Object Class Segmentation using Reliable Regions

Vida Vakili and Olga Veksler

Computer Science Department
University of Western Ontario
London, Canada

Abstract. Image segmentation is increasingly used for object recognition. The advantages of segments are numerous: a natural spatial support to compute features, reduction in the number of hypothesis to test, region shape itself can be a useful feature, etc. Since segmentation is brittle, a popular remedy is to integrate results over multiple segmentations of the scene. In previous work, usually all the regions in multiple segmentations are used. However, a typical segmentation algorithm often produces generic regions lacking discriminating features. In this work we explore the idea of finding and using only the regions that are reliable for detection. The main step is to cluster feature vectors extracted from regions and deem as unreliable any clusters that belong to different classes but have a significant overlap. We use a simple nearest neighbor classifier for object class segmentation and show that discarding unreliable regions results in a significant improvement.

1 Introduction

Over the past couple of decades, most object recognition approaches were based on the sliding window scanning [1–5], localizing the bounding box of an object. While very successful for some applications, such as face and pedestrian detection [2], there are several well known issues with the sliding window approach. First, it is mostly appropriate for object classes that are well-approximated by a rectangle, such as cars, but not for object classes with thin parts, such as giraffes. Another problem is that there is no precise pixel-level segmentation of the detected object. Also, sliding window approach is expensive computationally, since a large number of sub-windows at different scales needs to be processed in a single image.

In recent years, there has been a lot of interest in using results of a generic image segmentation algorithm to help obtain pixel-precise object segmentation [6–15]. Segmentation is used to subdivide an image into a set of regions. These regions do not have predefined shapes like rectangular patches and their boundaries are more likely to align with the boundaries of objects in the scene. Therefore the shape and boundary properties of regions can be used for feature extraction. Furthermore, it has been shown in [16] that regions from unsupervised bottom-up segmentation can provide a better spatial support for object detection than rectangular windows. Another advantage of using image regions is scalability and potential savings in computational efficiency. Image regions provide a usually much smaller set of hypothesis to examine compared to the sliding window approach, and at “natural" scales that are obtained through segmentation.

Ideally, a good segmentation algorithm would cleanly separate the objects or at least the object parts. However, the state-of-the art in segmentation is still very far from this goal. With hierarchical segmentation there is a higher chance that a complete object or an object part is contained at some hierarchy level. Therefore some methods use a single hierarchical segmentation algorithm [17, 14] for object recognition. However, even hierarchical algorithms suffer from instability, the results depends on the choice of the feature space and parameter settings. Another strategy is to use smaller image regions, or "superpixels", as in [15], and regularize the classifier by aggregating histograms in the neighborhood of each superpixel. However, if superpixels are too small, they may not have enough spatial support for feature extraction. If superpixels are bigger, then there is a larger chance for an error due to inaccuracies in superpixel boundaries, since superpixels are used as indivisible units.

A more commonly used strategy is to integrate the results over multiple different segmentations of the same scene [6, 18, 13, 19]. Usually several unsupervised bottom-up segmentation algorithms, each with different parameters are used. With more segments, there is a higher chance that some segments contain whole objects or easily recognizable object parts. In support of using multiple segmentations, in [16] they show that usually regions generated by a single segmentation of a scene do not contain entire objects.

Previous work that relies on multiple segmentations either makes an assumption that at least one segmentation separates the object from the background [6, 19], or uses all segments as reliable for object class recognition [13]. Clearly many regions from multiple segmentations are rather generic blobs that do not contain features useful for discrimination. This is because most bottom-up segmentation algorithms are implicitly or explicitly designed to partition an image into a set of uniform patches in some feature space. The main idea of our approach is to find reliable regions and discard the ones that are not reliable for object recognition. This should result in more stable decisions at the time of object recognition.

In the training stage, first multiple image segmentations are obtained by varying the parameters in FH [20] and Mean Shift [21] segmentation algorithms. Each segment is described by a large set of features based on color, texture, etc. Then discriminative features are selected using the Relief-F algorithm [22]. For robustness, we do not want to decide on reliability individually for each segment. It is more reliable to make this decision for a whole cluster containing similar segments. A single object class is usually segmented into different possibly overlapping parts, each part may bear little similarity to the other parts. Therefore we cluster each object class (separately from the other classes) using the k-means algorithm, with the aim that each cluster contains similar segments of the object class. Clusters from different objects could have significant intersection, which makes them unreliable for recognition. The overlapping clusters contain segments that are generic blobs not corresponding to an easily recognizable object part. These clusters are discarded.

In the test stage, first multiple image segmentations are generated from the test image and features are extracted for each segment. Reliable segments are found based on their distance to the reliable clusters obtained at the training stage. Since our goal is to learn to determine unreliable segments, and not necessarily to come up with the state of the art object class segmentation system, we use a simple nearest neighbor classifier as opposed to more sophisticated techniques. Initially, each reliable segment is classified independently, and the results may not be spatially coherent. To encourage coherency and integrate the results of multiple segmentations, we use graph-cut optimization [23] to generate the final result. This leads to a significant improvement in accuracy, compared to single segmentation results.

Experimental results on MSRC database show that discarding unreliable regions results in a significant accuracy improvement. Even though the goal was not to come up with the state-of-the art classifier, surprisingly, a simple nearest neighbor classifier with reliable regions is not too far from the state-of-the art methods.

2 Related Work

Our work is most related to the methods that use multiple segmentation for various object recognition tasks. An early work that uses segmentation for recognition is [24]. Their goal is associating labels to image regions, and use multiple segmentation algorithms to evaluate which one is better for their task.

The method introduced in [6] seeks to discover visual object categories and their segmentation in a collection of unlabeled images starting with regions generated with several unsupervised image segmentation approaches. However, they make an assumption that is often violated in practice. They assume that at least one of the regions contains the full object.

Multiple segmentations were also used by [18] to label each pixel with its rough geometric surface class. Multiple segmentations contribute to a wider set of hypothesis to explore as compared to a single segmentation. Their work popularized the use of multiple segmentations for various computer vision tasks.

In [19], they use multiple stable segmentations to localized objects in a weakly supervised framework. A segmentation is considered stable if small perturbations of the image do not cause significant change in segmentation results. The idea is that stable segments might be more adequate for object recognition. However, stable regions could still correspond to generic blobs, not necessarily be useful for object recognition.

In [13], they classify the regions generated from multiple segmentations in supervised manner and then integrate all segments to get the final result. However in their approach they have an assumption that all the regions generated by multiple segmentations are beneficial for object recognition, which is likely not be true in practice.

The importance of incorporating multiple segmentations into object recognition, as opposed to a single segmentation, is investigated in [16]. They show two important results: good spatial support is important for recognition and multiple segmentations provide better spatial support for objects than bounding boxes and a single segmentation.

3 Proposed Approach

In this section, we describe our approach to the object class segmentation with reliable regions. In the training stage, reliable regions for each object class are determined and retained, and unreliable regions are discarded. In the test stage, unreliable regions are discarded as well. Reliable regions are classified with the nearest neighbor classifier using the pool of reliable regions obtained at the time of training. Finally region classifications are combined and regularized using graph cuts to get the final object map.

3.1 Obtaining Multiple Segmentations

To obtain multiple segmentations, we use two popular and efficient segmentation algorithms, FH [20] and the Mean-Shift [21]. Multiple segmentations are generated by varying the parameters of each algorithm. We obtain 18 distinct segmentations for the FH algorithm, and 16 for the mean-shift algorithm. Figure 1 shows some examples of segmentation images that are generated for one of the input images. This figure illustrates the possible variation in segmentation regions and different characteristics of the segmentation algorithms.

3.2 Features from Regions

We explored features that are based on size, location, color, texture, probability of the boundary, and edge histograms. Location, shape, color and texture features are the same as in [18].

Probability of the boundary is based on the boundary detector introduced in [25], which is better at detecting object boundaries than the conventional edge detection methods because it is less distracted by texture gradients. We take the mean value of probability of the boundary within a two pixel wide radius of the region boundary.

Distribution of intensity gradients can be useful for shape and appearance description. Therefore two kinds of edge descriptors are used to capture edge cues: Histogram of Oriented Gradient (HOG) [3] and Edge Orientation Histogram (EOH) [26]. In our work HOG and EOH are computed for 8 bins for all the pixels in each region and also for pixels around the boundary of regions within two pixels radius. They are normalized by the size of the region or the size of the boundary of the region, correspondingly.

The initial set of features has size 219. In order for clustering to have a reliable performance, it is necessary to reduce dimensionality [27]. We use the

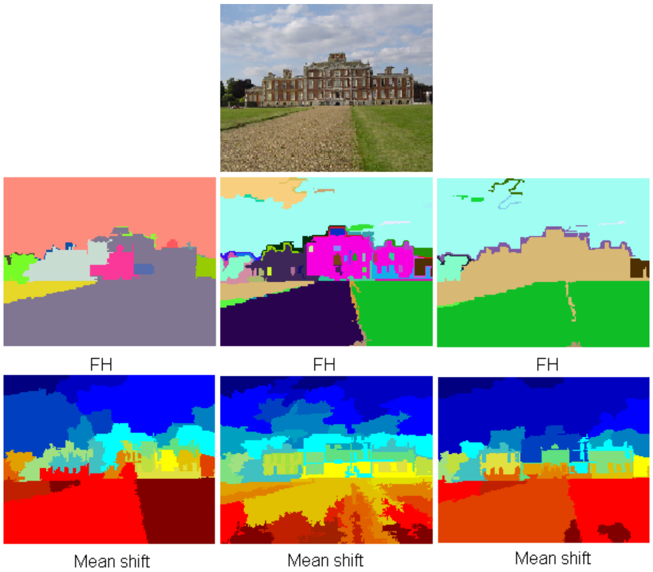

Fig. 1. Some examples of segmentation results from FH and Mean Shift algorithms with different input parameters.

Relief-F algorithm [22] for feature selection in this work. We retain the top 15 features ranked by Relief-F. These features are based on color, location, edge histogram and texture filters. It is interesting to note that boundary features were not selected.

3.3 Clustering

For robustness, we want to make a decision on the reliability of a whole group of similar segments, rather than decide for each segment separately. Therefore we need to perform segment clustering within each object class. A single object class is usually segmented into several possibly overlapping parts, sometimes bearing little similarity to each other. As illustrated in Figure 2 a cow generally gets segmented into the head, body, and legs. Therefore segments resulting from multiple segmentations of multiple instances of a single object class typically consist of several distinct clusters.

To facilitate discovery of reliable segments, first the clustering structure resulting from multiple segments of a single object class must be captured. We use k-means [28] to cluster each object class. The number of clusters for each object is chosen between 4 to 8 so that the resulting number of clusters maximizes the silhouette coefficient.

Figure 3 shows some representative samples for four out of six clusters from the cow class and three out of five clusters from the car class. Each cluster corresponds to a separate column. For cow class, the clusters tend to correspond

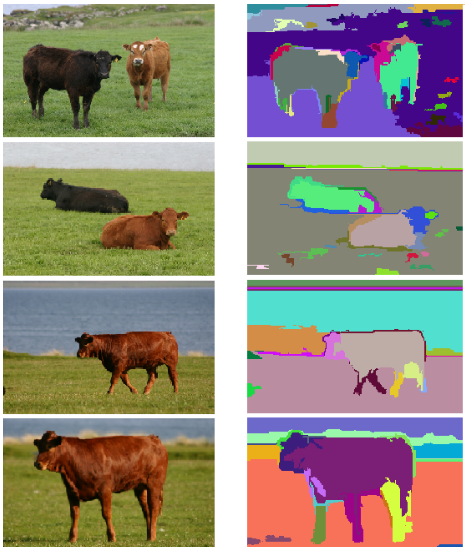

Fig. 2. First column shows the original images and the second column their typical segmentations.

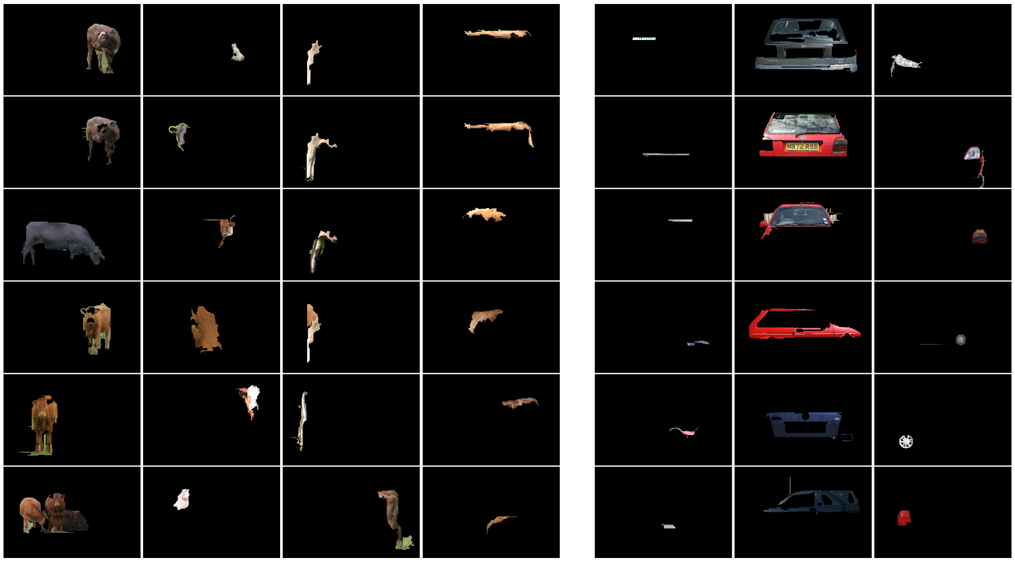

Fig. 3. Left: Four clusters from the cow class, right: 3 clusters from the car class. Each column corresponds to a separate cluster. Six representative members are shown for each cluster.

to bodies, heads, vertical segments, horizontal segments, respectively. For the car class, the clusters seem to correspond to the horizontal structures, body, and the “lights and wheels" parts.

3.4 Discarding Unreliable Segments

So far we used all regions generated from multiple segmentations in the clustering, however clusters from different objects could have lots of intersection with each other which makes them unreliable for object detection.

To get rid of unreliable clusters, those clusters that have a large amount of intersection with clusters of different object classes should be discarded. In our work, first the average intersection of each of the clusters with all other clusters of different classes is calculated and then for each object class two or more clusters with the least amount of intersection are retained. These clusters are reliable and will lead to more certain and stable decision at the time of object recognition.

There are many different ways to measure cluster overlap. Since the number of training samples is relatively small and the dimensionality of our data is relatively high, we avoid measuring cluster overlap in the original feature space to avoid the problems associated with high dimensionality. Instead we project the two clusters for which we want to measure the overlap into the best one dimensional subspace so that the samples from each cluster stay as separated as possible. This step is performed with Fisher linear discriminant [29]. According to Fisher linear discriminant, two clusters should be projected on the line in the direction V which maximizes:

$$J(V) = \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2} \tag{1}$$

μ is the mean and s^2 is the scatter of projection. The best V for projection is found with eigenvector analysis.

After two clusters are projected on the best line that separates them, the value of equation (2) can be used to quantify the overlap in the projected space:

$$J(c_i^l, c_j^k) = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}, \tag{2}$$

where c_i^l denotes cluster i from object class l; σ_1^2 and σ_2^2 are the variances of projections of two clusters. The larger is the value in equation (2), the better is the separation between the two clusters.

To measure how reliable one cluster is, the average of equation (2) is calculated over all clusters of other classes, see equation (3):

$$J(c_i^l) = \frac{\sum_{j,k \neq l} J(c_i^l, c_j^k)}{\iota}, \tag{3}$$

where ι is the number of clusters such that their object label is not equal to l. For each object class we choose two or more clusters with the best value of $J(c_i^l)$.

In Figure 3, for the cow class, the first and second clusters (corresponding to the first and second column) are reliable, and the last two are unreliable. For the car class, the first cluster is unreliable and the last two are reliable. It can be seen from the figure that the reliable clusters contain segments that are more easily recognizable as parts of the object class. Unreliable clusters tend to have segments that are more generic and shared between classes, such as blobs of certain eccentricity and orientation.

4 Object Class Segmentation

So far the process of creating a pool of reliable clusters from training images was discussed. Given a novel image to segment into its constituent object classes, we first generate multiple segmentations using FH and mean-shift algorithms. For each segment we extract fifteen features selected as discriminative at the training stage. We discard unreliable segments using the following procedure. We compute the distance from the current segment to the cluster centers obtained at the training stage. If the current segment is closest to an unreliable cluster center, it is itself deemed as unreliable and discarded. Otherwise it is classified using the k-nearest neighbor classifier. We set $k = 8$ for the experiments.

Notice that since we discard unreliable segments for a test image, there is a chance that a pixel does not get classified because it is contained only in unreliable segments. We found experimentally that the percentage of such pixels is small. In most cases, if a pixel is in unreliable generic blob in one segmentation, it often does belong to a reliable region for another segmentation, usually at a different scale. The small percentage of pixels that do not get classified at this stage do get classified at the next stage, which we now describe.

Since all the segments are classified independently, to improve coherence and integrate the results of multiple segmentations, we use graph cut optimization framework [23]. We seek a final labeling f that minimizes the following energy:

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} w_{pq} \cdot \delta(f_p \neq f_q) \tag{4}$$

The first term in equation (4) is called a data term and it is the penalty for pixel p to be assigned label f_p. D_p measures how well label f_p suits pixel p. This constraint ensures that the current labeling f is consistent with the observed data.

We set the data term as follows. The more often pixel p was classified as object class l, the smaller is $D_p(l)$, making object class l more likely for p. Specifically, $D_p(f_p)$ is defined as the negative number of segments that contain pixel p and get classified as f_p.

The second term in equation (4) is known as the smoothness term. The smoothness term measures the extent to which f is not smooth. We use the Potts Model. Here $\delta(f_p \neq f_q)$ is 0 if $f_p = f_q$ and 1 otherwise. A typical choice for w_{pq} is $\lambda e^{-\frac{(I_p - I_q)^2}{2\sigma^2}}$, where I_p and I_q are the intensities of pixels p and q.

Parameter σ is related to the level of variation between neighboring pixels. The parameter λ is used to control the relative importance of the data term versus the smoothness term. We set λ to 10 in this work. This energy function can be approximately optimized with the alpha-expansion algorithm of [23]. The code for alpha-expansion is taken from Boykov et al. [23, 30, 31].

Notice that if a pixel belongs only to unreliable segments, i.e. it did not get classified with any object class for any segmentation, then its data terms are equal to zero for all classes. Such pixel gets classified based only on propagation from it surrounding context due to the smoothness term.

5 Experimental Results

We tested our approach on the MSRC 21-class data set [32]. In this data set pixels close to object boundaries and in non-object regions are labeled void in the ground truth images. These pixels are ignored in evaluation. Since ground truth may contain different labels for any given segment, for training, we use only the segments that have more than 70% label consistency, that is at least 70% of pixels should belong to a single class.

Table 1 gives the quantitative results using graph cuts for integrating multiple segmentations. The second row shows the results when only reliable regions are used for object detection and the first row is the results of using all regions generated by all of the segmentations. As it can be seen, the results are better by 11% in pixel-averaged and class-averaged accuracy when just the reliable regions are used. For all object classes except the sheep, sky and boat object classes, using just the reliable regions outperform the approach where all regions are used. This shows that lots of regions do not have informative features to distinguish an object or an object part. Using all of these regions instead of just the reliable ones can mislead object recognition.

We also measure the improvement graph cut integration of multiple segmentations gives over single segmentation results. We compute accuracy improvement of graph cut results over each of the 34 single segmentation result. On average, the class-averaged accuracy increases by 7.8% with graph cuts, with standard deviation of 0.03%. The smallest improvement is 4.0% and the best improvement is 14.5%.

Figure 4 shows some successful qualitative results. Some partial successful examples or failures are shown in Figure 5. The first column contains the input images, the second column shows the ground truth segmentation. The third and fourth columns present maps of the most likely class at each pixel using all regions and reliable regions, respectively. The black pixels in the result images are considered as void in ground truth images and are not considered in our evaluation. These are the pixels for which no accurate ground truth could be constructed by the authors of the dataset. As can be seen from Figure 4 and Figure 5 using all regions leads to less accurate results as opposed to using only the reliable regions.

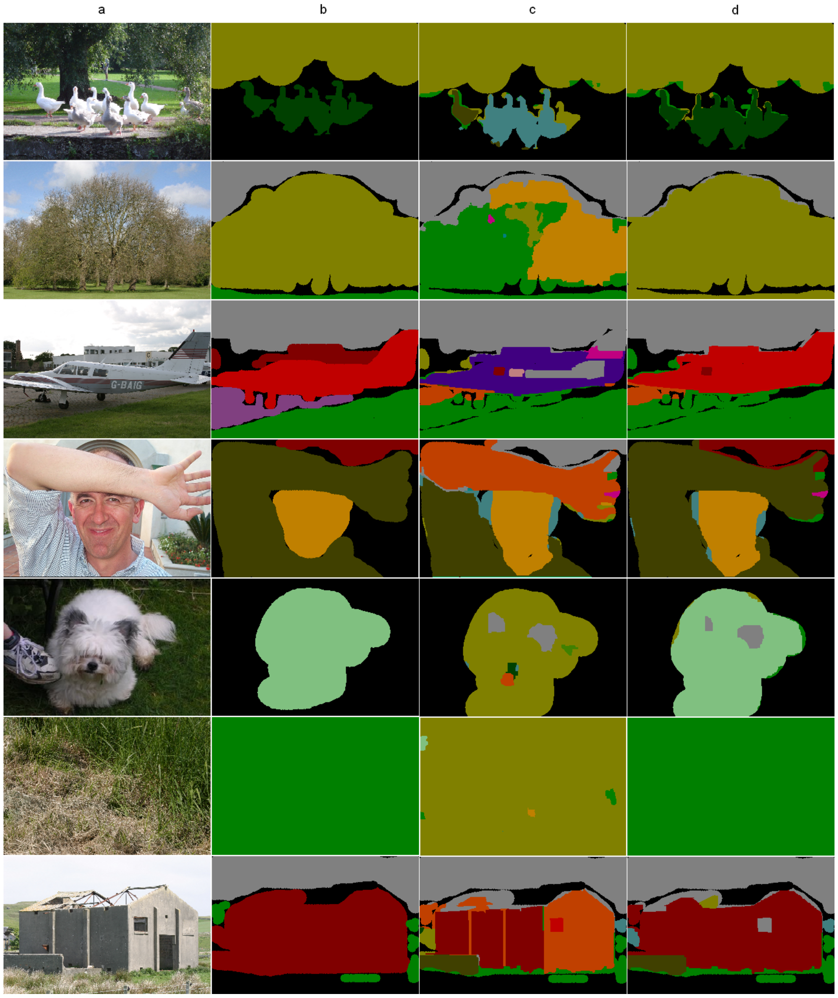

Fig. 4. Successful qualitative results generated using graph cuts for integrating multiple image segmentations. a) Original images. b) Ground truth images. c) Results generated using all the regions. d) Results generated using reliable regions. The black pixels in all result images are not labeled in ground truth images and therefore are not considered in our evaluation. This picture is best viewed in color.

Table 1. Quantitative performance using multiple segmentations. In the first row, all regions from multiple segmentations are used for recognition while in the second row just reliable regions are used.

	Pixel Average	Class Average	building	grass	tree	cow	sheep	sky	aeroplane	water	face	car	bike	flower	sign	bird	book	chair	road	cat	dog	body	boat
Graph cut (all regions)	57	44	47	81	60	23	49	96	31	46	60	41	39	65	35	22	43	24	63	30	11	33	31
Graph cut (reliable regions)	68	55	60	90	70	50	43	87	46	87	64	54	46	71	43	26	67	25	75	39	50	37	24

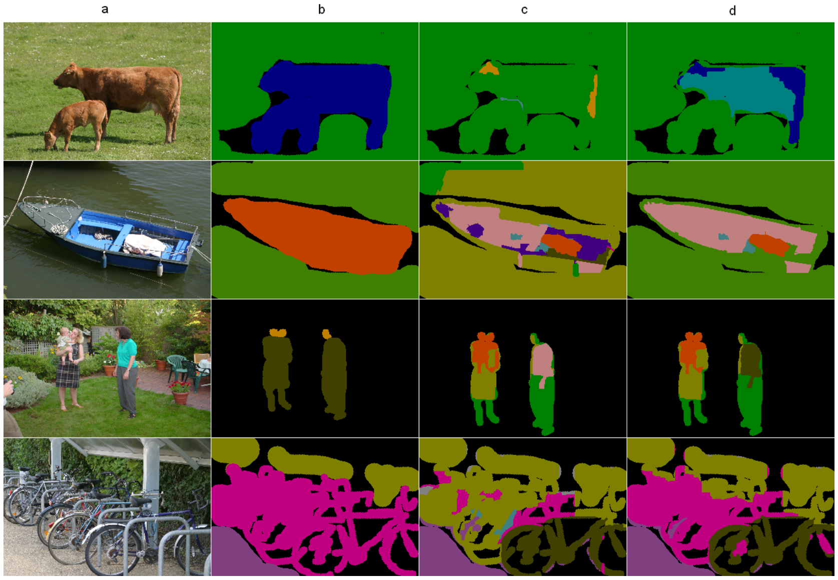

Fig. 5. Partially successful or failure results generated using graph cuts for integrating multiple image segmentations. a) Original images. b) Ground truth. c) Results generated using all the regions. d) Results generated using reliable regions. The black pixels in all result images are not labeled in ground truth images and therefore are not considered in our evaluation. This picture is best viewed in color.

5.1 Comparison to Previous Work

The purpose of this work was not to come up with the state-of-the art algorithm but to show that discarding unreliable regions from multiple segmentations gives better results for object recognition. This was shown using a simple nearest neighbor framework. Even though the goal was not a state of the art algorithm, we compare the performance of the proposed system to the state-of the art recent work [7, 8, 13]. Notice that they [7, 8, 13] use much more sophisticated machine learning algorithms and features. Nevertheless our overall performance is not that much worse, which is surprising, given the simplicity of our classifier. Furthermore, we are even able to achieve better accuracy for some classes. Our approach is superior on "water", "sign" , "bird" "chair", "dog" and "boat" object classes.

Table 2. The comparison of our approach with that of Shotton et al. [7], Verbeek and Triggs [8] and Pantofaru et al. [13].

	Pixel Average	**Class Average**	**building**	**grass**	**tree**	**cow**	**sheep**	**sky**	**aeroplane**	**water**	**face**	**car**	**bike**	**flower**	**sign**	**bird**	**book**	**chair**	**road**	**cat**	**dog**	**body**	**boat**
Shotton et al. [7]	72	58	62	98	86	58	50	83	60	53	74	63	75	63	35	19	92	15	86	54	19	62	7
Verbeek and Triggs [8]	74	64	52	87	68	73	84	94	88	73	70	68	74	89	33	19	78	34	89	46	49	54	31
Pantofaru et al. [13]	74	60	68	92	81	58	65	95	84	81	75	65	68	53	35	23	85	16	83	48	29	48	15
reliable regions	68	55	60	90	70	50	43	87	46	**87**	64	54	46	71	**43**	**26**	67	**25**	75	39	**50**	37	**24**

The running time of our approach could be improved significantly, it is currently in minutes. The most time consuming part is obtaining multiple segmentations and extracting features.

6 Conclusion

We develop a novel object recognition algorithm based on selecting reliable regions from multiple segmentations. We produce pixel-wise segmentation of the objects in the image. To increase the robustness of our system to poor segmentations and also in order integrate the results of multiple segmentations, information from multiple image segmentations is integrated using graph cut optimization framework. The main contribution of our work is a clustering approach to find the reliable regions for object recognition. Experiments show that when using only the reliable regions, the results improve by approximately 11% both in pixel-averaged and class-averaged accuracy.

References

1. Rowley, H., Baluja, S., Kanade, T.: Human face detection in visual scenes. Advances in Neural Information Processing Systems (1996) 875–881
2. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision **57** (2004) 137–154
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Conference on Computer Vision and Pattern Recognition. (2005) 886ff
4. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **30** (2008) 36–51
5. Lampert, C., Blaschko, M., Hofmann, T., Zurich, S.: Beyond sliding windows: Object localization by efficient subwindow search. In: Conference on Computer Vision and Pattern Recognition. (2008)
6. Russell, B., Efros, A., Sivic, J., Freeman, W., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: Conference on Computer Vision and Pattern Recognition. (2006) 2:1605–1614
7. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. European Conference on Computer Vision (2006) 1–15
8. Verbeek, J., Triggs, B., INRIA, M.: Region classification with markov field aspect models. In: Conference on Computer Vision and Pattern Recognition. (2007) 1–8
9. Verbeek, J., Triggs, B.: Scene segmentation with conditional random fields learned from partially labeled images. In: Proc. NIPS. (2008)
10. He, X., Zemel, R.: Learning hybrid models for image annotation with partially labeled data. Advances in Neural Information Processing Systems (NIPS) (2008)
11. Schroff, F., Criminisi, A., Zisserman, A.: Object class segmentation using random forests. (2008)
12. Gould, S., Rodgers, J., Cohen, D., Elidan, G., Koller, D.: Multi-class segmentation with relative location prior. International Journal of Computer Vision **80** (2008) 300–316
13. Pantofaru, C., Schmid, C., Hebert, M.: Object recognition by integrating multiple image segmentations. In: European Conference on Computer Vision, Springer (2008) 481–494
14. Gu, C., Lim, J., Arbelaez, P., Malik, J.: Recognition Using Regions. In: Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, In Proc. CVPR (2009) 1030–1037
15. Fulkerson, B., Vedaldi, A., Soatto, S.: Class segmentation and object localization with superpixel neighborhoods. In: IEEE International Conference on Computer Vision. (2009) 670–677
16. Malisiewicz, T., Efros, A.: Improving spatial support for objects via multiple segmentations. In: British Machine Vision Conference (BMVC). Volume 2., BMVC (2007)
17. Todorovic, S., Ahuja, N.: Learning subcategory relevances for category recognition. In: Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2008) 1–8
18. Hoiem, D., Efros, A., Hebert, M.: Recovering surface layout from an image. International Journal of Computer Vision **75** (2007) 151–172
19. Galleguillos, C., Babenko, B., Rabinovich, A., Belongie, S.: Weakly supervised object localization with stable segmentations. In: European Conference on Computer Vision. (2008) I: 193–207

20. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. International Journal of Computer Vision **59** (2004) 167–181
21. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on pattern analysis and machine intelligence (2002) 603–619
22. Kira, K., Rendell, L.: A practical approach to feature selection. In: Proceedings of the ninth international workshop on Machine learning table of contents, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA (1992) 249–256
23. Boykov, Y., Veksler, O., Zabih, R.: Efficient approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (2001) 1222–1239
24. Barnard, K., Duygulu, P., Guru, R., Gabbur, P., Forsyth, D.: The effects of segmentation and feature choice in a translation model of object recognition. In: Conference on Computer Vision and Pattern Recognition. (2003) 675–682
25. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Transactions on Pattern Analysis and Machine Intelligence (2004) 530–549
26. Levi, K., Weiss, Y.: Learning object detection from a small number of examples: The importance of good features. In: Conference on Computer Vision and Pattern Recognition. Volume 2. (2004)
27. Berkhin, P.: A survey of clustering data mining techniques. Grouping Multidimensional Data (2006) 25–71
28. Hartigan, J.: Clustering algorithms. Wiley New York (1975)
29. Fisher, R.: The use of multiple measures in taxonomic problems. Ann. Eugenics **7** (1936) 179–188
30. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence (2004) 1124–1137
31. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004) 147–159
32. Shotton, J., Winn, J., Rother, C., Criminisi, A.: The MSRC 21-class object recognition database. (2006)