i = \infty$ to create their being labeled +1. Algorithm 1 Illustrates the bottom-up algorithm.

ALGORITHM:1

Input: A list of energy of each part pi for merge
Output: A list of energy tuples $f(E_i^m ;E_i^s)$
1. Energy Tuple List
2. Get Energy Tuples(pr), where v_0^r is the root node
3. Sub-function that recursively computes energy terms
4. Function Get Energy Tuples(pi)
5. If v^d_i is a leaf node then
6. Return (0;1)
7. end if
8. Get Energy Tuples(p_j)
9. Get Energy Tuples(p_k)
10. Add new energy tuple to Energy Tuple List
11. Return new energy tuple

12. End function.

In the top-down algorithm start from the root and then perform depth first search: if the merging energy of a node is lower than its splitting energy, label this node and all its descendants +1; else, label this node −1 and search its children. Algorithm 2 illustrates the top-down algorithm

ALGORITHM 2
Input: A list of energy tuples
Output: A complete label assignment
1. Y=NULL
2. OpenNodeQueue.Enqueue(p_r), where v^0_r is the root node
3. While OpenNodeQueue is not empty do
4. p_i OpenNodeQueue.Dequeue()
5. if $E^m_i < E^s_i$ then
6. y_i = +1 and add it to Y
7. For all descendant part p^i_d of p_i do
8. y^i_d = +1 and add it to Y
9. Else y_i = -1 and add it to Y
10. OpenNodeQueue.Enqueue(p_i)
11. OpenNodeQueue.Enqueue(p_k)
12. stop.
If in system select the set of the nodes, such that its label
is +1 and its parent is labeled −1, to form final segmentation. In both algorithms, every node is checked only once with same operations, and need only linear space proportional to the number nodes for T_E and Y , so the time and space complexity are both $O(|V|)$

ITERATIVE HIERARCHICAL MERGE TREE MODEL

The performance upper bound of the hierarchical merge tree model is decided by the quality of the tree structure. If all true segments exist as nodes in the tree, they may be picked out by the logical thinking rule using predictions from well trained boundary classifiers. However, if a desirable segments not represented by any node in the tree, the model is not able to recover the segment. Hence, the merging saliency function, which is used to work out merging priorities, is critical to the entire performance. With a good merging saliency function, push the upper bound of performance and therefore improve segmentation accuracy. Statistics over the boundary strengths can be used to indicate merging saliency. In system use the negated median of boundary pixel strengths as the initial representation of saliency. Since a boundary classifier is essentially designed to measure region merging probability, and it has advantages over easy boundary statistics as a result of it takes numerous options from each boundary and regions, propose to use the merging probabilities predicted by boundary classifiers as the merging saliency to construct a merge tree. The training of a boundary classifier needs samples generated from a merge tree, but would like to use a boundary classifier to construct emerge tree. Therefore, propose an iterative approach that alternately collects training samples from a merge tree forth training of boundary classifiers and constructs a merge tree with the trained classifier. As illustrated in Fig. 2(a), initially use the negated median of boundary strengths to construct a merge tree, collect region merging samples, and train a boundary classifier f 0b . Then, the boundary classifier f 0bis used to generate a brand new merge tree from an equivalent initial super pixels So, from which new training samples are generated. next combine the samples from the current iteration and from the previous iterations, remove duplicates, and train the next classifier f 1b . This process is continual for T iterations or till the segmentation accuracy on a validation set no longer improves. In practice, fix the iteration number to T = 10for all data sets. Eventually, have a series of boundary classifiers T_t=0 from each training iteration. The training algorithm is illustrated in Algorithm 3

ALGORITHM 3
Input: Original images , boundary maps , and iteration number
Output: Boundary classifiers
1. Generate initial super pixels
2. for t : 0; 1; : : : ; T do
3. Use the negative median of boundary strengths to construct a merge tree

4. Then the boundary classifier is used to generate a new merge tree, from which new training samples are generated.
5. Then we combine the samples from the current iteration and from the previous iterations, remove duplicates, and train the next clas-s_i_e_r.
6. stop

when testing will perform, Then take the number of trained classifiers and iterate in a way that the same as the training method, as shown in Fig. 2 at every iteration t, accept the previous boundary classifier f_b^{t-1} to construct a merge tree over the same initial super pixels So and use the present classifier f_b^t to predict every merge score within the merge tree, based on that final segmentation s_t is inferred. Finally, translate every segmentation into a binary closed contour map by assigning boundary pixels 1 and others 0 and average them for every image for all iterations to be result. The testing algorithmic illustrated in Algorithm 4. The explanation for the iterative approach is two-purpose. First, by collecting samples that were not seen in any before iterations, can expand the merge sample space and intern explore the space of merge trees generated by the classifiers trained using the increased sample set towards the "correct" merge tree. Second, like a bagging algorithm, segmentation averaging through iterations tends to emphasize accurate boundaries by phasing out non-systematic errors duet incorrect tree structures or classifier mispredictions.

ALGORITHM 4
Input:Original imagesi, boundary maps , boundary classifiers , and iteration number
Output: Hierarchical segmentation contour map
1. Generate initial super pixels
2. for t: 0; 1; T do
3. If t == 0 then
4. Use the negative median of boundary strengths to construct a merge tree
5. Predict energy for each merge
6. Generate segmentation using bottom-up/top down algorithm
7. Generate binary contour map
8. Stop

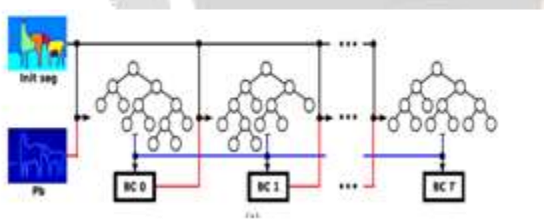


Fig.2 (a) Training procedure of the iterative hierarchical merge tree model.

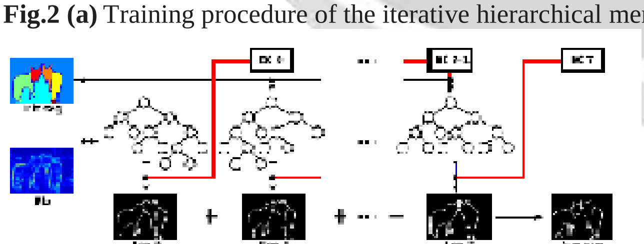


Fig.2 (b) Testing procedure of the iterative hierarchical merge tree model.

Fig.2.(a) the training and (b) the testing procedure of the iterative hierarchical merge tree model. Starting with the fixed initial super pixels("Init Seg"), the first iteration uses boundary probability ("Pb") statistics for merge tree generation, and the training procedure iteratively augments the training set by incorporating new samples from merge trees and trains a new boundary classifier ("BC"), which is used for merge tree generation from the same initial super pixels in the next iteration. At testing time, boundary probability statistics and boundary classifiers

learned at each iteration are used to generate merge trees from the same initial super pixels, and each boundary classifier is used to score merge cliques in the previous iteration; segmentations are generated from each merge tree and accumulated to generate the final contour hierarchy. The black lines show the use of initial super pixels, the red lines show the use of boundary classifiers, and the blue lines show the flow of sample data collected from tree structures.

SYSTEMANALYSIS

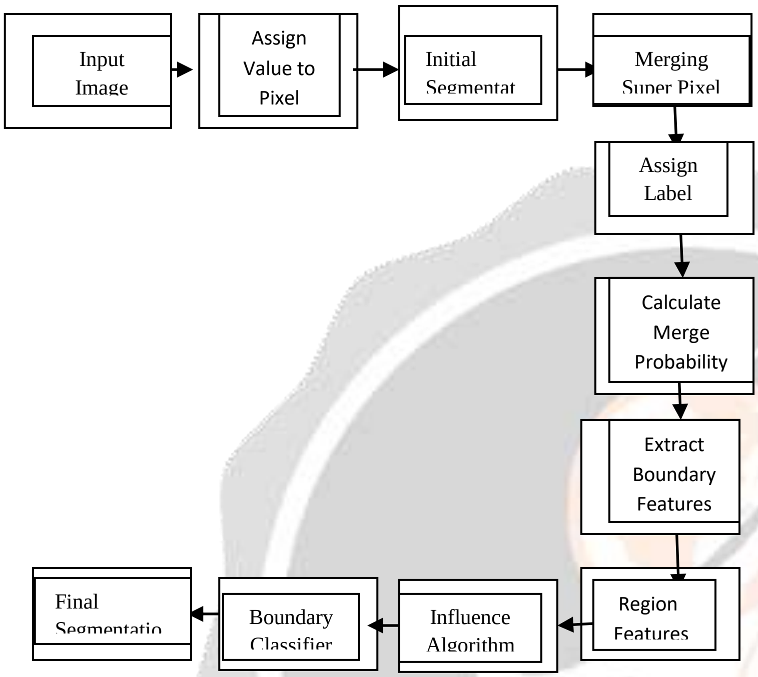


Fig 3: System Block Diagram

CONCLUSION

Like this is a hierarchical image segmentation framework, namely the hierarchical merge tree model, that limits the search space to one that is induced by tree structures and thus linear with respect to the number of initial super pixels. The framework allows the use of various merging saliency heuristics and features, and its supervised nature grants its capability of learning complex conditions for merging decisions from training data without the need for parameter tuning or the dependency on any classification model. Globally optimal solutions can be efficiently found under constraints to generate final segmentations thanks to the tree structure. Also introduce a modification to the hierarchical merge tree model that iteratively trains a new boundary classifier with accumulated samples for merge tree construction and merging probability prediction and accumulates segmentation to generate contour maps.

REFERENCES

[1] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” IEEE Trans. Pattern Anal. Mach.Intell., vol. 33, no. 5, pp. 898–916, May 2011.

[2] D. Marr and E. Hildreth, “Theory of edge detection,” Proc. Roy. Soc.London B, Biol. Sci., vol. 207, no. 1167.pp. 187–217, 1980.

[3] J. Canny, “A computational approach to edge detection,” IEEE Trans.Pattern Anal. Mach. Intell., vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[4] M. C. Morrone and R. A. Owens, “Feature detection from local energy,”Pattern Recognit.Lett., vol. 6, no. 5, pp. 303–313, Dec. 1987.

[5] W. T. Freeman and E. H. Adelson, “The design and use of steerable filters,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 13, no. 9,pp. 891–906, Sep. 1991.
[6] R. Xiaofeng and L. Bo, “Discriminatively trained sparse code gradients for contour detection,” in Proc. Adv. Neural Inf. Process. Syst., 2012,pp. 584–592.
[7] P. Dollár and C. Zitnick, “Structured forests for fast edge detection,” in Proc. Int. Conf. Comput. Vis., 2013, pp. 1841–1848.
[8] M. Seyedhosseini and T. Tasdizen, “Semantic image segmentation with contextual hierarchical models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 5, pp. 951–964, May 2015.
[9] S. Belongie, C. Carson, H. Greenspan, and J. Malik, “Color- and texturebased image segmentation using EM and its application to contentbased image retrieval,” in Proc. Int. Conf. Comput. Vis., Jan. 1998,pp. 675–682.
[10] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24,no. 5, pp. 603–619, May 2002.
[11] A. Vedaldi and S. Soatto, “Quick shift and kernel methods for mode seeking,” in Proc. Eur. Conf. Comput. Vis., 2008, pp. 705–718.
[12] J. Shi and J. Malik, “Normalized cuts and image segmentation,”IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 8, pp. 888–905,Aug. 2000.
[13] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” Int. J. Comput. Vis., vol. 59, no. 2, pp. 167–181, Sep. 2004.
[14] H. Zhu, F. Meng, J. Cai, and S. Lu, “Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation,”J. Vis. Commun. Image Represent., vol. 34, pp. 12–27, Jan. 2016.
[15] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV), vol. 2. Jul. 2001, pp. 416–423.
[16] J. Shotton, J. Winn, C. Rother, and A. Criminisi, “TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation,” in Proc. Eur. Conf. Comput. Vis., 2006, pp. 1–15.
[17] S. Gould, R. Fulton, and D. Koller, “Decomposing a scene into geometric and semantically consistent regions,” in Proc. Int. Conf. Comput. Vis., Sep./Oct. 2009, pp. 1–8.
[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman.The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.acessed on Dec. 9, 2014. [Online]. Available: http://www.pascalnetwork.
org/challenges/VOC/voc2012/workshop/index.html
[19] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from RGBD images,” in Proc. Eur. Conf. Comput.Vis., 2012, pp. 746–760.
[20] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan, “Multi-task low-rank affinity pursuit for image segmentation,” in Proc. Int. Conf. Comput.Vis., Nov. 2011, pp. 2439–2446.
[21] H. Zhu, J. Zheng, J. Cai, and N. M. Thalmann, “Object-level image segmentation using low level cues,” IEEE Trans. Image Process., vol. 22,no. 10, pp. 4019–4027, Oct. 2013.
[22] S. Kim, C. D. Yoo, S. Nowozin, and P. Kohli, “Image segmentation using higher-order correlation clustering,” IEEE Trans. Pattern Anal.Mach. Intell., vol. 36, no. 9, pp. 1761–1774, Sep. 2014.
[23] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, “Multiscale combinatorial grouping,” in Proc. IEEE Conf. Comput.Vis.Pattern Recognit., Jun. 2014, pp. 328–335.
[24] Y. Yu, C. Fang, and Z. Liao, “Piecewise flat embedding for image segmentation,” in Proc. Int. Conf. Comput. Vis., Dec. 2015,pp. 1368–1376.
[25] Z. Ren and G. Shakhnarovich, “Image segmentation by cascaded region agglomeration,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,Jun. 2013, pp. 2011–2018.
[26] J. Nunez-Iglesias, R. Ken nedy, T. Parag, J. Shi, and D. B. Chklovskii,“Machine learning of hierarchical clustering to segment 2D and 3D images,” PLoS One, vol. 8, no. 8, p. e71715, 2013.
[27] E. Borenstein and S. Ullman, “Combined top-down/bottom-up segmentation,”IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 12,pp. 2109–2125, Dec. 2008.
[28] J. Carreira and C. Sminchisescu, “CPMC: Automatic object segmentation using constrained parametric min-cuts,” IEEE Trans. Pattern Anal.Mach. Intell., vol. 34, no. 7, pp. 1312–1328, Jul. 2012.
[29] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik,“Semantic segmentation using regions and parts,” in Proc. IEEE Conf.Comput. Vis. Pattern Recognit., Jun. 2012, pp. 3378–3385.
[30] D. Weiss and B. Taskar, “SCALPEL: Segmentation cascades with localized priors and efficient learning,” in Proc. IEEE Conf. Comput.Vis. Pattern Recognit., Jun. 2013, pp. 2035–2042.

[31] A. Joulin, F. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,Jun. 2010, pp. 1943–1950.
[32] S. Vicente, C. Rother, and V. Kolmogorov, "Object cosegmentation,"inProc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2011,pp. 2217–2224.
[33] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade, "Distributed cosegmentation via submodular optimization on anisotropic diffusion," in Proc. Int. Conf. Comput. Vis., Nov. 2011, pp. 169–176.
[34] T. Liu, E. Jurrus, M. Seyedhosseini, M. Ellisman, and T. Tasdizen,"Watershed merge tree classification for electron microscopy image segmentation," in Proc. Int. Conf. Pattern Recognit., Nov. 2012,pp. 133–137.
[35] T. Liu, C. Jones, M. Seyedhosseini, and T. Tasdizen, "A modular hierarchical approach to 3D electron microscopy image segmentation,"J. Neurosci. Methods, vol. 226, pp. 88–102, Apr. 2014.LIU et al.: IMAGE SEGMENTATION USING HIERARCHICAL MERGE TREE 4607
[36] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson,and K. Siddiqi, "TurboPixels: Fast superpixels using geometric flows,"IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 12, pp. 2290–2297,Dec. 2009.
[37] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in Proc. Eur. Conf. Comput. Vis.,2010, pp. 211–224.
[38] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk,"SLIC superpixels compared to state-of-the-art superpixel methods,"IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 11, pp. 2274–2282,Nov. 2012.
[39] S. Beucher and C. Lantuéjoul, "Use of watersheds in contour detection,"inProc. Int. Workshop Image Process., Real-Time Edge Motion Detection/Estimation, 1979.
[40] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis,vol. 3. New York, NY, USA: Wiley, 1973.
[41] W. M. Rand, "Objective criteria for the evaluation of clustering methods,"J. Amer. Statist. Assoc., vol. 66, no. 336, pp. 846–850, 1971.
[42] M. Meil˘a, "Comparing clusterings: An axiomatic view," in Proc. Int.Conf. Mach. Learn., 2005, pp. 577–584.
[43] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," Comput. Vis. Image Understand., vol. 110, no. 2, pp. 212–225, May 2008.
[44] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32,2001.
[45] J. Pearl, "Reverend Bayes on inference engines: A distributed hierarchical approach," in Proc. AAAI Conf. Artif. Intell., 1982, pp. 133–136.
[46] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," IEEE Trans. Inf. Theory, vol. 47, no. 2,pp. 498–519, Feb. 2001.
[47] T. Malisiewicz and A. A. Efros, "Improving spatial support for objectsvia multiple segmentations," in Proc. Brit. Mach. Vis. Conf., 2007.
[48] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," IEEE Trans. Pattern Anal.Mach. Intell., vol. 29, no. 6, pp. 929–944, Jun. 2007.
[49] M. Donoser and D. Schmalstieg, "Discrete-continuous gradient orientation estimation for faster image segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 3158–3165.
[50] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," Int. J. Comput. Vis., vol. 43, no. 1, pp. 7–27, 2001.