Piecing Together the Segmentation Jigsaw using Context

Xi Chen[†], Arpit Jain[†], Abhinav Gupta[§], Larry S. Davis[†]
[†]University of Maryland College Park, MD,20740
[§]Carnegie Mellon University, PA, 15213

Abstract

We present an approach to jointly solve the segmentation and recognition problem using a multiple segmentation framework. We formulate the problem as segment selection from a pool of segments, assigning each selected segment a class label. Previous multiple segmentation approaches used local appearance matching to select segments in a greedy manner. In contrast, our approach formulates a cost function based on contextual information in conjunction with appearance matching. This relaxed cost function formulation is minimized using an efficient quadratic programming solver and an approximate solution is obtained by discretizing the relaxed solution. Our approach improves labeling performance compared to other segmentation based recognition approaches.

1. Introduction

We describe an approach that jointly segments and labels the principal objects in an image. Consider the image in figure 1. Our goal is to locate and pixel-wise label the principal objects such as car, building, road and sidewalk. One approach is to first segment the image, then perform recognition using appearance and context. However, there are generally no reliable algorithms for segmentation. For example, for the image shown in Figure 1, segmentation algorithms will generally not combine the roof and the body of the car into one segment due to differences in appearances. Therefore, there has been a recent trend to simultaneously address segmentation and recognition.

For example, some recent approaches construct the segments by selectively merging superpixels while simultaneously labeling these elements. However, at the superpixel level global image features such as shape cannot be easily employed. So, while these approaches show high performance for "stuff"-like objects such as grass - they often fail to identify objects which require shape cues for identification. To harness shape features, approaches such as [5, 14] have instead started with an initial segmentation and then refined these segments iteratively. However, the modifications are generally local in nature and tend to get stuck in local minima.

To overcome these problems, recent approaches have advocated the use of multiple segmentations [7, 20]. Recognition, then, involves selecting the best segments. These methods use only appearance features to select segments and the best overall labeling is constructed in a greedy manner. They ignore context, which is important for accurate segment selection and labeling. For example, the window of the car is labeled as "airplane" because the context from other scene elements such as road, sidewalk and building are ignored.

We propose an approach to select the best segmentation and labeling in a single optimization procedure that utilizes context to perform segment selection and labeling coherently. To overcome the fragmentation problem, we allow connected segments to be merged based on local color, texture and edge properties. We also include mid-level cues to constrain the solution space - for example, the segment merging step leads to overlapping segments, and we restrict global solutions to exclude overlapping segments (avoiding the possibility of multiple labeling for pixels). By incorporating contextual relations between region pairs, we find the subset of segments that best explains the image. For example, in Figure 1, our approach correctly selects the combined region of window and body segments and labels it as "car". The labeling of the window segment as "airplane" is not chosen due to contextual constraints from sidewalk, road and building.

The contributions of our paper are: (a) An approach to incorporate contextual information in a multiple segmentation framework, and (b) Increasing the spatial support[1] of image labeling by constructing additional segments from a base pool, at the cost of only a small increase in segment pool size.

[1]Spatial support measures the quality of pool of segments as compared to ground truth. The score is higher if the segments in the ground-truth find segments in the pool with high overlap.

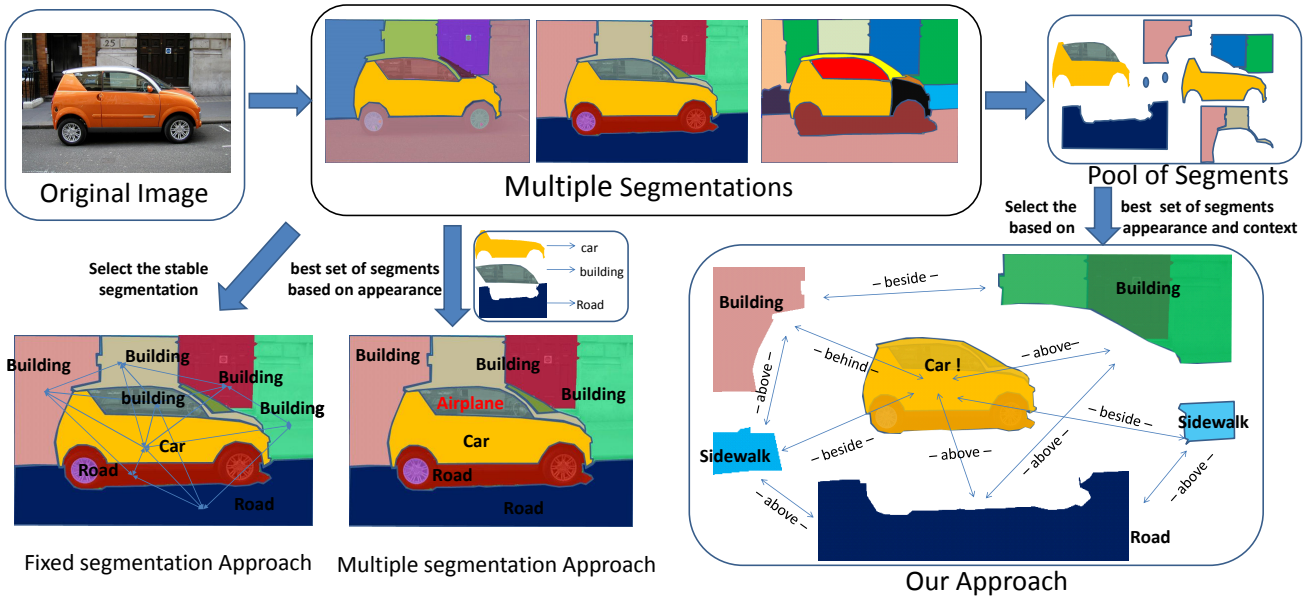

Figure 1. Comparison of our approach to fixed and multiple segmentation algorithms. Our approach solves the problem of segmentation and recognition jointly using appearance and context. The figure shows how global contextual relations help to select the whole car segment subset over other fragmented pieces of car, as their association does not satisfy context.

2. Related Work

The problem of image parsing has a long history in computer vision dating back to the 1970's. Unlike Marr's sequential processing pipeline, where segmentation from bottom-up cues preceded recognition, Tenenbaum and Barrow proposed Interpretation-Guided Segmentation [27] which labeled image regions using constraint propagation to arrive at a globally consistent scene interpretation. This was followed by development of complete scene understanding systems such as ACRONYM [1] and VISONS [8]. During the last decade, researchers in visual recognition have made significant advances in object recognition due to better appearance modeling techniques and visual context. These approaches can be broadly categorized into three categories based on how interactions between segmentation and recognition are modeled:

Pixel Based Approaches: These approaches model the problem of visual recognition at the pixel level [9, 25, 26, 29] and therefore the problem of segmentation is solved implicitly (neighboring pixels belonging to different class represent boundary pixels). One of the major shortcomings of pixel-based approaches is that many objects (such as cars) are defined in large part by their shape and therefore categorization at the pixel-level using local appearances without global shape analysis performs poorly.

Fixed Segmentation Approaches: These approaches classify individual regions in some fixed image segmentation based on region color, texture and shape [6, 4, 11]. However, obtaining semantically meaningful segmentations without top-down control is well beyond the state of the art.

Image Parsing (Joint Segmentation and Recognition): These approaches jointly solve segmentation and recognition. Approaches such as [23, 20] obtain multiple segmentations of the image and model the problem of segmentation and recognition as the selection of segments based on their matches to semantic classes. On the other hand, approaches such as [5, 14, 18] start from an imperfect segmentation and then refine it iteratively by optimizing a cost function defined on segments and appearance matchings. One of the shortcomings of these approaches is that they tend to get stuck in local minima due to local refinement. [16, 15] proposed super pixel based approaches where the class labels are inferred based on local appearance and context using CRFs. Such approaches fail to incorporate higher level shape information; additionally learning CRF's parameters has proven to be difficult. In [28] segmentation was combined with the responses of sliding window object detectors for image labeling to avoid fragility of segmentation.

3. Overview

Multiple segmentation approaches construct a pool of initial segments by varying the controlling parameters of a segmentation algorithm or by starting from a coarse segmentation and iteratively refining the segmentation by merging or further segmenting initial segments. They gen-

erally assume that each object will be well segmented at some parameter setting or level. [19] pointed out that merging small connected subsets (pairs and triples) of base segments improves recognition performance. However, the algorithm in [19] employed manually choosing the segments to merge. One could simply join all possible pairs and triples of connected segments but this would lead to an explosion in the segment pool size. In contrast, we construct a "good" set of mergings using a classifier which rejects combination which are unlikely to correspond to "complete" objects (section 4).

We organize these segments into a hierarchical segment graph for recognition. The graph structure allows us to impose constraints that reduce the combinatorics of the search process - for example, that a solution cannot include overlapping segments, since this could lead to pixels being given multiple labels.

Given the segment graph, we compute pairwise and higher-order constraints on selection of segments. We then formulate a cost function which accounts for local appearance and enforces pair-wise contextual relationship consistency (such as sky above water, road below car, etc). Directly optimizing this cost function is NP hard so the cost function is approximately minimized by first relaxing the selection problem. The relaxed problem can be solved efficiently by quadratic programming (QP). The relaxed solution is then discretized to obtain the final labeled segmentation (section 5). Finally, we evaluate the performance of our approach with previously reported methods (section 6).

4. Constructing the Segment Graph

Obtaining the Initial Segment Pool: We use the hierarchical segmentation algorithm from [24] to construct the segment pool. To increase the robustness of the segmentation algorithm, we use the stability based clustering analysis of [22]. Stability analysis selects segments which are stable under small perturbations (noise) to the image.

In the first step, image is segmented and the segments in the first hierarchical level are added to the segment pool. Then each of these segments is iteratively segmented and the smaller segments are added to the segment pool until any of the following conditions are met. (1) The segment size is too small ($< 2\%$ of total image pixels). (2) The integrated edge strength along the boundary of the segment (obtained by Berkeley edge detector [21]) is below a threshold. (3) The number of leaf nodes in the segment subgraph rooted at the original segment exceeds a threshold.

This procedure gives us initial segment pool over which we will perform segment selection.

Merging Segments: The base segmentation algorithm seldom produces segments that directly correspond to the objects in the image. Hence, we merge small (2 and 3) connected sets of segments from the segment pool to obtain a better collection of segments. But allowing all possible segment merges would explode the size of the pool. To limit the number of pairs and triples merged, we learn a function that scores these small subsets from a training set of fully labeled images.

A Support Vector Regression (SVR) [2] model using radial basis functions is learned from the training images to score potential merges. We compute color, texture and edge features similar to those used by Hoiem et. al. [10] for each segment of an object. Based on these features, the SVR predicts whether the segments should be merged or not. Training images are segmented using the segmentation algorithm described above and a segment pool is obtained for each image. Objects which are broken into multiple segments are determined using the ground truth segmentation. These fragmented objects provide positive examples and the negative examples are obtained using random samplings from the training data. For a testing image, each adjacent pair and connected triple[2] of segments is evaluated for merging using the regression model learned, providing a score for each merging. The pairs and triples with scores above a threshold are added to the segment pool.

We evaluated the merging scheme on the 256 test images in the MSRC dataset. Figure 2 shows the spatial support in the pool with increasing pool size. The pool size is increased by lowering the threshold at which mergings are accepted. To demonstrate that the SVR learns an informative merging function, we compare the spatial support metric when the segment pool is enlarged using random merges (red curve in Figure 2). Although spatial support increases (which it obviously must), it does so at a much slower rate than the SVR.

Construction of the Segment Graph: The pool of segments are then arranged in a hierarchical graph structure to which our inference algorithm will subsequently be applied. The graph structure is constructed as follows: The root node is assigned to the whole image. A segment S_i is a child of segment S_j if segment $S_i \subset S_j$. If two segments S_i and S_j are subsets of a S_k then both the segments are children of segment S_k. The segments which have no smaller segment subsets are leaf nodes.

5. Piecing together the Segments

Our goal is to select a set of segments from the pool such that each segment has high overlap with a ground-truth segment and is assigned its correct label.

We formulate a cost function which evaluates any possible selection and labeling of segments from the pool. Each segment, S_i in the pool is associated with a binary variable X^i which represents whether or not the segment is selected.

[2] triples of segments are constructed by evaluating mergings of a segment from the initial pool with an adjacent segment formed from the pairwise merging step.

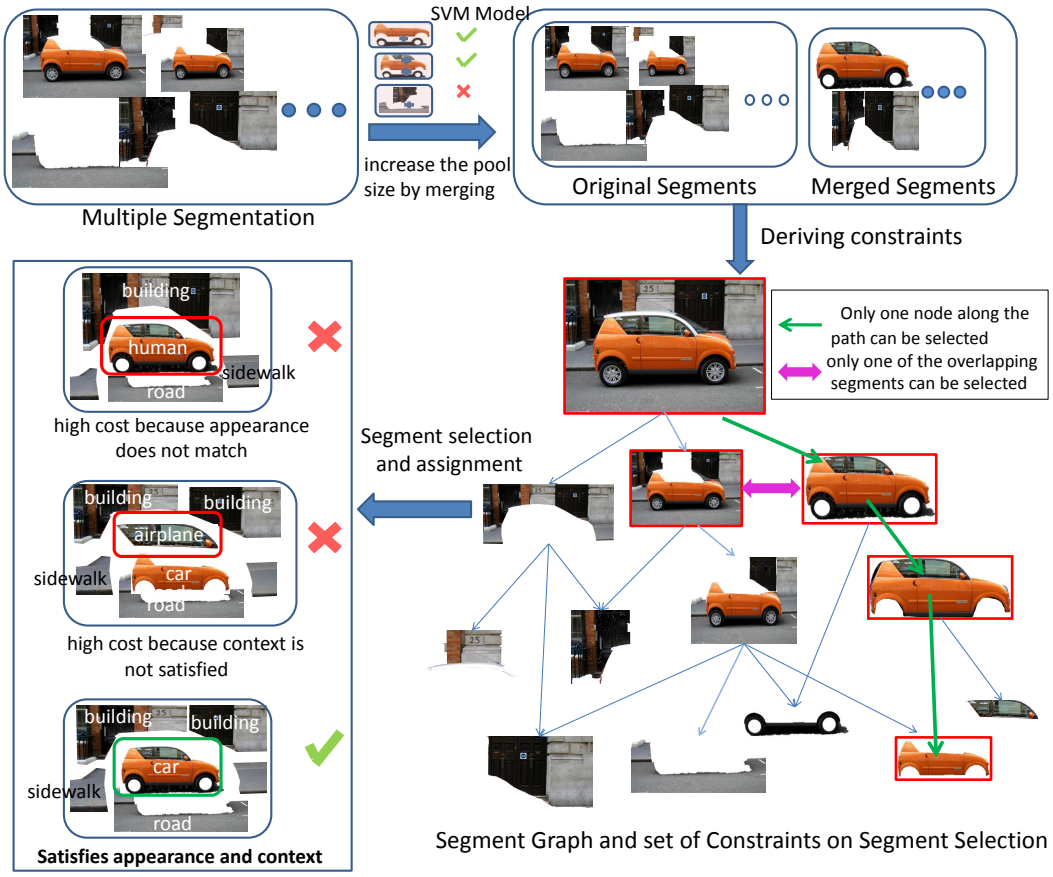

Figure 3. Our approach: We first create a pool of segments using multiple segmentations of an image and merging some of the connected pairs and triples of these segments. These segments are arranged in a graph structure where path constraints are used to obtain selection constraints. An example of a path constraint is shown using green edges: only one segment amongst all the segments in the path can be selected. The magenta arrow shows that two segments which overlap cannot be selected simultaneously. Finally, the QP framework is used to find the set of segments, together with their labels, which minimizes the cost function given the constraints

With each selected segment we also associate a set of C binary variables, $(X_1^i ... X_C^i)$, which indicates the label associated with the segment. $X_j^i = 1$ represents that segment i is labeled with class j. Our goal is to choose X^i such that the cost-function $\mathcal{J}$ is minimized, where $\mathcal{J}$ is defined as:

$$\mathcal{J} = \sum_{i,j} -w_1 A_{ij} X_j^i - \sum_i w_2 S_i X^i + \sum_{i,j} \sum_{k,l} w_3 X_j^i P_{ijkl} X_l^k \quad (1)$$

The cost function consist of three terms. The first term uses an appearance based classifier to match the appearance of selected segments with their assigned labels. The second term is the explanation reward term which rewards the selection of segments proportional to their size. The third term is a context satisfaction term which penalizes assignments which do not satisfy the contextual relationships learned from the training data. We discuss each of these terms below. The weight w_1,w_2,w_3 are obtained by cross validation on a small dataset and for our experiments we use 1, 1.5 and 0.5 respectively.

5.1. Constraints on Segment Selection

While there are 2^{N_S} possible selections (where N_S is the number of segments in the pool), not all subsets represent valid selections. For example, if segment i is selected and assigned label j, then other segments which overlap with segment i should not be selected to avoid multiple labeling of pixels. Figure 3 shows the overlap constraint by a magenta arrow where the two car segments which overlap cannot be chosen simultaneously. Similarly, two segments along a path from the root to any leaf node cannot be selected together. Figure 3 shows one such path constraint in green, where selection of the car and its subset segments simultaneously is prohibited.

These constraints are represented as follows:

$$0 \leq X^i + X^k \leq 1 \;\; \forall (i,k) \in O \quad (2)$$

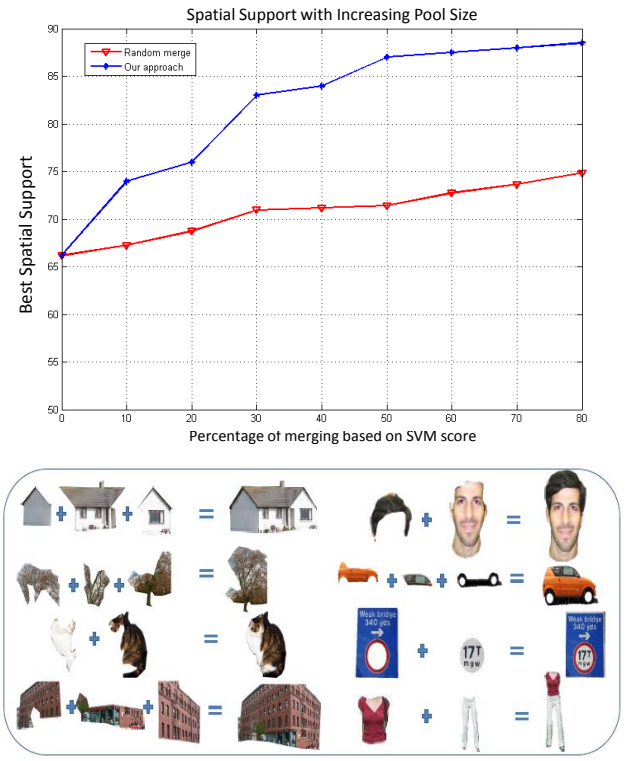

Figure 2. Graph on top shows the improvement in spatial support with increase in pool size. Image below the graph shows the instances where SVR model correctly merged fragmented segments of objects in the pool to complete the object segment.

$$0 \leq X^{p_1} + X^{p_2} .... X^{p_m} \leq 1 \quad \forall p \in \mathcal{P} \tag{3}$$

where O represents the set of pairs of regions in the graph that overlap spatially and $\mathcal{P}$ represents the set of paths from the root to the leaves in the segment graph. Additional constraints that are enforced while minimizing the cost function $\mathcal{J}$ include:

$$0 \leq X^i \leq 1 \tag{4}$$

$$\sum_j X^i_j = X^i \tag{5}$$

These constraints allow only one label to be assigned to each selected segment.

5.2. Cost Function

We now explain the individual terms in the cost function.

Appearance Cost: The first term in the cost function evaluates how well the appearance of the selected segment i associated with label j matches the appearance model for class j. For computing A_{ij}, we learn an appearance model from training images using a discriminative classifier over visual features. We use the appearance features from [10] and learn a discriminative probabilistic-KNN model as in [13, 12] for classification.

Explanation Reward: This term rewards selecting a segment proportional to its size, represented by S_i. This term avoids the trivial solution where no segment gets selected by the algorithm.

Contextual Cost: The third term evaluates the satisfaction of contextual relationships for a given selection of segments and their label assignment. We model context by pair-wise spatial and contextual relationships as in [6]. If segment i is assigned to class j and segment k is assigned to class l, P_{ijkl} measures the contextual compatibility based on co-occurrence statistics of classes j and l. We also evaluate spatial contextual compatibility by extracting the pairwise-differential features as in [6] for segments i and k and comparing them with a learned model of differential features for labels (j, l). For example, if the labeling is such that sky occurs below water then the penalty term is kept high and vice-versa. The penalty term is defined as:

$$P_{ijkl} = C_1 \exp(\frac{(d_{i,k} - \mu_{j,l})^2}{2\sigma^2_{j,l}}) + C_2 \exp(-\alpha M_{j,l}) \tag{6}$$

where C_1, C_2 and α are constants. $d_{i,k}$ is the differential feature between segment i and segment k. $\mu_{j,l}$ is the mean differential feature obtained from training between class labels j and l. The term $M_{j,l}$ represents the co-occurrence of classes j and l, also obtained from training. We employ eight differential features - Δx,Δy,$\Delta \mu_{red}$,$\Delta \mu_{green}$,$\Delta \mu_{blue}$,$\Delta \mu_{brighter}$, adjacency and overlap.

5.3. Optimization

For optimizing the cost function, we relax the binary variables X^i and X^i_j to lie in [0, 1]. We use the Integer Projected Fixed Point (IPFP) algorithm [17] to minimize the cost function. The solution generally converges in 5-10 steps, which makes it very efficient, while outperforming current state-of-the-art methods for inference. IPFP solves quadratic optimization functions of the form:

$$x'^* = argmax(x'^T M x') \;\; s.t. \, Ax' = 1, \; x' \geq 0 \tag{7}$$

To use the IPFP algorithm, we transform the original equation 1 into 7 through the following substitution: $x' = \begin{pmatrix} 1 \\ X \end{pmatrix}$ and $M = \begin{pmatrix} 0 & (A+S)^T/2 \\ (A+S)/2 & -P \end{pmatrix}$. The path constraints discussed in section 5.1 are incorporated as constraints in a linear solver during step 2 of the optimization algorithm.

6. Experiments

We evaluated the performance of our algorithm on three standard dataset: Label Me subset (used in [11]), PASCAL VOC 2009 [3] and MSRC [26].

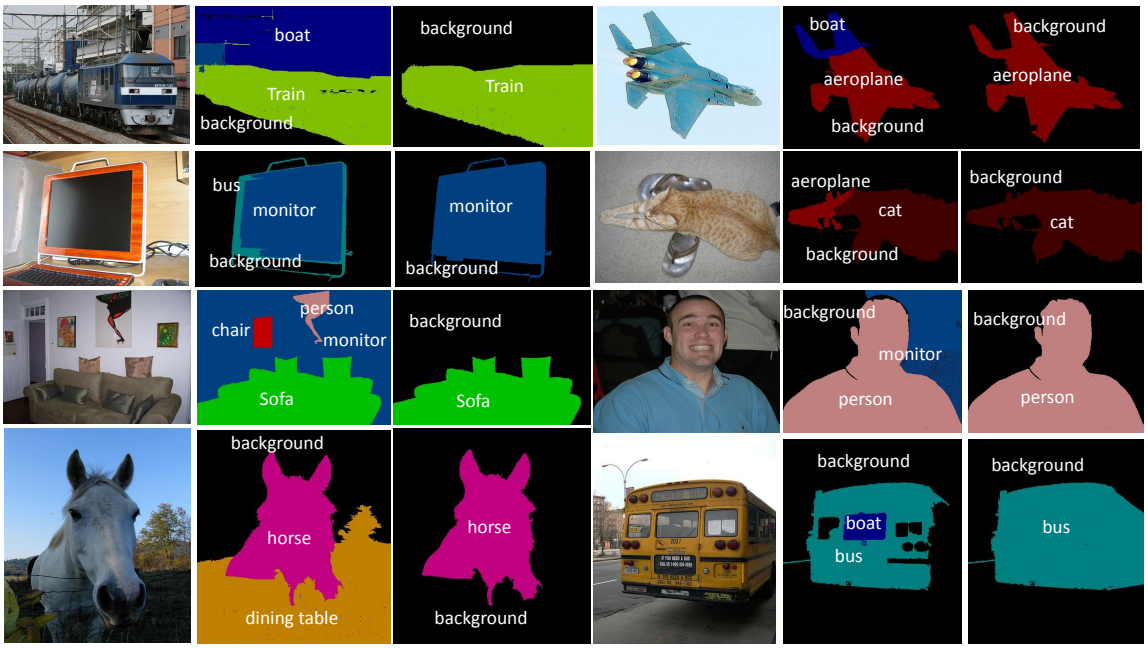

Figure 4. PASCAL VOC'09 labeling results. Columns (a) and (d) - original images. Columns (b) and (e) show the performance of appearance based approach without context. Columns (c) and (f) show the performance of our algorithm with context. Best viewed in color.

Table 1. Performance comparison of our algorithm against previous approaches on PASCAL VOC09 dataset.

	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining Table	Dog	Horse	Motor Bike	Person	Potted Plant	Sheep	Sofa	Train	TV/Monitor	Average
Hierarchical CRF [15]	77.7	38.3	9.6	**24.0**	35.8	**31.0**	59.2	36.5	21.2	8.3	1.7	22.7	14.3	17.0	26.7	21.1	15.5	16.3	14.6	48.5	33.1	27.3
Hierarchical CRF with CO [15]	82.3	**49.3**	**11.8**	19.3	37.7	30.8	**63.2**	46.0	23.7	10.0	0.5	23.1	14.1	22.4	33.9	35.7	**18.4**	12.1	**22.5**	**53.1**	37.5	30.8
Ours (w/o Context,w/ Merging)	76.4	25.6	8.0	14.2	**47.3**	8.1	30.5	53.7	50.1	**18.6**	9.1	48.5	10.9	15.8	33.8	47.3	10.2	15.7	11.2	48.6	35.2	29.5
Ours (w/ Context, w/o Merging)	61.2	37.3	5.5	20.6	36.0	14.6	30.8	55.3	46.8	10.6	4.2	40.2	11.3	17.3	29.0	36.1	9.1	29.3	12.8	47.4	38.2	28.3
Ours (Context, w/ Merging)	**85.8**	39.8	7.6	18.4	45.0	8.4	44.6	**66.1**	**54.2**	11.2	**10.3**	**52.7**	**15.2**	**23.5**	**39.2**	**50.8**	11.5	**31.5**	19.8	40.4	**48.9**	**34.5**

LABEL-ME: [11] used a subset of LABEL ME containing 350 images - 250 training and 100 testing. The dataset contains 19 classes. Performance is measured using the two standard measures from [11]. For comparison, we also evaluate four approaches in addition to those compared in [11] (1) Our multiple segmentation framework, but without contextual information. (2) A fully connected MRF-model similar to [4], which performs recognition using context on a fixed segmentation obtained using stability analysis. (3) A Texton-boost approach [3] without the CRF model, and 4) our method applied to the initial segment pool, but without the SVR merged segments.

Figure 5 shows a few qualitative examples of our approach. When context is not utilized many small segments are mislabeled and matched to wrong object classes. However, when context is added many of these errors are eliminated.

[3]http://jamie.shotton.org/work/code/

Table 2 shows the quantitative performance of our approach compared with these four methods and [11] using the two standard evaluation metrics. Our approach has a pixel-wise accuracy of 75.6%; when only appearance is used the performance falls to 65.23%. This shows that contextual information is critical not only for recognition but also for segment selection. As expected, the fixed segmentation MRF model has a low pixel-wise accuracy of 54.2%. The publicly available version of Texton-boost achieves just 49% pixel-wise accuracy. This is because Texton-boost relies on pixel-based appearance models. These are adequate for modeling regions like 'grass' and 'sky' but perform poorly for objects whose recognition requires cues such as shape.

PASCAL VOC 2009: The PASCAL VOC 2009 dataset [3] consists of 1499 images which is split into 749 images for training and 750 images for validation. We follow the protocol used by [15] to compare against the state

Table 2. Performance comparison of our algorithm against other approaches on LabelMe dataset.

	Texton-boost	MRF based	Jain et. al. [11]	Ours (no Context, Merging)	Ours (Context, no Merging)	Ours (Context,Merging)
pixel wise	49.75	54.2	59.0	65.23	71.9	75.6
class wise	20	30.2	–	38.5	43.5	45

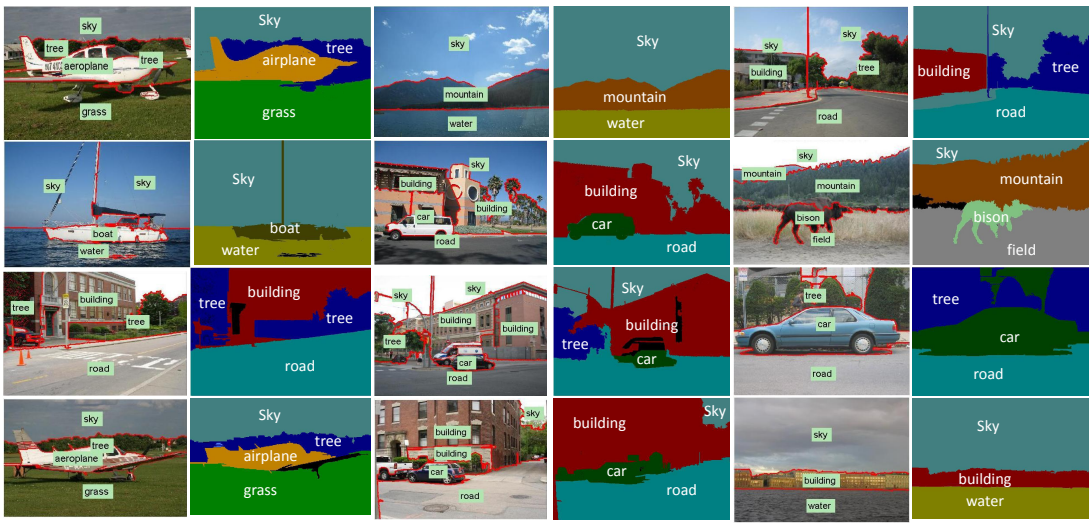

Figure 5. LabelMe dataset results - columns 1, 3 and 5 show the original image with object labels obtained by our algorithm and columns 2, 4 and 6 show the corresponding image segmentation.

of the art, and use the same evaluation metric as [15]. Table 1 shows the class wise performance of our approach compared with the other approaches. Our approach outperforms previous approaches on many classes which shows that it generalizes to a large number of object classes. Our better performance on classes like Car, Cat, Horse, Sheep, Cow, Monitor, Dog and Person supports our contention that a multiple segmentation approach performs better on object classes for which shape is important. Table 1 also shows that both context and merging improves recognition by choosing segments which have better spatial support.

Figure 4 shows some qualitative results on VOC 2009. Columns (b) and (e) show the labeling performance of our algorithm solely based on appearance. The algorithm using only appearance leads to a variety of errors such as the wing of the aeroplane being labeled as boat, the ground in the horse image as dining table, and the painting above the sofa as a person. Columns (c) and (f) show the performance of our approach with context. Figure 6 compares qualitative results of our algorithm with and without mergings and elucidates the importance of merging for better recognition. For example, in the sign image, the parts of the sign board are labeled as water and building but after merging them, it is correctly labeled as sign board.

MSRC dataset: Our algorithm achieved 75% (pixel-wise) and 68.7%(classwise) on the MSRC dataset, which is comparable to state-of-the-art results except [15]. MSRC is relatively simple and does not significantly benefit from the use of multiple segmentations. Our approach performs better than [15] for classes like bird, car and cow, where multiple segmentation and merging helps by creating segments whose shapes are closer to class models, but performs poorer on "stuff" classes such as grass and sky.

7. Conclusion

We described an approach for simultaneous segmentation and labeling of images using appearance and context. The optimization criteria developed was solved by relaxing the discrete constraints and employing a quadratic programming method. The relaxed solution was then discretized (and additional constraints were introduced) using a greedy algorithm. Experiments on three well studied datasets demonstrated the advantages of the method.

Acknowledgement: This research is supported by ONR Grant N000141010766 and MURI Grant N00014101093.

References

[1] R. Brooks, R. Greiner, and T. Binford. Model-based three-dimensional interpretation of two-dimensional images. *In Proc. Int. Joint Conf. on Art. Intell.*, 1979.

[2] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. 2001.

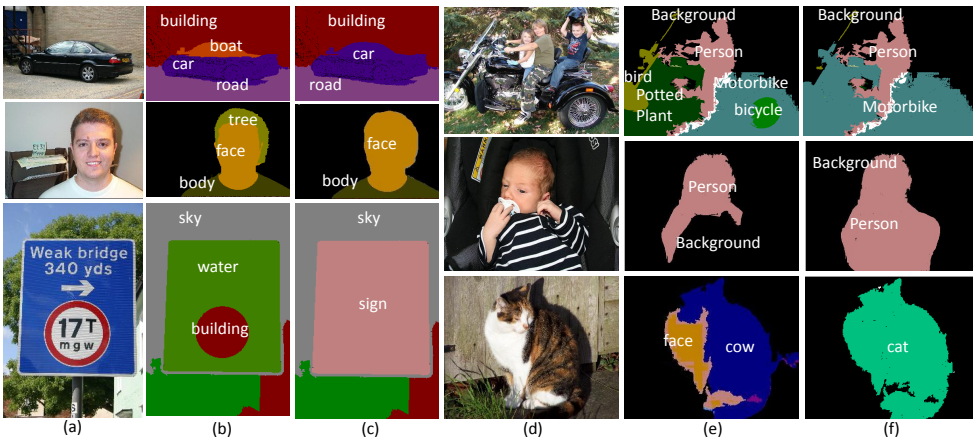

Figure 6. Qualitative results of our algorithm with and without merging. Columns (a) and (d) are original images. Columns (b) and (e) show the labeling performance without merging. Columns (c) and (f) show performance with merging. Best viewed in color.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2009 (voc2009) results.

[4] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *Proc. CVPR*. 2008.

[5] S. Gould, T. Gao, and D. Koller. Region-based segmentation and object detection. In *NIPS*. 2009.

[6] A. Gupta and L. S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *Proc. ECCV*. 2008.

[7] A. Gupta, A. Efros, and M. Hebert. Block world revisited: Image understanding using qualitative geometry and mechanics. In *In ECCV*. 2010.

[8] A. Hanson and E. Riseman. Visions: A computer system for interpreting scenes. *In Computer Vision Systems.*, 1978.

[9] X. He, R. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *ECCV*. 2006.

[10] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*. 2005.

[11] A. Jain, A. Gupta, and L. S. Davis. Learning what and how of contextual models for scene labeling. In *ECCV*. 2010.

[12] P. Jain and A. Kapoor. Probabilistic nearest neighbor classifier with active learning. *Microsoft Research, Redmond*.

[13] P. Jain and A. Kapoor. Active learning for large multi-class problems. In *IEEE CVPR*. 2009.

[14] M. P. Kumar and D. Koller. Efficiently selecting regions for scene understanding. In *NIPS*. 2010.

[15] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Graph cut based inference with co-occurrence. In *ECCV*. 2010.

[16] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*. 2009.

[17] M. Leordeanu, M. Hebert, and R. Sukthankar. An integer projected fixed point method for graph matching and map inference. In *Advances in NIPS*. 2009.

[18] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding:classification, annotation and segmentation in an automatic framework. In *CVPR*. 2009.

[19] T. Malisiewicz and A. A. Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*. 2007.

[20] T. Malisiewicz and A. A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*. 2008.

[21] D. Martin, F. C., and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Tran. on PAMI*, 26:530–549, 2004.

[22] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *Proc. ICCV*. 2007.

[23] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *ECCV*. 2006.

[24] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt. Hierarchy and adaptivity in segmenting visual scenes. *In the journal of Nature*, 442(7104):719–864, 2006.

[25] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Proc. IEEE CVPR*. 2008.

[26] 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. ECCV*. 2006.

[27] J. M. Tenenbaum and H. G. Barrow. Experiments in interpretation guided segmentation. *Journal of Artificial Intelligence*, 8(3):241–274, 1977.

[28] A. Torralba, K. Murphy, and W. Freeman. Contextual models for object detection using boosted random fields. In *NIPS*. 2005.

[29] Z. Tu. Auto-context and its application to high-level vision tasks. In *CVPR*. 2008.