Universitat Autònoma de Barcelona

MASTER IN COMPUTER VISION AND ARTIFICIAL INTELLIGENCE
**REPORT OF THE RESEARCH PROJECT**
COMPUTER VISION

# Object Localization Enhancement by Multiple Segmentations Fusion

**Author: Ahmed Mounir Gad**
**Date: 13/07/2010**
**Advisor: Ramon Baldrich**

# Object localization enhancement by multiple segmentations fusion

**Ahmed Gad** AMOUNIR@CVC.UAB.ES
*Computer Vision Center*
*Edificio O, Campus UAB*
*Bellaterra, 08193, Spain*

**Advisor:** Ramon Baldrich

## Abstract

Despite being a very complex task, image segmentation is used as a vital prerequisite in several computer vision applications, including object tracking, object recognition and the more challenging object localization. Image segmentation is known to be unstable and is highly affected by image perturbations such as shadows, shading and highlights. In our work, we combine different cues from multiple segmentations of each image to obtain better segmentations and, hence, better object localization. We propose the idea of combining multiple segmentations in a bottom up fashion to obtain enhanced and more reliable image segmentation using the criteria of "segment goodness". We also propose the "voting" technique for integrating image partitions in a top-down fashion to obtain enhanced object localization. Finally, we discuss some possible extensions for our method that will potentially boost its performance. Our results exceed the previously published, state-of-the-art results on a challenging dataset: The PASCAL VOC 2007 object segmentation challenge.

**Keywords:** image segmentation, object localization.

## 1. Introduction

Image segmentation is a computer vision process where the focus is on partitioning an image into a set of non-overlapping regions. This is an extremely challenging task as on real images, the varying shapes of the objects provoke several effects related with the illumination such as shadows, shadings and highlights. These effects are one of the main problems that should be solved in order to obtain an efficient segmentation.

Image segmentation algorithms can be divided in several ways, however, all of the existing approaches can first be divided into two main hierarchies: bottom-up approaches and top-down approaches.

Bottom-up segmentation approaches mainly examine the image and try to figure out how to divide it into coherent and meaningful segments. Comprehensive surveys as presented by Yz and S. K. Mitra (2001) and Cheng et al. (2001) drew the basis for the current classification of bottom-up segmentation techniques. From all of these existing methods, segmentation methods can be divided into four main categories: feature-based, image-based, physics based and hybrid approaches.

Feature based approaches mainly focus on the photometric information of an image represented by its histogram as in the work by Agarwal et al. (2005) and Yang et al. (2006). Image-based approaches exploit the spatial coherence of colour in an image and an example is given from the work of Freixenet et al. (2002). Physics-based methods use physics and psychophysics information to perform the segmentation. Finally, hybrid techniques combine methods of the previous categories.

Top-down approaches, on the other hand, are guided primarily by high-level information and the use of class-specific criteria. The motivation for using these class-specific criteria, as shown by Borenstein and Ullman (2002), has two parts. The first is that although recent image segmentation algorithms provide impressive results, they still often fail to capture meaningful and at times crucial parts of the objects in the image. The second is that these methods are analogous to human vision in the sense of indicating high-level, class-based criteria to segment the images in a meaningful manner. Examples of using top-down

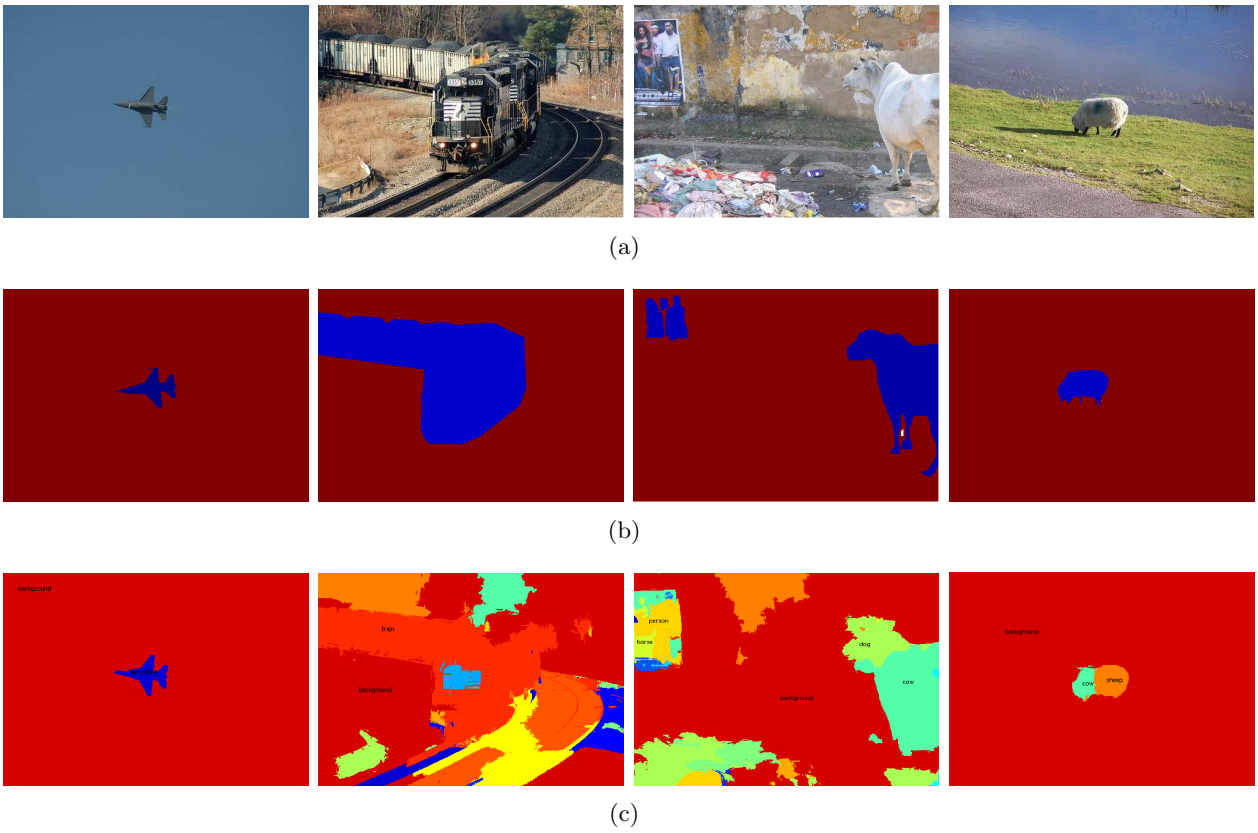



Figure 1: (a) Original images (b) Ground truth (c) Obtained object class segmentation

approaches to perform segmentation are shown in the work of Borenstein and Ullman (2002), Leibe et al. (2004), Winn and Jojic (2005) and Yuille and Hallinan (1993).

The importance of image segmentation has been shown in a variety of computer vision problems. In other words, a robust and efficient segmentation is usually an important preprocessing step for several computer vision tasks such as object tracking or object recognition. However, extensive experiments by Unnikrishnan et al. (2007) show that when using a single region generated by an image segmentation algorithm, the segmentation quality is highly variable and dependant on image data, the segmentation algorithms and the parameters used to create this segmentation. This was a motivation for the emerging new trend in object recognition that uses the segments generated from multiple segmentation algorithms and tries to merge them efficiently to recognize objects in the scene Russell et al. (2006), Pantofaru et al. (2008) and Malisiewicz and Efros (2007).

This trend of using multiple segmentations was also our motivation in our work. In our work we focus on two parts. The first one is building a robust and efficient segmentation method that uses the information provided by several other segmentation algorithms to build more reliable image segmentations in a bottom-up fashion. The second is using this new segmentation along with the other segmentation results to recognize and localize objects in the image by performing object class segmentation to this image then combining the localization results in a top-down approach.

The goal of object class image segmentation is to produce a pixel-level segmentation of the input image. In figure 1 we show an example of object class image segmentation. This is a higly constrained problem where most state-of-the-art approaches focus on exploiting contextual information available around each pixel. Afterwards, they measure feature statistics from this information (in this case histograms of visual words) and use the bag-of-words technique in order to determine the class that each pixel falls into.

Instead of working on pixels, Fulkerson et al. (2009) present a method to perform object class segmentation by aggregating the histograms in the neighbourhood of the small regions obtained from a conservative

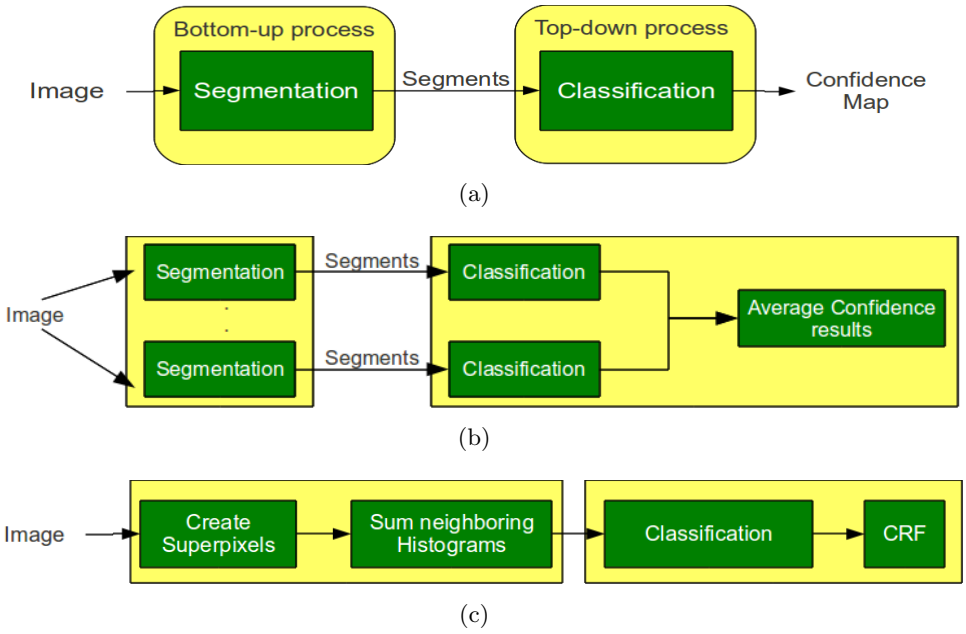



Figure 2: Object class segmentation frameworks (a) Basic framework (b) Pantofaru et al. (2008) framework (c) Fulkerson et al. (2009) framework

oversegmentation, or "superpixels" as described by Ren and Malik (2003) and Moore et al. (2008) as their elementary unit of any detection, categorization or localization scheme. They not only use the information obtained from superpixels but also they use of neighbouring superpixels to provide more contextual information. However, in our work we show that even when adding neighbourhood information, superpixels are still not the best choice for describing an object.

In our work, instead of using pixels or superpixels, we use regions emerging from several segmentation techniques. Our results show that even superpixels do not capture enough information about the objects being recognized and localized that can be better captured by using larger segments. We learn a model from each segmentation method to classify the regions of this segmentation method. Afterwards, we combine all of the results coming from each model separately into a new model by the technique that we call, the "voting technique". Our results show that combining results from outputs of several classifications give a significant improvement over using each classification solely. The results are even better than those obtained while using superpixels with joining neighbourhood information for classification.

This thesis is organized as follows. In section 2, we highlight some of the existing literature in the field of segmentation and object recognition and localization. We then explain more details about the segments neighborhood in section 3 which is the core of the class segmentation method of Fulkerson et al. (2009). It is also considered as the basis of our method. In section 4, we thoroughly illustrate our core approach. In section 5, we show the performed experiments and the results we have. In Section 6 we sum up our work with some conclusions. Finally, in section 7 we introduced some of the extensions of our method that will potentially improve its performance.

## 2. Related Work

In this section we briefly explain the existing literature in addressing the problems we are proposing in our work. We are addressing two different problems: First, the segmentation problem. We are building a new segmentation method based on segments coming from different other segmentations by combining these segments in a bottom up fashion. Second, the recognition problem. We are performing object class image segmentation which is the process of producing a pixel level segmentation for the whole image. In

other words, recognizing which class does every pixel of the image lies into. Figure 2(a) shows the basic framework used to produce an object class image segmentation.

### 2.1 Image segmentation

The problem of image segmentation has been the subject of considerable research activity over the last decades. The aim of image segmentation is to identify the homogeneous regions in the image. In other words, it is the process of partitioning an image into a set of disjoint and homogeneeous regions. This task can be equivalent to the task of identifying boundaries between the regions.

Segmentation is an extremely important preprocessing task for several applications of image processing and computer vision. It represents the first step of low-level processing of imagery. Consequently, many literature exist trying to address the segmentation problem. Existing segmentation methods can be divided into two main sections. Top-down and bottom-up segmentation techniques.

According to the color image segmentation survey by Yz and S. K. Mitra (2001), existing bottom-up segmentation methods can be divided into feature space-based, image-domain based, physics-based and hybrid techniques.

According to Yz and S. K. Mitra (2001), feature space based techniques rely on the following observation. "If we assume that color is a constant property of the surface of each object within an image and we map each pixel of the color image into a certain color space, it is very likely that different objects present in the image will manifest themselves as clusters or clouds of points." An example of the techniques that are considered as feature-based is the clustering. Clustering can be defined as a non-supervised classification process where one has to generate classes or partitions without any prior information. Shi and Malik (2000) consider image segmentation as a clustering problem via graph partitioning.

Another way to approach the image segmentation problems is to use the image domain based techniques. According to Yz and S. K. Mitra (2001) existing techniques are divided into split-and-merge, region growing, edge based and neural network-based techniques. One of the most commonly used image based techniques for image segmentation is the one introduced by Felzenszwalb and Huttenlocher (2004). They define a predicate for measuring the evidence for the boundary between two regions using a graph based representation of the image. Afterwards, they develop an efficient segmentation algorithm based on this predicate.

Physics based techniques, the proposed methods focus on analyzing how light interact with coloured materials and try to introduce models of this physical interaction in the segmentation algorithms. A milestone in physics-based segmentation was the "dichromatic reflection model" introduced by Shafer (1992).

Finally, hybrid methods focus on combining the previous techniques together. An example of using hybrid techniques for performing segmentation is the work presented in this thesis. The method proposed focuses on getting the best segments out of several bottom up segmentation methods of different categories and tries to combine them together.

Top-down segmentation algorithms, also known as class-specific segmentation algorithms, were developed to overcome the difficulties in low-level segmentation. Mainly, they add more semantics to the top-down segmentations by grouping segments that belong to the same object based on some prior information about the characteristics of the object. The work by Borenstein and Ullman (2002) and Winn and Jojic (2005) provide an example for top down segmentation algorithms.

Moreover, several approaches also exist that combine bottom-up with top-down cues to yield a better segmentation. An example is the work by Levin and Weiss (2006). They train a fragment based segmentation algorithm which takes into account both bottom up and top down cues simultaneously. They formulate the problem in the framework of conditional random fields and derive a feature induction algorithm for CRF which allows it to efficiently search over thousands of candidate fragments.

### 2.2 Objects recognition and localization

Recently, significant progress has been made in image-level object categorization. The "classification" task works on a certain image and tries to determine whether it contains at least a single instance of a certain object or not. The bag of words approach proved to be a suitable technique for this task. However, this image level object categorization fails to determine spatial information about the objects. This led to significant interest on the related fronts of localization and pixel-level categorization. Challenges like PASCAL VOC classification, detection and segmentation encourage further research in object recognition and localization.

Sliding window classifiers have been well explored for the task of detecting the location of an object inside an image. Lampert et al. (2008) propose a simple yet powerful branch-and-bound scheme that allows efficient maximization of a large class of classifier functions over all possible subimages. Their method works in sublinear time and is applicable to object detection and retrieval. Fulkerson et al. (2008) perform bag of features classification within a local region. However, the size of the region is fixed (rectangular window).

Moreover, many approaches exist to class segmentation which work at the pixel level or more local features like textons as the work by Shotton et al. (2006). Shotton et al. (2008) construct semantic forests for extremely fast classification.

The basis of our work is the work of Fulkerson et al. (2009) and Pantofaru et al. (2008). Fulkerson et al. (2009) advocate the use of superpixels as the basic unit of their class segmentation or localization scheme. They construct a classifier on the histogram of local features found in each superpixel. They regularize this classifier by aggregating histograms in the neighborhood of each superpixel. Finally, they refine their results by using a classifier in a conditional random field operating on the superpixel graph. Figure 2(c) shows a graphical explanation for Fulkerson et al. (2009) framework.

Pantofaru et al. (2008) works similarly to that of Fulkerson et al. (2009). The difference is that it extracts superpixels like objects by intersecting multiple segmentations and then classifies them by averaging the classification results from all of the member regions. A clearer explanation is shown in figure 2(b). The work of Fulkerson et al. (2009) provides the insight for our method. We explain it in more details in the next section.

## 3. Segment neighbourhoods

Fulkerson et al. (2009) introduced the concept of superpixel neighbourhoods. They note that superpixels by their nature do not carry any meaningful information about the object where they are contained. However, establishing a relation between the superpixel and its neighbour helps make the superpixel more meaningful.

They assume the superpixels in an image form a graph $G(S, E)$ where superpixels $s_i \epsilon S$ are considered the nodes of the graph, $E$ is the set of edges formed between pairs of adjacent superpixels ($s_i$, $s_j$) in the image.

Let $D(s_i, s_j)$ be the length of the shorteset path between two superpixels. In this case, $H_i^N$ is the histogram obtained by merging the histograms of the superpixel $s_i$ and neighbours who are less than N nodes away in the graph

$$H_i^N = \sum_{s_j | D(s_i, s_j) \leq N} H_j^0. \tag{1}$$

Fulkerson et al. (2009) use the histogram obtained from equation 1 for the classification. They use different values of $N$ to construct the histograms. This means, Fulkerson et al. (2009) assume that increasing the value of $N$ means considering the histogram resulting from a larger segment instead of individual superpixels. This segment is constructed by the value of $D$ which is the shortest path between two superpixels and is calculated from the minimum difference of colours of each two neighboring superpixels.

We also tried using the same concept but for joining segments instead of superpixels. This concept improved some of the segmentations like the mean-shift and the graph based. On the other hand, joining

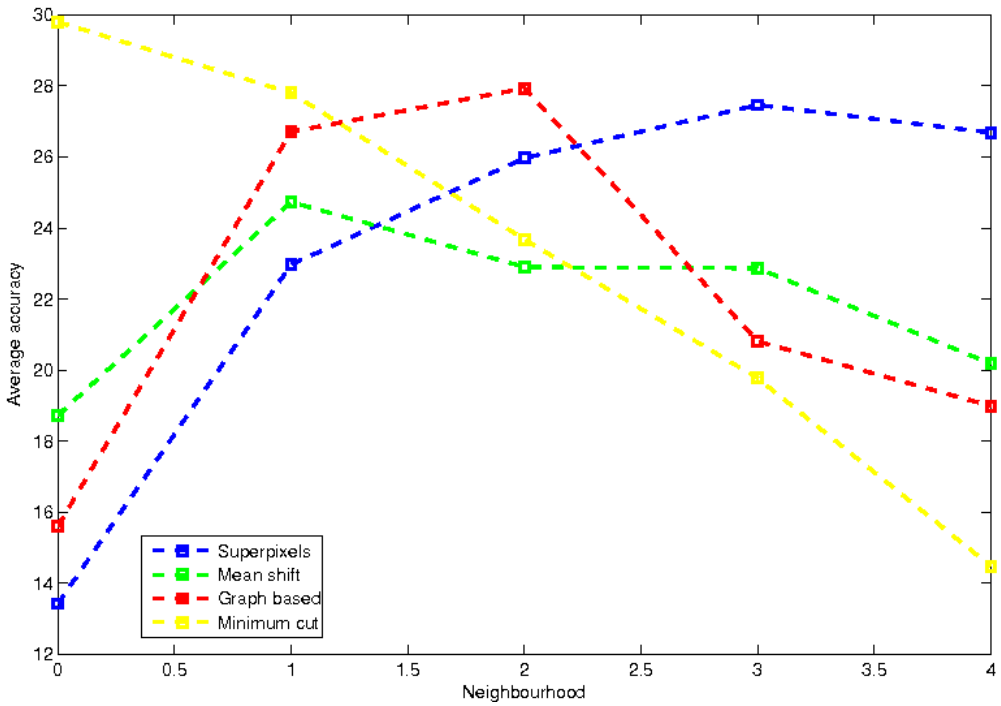



Figure 3: Effect of neighbourhood on the average accuracies

segments by summing the neighbour's histograms got worse results for some other segmentations like the normalized-cut. From these observations we concluded that joining segments don't always work. Sometimes when the segment is already large, joining the segments makes them carry other useless information from other different objects. In this case, the results start decreasing gradually.

In figure 3, we see the effect of increasing the value of the pixels neighbourhood on the final accuracy of the segmentation. We conclude that when the segments are larger and carrying more information, the classification results are improved. However, the segments should only be merged with other segments that correspond to the same object. Otherwise, further increasing of the size of segments, the accuracy tends to decrease dramatically.

From these conclusions we arrive at our insight for obtaining better segmentations that can help in improving the object localization consequently. We first have to favor large segments from all segmentations. These segments are more meaningful about the objects and help the classifier to know a better clue about each class of objects. However, these segments should still be meaningful. In other words, they should be good segments. We provide a definition of "goodness" for segments in later sections.

## 4. Core approach

In this section, we describe the core of our approach for the object class segmentation problem. The process involves five main steps. First, we generated multiple segmentations for each image in our dataset. Second, we use these segmentations to build a more reliable segmentation. We tried several techniques to combine these segmentations together into a better final segmentation. We'll describe each of these techniques in the next sections.Third, we describe and classify each region from every segmentation technique including our generated segmentation. Afterwards, we combine the regions' classifications into an object map indicating which class does each of the pixels lie in. Finally, we refine our results with a conditional random field. A graphical explanation of our object class segmentation framework is explained by figure 4

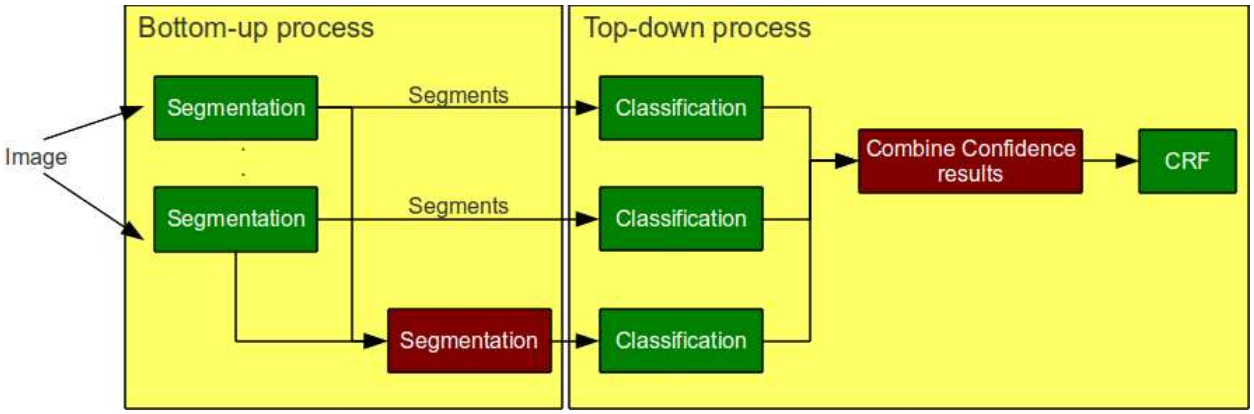



Figure 4: Our object class segmentation model. Our contributions are highlighted in red.

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

Figure 5: (a) Original image (b) Image segmented using mean shift. Number of segments = 297 (c) Image segmented using Graph Based technique by Felzenszwalb and Huttenlocher (2004). Number of segments = 2114 (d) Image segmented using normalized cut. Number of segments = 100

### 4.1 Generating multiple segmentations

Every image segmentation technique uses unique cues to construct segments cues that differ from those used by other segmentation methods. In order to build a more reliable segmentation, we constructed a number of segmentations each containing a number of properties that we desire in our final segmentation and that are not obtained from a single segmentation method solely.

We generate a segmentation based on mean-shift by Comaniciu et al. (2002). In this segmentation we guarantee that the segments mainly vary in the colour distribution along the whole segment. We also generate another segmentation based on the graph-based method by Felzenszwalb and Huttenlocher (2004) to guarantee a variation along the edges and we generated a segmentation based on minimum cut by Shi and Malik (2000) for a specific number of segments to guarantee some large segments.

Using these segmentations, we obtain mose of the qualities that we desire in our final segmentation result so that when we mix them we will get a better final segmentation.

In figure 5 we show an image segmented using all different segmentations and we show the number of segments generated by each segmentation method. Our goal now is to first, combine these segmentations in a bottom-up fashion to obtain a new more reliable segmentation. Second, combine all these segmentations in a top-down fashion into a better, final object class segmentation.

### 4.2 Building a more reliable segmentation

After obtaining different segmentations for each image, we use these segmentations to build a more reliable segmentation. Our aim is to select those segments that are considered good enough and split those which aren't based on some other cues.

Assume we have $N$ segmentations $S_i$, where $i = 1, ..., N$ and each of the segmentations correspond to $S_i = \bigcup_{j=1}^{C_i} T_j$, where $T$ is a segment in the segmentation $S$ and $C$ is the number of segments inside a certain segmentation. We want to build a final segmentation $S^* = \bigcup_{k=1}^{C^*} T_k^*$ based on the two observations we had before:

1. The segments of the final segmentation should be as large as possible. In other words, we need to minimize $C^*$

2. The segments should be "good" segments. In our context, we defined a "good segment" as the largest set of connected pixels that lie in the same class and are coherent in the colour distribution.

Consequently, our motivation for the mixing problem was to start looking for the large segments and then check if this segment is a valid segment or not. we tried two techniques to combine segments. The two techniques perform greedily favoring the largest segment. The techniques are described by algorithms 1 and 2.

**Algorithm 1** mix_all_segmentations(ip_img, curr_segment, all_segs)

```
[largest_seg, seg_id] = get_largest_segment(ip_img, all_segs);
if good_segment(largest_seg) then
  include_segment(seg_im, largest_seg);
else
  include_segment(seg_im, mix_all_segmentations(largest_seg, all_segs - seg_id));
end if
include_segment(seg_im, mix_all_segmentations(ip_img - largest_seg, all_segs));
return seg_im;
```

**Algorithm 2** mix_all_segmentations(ip_img, all_segs)

```
Add all segments from all segmentations to a heap;
while ! empty(segments_heap) do
  segment T = largest_segment()
  if segment contains pixels already in the final segmentation then
    remove those pixels
    add the segment again to the heap
  else
    if good_segment(T) then
      include in the final segmentation
    else
      discard this segment
    end if
  end if
end while
return seg_im;
```

As shown in the pseudo-code in Algorithm 1, we take the largest segment from all segmentations and try to check if this segment is a good segment. If the segment is not good, we try to segment it recursively using the other segmentations. Afterwards, we segment the remaining part of the image similarly. However, in Algorithm 2, we always favor large segments on the image level. In other words, if the largest segment isn't a good segment, we choose the second largest segment in the whole image and check if it's a good segment. In figure 6, we also provide a graphical explanation for the two mixing techniques.

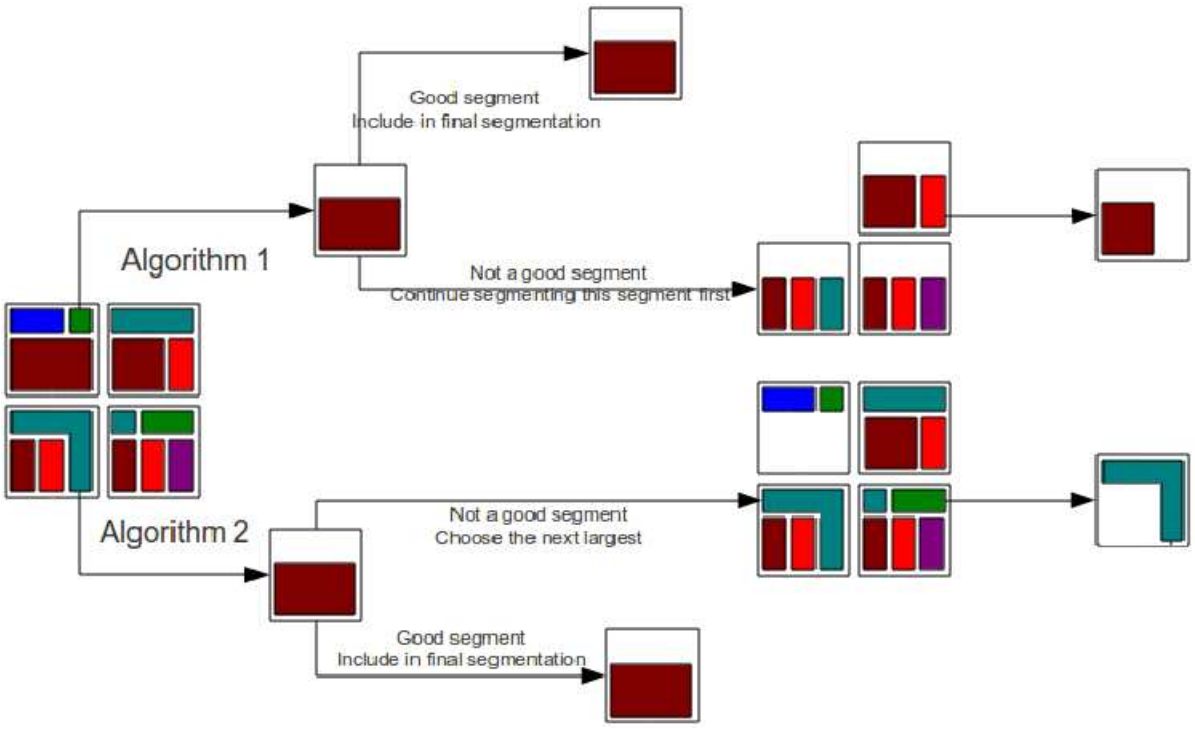



Figure 6: 2 Steps in mixing 4 different segmentations with the two algorithms segments are represented by large colorful squares to better clarify the idea.

The only part remaining now is how to determine whether the segment is good or not. To do that, we had to find a way for good segments evaluation. First, we investigated in the the possibility of assuming that the color of each region should follow a normal distribution. In figure 7, we show a region of the image and the colour histogram of the three channels. The check for normality didn't work the way we expected. Even while smoothing the histograms, still sometimes the normality test fails for a perfectly "good" region. Figure 7(c) shows an example that good regions may not follow a normal distribution on real images.

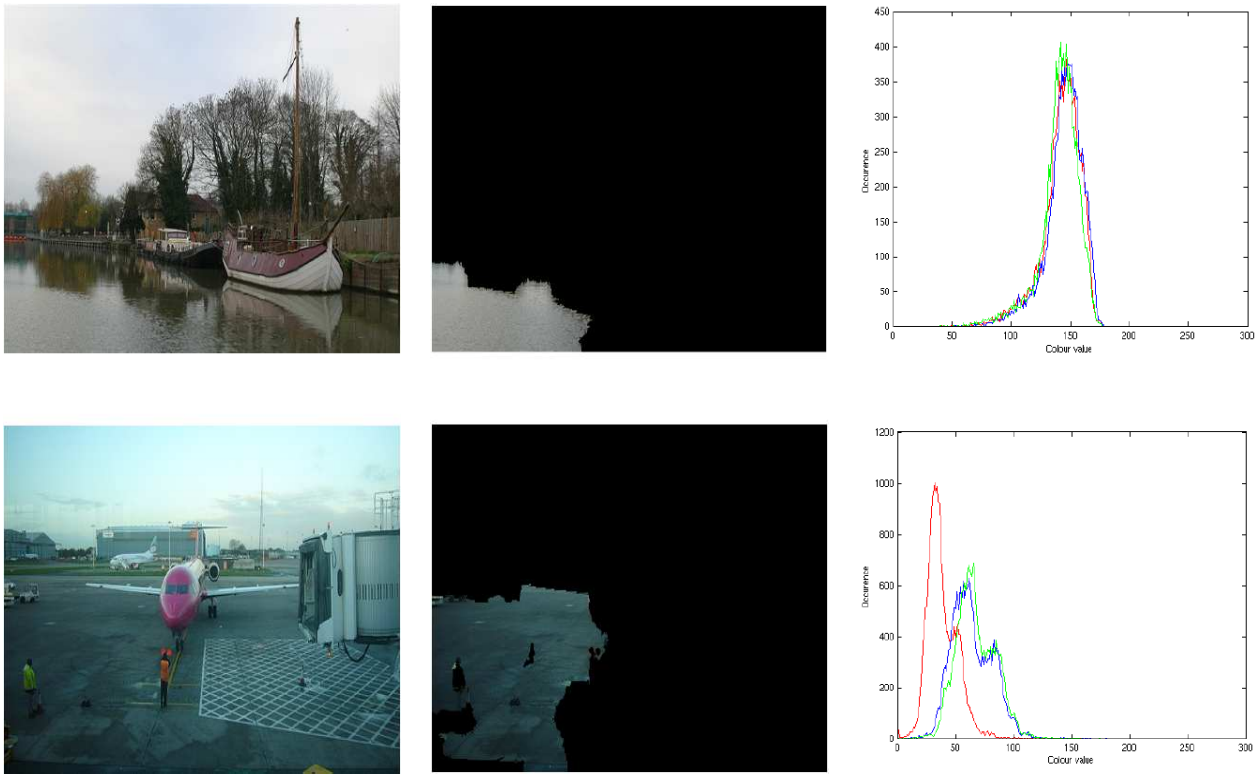
Second, we tried the unimodality test. We checked if the distribution of the histogram of each channel follow unimodal function. We tried Dip test for unimodality by Hartigan and Hartigan (1985). The resulting segments were slightly better than those provided by the normality test but are still too much. Figure 7(c) also shows an example for failure of unimodality test to determine a really good segment.

Third, we tried using the outliers detection method which will be described later. Fourth we tried the RAD method which gives our state of the art results. Finally, we tried the alternative segmentation technique. We'll describe the previous three techniques in the next sections.

#### 4.2.1 Good Segments evaluation using outliers finding

As defined by Grubbs (1969), "An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs." We consider a segment good if the number of outliers inside the segment is less than a certain value. We did a series of experiments to empirically determine the best value for the tolerance of outliers inside the segment. We finally decided that a segment is considered good if the number of outliers lying inside is less than 2% of the number of pixels in the segment.

We use the Z-Test to find the outliers of a certain segment. The Z-Test indicates that the point is considered an outlier if it satisfies the following formula:




(a) (b) (c)

Figure 7: (a) Original images (b) Current largest segments (c) Histograms from every channel. As we can see the first image can be approximated into a normal distribution or a unimodal distribution while the second one fails in that test.

$$\frac{|X - \mu|}{\sigma} > 3,$$

where X is a label for each point of the image, $\mu$ is the mean of the color values of each segment and $\sigma$ is the standard deviation of these colors.

Figure 8(e) shows the results from mixing the different segmentations using outlier finding. It also shows the number of generated segments which is quite large due to our highly restricting criteria for segments evaluation.

Although outliers detection worked well in detecting the segments in Figure 7(b), they didn't work on detecting the segment in Figure 9(b). In this segment the variations in colour are really large and the different colours are not outliers in this case. Consequently, we had to find another evaluation method that works well in the presence of shadows and illuminations. This method is the RAD evaluation method explained in the next section.

#### 4.2.2 Good Segments evaluation using RAD

We use the same method presented by Vazquez et al. (2008) for obtaining image segmentations in the presence of shadows and highlights. This method is based on the insight that the distributions formed by a single-colored object have a physically determined shape in the colour histogram space. To capture these ridges they proposed a new ridge based distribution analysis (RAD) to find the set of ridges representative of the dominant colour.

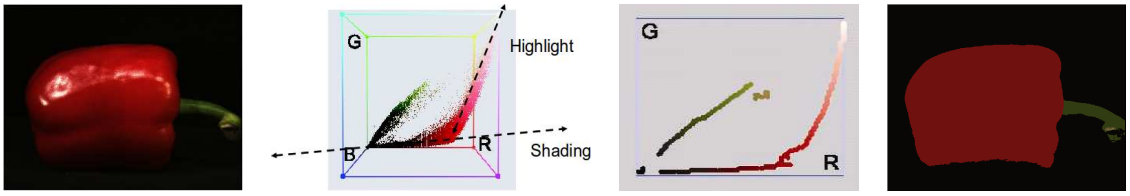

Figure 8: (a) Original image (b) Image segmented using mean shift. Number of segments = 200 (c) Image segmented using Graph Based technique by Felzenszwalb and Huttenlocher (2004). Number of segments = 2223 (d) Image segmented using normalized cut. Number of segments = 100 (e) Image segmented using mixing and evaluation by finding outliers. Number of segments = 327 (f) Image segmented using mixing and evaluation by Alternate segmentation. Number of segments = 941 (g) Image segmented using mixing and evaluation by RAD. Number of segments = 106

We model the region "segment" as a set of dominant colours (DC). This DC is described by a distribution in histogram-space. We assume that if this segment contains only 1 DC then this segment is a "good" segment as this segment contains only 1 semantic object. Based on this technique, the evaluation method becomes very robust to shadows and highlights and becomes closely related to the physical features of each segment than the previous method.

To perform this evaluation we need to use the first step done by Vazquez et al. (2008) in their segmentation technique which is extracting ridges as a representative of a dominant structure (DS). Afterwards, we check if we have only one single ridge in the image then we consider this segment as a good segment.


(a) (b) (c)

Figure 9: (a) Original image (b) Current largest segment where outlier isn't working (c) The image colour histogram for every channel




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

Figure 10: (a) Original image (b) The image's histogram. The effects of shading and highlights are clearly visible in the red colours of the histogram. (c) Ridges found with RAD (d) Final segmented image. Image by Vazquez et al. (2008)

**A Ridge based Distribution Analysis method** In this section we will briefly explain the algorithm used by Vazquez et al. (2008) to extract DCs from histogram space. They introduced a technique to find dominant structures (DS) for a d-dimensional feature space. In our context here, a dominant color is the dominant structure of the 3D chromatic histogram. We only consider the segment good if it contains a single dominant colour.

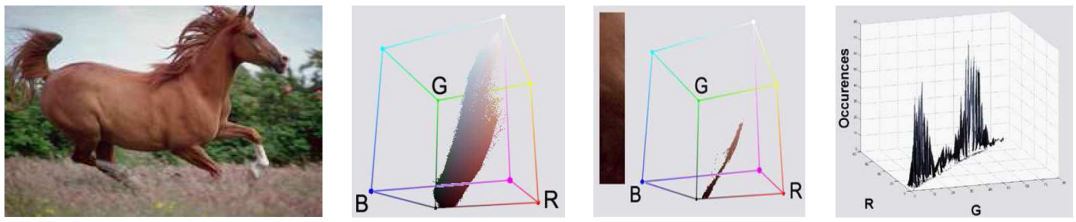
First, we start by reviewing the dichromatic reflection model by Shafer (1992):

$$f(x) = m^b(x)c^b + m^i(x)c^i, \tag{2}$$

in which f = R, G, B, $c^b$ is the body reflectance, $c^i$ is the surface reflectance, $m^b$ and $m^i$ are geometry dependant scalars representing the magnitude of body and surface reflectance.

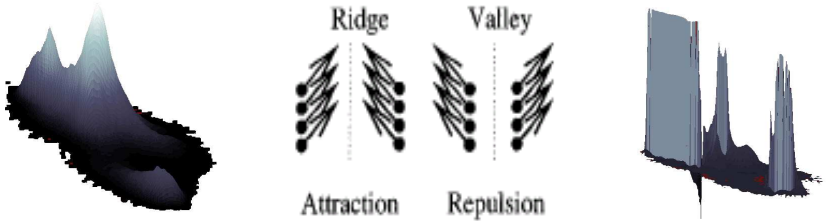
The two parts of the dichromatic reflectance model are clearly visible in the histogram of figure 10(b). First, due to the shading variations, the distribution of the red pepper traces an elongated shape in histogram-space. Second, the surface reflectance forms a branch which points in the direction of the reflected illuminant. From the image, we can clearly condlude that the distribution of a single DC doesn't really follow two straight lines as expected from the dichromatic reflectance model due to the gamma, compressions, camera aberrations or interreflections. Instead, they form a ridge-like structure in the histogram space.

Another conclusion is what we can get from figure 11. Consider figure 11(c) which contains a patch of the horse image. The 2D Red-Green histogram of the patch is shown in figure 11(d) to see the occurrences of the dichromatic combinations. It's clearly seen that the density of the term $m^b$ varies significantly. The distribution is even broken into two parts although they both belong to the same DC.




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

Figure 11: (a) Original image (b) The image's RGB histogram. (c) A patch of a) and it's RGB histogram (d) 2D histogram of c) to illustrate discontinuities on a DC. Image by Vazquez et al. (2008)




(a) (b) (c)

Figure 12: (a) a 3D histogram (b) Points attracted to ridges and repulsed from valleys. (c) MLSEC-ST applied to a). Image by Vazquez et al. (2008)

From the previous two examples we conclude that we have two main shortcomings from the dichromatic reflection model in describing what actually happens with the real world images. The first one is how to deal with non-linearities and how to get an approximation of the actual structure formed by the image's colour histogram. The second, is dealing with the gaps in the colour histogram obained within dominant structures of the same dominant colour.

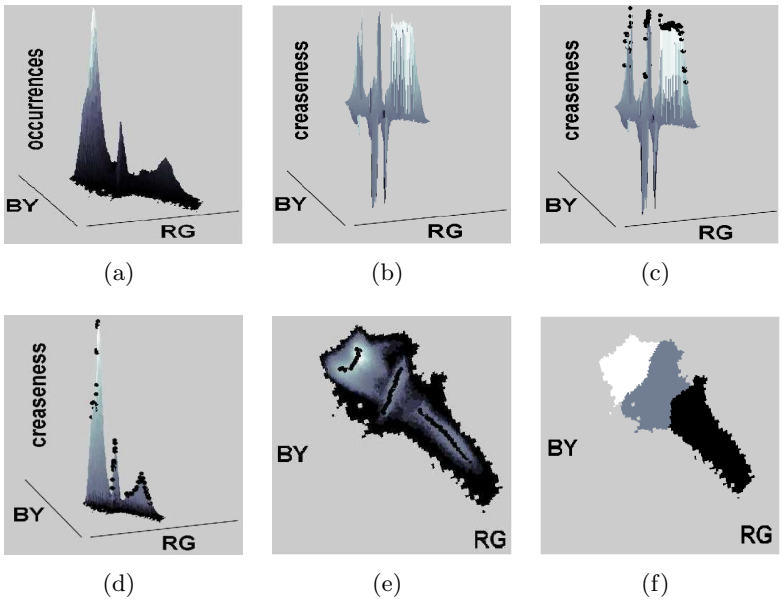
To solve these problems, Vazquez et al. (2008) applied a multilocal creaseness algorithm to overcome the gaps problem in the image's colour histogram. Afterwards, they introduced a ridge extraction technique to avoid limitations of linear assumption.

**Problem 1: Filling the gaps in the colour histogram** In order to deal with the discontinuities in the image's colour histogram we applied the MLSEC-ST operator introduced by Lopez et al. (1999) to enhance ridge points before the ridge detection step. This method is used due to its good performance comnpared with other ridge detection methods on irregular and noisy landscapes as explained by Lopez et al. (1999).

The structure tensor (ST) computes the dominant gradient orientation in a neighbourhood of size proportional to $\sigma_d$. In other words, the operator enhances the situations where either a big attraction (ridge) or a big repulsion (valley) exist. Figure 12 shows an example of applying the MLSEC-ST operator on a 3D histogram.

**Problem 2: Dealing with non-linearities** To deal with non-linearities, we used the technique proposed by Vazquez et al. (2008) for ridge detection. Their algorithm is briefly explained by algorithm 3 and figure 13.

After applying the ridge extraction algorithms. We check if there is more than one ridge. In this case, clearly there is more than one dominant colour in this region and, we discard the segment and look for an alternative segmentation of it. Otherwise, we accept the segment is it is in our final segmentation.




(a) (b) (c)

(d) (e) (f)

Figure 13: (a) Opponent red-green and blue-yellow histogram. (b) Creaseness representation of a). (c) Ridges found in b). (d) Ridges fitted on the original distribution. (e) Top view of d). (f) Dominant structures of a). Image from Vazquez et al. (2008)

**Algorithm 3** EXTRACT_RIDGES

Select a local maximum $m_i$
Add this local maximum to the list of ridge points. R = $m_i$
**if** We reach a flat area **then**
  stop;
**else**
  repeat previous steps
**end if**

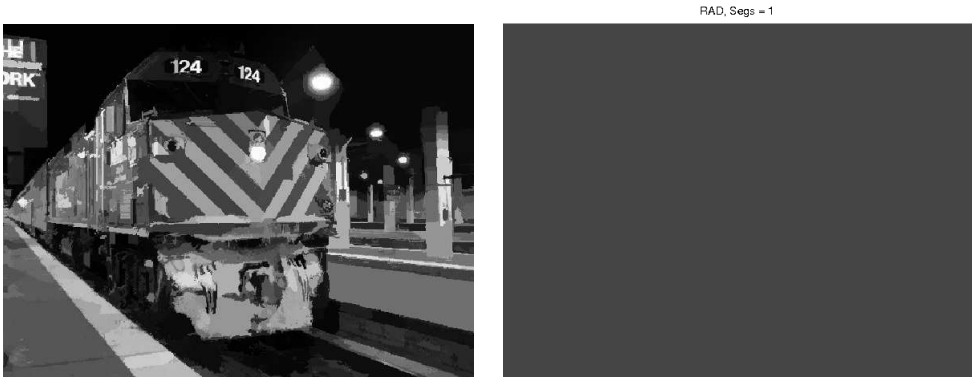
In figure 8(g) we show one of our results from mixing segmentations using segment evaluation with the described RAD criteria. Clearly, the number of segments is reasonable. It is much less than those provided from the mixing and evaluation using outliers detection method. Also the segments are meaningful as they depend mainly on the image's colour histogram as an evaluator which gives an important clue about the distribution of colours in this segment.

It is worth mentioning here that despite being satisfactory while working with shadows and highlights, RAD wasn't in our choices to perform the segmentation on the PASCAL VOC 2007 dataset. RAD depends mainly on the MLSEC-ST operator. This operator works much better with more colourful images that contain landscapes as explained by Lopez et al. (1999). Consequently, RAD gives better segmentations for some images but very bad segmentations for others. In figure 14 we show an example of a really bad segmentation obtained by RAD on one of the images. This behaviour is always repeated in the case that the illumination of the image does not change considerably.

#### 4.2.3 Good Segments evaluation using alternative segmentations

Finally, we tried another method to evaluate the goodness of a segment which is to try segmenting the same segment using another segmentation method which differ in the way of extracting a segment.

Example: Mean-Shift segmentation mainly relies on the distribution of colours in the segment, where graph-based segmentations mainly rely on the distribution of edges in the segment.




(a) (b)

Figure 14: (a) Original image (b) Image segmented by RAD yielding an undesired behaviour.

This is mainly what we do for the evaluation of the segments using the alternative segmentation method. In figure 8(f) we show the one if the results obtained from mixing segmentations using the alternative segmentation method.

### 4.3 Describing and classifying regions from each segmentation

Relying on the same framework by Fulkerson et al. (2009), we construct a bag of features classifier which operates on the regions defined by the segments obtained from each segmentation method. we use SIFT descriptors by for each pixel in the image at a fixed scale and orientation.

We quantize the extracted descriptor using a K-means dictionary and we aggregate them into one $l^1$ normalized histogram. For training the classifier, we assign to each segment the most frequent class label it contains. Afterwards, we train a one-vs-rest support vector machine (SVM) with an RBF-$\chi^2$ kernel on the labeled histograms for each of the object categories.

The $\chi^2$ distance between histograms $h_i^0$ and $h^0$ is calculated using the following formula:

$$d^2_{\chi^2}(h^0, h_i^0) = \sum_{m=1}^{M} \frac{(h^0(m) - h_i^0(m))^2}{h^0(m) + h_i^0(m)}. \tag{3}$$

From the previous equation it can be concluded that the larger the segment is, the more discriminative this distance will become. This is because the larger the segment will be the more the information it will carry from the dictionary and the more the variance will be in the final value of the distance which will lead to better classification.

### 4.4 Combining region classifications

Similar to Pantofaru et al. (2008), our approach towards combining multiple segmentations revolve around two main principles. The first is that pixels grouped together by every segmentation should be classified consistently. We use the same term used by Pantofaru et al. (2008) to describe pixels that are grouped together by each segmentation. We call them (IofRs) for Intersection of Regions.

The second main principle is that to know an exact clue about these IofRs we have to look at the whole segment obtained from each of the original segmentations. In other words, the original regions provide full support for extracting the features of those IofRs.

To combine the IofRs by looking at whole segments, our input is the output of the SVM. This output describes $P(c_r = k|r)$ where $c$ is the class label for region $r$ and $k \in K$ is a specific class label in the set of class labels $K$.

Now let $i$ be an IofR, $r_i^S$ is the region that contains $i$ in segmentation $S$.

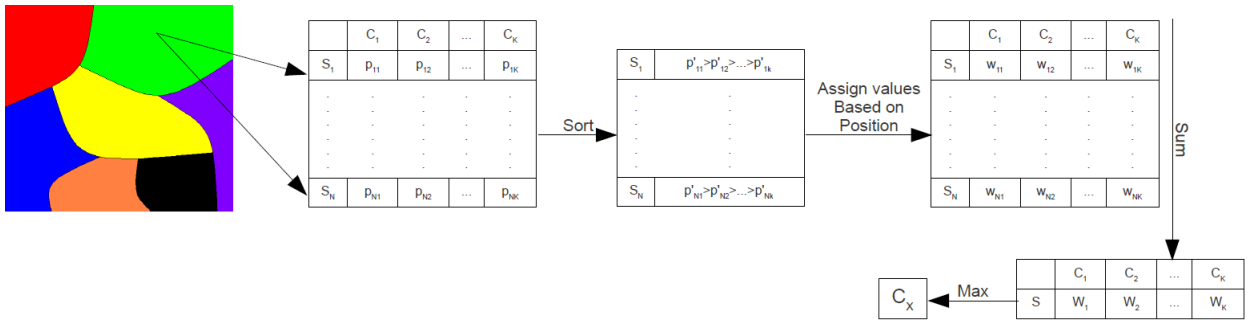



Figure 15: Voting scheme illustrated

$$\forall_{p\in K, q\in K}, P(c_i = p|I) > P(c_i = q|I) \propto \frac{\sum_{m=1}^{N} P(c_i^{S_m} = p|r_i^{S_m}, I) > P(c_i^{S_m} = q|r_i^{S_m}, I)}{N}$$

Where $I$ is the original image and $N$ is the number of available segmentations that we created. This formula shows that an IofRs lies in class p with a higher probability than a class q if in several segmentations p lies in a better location than q. We also try to take the position of p and q into consideration. In other words, if p lies in the second location in one of the segmentations and on the first location on another segmentation, it's better than q if q lies in the first location in the first segmentation and the tenth location in the later one.

In figure 15 we show a graphical explanation of our technique. Also a pseudo code is shown in algorithm 4.

**Algorithm 4** class_label = get_class_label(iofr, seg_confs)

```
final_confidences = zeros(classes_count);
for all confidence segconf in seg_confs do
    [vals, indices] = sort(segconf);
    final_confidences(indices) += 1/indices;
end for
[val, class_label] = max(final_confidences);
return class_label;
```

### 4.5 Refining results with a CRF

This is the final step of our framework. In order to recover more precise boundaries and reduce the effects of misclassifications inside the image, we finally refined our results with a conditional random field.

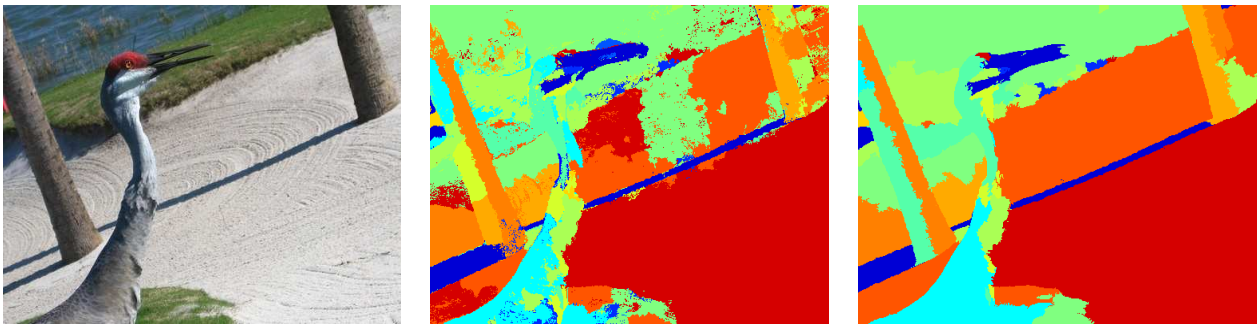
In our framework, we didn't focus on enhancing the way of introducing the concept of CRF. Instead, we used the same technique explained by Fulkerson et al. (2009). We assumed the final segmentation $G(IofR, E)$ is an adjacency graph. $P(c|G; w)$ is the conditional probability of the set of class label assignments c where w is a weight.

$$-log(P(c|G; w)) = \sum_{IofR_i \in S} \psi(c_i|IofR_i) + \omega \sum_{(IofR_i, IofR_j)\in E} \phi(c_i, c_j|IofR_i, IofR_j). \quad (4)$$

Our unary potentials $\psi$ are obtained from probability outputs provided by our SVM for each segment:

$$\psi(c_i, s_i) = -log(P(c_i|s_i)),$$

and our pairwise edge potentials $\phi$ are obtained using the following formula:



(a) (b) (c)

Figure 16: (a) Original image (b) Segmented image before applying the CRF operator. (c) Refined image after applying the CRF operator.

$$\phi(c_i, c_j | s_i, s_j) = (\frac{L(s_i, s_j)}{1 + \parallel s_i - s_j \parallel})[c_i \neq c_j],$$

where $\parallel s_i - s_j \parallel$ is the norm of the colour difference between the two neighboring segments in the LUV colourspace. $L(s_i, s_j)$ is the shared boundary length between segments $s_i$ and $s_j$ and acts as a regularizing term discouraging small regions. In figure 16 we show an image before and after applying the CRF operator in order to better clarify the idea.

## 5. Experiments

Our results are divided into three main parts. The first part is an evaluation of our new segmentation technique which is constructed from combining other segmentations in a bottom-up fashion. We provide the results from using this method solely. In the second part, we evaluated our votation technique applied only on the old segmentations that we constructed. Finally, we add our new segmentation along with the others and use the votation scheme to get our final state of the art results.

We used VLBlocks by Fulkerson et al. (2009) in our experiments. We added the extra modules for mixing segmentations in various ways.

Figure 17 shows some of the qualitative results resulting from our framework.

### 5.1 Experimental setup

In this section we explain the different parameters we used to perform our experiments.

We evaluated our method on one a standard, difficult data set: the PASCAL VOC 2007 segmentation competition dataset. This dataset contains multiple object classes with extreme variation in deformation, scale, illumination, pose and occlusion. While the challenge specifies that the detection challenge training data may also be used, we use only the 422 fully segmented images to train our core approach. The performance measure for this dataset is the average pixel accuracy: for each category the number of correctly classified pixels is divided by the ground truth pixels plus the number of incorrectly classified pixels. We also show the total percentage of pixels correctly classified.

We extracted SIFT descriptors at each pixel using the quick SIFT technique in the VLFeat framework by Vedaldi and Fulkerson (2008). The patch size for the SIFT descriptors is 12 pixels. We quantize our descriptors into a K-means dictionary learned using the training data. We chose $K = 400$ for all of the performed experiments. Fulkerson et al. (2009) stated in their experiments that changing the value of $K$ still yields almost the same results. We didn't try to change K but we used the same framework and we assumed their statements to be true.

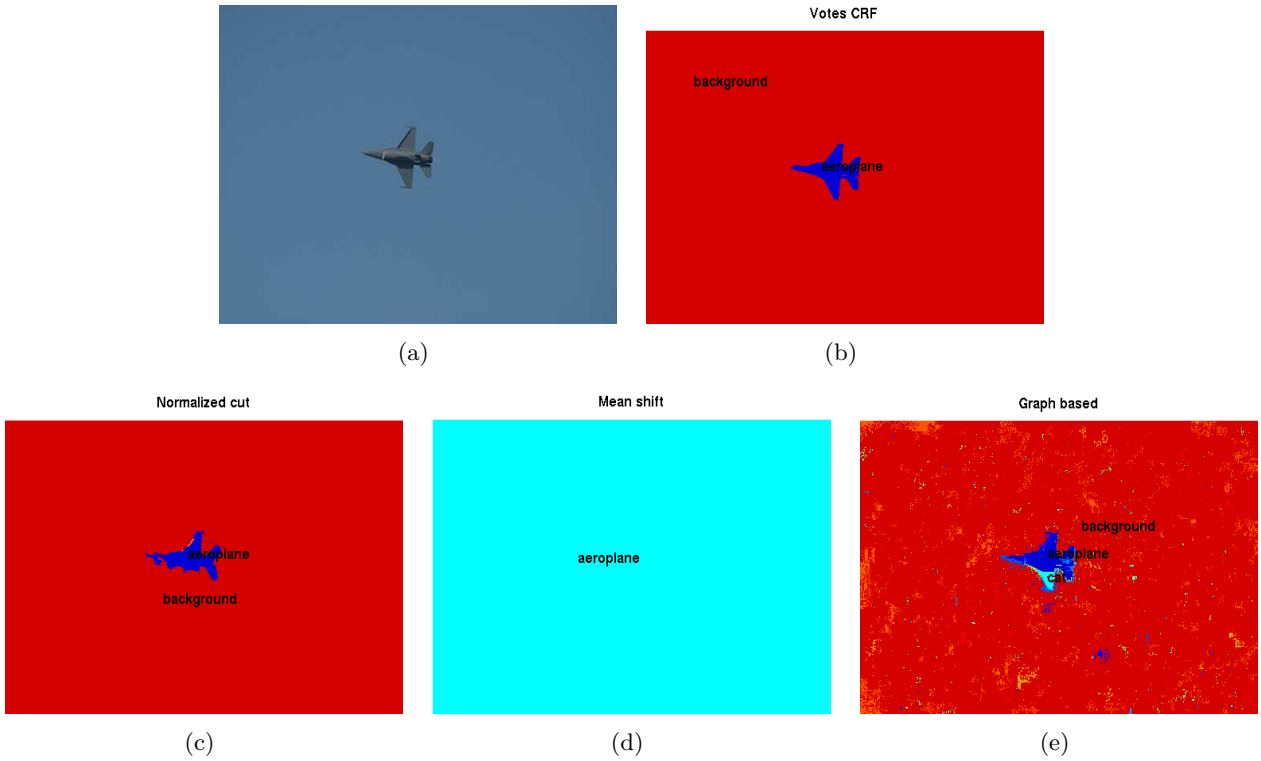



Figure 17: (a) Original image (b) Final image segmented resulting from our framework by applying the votation technique on the other segmentations. (c) Image segmented by normalized cut then classified. (d) Image segmented by mean shift then classified. (e) Image segmented by Graph-Based techniques then classified.

Histograms with varying $N$ are extracted as described in section 3. For training, labels are assigned to training segments coming from each segmentation by the majority of class votes that we get from our training ground truth. We randomly select an equal number of training histograms from each category as the training data for our SVM.

We learn a one versus rest SVM model for each class with an RBF-$\chi^2$. We used the libSVM tool by chung Chang and Lin (2001). Finally, we refined our final labelled results with a CRF model as described in section 4.5.

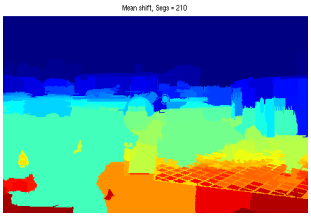
### 5.2 Generating multiple segmentations

In this section we will explain the different segmentations that we generated to help us in the mixing process. We generated three types of segmentations, Mean-shift segmentation by Comaniciu et al. (2002), Graph-based segmentation by Felzenszwalb and Huttenlocher (2004) and normalized cut segmentation by Shi and Malik (2000).

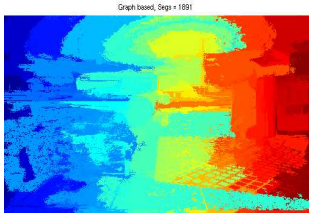
For the mean-shift segmentation we used the EDISON wrapper by Christoudias et al. (2002). It is available online and has a good performance. This code requires the following parameters: hs, indicating the spatial bandwith for mean shift analysis, $hr$, indicating the range bandwidth for mean shift analysis and $M$ which is the minimum size of final output regions. For our experiments we used $hs = 11$, $hr = 8$ and $M = 40$.

For the efficient Graph-Based image segmentaton we used the code by Felzenszwalb and Huttenlocher (2004). The GB segmentation has two versions, the adjacent neighborhood based and the $k$ nearest neighborhood based. We chose the adjacent neighborhood based. The following parameters are required:

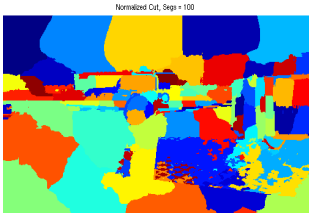

(a)



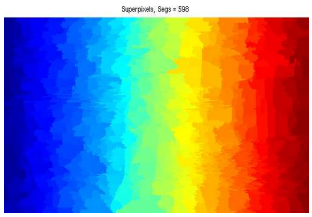

(b)




(c)




(d)




(e)

Figure 18: (a) Original image (b) Image segmentation map using mean shift. (c) Image segmentation map using Graph Based technique by Felzenszwalb and Huttenlocher (2004). (d) Image segmentation map using normalized cut. (e) Image segmentation map with superpixels using quickshift.

threshold, larger prefer larger segmented area. minsize, the minimum size of segmentation component. $nRadius$, the radius of neighbourhood of a pixel. We used the following values. $threshold = 0.5$, $minsize = 50$, and $nRadius = 2$.

For the minimum cut we used the version explained by the work of Cour et al. (2005). They have publicly available source code on their website. The code only requries the number of requested segments as its parameter. We chose the value of 100 to be the number of requsted segments each time.

We also generated an oversegmentation to obtain superpixels using quickshift by Vedaldi and Soatto (2008).These superpixels are controlled with the following three parameters: $\lambda$, the tradeoff between color imprtance and spatial importance, $\sigma$, the scale at which the density is estimated and $\tau$ the maximum distance in the geature space between members of the same reguin. In our experiments we used the folwing parameters, $\lambda = 0.5$, $\sigma = 2$ and, $\tau = 8$.

In all our segmentations we chose our values after we applied several tries and manually tuned the parameters values until we found the parameters that we thought it better describes the segments and their boundaries. Figure 18 shows the maps generated from segmentation of the image in figure 18(a).

### 5.3 Superpixels versus meaningful segments

In this section we show how using superpixels alone always obtains worse results than those obtained from meaningful segments. We show that even when the segmentations give a greater number of segments, still, the performance of the meaningful segments is higher than those obtained by the superpixels.

As we can see from table 1 and also from figure 18(e) that the superpixels alone don't carry any meaningful information. Therefore, when using them in classification, the segments tend to have very low accuracy. Even lower than the graph based which although contains more segments than those provided by the superpixels, the segments are still more meaningful segments which leads to higher accuracy achievement.

### 5.4 Segments resulting from our new segmentation method

In this section we show our results of combining multiple segmentations in a bottom-up manner using each of the 3 previously explained techniques. We compare our mixed segmentations using 2 metrics. The first

| | Background | Aeroplane | Bicycle | Bird | Boat | Bottle | Bus | Car | Cat | Chair | Cow | Dinningtable | Dog | Horse | Motorbike | Person | Pottedplant | Sheep | Sofa | Train | TVmonitor | Avg Accuracy | Avg segcount |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Superpixels | 14 | 21 | 7 | 7 | 22 | 12 | 6 | 7 | 26 | **12** | 12 | 9 | 10 | 10 | 11 | 2 | 7 | 18 | 35 | 13 | 22 | 13 | 1201 |
| Mean-shift | 16 | **25** | 12 | 8 | 16 | 7 | 15 | 15 | **52** | 10 | 9 | 12 | **20** | 5 | 19 | 16 | 23 | **23** | **43** | 17 | 31 | 19 | 393 |
| Graph-based | 18 | 21 | 20 | 2 | 17 | 6 | 11 | 21 | 39 | 2 | 15 | 14 | 3 | 5 | 37 | 8 | **33** | 5 | 7 | 29 | 15 | 16 | 2444 |
| Normalized cut | **37** | 17 | **25** | **24** | **27** | **16** | **41** | **38** | 50 | 9 | **26** | **42** | 12 | **15** | **66** | **30** | 10 | 19 | 27 | **58** | **35** | **30** | **100** |

Table 1: Comparison on the average accuracy for superpixels using quickshift by Vedaldi and Soatto (2008), mean-shift segmentation by Comaniciu et al. (2002), graph based segmentation by Felzenszwalb and Huttenlocher (2004) and normalized cut segmentation by Shi and Malik (2000). We show the accuracy of each class and the final average accuracy. We also show the number of segments generated from each segmentation method.

| | Background | Aeroplane | Bicycle | Bird | Boat | Bottle | Bus | Car | Cat | Chair | Cow | Dinningtable | Dog | Horse | Motorbike | Person | Pottedplant | Sheep | Sofa | Train | TVmonitor | Avg Accuracy | Avg segcount |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Outliers | 22 | 9 | 15 | 8 | **25** | 14 | 18 | 11 | 43 | 11 | 13 | 16 | **31** | **22** | 27 | 6 | 31 | 15 | **35** | 27 | 18 | 20 | 411 |
| AltSeg | 10 | **33** | 17 | 3 | 2 | **22** | 14 | 9 | 44 | 7 | 10 | 31 | 5 | 7 | 20 | 2 | 24 | 10 | 8 | 7 | 20 | 15 | 1769 |
| RAD | **26** | 12 | 11 | **26** | 17 | 16 | **28** | 17 | **53** | **14** | 14 | 40 | 28 | 21 | 50 | 20 | **50** | **38** | 30 | 50 | 32 | 28 | 121 |
| Worst seg (Superpixels) | 14 | 21 | 7 | 7 | 22 | 12 | 6 | 7 | 26 | 12 | 12 | 9 | 10 | 10 | 11 | 2 | 7 | 18 | 35 | 13 | 22 | 13 | 1201 |
| Best seg (Normalized cut) | 37 | 17 | **25** | 24 | 27 | 16 | 41 | **38** | 50 | 9 | **26** | **42** | 12 | 15 | **66** | **30** | 10 | 19 | 27 | **58** | **35** | **30** | **100** |

Table 2: Comparison on the average accuracy for mixing segmentations and evaluating good segments using outliers detection, opposite segmentations and RAD. We also include the worst segmentation (Superpixels) and the best segmentation (Normalized cut) to include them in the comparison. We show the accuracy of each class and the final average accuracy. We also show the number of segments generated from each segmentation mixing method.

one is the average number of segments per segmentations. The second is the average accuracy obtained from using each one of the segmentations.

In table 2 we show the values of the accuracy obtained by different segmentation mixing techniques for each class. In figure 19 we show the effect of using segment neighbourhoods on the final value of the accuracy.

### 5.5 Segments benchmarks from our new segmentation method

We also performed a series of experiments on the Berkeley Segmentation Dataset and Benchmark by Martin et al. (2001). We evaluated our multiple segmentations with the newly created segmentation based on RAD for evaluation. We measured 4 different error measurements for evaluating our segmentations quality. However, the only problem is that we didn't try several different parameters either for each of the segmentation methods or for the RAD criteria for segments evaluation.

The 4 different error measures we tried are: First, the Probabilistic Rand Index (PRI) by Unnikrishnan et al. (2005) which counts the fraction of pairs of pixels whose labellings are consistent between the

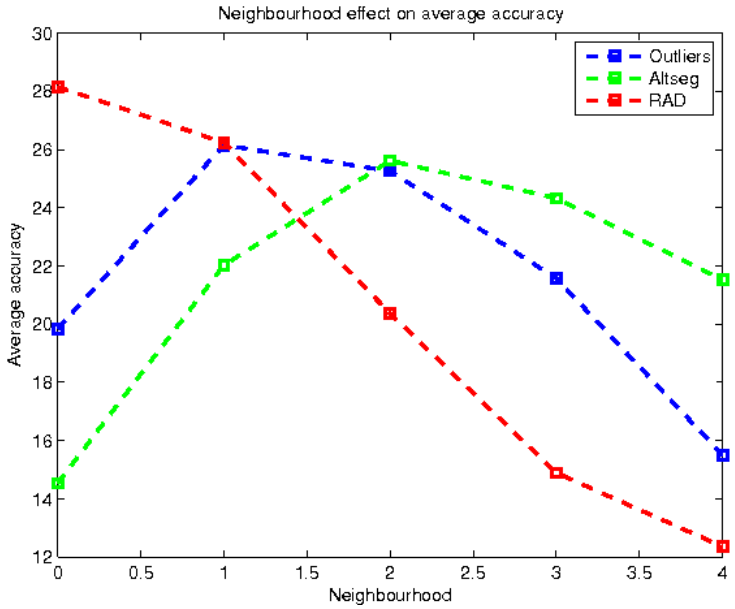



Figure 19: Effect of neighbourhood on the average accuracies

| | PRI | VoI | GCE | BDE |
|---|---|---|---|---|
| Mean-Shift | 0.7424 | 4.5293 | **0.0842** | **14.2716** |
| Graph-Based | 0.7082 | 5.1148 | 0.1150 | 17.2428 |
| Normalized-cut | 0.7079 | 4.1370 | 0.1153 | 14.7337 |
| Mix+RAD | **0.7483** | **3.5364** | 0.1433 | 14.8047 |

Table 3: Segmentation benchmarks results for the Berkeley Segmentation Dataset and Benchmark. 4 error measures were considered for 4 different segmentations.

computed segmentation and the ground truth, averaging across multiple ground truth segmentations to account for scale variation in human perception. The second error measure is the Variation of Information (VoI) by Meila (2003). It defines the distance between two segmentations as the average conditional entropy of one segmentation given the other, and thus roughly measures the amount of randomness in one segmentation which cannot be explained by the other. The third is the Global Consistency Error (GCE) by Martin et al. (2001) that measures the extent to which one segmentation can be viewed as a refinement of the other. Segmentations which are related in this manner are considered to be consistent, since they could represent the same natural image segmented at different scales. The fourth measure is the Boundary Displacement Error (BDE) by Freixenet et al. (2002). It measures the average displacement error of boundary pixels between two segmented images. Particularly, it defines the error of one boundary pixel as the distance between the pixel and the closest pixel in the other boundary image.

In table 3 we show the segmentations benchmark results. We see that our method improves both the PRI and the VoI. Even though, we still need to repeat the experiments several times with different parameters to be able to generalize our approach.

### 5.6 Class segmentation results using the voting technique alone

In this section, we describe the results obtained using the voting technique for mixing several segmentations. We show how mixing segmentations using the voting technique improves the overall average accuracy and also improves the accuracy for many classes.

| | Background | Aeroplane | Bicycle | Bird | Boat | Bottle | Bus | Car | Cat | Chair | Cow | Dinningtable | Dog | Horse | Motorbike | Person | Pottedplant | Sheep | Sofa | Train | TVmonitor | Avg Accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Votation Best Mix | **48** | 20 | 21 | **16** | 10 | **30** | **32** | 42 | 56 | 23 | 19 | **35** | 51 | 18 | 63 | **52** | **28** | 20 | 29 | 40 | 34 | **33** |
| Votation Worst Mix | 34 | 14 | **22** | 5 | **18** | 25 | 28 | 21 | 55 | 25 | 17 | 25 | 30 | **28** | 54 | 34 | 36 | **21** | 30 | 38 | 33 | 28 |
| Pantofaru et al. (2008) Best Mix | 38 | **30** | 21 | 11 | 14 | 9 | 25 | 24 | **61** | **36** | **21** | 14 | **53** | 24 | **65** | **52** | 27 | 15 | **31** | **42** | **42** | 31 |
| Pantofaru et al. (2008) Worst Mix | 38 | 14 | 14 | 14 | 8 | 6 | **32** | **45** | 52 | 19 | 12 | 18 | 30 | 24 | 56 | **52** | 24 | 15 | 24 | **42** | 32 | 27 |

Table 4: Comparison on the average accuracy for mixing segmentations and evaluating good segments by mixing segmentations using votations and the technique proposed by Pantofaru et al. (2008). In both cases the final best mix was obtained by mixing all segmentations together and the worst mix was obtained by only mixing the graph based segmentation with the mean shift.

We tried to combine many different segmentations together and check for the results using our voting technique and the technique explained by Pantofaru et al. (2008). We show the results in table 4.

We compare our segments combination method with the one proposed by Pantofaru et al. (2008). We get better results in the final overall accuracy. Also notice that The best segmentation result is higher than the best accuracy achieved without combining. Even the worst results are still good enough and are way better than the results before mixing. The best results are obtained on combining all segmentation results together while the worst results are obtained when combining the mean shift with the graph based segmentations. Note that still the results from combining those two improved more than 10 percent over the results of the graph based alone and almost 10 percent over the results of the mean shift alone.

### 5.7 Class segmentation by using the voting technique with our new segmentation method

This is our final work until now. We combined all of the information that we have had to get the best possible results. We used the voting technique for mixing segmentations along with our newly created segmentation. We also tried many possible permutations for segments. We show our results in table 5.

In the table 5 we show our final results obtained from mixing all of the segmentations and also by finding the best results from among all of the segmentations. We compared to all state of the art methods and we give at least 2% better accuracy from each one for our best and 1% from our original. We even compared with the TKK entry which had another entry in the detection in 2007 and still our method gave better results. The best results were given by combining the mean shift, graph based, normalized cut and the newly created segmentation with RAD criteria for evaluation.

## 6. Conclusions

We have demonstrated the effect of using multiple segmentations on improving the segments and hence improving the applications that use image segmentations as a prerequisite. We presented a novel framework for recognizing and segmenting objects. Our approach relies on multiple bottom-up image segmentations to build another intuitive more accurate image segmentation. These bottom-up segmentations then support top-down object recognition and localization. We have shown how these techniques improve the average accuracy of a challenging dataset. The PASCAL VOC 2007 object segmentation challenge.

We also showed that superpixels are usually not the best level of representation for objects. They provide very basic information about each segment that don't usually vary between different object classes. Hence, we showed the effect of using larger segments on the final object segmentation which was in all cases better than using superpixels.

| | Background | Aeroplane | Bicycle | Bird | Boat | Bottle | Bus | Car | Cat | Chair | Cow | Diningtable | Dog | Horse | Motorbike | Person | Pottedplant | Sheep | Sofa | Train | TVmonitor | Avg Accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Best Mix | 49 | 21 | 20 | 10 | 15 | **9** | 32 | 48 | 56 | 28 | 13 | **37** | 56 | 19 | 61 | 48 | 33 | 32 | **45** | 44 | 38 | **34** |
| Mix all | 50 | 16 | 19 | 15 | 11 | 7 | 32 | 47 | **62** | **29** | **14** | 31 | 53 | 20 | 63 | 58 | 27 | 26 | 39 | 43 | 38 | 33 |
| Viitaniemi (2007) | 23 | 19 | 21 | 5 | **16** | 3 | 1 | **78** | 1 | 3 | 1 | 23 | **69** | **44** | 42 | 0 | **65** | **40** | 35 | **89** | **71** | 30 |
| Fulkerson et al. (2009) | 56 | 26 | **29** | **19** | **16** | 3 | **42** | 44 | 56 | 23 | 6 | 11 | 62 | 16 | **68** | 46 | 16 | 10 | 21 | 52 | 40 | 32 |
| Ladicky et al. (2007) | **78** | 6 | 0 | 0 | 0 | 0 | 9 | 5 | 10 | 1 | 2 | 11 | 0 | 6 | 6 | 29 | 2 | 2 | 0 | 11 | 1 | 9 |
| Pantofaru et al. (2008) | 59 | **27** | 1 | 8 | 2 | 1 | 32 | 14 | 14 | 4 | 8 | 32 | 9 | 24 | 15 | **81** | 11 | 26 | 1 | 28 | 17 | 20 |

Table 5: Comparison of the final accuracy from mixing all segmentations, best mix of segmentations, TKK entry in the PASCAL VOC 2007, Fulkerson et al. (2009) technique based on Superpixels neighbourhoods, Oxford Brookes PASCAL 2007 entry and Pantofaru et al. (2008) results based on integrating multiple segmentations. The best mix was obtained by only mixing each each segmentation (graph based with mean shift and normalized cut) with the new segmentation method obtained by mixing and evaluating segments using the same RAD technique.

Furthermore, we suggested a number of extensions that can be applied to our method to make it perform even better. We briefly studied the effect of these extensions and clearly described how they can be plugged into our framework.

## 7. Extensions

In this section we'll discuss some of the possible extensions that can be added to our method to help in boosting the results even higher. We show these extensions with a simple analysis on how they can be done and how it'll affect the results.

### 7.1 Adding a weight to each segmentation method

As we can see from the previous tables and figure 20, the accuracy of some classes for some segmentations are high even if the average accuracy of this segmentation is low. We can make use of this observation by adding a weight on the votation process on each segmentation method for a certain class. In other words, it's very obvious from figure 20 that the Graph-based segmentation despite having a generally low performance with all classes, it gives a high accuracy with the Pottedplant class. In this case, we can make use from this observation by giving a higher weight on the decision of the Graph-Based segmentation with the pottedplant class than for instance the Normalized-cut which gives a low accuracy with this class particularly.

### 7.2 Incorporate image level priors

Adding Image level priors or in other words, the classification results will also guide the segmentation in a decent way. In the case we know the image contains a certain object with a certain probability, in this case we can boost the votation of this class depending on the confidence of the existance of that class inside the image.

For example, if the classification results show that an aeroplane exists in that image with a probability of 0.8, we can add a certain value to all of the votes for aeroplane in the whole image where we boost the locations where the existance of an aeroplane is quite clear.

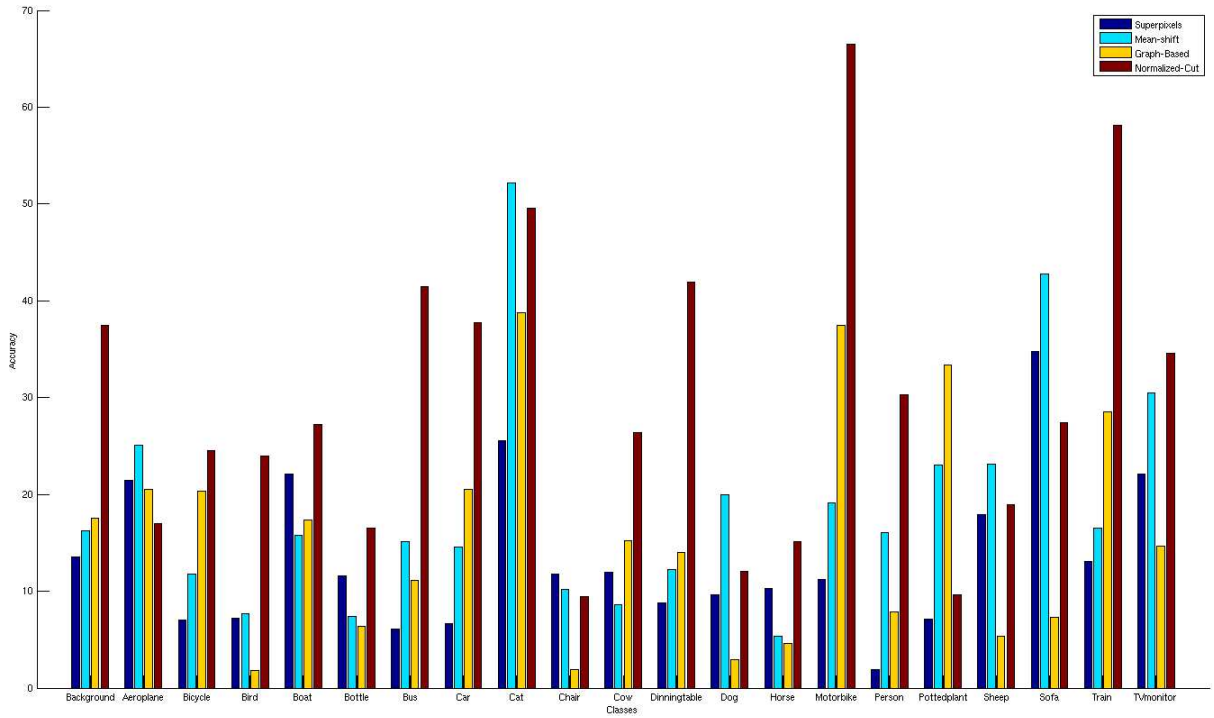



Figure 20: Classes accuracy for different segmentations

### 7.3 Use object detection to guide segmentation

This is also another way of guiding the class segmentation with the votation process. If we know that a certain object exists in a certain bounding box, In this case the only votation inside this bounding box is occuring between that object and the background. In other words, We now only consider the votes of the object and the background to take our final decision discarding any other noise.

### 7.4 Use other ways for segments goodness measure

There is several existing methods that try to measure the goodness of the segments. For instance, in the JSEG segmentation by Deng and Manjunath (2001), the authors choose a criterion of good segmentations using spatial relation existing between the pixels in the image space.

Another way to measure goodness of the segments is by using the color boosting algorithm introduced by Weijer et al. (2005). For each segment to be a good segment, it shouldn't contain more than a certain threshold of salient points that can be determined from the boosting algorithm.

There are also several other methods that we can use for good segments evaluation like the methods proposed by of Kim et al. (2008), Ge et al. (2006), Cheng et al. (2002) and Hanbury (2006).

### 7.5 Create a better RAD segmentation

We can still investigate in a suitable method for adding spatial locality to the RAD dominant colours to help them better identify correct segments. We can also try several creaseness operators and determine which ones work better on each type of images.

## References


Shilpa Agarwal, Shweta Madasu, Madasu Hanmandlu, and Shantaram Vasikarla. A comparison of some clustering techniques via color segmentation. In *ITCC '05: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II*, pages 147–153, 2005.

Eran Borenstein and Shimon Ullman. Class-specific, top-down segmentation. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part II*, 2002.

H. D. Cheng, X. H. Jiang, Y. Sun, and Jing Li Wang. Color image segmentation: Advances and prospects. *Pattern Recognition*, 34:2259–2281, 2001.

H. D. Cheng, X. H. Jiang, and Jingli Wang. Color image segmentation based on homogram thresholding and region merging. *Pattern Recognition*, 35:373–393, 2002.

Christopher M. Christoudias, Bogdan Georgescu, Peter Meer, and Christopher M. Georgescu. Synergism in low level vision. In *In International Conference on Pattern Recognition*, pages 150–155, 2002.

Chih chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines, 2001.

Dorin Comaniciu, Peter Meer, and Senior Member. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

Timothee Cour, Florence Benezit, and Jianbo Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 1124–1131, 2005.

Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23:800–810, 2001.

Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59, 2004.

Jordi Freixenet, Xavier Mu noz, D. Raba, Joan Martí, and Xavier Cufí. Yet another survey on image segmentation: Region and boundary information integration. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part III*, pages 408–422, 2002.

B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *Proc. ICCV*, 2009.

Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto. Localizing objects with smart dictionaries. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 179–192, 2008.

Feng Ge, Song Wang, and Tiecheng Liu. Image-segmentation evaluation from the perspective of salient object extraction. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1146–1153, 2006.

F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11:1–21, 1969.

Allan Hanbury. Automatic image segmentation by positioning a seed. In *In Computer Vision European Conference on Computer Vision 2006, volume 3952*, pages 468–480, 2006.

J. A. Hartigan and P. M. Hartigan. The dip test of unimodality. *The Annals of Statistics*, 13, 1985.

Chiho Kim, Bum-Jae You, Mun-Ho Jeong, and Hagbae Kim. Color segmentation robust to brightness variations by using b-spline curve modeling. *Pattern Recogn.*, 41(1):22–37, 2008.

L. Ladicky, P. Kohli, and P. Torr. Oxford brookes entry, pascal voc2007 segmentation challenge, 2007.

Christoph H. Lampert, Matthew B. Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR*, pages 1–8, 2008.

Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *In ECCV workshop on statistical learning in computer vision*, pages 17–32, 2004.

Anat Levin and Yair Weiss. Learning to combine bottom-up and top-down segmentation. In *In ECCV*, pages 581–594, 2006.

Antonio M. Lopez, Felipe Lumbreras, Joan Serrat, and Juan J. Villanueva. Evaluation of methods for ridge and valley detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:327–335, 1999.

Tomasz Malisiewicz and Alexei A. Efros. Improving spatial support for objects via multiple segmentations. In *British Machine Vision Conference (BMVC)*, September 2007.

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 Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

Marina Meila. Comparing clusterings by the variation of information. pages 173–187. 2003.

A.P. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel lattices. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 23-28 2008.

Caroline Pantofaru, Cordelia Schmid, and Martial Hebert. Object recognition by integrating multiple image segmentations. In *European Conference on Computer Vision*, volume III, pages 481–494, 2008.

Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *In Proc. 9th Int. Conf. Computer Vision*, pages 10–17, 2003.

Bryan C. Russell, Alexei A. Efros, Josef Sivic, William T. Freeman, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proceedings of CVPR*, June 2006.

Steven A. Shafer. Using color to separate reflection components. pages 43–51, 1992.

Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22, 2000.

J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *In ECCV*, pages 1–15, 2006.

J.D.J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. pages 1–8, 2008.

R. Unnikrishnan, C. Pantofaru, and M. Hebert. A measure for objective evaluation of image segmentation algorithms. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 34, 2005.

Ranjith Unnikrishnan, Caroline Pantofaru, and Martial Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:929–944, 2007.

Eduard Vazquez, Joost Weijer, and Ramon Baldrich. Image segmentation in the presence of shadows and highlights. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 1–14, 2008.

A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

V. Viitaniemi. Helsinki university of technology, pascal voc2007 challenge, 2007.

J. Van De Weijer, Th. Gevers, A.D. Bagdanov, I. A Edge, Feature Detection, and I. D Statistical. Boosting color saliency in image feature detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28: 150–156, 2005.

J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 756–763, 2005.

Allen Y. Yang, John Wright, S. Shankar Sastry, and Yi Ma. Unsupervised segmentation of natural images via lossy data compression. Technical Report UCB/EECS-2006-195, EECS Department, University of California, Berkeley, 2006.

Alan Yuille and Peter Hallinan. Deformable templates. pages 21–38, 1993.

L. Lucchese Yz and Y S. K. Mitra. Color image segmentation: A state-of-the-art survey, 2001.