



Task Specific Image Partitioning for Region Based Image Retrieval

Deepu Baby
M Tech Student, VJCET
India

Sabitha Raju
Asst.Professor, VJCET
India

Abstract— Image partitioning is an important pre-processing Step for many of the state-of-the-art algorithms used for performing high-level computer vision tasks. Typically, partitioning is conducted without regard to the task in hand. Task-specific image partitioning framework to produce a region-based image representation that will lead to a higher task performance than that reached using any task-oblivious partitioning framework and existing supervised partitioning framework, albeit few in number. This method partitions the image by means of correlation clustering, maximizing a linear discriminant function defined over a superpixel graph. The parameters of the discriminant function that define task specific similarity/dissimilarity among superpixels are estimated based on structured support vector machine (S-SVM) using task specific training data. Region based image retrieval is used to retrieve similar images from a database of images, where the images in database are partitioned by using the same partitioning method. Similar images are finding by region wise similarity.

Keywords— Image partitioning, superpixels, correlation clustering, support vector machine, image similarity

I. INTRODUCTION

Region based image representations (RBIRs) have been shown to be effective in improving the performance of algorithms for high-level image/scene understanding, which encompasses tasks such as object class segmentation, scene segmentation, surface layout labeling, and single view 3D reconstruction. The effectiveness comes as a result of promoting the following three merits of using the RBIRs. First, the coherent support of a region, commonly assumed to be of a single label, serves as a good prior for many labeling tasks. Second, these coherent regions allow a more consistent feature extraction that can incorporate surrounding contextual information by pooling many feature responses over the region. Third, compared to pixels, a small number of larger homogeneous regions can significantly reduce the computational cost in the successive labeling task. The image partitioning framework for obtaining RBIRs that realizes these benefits and improves the task-specific labeling performance.

After partitioning the image into regions, a region based image retrieval is performed from a database of images. For that a query image is partitioned to regions by using any particular partitioning method and the region based image retrieval is performed where the images stored in the DB is also partitioned by similar partitioning method.

II. SYSTEM OVERVIEW

Steps in the framework includes,

1. DB of images are created by using partitioning methods
2. Query image is partitioned to regions
3. Region wise similarity assessment
4. Overall similarity between query image and db image
5. Similar images are extracted

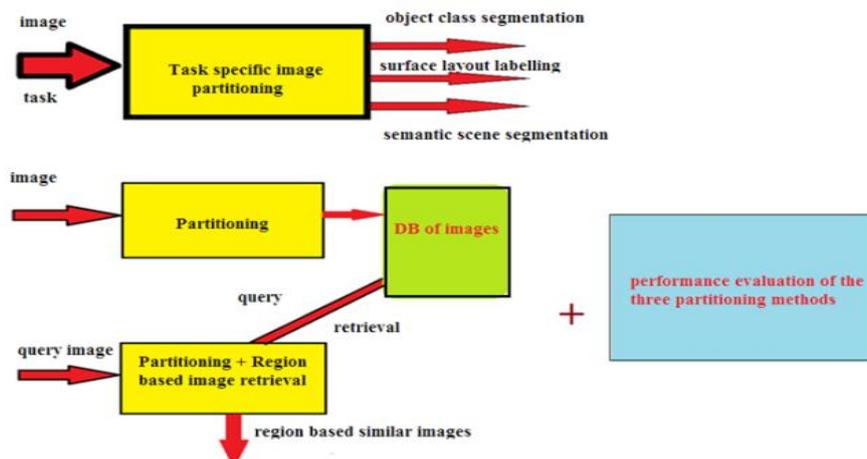


Figure 1 Framework

III. MODULES

The system includes four main components:

1. Superpixel Graph Creation
2. Correlation Clustering Over Superpixel Graph.
3. Structured Support Vector Machine
4. Cutting Plane Algorithm
5. Label Loss Function
6. Database of Partitioned images
7. Region Similarity
8. Image Similarity

A. Superpixel Graph Creation

The image partitioning is based on superpixels, which can significantly reduce computational cost and allow feature extraction to be conducted from a larger homogeneous region. Superpixels preserve almost all boundaries between different regions, independent of the task. The correlation clustering merges superpixels into disjoint regions of homogeneity over a superpixel graph

B. Correlation Clustering Over Superpixel Graph

Image partitioning is based on superpixels, which can significantly reduce computational cost and allow feature extraction to be conducted from a larger homogeneous region. Superpixels preserve almost all boundaries between different regions, independent of the task. Correlation clustering merges superpixels into disjoint regions of homogeneity over a superpixel graph

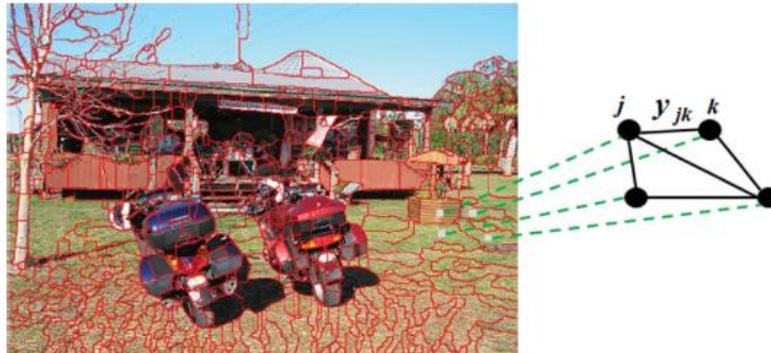


Figure 2 Illustration of a part of the graph built on superpixels

Define an undirected graph $G = (V, \varepsilon)$ where a node corresponds to a superpixel and a link between adjacent superpixels corresponds to an edge. A binary label y_{jk} for an edge (j, k) between nodes j and k is defined such that

$$Y_{ik} = \begin{cases} 1, & \text{if nodes } i \text{ and } k \text{ belong to same region} \\ 0, & \text{otherwise} \end{cases} \quad (3.1)$$

A discriminant function, which is the negative energy function, is defined over image x and all edge labels y as

$$\begin{aligned} F(x, y; w) &= \sum_{(j,k) \in \varepsilon} Sim_w(X, j, k) y_{jk} \\ &= \sum_{(j,k) \in \varepsilon} \langle w, \Phi_{jk}(x) \rangle y_{jk} \\ &= \langle w, \sum_{(j,k) \in \varepsilon} \Phi_{jk}(x) y_{jk} \rangle \\ &= \langle w, \Phi(x, y) \rangle \end{aligned} \quad (3.2)$$

where the similarity measure between nodes j and k , $Sim_w(x, j, k)$, is parameterized by w and takes values of both signs such that a large positive value means strong similarity while a large negative value means high degree of dissimilarity. The discriminant function $F(x, y; w)$ is assumed to be linear in both the parameter vector w and the joint feature map. An image segmentation is to infer the edge label, y , over the pair wise superpixel graph G by maximizing F such that

$$\hat{y} = \underset{y \in Y(G)}{\operatorname{argmax}} F(x, y; w) \quad (3.3)$$

C. S-SVM Training

For task-specific image partitioning, the parameter vector w is estimated from the training data for each task. The proposed discriminant function is defined over the superpixel graph, and therefore, the ground-truth task labels of the pixels need to be transformed to the ground-truth edge labels of the superpixel graph. Note that different from the ground-truth edge-labelling over the superpixel graph, the ground-truth partitioning is directly defined by the ground-truth task labels. First, assign a single dominant task label to each superpixel by majority voting over the superpixel's constituent pixels and then obtain the ground-truth edge labels on the superpixel graph according to whether dominant labels of neighbouring superpixels are equal or not.

Using this ground-truth edge labels of the training data, use the S-SVM to estimate the parameter vector for task-specific correlation clustering. Use the cutting plane algorithm with LP relaxation for loss-augmented inference is used to solve the optimization problem of the S-SVM, since fast convergence and high robustness of the cutting plane algorithm in handling a large number of margin constraints are well-known.

D. Structured Support Vector Machine

Given N training samples $(y^n, y)_{n=1}^N$ where y^n are the ground-truth edge labels for the n th training image, the S-SVM optimizes w by minimizing a quadratic objective function subject to a set of linear margin constraints:

$$\min_{w, \xi} \frac{1}{2} \|w\|^2 + C \sum_{n=1}^N \xi_n \quad (3.4)$$

$$s.t \langle w, \delta \Phi(x^n, y) \rangle \geq \Delta(y^n, y) - \xi_n, \forall_n, y \in (\mathcal{G}) \setminus y^n, \xi_n \geq 0, \forall_n$$

In the S-SVM, the margin is scaled with a loss $\Delta(y^n, y)$ which is the difference measure between prediction y and ground-truth label y^n of the n th image. The S-SVM offers good generalization ability as well as the flexibility to choose any loss function

E. Cutting Plane Algorithm

The exponentially large number of margin constraints and the intractability of the loss-augmented inference problem make it difficult to solve the constrained optimization problem of. Therefore, by apply the cutting plane algorithm also known as the column generation algorithm, to approximately solve the constrained optimization problem. In each iteration, the most violated constraint for each training sample is approximately found by performing the loss-augmented inference using the LP relaxation. The computational cost for inference can be greatly reduced when a decomposable loss such as the Hamming loss is used; if the loss function is decomposed in the same manner as the joint feature map, add the loss function to each edge score in the inference. Then check if the constraint found tightens the feasible set, and if it does, then the parameter vector w and ξ are updated by solving the restricted problem of (3.4) on the current set of active constraints that includes it. The LP relaxations for loss-augmented inferences are considered to be well suited to structured learning

F. Label Loss Function

A loss function $\Delta: Z \times Z \rightarrow R^+$ is defined as a nonnegative function satisfying the following properties for all n ,

$$\begin{aligned} \Delta(y^n, y) &> 0, \text{ if } y \neq y^n \\ \Delta(y^n, y) &= 0, \text{ if } y = y^n \end{aligned}$$

A loss function should be decomposable to effectively perform loss-augmented inference in the cutting plane algorithm. The most popular decomposable loss function is the Hamming distance which is equivalent to the number of mismatches between y^n and y . Unfortunately, the number of edges with label 1 in the proposed correlation clustering is considerably higher than that of edges with label 0. This imbalance makes other learning methods such as the perceptron algorithm inappropriate, since it leads to the clustering of the whole image as one segment. This imbalance occurs when using the Hamming loss in the S-SVM; therefore, use the following adjusted loss function:

$$\begin{aligned} \Delta(y^n, y) &= \sum_{j,k \in \mathcal{E}} \Delta_{jk}(y_{jk}^n, y_{jk}) \\ &= \sum_{j,k \in \mathcal{E}} R y_{jk}^n + y_{jk} - (R + 1) y_{jk}^n y_{jk} \end{aligned} \quad (3.5)$$

Where Δ_{jk} is the label loss on the edge between nodes j and k , and R is the relative weight of the false negative to that of the false positive 1. Note that the additive decomposition of the loss allows us to cast the loss into the additive edge score when performing the loss-augmented inference. Moreover, R controls the relative importance between the incorrect merging of the superpixels and the incorrect separation of the superpixels by imposing different weights to the false negative and the false positive, as shown in fig 3.4. Here, set R to be less than 1 to overcome the problem due to the imbalance. However, the proposed loss is appropriate for fractionally-predicted labels during LP-relaxed inference while their loss is appropriate for only integer solutions.

y_{jk}^n	0	1	0	1
y_{jk}	0	1	1	0
Δ_{jk}	0	0	1	R

Figure 3 Label Loss at the Edge Level



Figure 4 input image

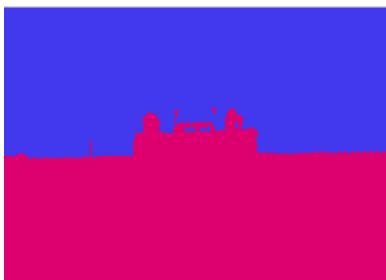


Figure 5 Surface Layout Labelling



Figure 6 Semantic Scene Segmentation



Figure 7 Object Class Segmentation

G. Database of Partitioned images

A Database of partitioned images is created. Any partitioning method, such that any task specific partitioning method can be used to make the database. The image is taken, then partitions it then stores it in database. Similarly a number of images are stored in the database.

H. Region Similarity

A Query image is inputting to the system. It is partitioned to regions. Then similarity between query image and images in database is calculated. Then the maximum similar images are retrieved. This similarity measurement includes region wise similarity and total similarity measurement. The similarity between two regions, R_i and R_j , is computed as follows:

$$Sim(R_i, R_j) = \alpha_{ij} \cdot \beta_{ij} \cdot h(d(R_i, R_j))$$

where $d()$ is a distance function, $h()$ is a so-called correspondence function relating distance values to similarity scores, α_{ij} and β_{ij} are similarity coefficients. The α_{ij} coefficient, used to favour match between large regions, takes into account the relative size of the regions with respect to those of the images they were extracted from. The β_{ij} coefficient takes into account the similarity in size between the two regions. If R_i is extracted from image I and R_j from image J , α_{ij} and β_{ij} can be defined as:

$$\alpha_{ij} = \frac{\text{size}(R_i) + \text{size}(R_j)}{\text{size}(I) + \text{size}(J)}$$

$$\beta_{ij} = 1 - \frac{|\text{size}(R_i) - \text{size}(R_j)|}{\text{size}(R_i) + \text{size}(R_j)} \quad (3.7)$$

The correspondence function $h()$ is used to transform distance values into similarity scores. The function $h: R_0^+ \rightarrow [0,1]$ has to satisfy the following properties: $h(0)=1$ and $d1 \leq d2 \rightarrow h(d1) \geq h(d2)$, $d1, d2 \in R_0^+$. In all the experiments, used $h(d) = e^{-d/\sigma_d}$ where σ_d^2 is the distance variance computed over a sample of regions.

Finally, distance between two regions is computed by way of the Bhattacharyya metric, used to compare ellipsoidal clusters:

$$d_s(R_i, R_j)^2 = \frac{1}{2} \ln \left(\frac{\left| \frac{C_{R_i}^{3:s} + C_{R_j}^{3:s}}{2} \right|}{\left| C_{R_i}^{3:s} \right|^{\frac{1}{2}} \left| C_{R_j}^{3:s} \right|^{\frac{1}{2}}} \right) + \frac{1}{8} \left[(VR_i - VR_j)^T \cdot \left(\frac{C_{R_i}^{3:s} + C_{R_j}^{3:s}}{2} \right)^{-1} \cdot VR_i - VR_j \right] \quad (3.8)$$

Where $|A|$ is the determinant of matrix A . Note that Equation 3.8 is composed of two terms, the second is Mahalanobis distance between regions, using the average covariance matrix. The first term is used to compare the covariance matrices of the two regions. Indeed, if the two regions have the same centroid, the second term of Equation 3.8 has a value of zero, and the first term is used to distinguish between the two regions. The overall distance between regions R_i and R_j is assessed by computing Equation 3.8 over all the frequency sub-bands:

$$d(R_i, R_j)^2 = \sum_s \gamma_s d_s(R_i, R_j)^2$$

$d(R_i, R_j)^2$ is computed by way of Equation 3.8 and coefficients γ_s are used to give different weights to sub-bands

I. Image Similarity

Having defined how the similarity between regions is computed, now need to assess the overall similarity between two images (e.g. the query image Q and a DB image T). When matching regions in Q with regions in T , have to satisfy two basic constraints:

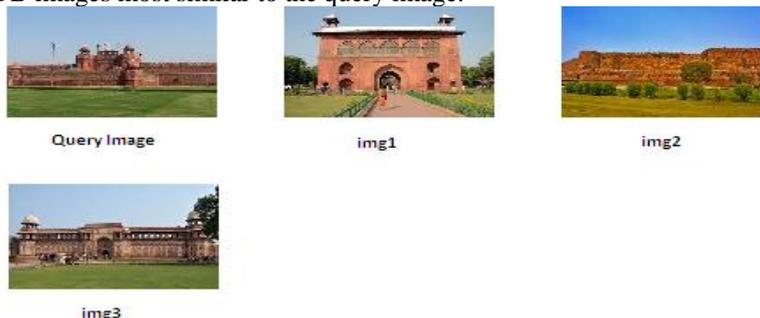
1. A region of Q cannot match with two different regions in T
2. Two different regions of Q cannot match with a single region of T

The overall similarity between two images Q and T as:

$$SIM(Q, T) = \sum_i sim(q_i, t_{j(i)}) \quad (3.9)$$

Of course, $\sum_i sim(q_i, t_{j(i)}) = 0$ if $t_{j(i)}$ is undefined. The user can now express a query by giving an input image. The

system will retrieve the DB images most similar to the query image.



J. Performance Evaluation

A performance evaluation has done for the three different partitioning methods based on,

- Partitioning Time
- Partitioning Accuracy
- Image Retrieval Time
- Image Retrieval Accuracy
- Database Creation Time

Partitioning Time means the average time taken for a partitioning method to give the partitioned image. The average is calculated by storing time for every instances of partition.

Let $t_1, t_2, t_3, \dots, t_n$ be the time measurements for different instances of partitions, then the partitioning time can be

calculated by

$$\text{Partition Time} = \frac{\sum t_i}{n} \quad (3.10)$$

Partitioning accuracy means the measure of how much accurate the partitioning results. The accuracy is calculated by storing feedback from for every instance

Let $a_1, a_2, a_3, \dots, a_n$ be the accuracy measurements for different instances of partitions, then the partitioning

accuracy can be calculated by

$$\text{Partition Accuracy} = \frac{\sum a_i}{n} \quad (3.11)$$

Retrieval Time means the average time taken for a partitioning method to give the retrieval results. The average is calculated by storing time for every instances of image retrieval.

Let $r_1, r_2, r_3, \dots, r_n$ be the time measurements for different instances of retrieval, then the retrieval time can be

calculated by

$$\text{Image Retrieval Time} = \frac{\sum r_i}{n} \quad (3.12)$$

Image Retrieval accuracy means the measure of how much accurate the retrieval results. The accuracy is calculated by storing feedback from for every instances of retrieval.

Let $ra_1, ra_2, ra_3, \dots, ra_n$ be the accuracy values for different instances of retrieval, then the retrieval accuracy can

be calculated by

$$\text{Image Retrieval Accuracy} = \frac{\sum ra_i}{n} \quad (3.13)$$

Database Creation Time means the time taken for a partitioning method to make a database of images. The time is calculated by storing time for each database creation.

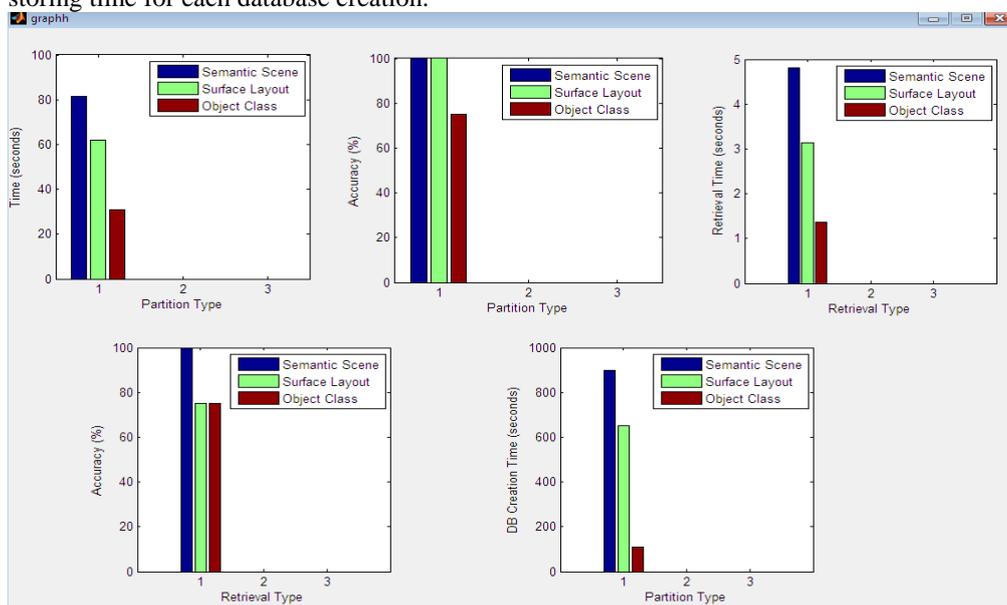


Figure 8 Performance Evaluation Graph

	Semantic Scene Segmentation	Surface Layout Labelling	Object Class Segmentation
Partition Time	80 seconds(average)	65 seconds(average)	30 seconds (average)
Partition Accuracy	Accurate	Accurate	Average
Retrieval Time(similarity measure)	6 seconds	3 seconds	1.5 seconds
Retrieval Accuracy	Accurate	Average	Average
Database Creation Time	900 seconds	600 seconds	100 seconds

Figure 9 Table showing evaluations

V. CONCLUSION

The system addressing the problem of task-specific image partitioning by supervised training and a region based image retrieval by creating a database of partitioned images and inputting a query image. The correlation clustering model aims to merge superpixels into regions of homogeneity with respect to the solution of any particular image labeling problem. The LP relaxation was used to approximately solve the correlation clustering over a superpixel graph where a rich pair wise feature vector was defined based on several visual cues. The S-SVM was used for supervised training of parameters in correlation clustering, and the cutting plane algorithm with LP-relaxed inference was applied to solve the optimization problem of S-SVM.

After partitioning, a database of images is created. When a user is inputting a query image, by calculating region similarity and total image similarity the most similar images can be retrieved. The partitioning framework is applicable to a broad variety of other high-level vision tasks. A performance analysis of three partitioning methods based on partition time, partition accuracy, database creation time, image retrieval time, image retrieval accuracy is also done.

REFERENCES

- [1] Sungwoong Kim, Sebastian Nowozin, Pushmeet Kohli and Chang D. Yoo, "Task-Specific Image Partitioning", in *IEEE Transactions On Image Processing*, vol. 22, no. 2, February 2013.
- [2] Stefania Ardizzoni, Ilaria Bartolini and Marco Patella, "Windsurf: Region-Based Image Retrieval Using Wavelets"
- [3] J. Shi and J. Malik, "Normalized cuts and image segmentation", in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug.2000
- [4] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis", in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002
- [5] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation", in *Int. J. Comput. Vis.*, 59, no. 2, pp. 167–181, 2004".
- [6] A. Vedaldi and S. Soatt, "Quick shift and kernel methods for mode seeking", in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 1–14
- [7] P. Kohli, L. Ladick'y, and P. H. S. Torr, "Robust higher order potentials for enforcing label consistency", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [8] L. Ladick'y, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical CRFs for object class image segmentation", in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep–Oct. 2009, pp. 739–746.
- [9] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions", in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep.–Oct. 2009, pp. 1–8.
- [10] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels", in *IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1253–1260.
- [11] M. P. Kumar and D. Koller, "Efficiently selecting regions for scene understanding", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2010, pp. 1–8