ABSTRACT
There is a general trend in recent methods to use image regions (i.e. super-pixels) obtained in an unsupervised way to enhance the semantic image segmentation task. This paper proposes a detailed study on the role and the benefit of using these regions, at different steps of the segmentation process. For the purpose of this benchmark, we propose a simple system for semantic segmentation that uses a hierarchy of regions. A patch based system with similar settings is compared, which allows us to evaluate the contribution of each component of the system. Both systems are evaluated on the standard MSRC-21 dataset and obtain competitive results.

Keywords
Semantic image segmentation, bag-of-regions

1. INTRODUCTION
The goal of semantic image segmentation and labeling is to produce a pixel-wise labeling of images according to a pre-defined set of semantic categories. It is a necessary pre-processing step for many applications, such as multimedia retrieval, scene description and parsing, as well as embedded computer vision systems such as driving assistance or video surveillance. Semantic image segmentation and labeling is a supervised learning problem in contrast to low-level region detection, at different stages of a more sequential framework [30, 8].

Local appearance is often described by Gaussian derivative filter outputs, colors, and locations computed for each pixel (called textons in this case) [26, 27], or SIFT and color statistics (histograms or moments) computed on image patches extracted either on a (multi-scale) grid [28, 8] or at detected interest point locations [30]. These low-level features can also be used to build a higher level representations such as Semantic Texton Forests [26], Bag-of-visual words [28] or Fisher Vectors [8]. High level representations are often fed into a classifier that predicts class labels at patch level [15, 8, 29], pixel level [27] or region level [30].

Local consistency is generally enforced via pairwise constraints between neighboring pixels and can be based on a simple Potts model that penalizes equally all class transitions between neighbor pixels [28] or can be contrast sensitive where the penalty depends also on the values of the pixels. To enforce region-level consistency, higher order potentials can be added to the CRF model [13, 15, 16], or alternatively, hard constrains ensure that all pixels within a low-level region have the same label. The latter can be done as a post processing step, e.g. averaging the class likelihood of each pixels within a region [8] or describing directly the regions and learning to predict class labels at region level [30]. Regions are described either as an aggregation of local patch information (e.g. Bag-of-visual words) or with global features, e.g. describing the shape of the region.

Finally, the global consistency can be integrated directly in the CRF model [28] or predictions of image-level classifiers, trained based on the presence/absence of the objects, are used to modify the local class probabilities [26] or to pre-select local classifiers for a given image [8].

Recently, in semantic image segmentation, there is a general trend to combine recognition (local appearance) with low-level segmentation methods, that produce regions, also called super-pixels, and to use these regions either before or after the appearance based classification step. These methods rely on an unsupervised (low-level) segmentation algorithm, such as Mean Shift [7], or the method from Berke-
Figure 1: Illustration of our hierarchical region based semantic image segmentation system. Images are first segmented into a hierarchical region tree that is transformed into a binary tree. Each region in the tree is represented by appearance features and global shape descriptors that are fed into classifiers. Region scores are aggregated yielding to class probabilities at pixel level and combined with image level prior information. Optionally, a CRF regularization can be considered as final labeling step. See section 2 for details.

A large portion of methods using super-pixels consider regions as their building blocks, for the computational cost aspect. Appearance models are built at the regions level (instead of pixel or patch level) building region descriptors that aggregate some lower-level information, like texture or color [11, 18, 29]. These methods can also implement a regularization mechanism that works directly at region level, i.e. a CRF model on super-pixels [20]. The complexity is significantly reduced.

Other methods use regions only at a later stage. Probabilities at pixel-level are regularized to ensure the label consistency within each region. This can be done by simple post-processing (e.g. averaging the class likelihood of pixels in a region [8]) or using super-pixels in the CRF to enforce the label consistency within regions, as for instance done by the P^n-Potts model [13] or the Harmony model [9].

The main limitation of methods working only with regions is that there is no possible recovery if a region groups multiple classes. To overcome this, many methods use low-level segmentation algorithms with parameters that produce over-segmentations. As a drawback, the size of super-pixels is often too small to convey enough information for the recognition, and regions rarely share boundaries with the class boundaries.

Alternative solutions include using multiple segmentations to obtain overlapping sets of regions [24, 13, 10], or building and exploiting a hierarchy of regions [15, 11, 18, 23, 17] or a graph of regions [5]. Such methods often consider a large pool of overlapping segments that are generally larger than super-pixels and use simple Hough voting followed with a verification strategy to choose segments from the pool [11]. CRF models are sometimes applied, based on a joint optimization over the choice of segments and semantic labels [10]. These approaches do not really exploit the hierarchy or the graph structure between image regions but rather uses the set of segments as a Bag-of-Regions (BoR).

A few approaches were proposed that go beyond a simple bag (or pool) of regions and exploit the relationship within the tree or graph of regions. In [18], each leaf node is represented by its ancestral set, that are all regions linking the leaf to the root node. For each ancestry of a leaf node, a different weight is learnt for each feature and each category. As far as we know, this is the only method that uses the tree structure at classification time, and this method does not scale, as all training region weights have to be stored.

Most methods use the region hierarchy at a post-processing stage. Ladicky et al. [15] define hierarchical associative CRFs where class labels are assigned to all regions in the hierarchy and the consistency between different levels of the tree is enforced with higher-order potentials. In the pylon model proposed by Lempitsky et al. [17], the hierarchical relationship is exploited by constraining the CRF to select only non overlapping segments from the tree for labeling. Similarly, Chen et al. [5] use the paths on the segment graph to add constraints in their cost function that is optimized using the Integer Projected Fixed Point algorithm.

These papers clearly show the benefit of using a region
hierarchy compared to using only information contained in the leaves of the tree either seen as a partitioning of the image [18] or as a flat CRF [15, 17].

Looking at the successful results obtained by the previously listed methods, there is no doubt that the semantic segmentation task strongly benefits from using super-pixel information, at one step or at another. The goal of our paper is to study the exact role that the regions play. More specifically, we are interested in answering the following questions: Do regions perform better than lower level entities, like patches or pixels? Can we exploit the hierarchy of regions into the classifier in a scalable way? What is the best way to enforce local, and global consistency? And more importantly, what is the most beneficial use of a set of regions, obtained from unsupervised segmentation, to enhance semantic image segmentation?

In this paper, we propose to systematically study the role of regions. For that purpose, we present a simple yet effective framework that allows to answer the questions raised above. Two full semantic segmentation systems are compared, a patch based and a region based one. Their recognition part uses exactly the same low-level information, either directly or aggregated within regions. Still, the region-based method does not have the drawbacks as the methods working directly on regions, i.e. it can recover from wrong super-pixels, by reasoning directly at the pixel-level for final labeling. As many of the previously cited papers, our system is based on the hierarchy of regions of [1]. The different systems are tested and compared on the MSRC dataset [27], one of the most used datasets in this area.

On this dataset, we first show the benefit of using regions to accumulate patch-level information to enhance the recognition part of our model. We then propose a simple way to use the hierarchy to improve shape-based recognition. We show the very competitive results obtained by combining the classifier with some simple image-level information, as a global classifier and a location prior. Finally, we show that integrating region information within the pairwise potential of the CRF model of [14] leads to performances close to state-of-the-art results, for both classifier settings: with and without aggregation in regions.

The rest of the paper is organized as follows. Section 2 describes our hierarchical region based semantic image segmentation system and its patch based counterpart. Section 3 presents the experiments and discusses the results. Conclusions are drawn in the last section.

2. PROPOSED FRAMEWORK

For the purpose of our benchmark we consider a segmentation framework that can take two different flavors: Region Based Semantic Image Segmentation (RB-SIS) and Patch based Semantic Image Segmentation (PB-SIS). This framework relies on several components (see Figure 1). Each image is segmented into a hierarchy of regions. Appearance (and, if relevant, shape) features are extracted and fed to a classifier either directly or first aggregated within a region. We will refer to this first phase as the local recognition step as it predicts locally the likelihood of a patch or a region to belong to a given class. These predictions are aggregated for each pixel over all patches (resp. regions) yielding a probability map for each class. A global classifier and a prior model on positions are combined to local cues. A CRF regularizes the obtained per pixel predictions. All these steps are detailed in the following sections.

2.1 Hierarchy of regions

In this section, we describe how an image is represented as a binary region tree. Any low level image segmentation could in principle be applied (e.g. Mean Shift [7]) and then transformed into a binary tree by an iterative merging process. Here we consider the algorithm proposed in [1], as this segmentation algorithm produces directly a tree of image regions and was shown to provide state-of-the-art low level image segmentation performances on the Berkeley Segmentation Dataset and Benchmark [2].

This approach [1] segments an image from the finest level (the leaf layer) to the coarsest level (the root node is the entire image) starting with the gPb contour detector [21]. The outputs of the gPb contour detector are weighted boundaries based on local image brightness, color and texture consistency. Then Oriented Watershed Transform and Ultrametric Contour Map (OWT-UCM) are used to generate the hierarchical region tree from the weighted boundaries (see for an example Figure 2 and 3(left)).

However, as we can see from Figure 3(left), in the region hierarchy of [1], a parent region can be segmented into any number of sub-regions. Using a general tree makes the exploitation of the tree structure more challenging. Therefore, we propose to transform the hierarchy of regions into a binary tree. This is done by splitting each layer that has more than two children into multiple layers, by iteratively merging two adjacent regions into a new parent (and hence adding an extra layer). This process is illustrated through an example in Figure 3.

Combining all the regions of all the layers in the binary tree produces the bag of regions (BoR) considered in our framework.

2.2 Local appearance features

We use Fisher Vector representations [25] of local image patches as in [8]. The Fisher Vector, which combines the benefits of generative and discriminative approaches, has shown [9] to be highly competitive even with linear classifiers.

These Fisher Vectors are built on low level image features.
In our case we use SIFT [19] and LCS [6] to describe the appearance feature of local image patches. The LCS (Local Color Statistics) features, are concatenated means and standard deviations in the R, G and B channels computed on the same 4x4 local grid used for SIFT. These patches (of size 24x24 in our case) were extracted densely (every 5 pixels) on a single or on a multi-scale (5 scales) image grid. The dimensionality of both features is reduced by PCA to 50. In the reduced feature space we build a visual vocabulary (Gaussian Mixture Model with 64 Gaussians) and transform for each patch the low-level representation into a Fisher Vector (FV) as described in [25, 8].

For the Patch Based Semantic Image Segmentation (PB-SIS) framework that does not use regions at early stages, we consider the patches individually, build a FV using SIFT or LCS color feature and train appearance classifiers at patch level as in [8]. For the Region Based Semantic Image Segmentation (RB-SIS) framework, each region is described by a single or on a multi-scale (5 scales) image grid. The dimensionality of both features is reduced by PCA to 50. In the reduced feature space we build a visual vocabulary (Gaussian Mixture Model with 64 Gaussians) and transform for each patch the low-level representation into a Fisher Vector (FV) as described in [25, 8].

For the related experiments is to test if using these features in combination with appearance improve segmentation. We do not compare with other shape features, as this is beyond the scope of our paper (and some preliminary experiments with several features from those mentioned above showed the superiority of the gPb descriptor in this experiment).

2.4 Bag-of-Triplets
As our regions are organized in a hierarchy, we would like to analyze if this structure can help segmentation. Seldomly studied, we look at how to integrate this hierarchical structure within the recognition process.

In order to achieve this, our framework goes beyond Bag-of-Regions (BoR) representation [11] and proposes a Bag-of-Triplets (BoT) representation to exploit the hierarchical region tree generated by [1]. Our BoT representation is built using the binary region tree described in section 2.1.

A triplet is a parent node with its two children nodes in the binary region tree. A binary region tree of n layers contains 2 × n − 1 nodes (regions) and n − 1 triplets. In order to represent a triplet, the FV (resp. shape feature) of a region is simply concatenated with the FVs (resp. shape features) of the parent region and its two children regions in the hierarchy. Such representation keeps a fixed length for all triplets and can be used to train a classifier. Note that both in learning and in labeling this feature is used to represent the parent region.

This representation can be seen as a trade-off between the method proposed in [11], that uses regions independently; and the method of [18], that uses only leaves for classification, but enriched by all the parents in the hierarchy. Our BoT uses the hierarchy only locally, but allows to use a classifier, while [18] is based on a voting strategy that requires to keep all training regions.

2.5 Pixel level class probability maps
To be able to obtain a probability map for each class, i.e. a map that encodes for each pixel the probability of belonging to each of the semantic categories, we learn the classifier either at patch level or at region level for each system (PB-SIS or RB-SIS) accordingly, and then aggregate class predictions over the overlapping patches (respectively regions).

In order to train these classifiers, we need annotations, i.e. each patch and each region in the dataset has to be individually labeled. This is done using the pixel level ground-truth masks as follows. Regions that have at least 25% of area overlapping with the ground-truth mask of an object class are considered as positive regions of that category. The remaining regions are used as negative examples. Similarly, for PB-SIS, we label the patches with the same 25% overlap rule between a patch and ground-truth masks.

Then, for each system (PB-SIS or RB-SIS) we train a one-versus-all classifiers per feature or triplets of features (in the case of BoT), per class. At test time, the classifiers are applied to all patches or regions of the test images.
respectively. The scores are converted into probabilities using a sigmoid function. The final pixel labeling is done by accumulating the information from all patches/regions containing that pixel using a mixture model with equal weights.

### 2.6 Image-level prior

Looking at the image as a whole brings additional information. We build a global FV per image aggregating all patches belonging to that image\(^2\).

Each training image is labeled globally for all the classes, depending on whether it contains at least some pixels of that class or not. Global classifiers, trained from these labels, can produce for each test image a probability of the presence of that class in the image. These global probabilities can be thresholded to allow only some classes in the final segmentation [8], or used to weight the local classifiers [28]. We follow the latter strategy.

As the location of a pixel can also hold class based information (e.g. sky is more probable at the top of an image, while grass more on the bottom), we build a location prior model (a location probability map per class) using the training annotations.

Finally, the probability that a pixel \( x_i \) belongs to category \( c \) is derived as:

\[
p(c|x_i) = (1 - \alpha - \beta)P_r^c(x_i) + \alpha P_c^c(x_i) + \beta P_f^c(x_i),
\]

where \( P_r^c \) is the class probability map obtained in the recognition step, \( P_c^c \) is a uniform map containing the likelihood of the class predicted by the global classifier trained for the class \( c \), \( P_f^c \) is the class location prior for the class \( c \), and \( \{\alpha, \beta, \alpha + \beta\} \in [0,1] \). The relative contribution of these maps (driven by the parameters \( \alpha \) and \( \beta \)) is learned using the validation set.

The final labeling of pixel \( i \) can be obtained by:

\[
y_i = \text{argmax}_{c \in \mathcal{C}} p(c|x_i).
\]

Note that using \( \alpha = \beta = 0 \), we obtain the segmentation based only on the local recognition.

### 2.7 Fully connected CRF model

Label consistency between pixels is often obtained using neighboring pixels or neighboring patches as pairwise potentials in CRF models. In our system, we use a fully connected dense CRF (denoted as ‘dCRF’), that models the pairwise dependencies between all pairs of pixels, and apply the inference presented in [14]. According to this model the following energy function is optimized per test image:

\[
E(x) = \sum_i \psi_u(x_i) + \sum_{i<j} \delta_{x_i,x_j} \psi_p(x_i,x_j),
\]

where the unary potential \( \psi_u(x_i) = -\log p(c|x_i) \) encodes the probability for pixel \( x_i \) to belong to class \( c \), according to the recognition part of our model. This can be patch or region based recognition with or without image level prior. The binary potential is defined as:

\[
\psi_p(x_i, x_j) = \omega_1 \exp \left( -\frac{|p_i - p_j|^2}{2\theta_i^2} - \frac{|I_i - I_j|^2}{2\theta_j^2} \right) + \omega_2 \exp \left( -\frac{|p_i - p_j|^2}{2\theta^2} \right)
\]

\(^2\)Note that using again the same FVs as in the recognition step leads to almost no additional cost at test time.

with \( p_i \) and \( I_i \) being the position and RGB value of pixel \( x_i \) respectively. The \( \omega \) and \( \theta \) parameters are optimized on the validation set. The first term of this pairwise potential encourages pixels with similar color to share the same label. The second term aims to remove the isolated regions.

In [14] \( \delta \) is a carefully designed function that defines the compatibility between labels. In our experiment we simplified this by using \( \delta_{x_i,x_j} = 0 \) if \( y_i = y_j \) (the labels assigned to \( x_i \) respectively \( x_j \) are the same), and 1 otherwise, which is also used in the Potts model.

In order to take into account the region information, we propose an adapted formulation of the dense CRF by adding the following additional term to the pairwise potential:

\[
\hat{\psi}_p(x_i, x_j) = \psi_p(x_i, x_j) + \omega_3 \exp \left( -\frac{|p_i - p_j|^2}{2\theta_i^2} - \frac{|R_i - R_j|^2}{2\theta^2} \right)
\]

where \( R_i \) is the leaf region that contains the pixel \( x_i \). Note that only the leaves (bottom layer of the hierarchy) are considered in the potential. Intuitively, this third term brings an additional constraint that encourages pixels belonging to the same super pixel to share the same label. This adapted dense CRF model is denoted as ‘dCRFSP’ afterward.

### 3. EXPERIMENTS

In order to evaluate the role and the contribution of each component of the proposed systems, we compare the region based and patch based segmentation systems using exactly the same patches and related appearance features. For simplicity, we refer to RB-SIS and PB-SIS as ‘R’ and ‘P’ in what follows.

We first compare the two systems using appearance features alone; then we add shape features and structure to the region based system. Next, we consider the improvement带来的 by the global image and location priors, and the CRF models. Finally, we compare our results to the results published on this dataset.

#### Dataset and evaluation.

We evaluate our systems on the MSRC-21 dataset, that contains 591 images of 21 categories, and we use the standard split of the dataset [27], i.e. 276 training, 59 validation and 256 test images. All parameters are optimized on the validation set. Average pixel-wise classification accuracy of each category is used to measure the performance on the test set.

#### Local appearance features.

First, we compare the two systems using patches extracted at either a single scale or 5 different scales. Color only (LCS), SIFT only and their combination (denoted as ‘APP’') are evaluated. Results in Table 1 show that when using appearance features alone, RB-SIS outperforms PB-SIS. As we can see, the differences are less important under multi-scale scenario. This can be explained by the larger patches extracted at higher scales conveying larger range information. This experiment clearly demonstrates that in the case of such a simple framework, regions are a great asset that improves recognition, and thus pixel-level labeling.

#### Shape features.

One advantage of using regions is that their shape can hold extra information. However, the result of using BoR shape only (first number in Table 2) shows that shape alone performs poorly compared to appearance feature based results (see Table 1). Their combination (weighted late fusion at pixel prediction level) brings...
Table 1: Both systems\(^2\) results on appearance only.

<table>
<thead>
<tr>
<th></th>
<th>One scale (1S)</th>
<th>Multi scale (MS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(-P)</td>
<td>(-R)</td>
</tr>
<tr>
<td>LCS</td>
<td>55.72</td>
<td>62.84</td>
</tr>
<tr>
<td>SIFT</td>
<td>46.10</td>
<td>61.98</td>
</tr>
<tr>
<td>APP</td>
<td>63.63</td>
<td>70.24</td>
</tr>
</tbody>
</table>

Table 3: Post processing for both systems

<table>
<thead>
<tr>
<th></th>
<th>REC + GL</th>
<th>dCRF</th>
<th>dCRFSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-P)</td>
<td>69.98</td>
<td>75.20</td>
<td>76.69</td>
</tr>
<tr>
<td>(-R)</td>
<td>72.99</td>
<td>75.88</td>
<td>76.02</td>
</tr>
</tbody>
</table>

Table 2: Results when using shape in RB-SIS

<table>
<thead>
<tr>
<th></th>
<th>shape (+)</th>
<th>(+P)</th>
<th>(+APP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoR</td>
<td>34.77</td>
<td>70.35</td>
<td>70.85</td>
</tr>
<tr>
<td>RT</td>
<td>42.70</td>
<td>71.18</td>
<td>72.90</td>
</tr>
</tbody>
</table>

For the rest of the experiments, we use the best version of the patch based system from Table 1 (Multi-scale APP-P) as our patch based recognition model (denoted as ‘REC-P’) and the best version of our region based system from Table 2 (Multi-scale APP-R combined with RT shape feature) as our region based recognition model (denoted as ‘REC+GL’). These results are shown again in the first column of Table 3.

The recognition models of both systems are first enhanced with simple image level priors, using the global image classification and the location prior (denoted as ‘+GL’), as described in section 2.6, and results are shown in the second column of Table 3.

Comparing the results with and without image level prior, we can see that both models strongly benefit from this simple global information, PB-SIS benefiting the most. Similar conclusion is also drawn in [20].

CRF based regularization. The pixel-level predictions obtained after considering the image level prior (‘REC+GL’) for both systems are then regularized using the two considered dense CRF models (‘dCRF’ and ‘dCRFSP’). Results are presented in the last two columns of Table 3.

We observe that this more sophisticated post-processing does not bring any notable performance improvement to our region based system. On the contrary, the patch based model benefit from CRF regularization, and results are improved further, even without using information about regions or region hierarchy in the regularization part of the model.

Visual examples. To complement our quantitative study, some segmentation results are shown in Figure 4. Although not always captured by the evaluation measure (partially because of the coarse ground truth) we observe that visually, the regularization induced by the CRFs produces nicer segmentations for both systems. In particular, PB-SIS using the adapted dense CRF (‘dCRFSP’), displayed in the fifth column, shows significant improvement compared with the results obtained without the CRF (third column).

Although the average accuracy of PB-SIS combined with dCRFSP regularization is slightly higher than that of RB-SIS, visual results of the latter (sixth column) suggest that the method obtains more consistent regions and better overlap with contour of objects (e.g. legs and ears of the sheep, the head of the bird and the legs of the chair) than the patch based method.

Comparison to the state-of-the-art. Finally we compare the full systems for PB-SIS and RB-SIS to some state-of-the-art methods. We also discuss the version of RB-SIS that uses no CRF regularization, as although being simple it also obtains competitive results.

Table 4 reports the per class and the average of all the per class accuracies, while Table 5 also shows global pixel accuracy that gives the percentage of correctly classified pixels over all categories\(^3\).

Table 5: Average class accuracy (ACA) and overall pixel accuracy (OPA) for our best methods and some of the state-of-the-art methods.

<table>
<thead>
<tr>
<th></th>
<th>dCRFSP-P</th>
<th>dCRFSP-R</th>
<th>[15]</th>
<th>[20]</th>
<th>[14]</th>
<th>[5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACA</td>
<td>77</td>
<td>76</td>
<td>75</td>
<td>78</td>
<td>78</td>
<td>69</td>
</tr>
<tr>
<td>OPA</td>
<td>86</td>
<td>83</td>
<td>86</td>
<td>79</td>
<td>86</td>
<td>75</td>
</tr>
</tbody>
</table>

From Table 4 and Table 5, we can see that our region based system without CRF based regularization (denoted as ‘REC+GL-R’), significantly outperforms [18] and [5], although both methods exploit a region hierarchy, and it has similar performances to some state-of-the-art CRF based regularization systems such as [15, 9].

Our best results are obtained with PS-SIS, when the region information is exploited within the dense CRF’s binary potential (dCRFSP-P) yielding to results close to the best state-of-the-art methods on both measures (see Table 4 and Table 5). We have the best results for 5 classes out of 21.

Note that our adapted dense CRF model (dCRFSP) uses simple constraints based only on the leaf regions exploited

\(^3\)The results of [14] and [5] appears only in Table 5 as only average results are reported in the papers.
in the pairwise potential. Hence adding more complex constraints to exploit the structure such as the ones proposed in [15, 17, 5] or learning the label compatibility parameters as in [14] would probably allow to further increase its performance.

4. CONCLUSION

In this paper, we have studied the different roles that regions could play and the benefit they bring in a semantic image segmentation and labeling system. For the purpose of this benchmark, we proposed a simple system for semantic segmentation that uses a hierarchy of regions. We considered the new way of using the structure of the hierarchy of regions and global constraints used in CRFs model are not always necessary.

When a CRF model is considered for regularization, we have seen that the patch based system becomes competitive and obtain good results. In this case, it is sufficient to use the regions and region structures only in later stages to ensure the label consistency within regions. This complement the conclusion of recent papers such as [15, 17, 5].

5. REFERENCES


Figure 4: Example segmentations. From left to right: original image; groundtruth; patch based results using appearance feature with image level priors (REC+GL-P); region based results using the shape and appearance feature with image level priors (REC+GL-R); patch based results after regularization with the adapted dense CRF (dCRFSP-P); and region based results with dense CRF regularization (dCRFSP-R).