Multiple Instance and Active Learning for weakly-supervised object-class segmentation.

Jaume Amores[1], David Gerónimo[1], Antonio López[1]
[1]Computer Vision Center, Universitat Autònoma de Barcelona
{jaume,davidg,antonio}@cvc.uab.es

Abstract

In object-class segmentation, one of the most tedious tasks is to manually segment many object examples in order to learn a model of the object category. Yet, there has been little research on reducing the degree of manual annotation for object-class segmentation. In this work we explore alternative strategies which do not require full manual segmentation of the object in the training set. In particular, we study the use of bounding boxes as a coarser and much cheaper form of segmentation and we perform a comparative study of several Multiple-Instance Learning techniques that allow to obtain a model with this type of weak annotation. We show that some of these methods can be competitive, when used with coarse segmentations, with methods that require full manual segmentation of the objects. Furthermore, we show how to use active learning combined with this weakly supervised strategy. As we see, this strategy permits to reduce the amount of annotation and optimize the number of examples that require full manual segmentation in the training set.

1. Introduction

In the object-class segmentation problem, the objective is to automatically segment out of the image instances of some object category, such as for example instances of cars in images. Current techniques for solving this problem are based on introducing a training set of object examples carefully segmented by hand, so that the system can learn an appearance model of the object category [1]–[5]. For this purpose, a user must delineate carefully the contour of every object example in the training set (fig. 1(a)), which usually takes several minutes per image. This high cost makes it impractical to introduce a large number of object examples to the system, which lowers the quality of the learned model, and also reduces the scalability in terms of the number of object classes that can be learned for segmentation.

In this work we analyze coarser forms of segmentation that are not completely accurate but allow us to save time per annotated image. This saving of time per image allows us to annotate more images spending the same total amount of time, which might eventually lead to obtaining more accurate models of our object. In particular, we study the use of bounding boxes as coarse form of segmentation (fig. 1(b)), which is much faster than an accurate delineation of the contour. The problem with coarse segmentations (such as bounding boxes) is that some pixels inside one bounding box belong to the background. In other words, a bounding box will contain *positive* pixels (located onto the object) and *negative* pixels (located onto the background), and we ignore a priori what are the positive ones.

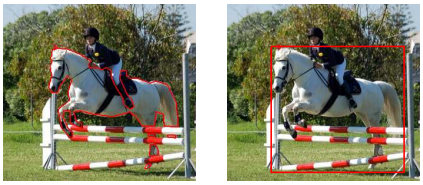
In order to learn a model with this type of loosely annotated data, we explore the use of Multiple Instance Learning (MIL) techniques [6]–[8] in our setting. This type of approaches try to automatically identify the relevant data (i.e., the positive pixels in our problem) and learn a model based on it. In this work, we perform a comparative analysis of three different MIL algorithms applied to object-class segmentation. In addition to this comparative analysis, we study whether or not using the proposed bounding box delineation together with MIL techniques and a high number of delineated examples (thanks to the reduction in cost per example) it is possible to equal or improve a full, accurate segmentation of the object. Finally, after learning a first model based on bounding box delineation and MIL, we study the use of Active Learning [9] for improving the initial model. Active Learning allows to automatically select the most interesting examples to be manually annotated (in our case examples that are to be fully segmented by hand) based on an initial model.

The proposed intermediate-level, coarse segmentation, and the analysis of its performance compared with full segmentation is novel and has not been performed

until now. In addition, there has not been any previous comparative analysis of different MIL paradigms for object-class segmentation, which is also a contribution of our work, and finally the use of Active Learning for optimizing the examples to be manually segmented is also novel and represents the third contribution of our work.

Related works in the literature are those of Pantofaru et al. [10] and Verbeek and Triggs [11], who also use MIL algorithms for object-class segmentation. They do not use bounding boxes as intermediate-level segmentation, and they just use the whole image without any type of indication of segmentation. Using the whole image works well for databases where the object occupies a big area of the image, but leads to very poor results in more difficult and realistic databases such as VOC where the object might occupy a tiny area of the image. In their works, the authors use generative MIL algorithms and do not compare against other alternatives. In our work, we perform a comparative analysis between different MIL algorithms applied to object-class segmentation. We do not only use generative approaches such as those of [10], [11], but also include discriminative approaches such as the Wrapper method described in the MIL literature [8]. As we observe in the results, using a discriminative type of algorithm leads to significantly better performance than the generative ones proposed in [10], [11].

In the following, we first describe the components of the proposed framework in section 2, we then explain the MIL algorithms studied in this work in section 3 and the Active Learning algorithm in section 4. Section 5 describes the results and comparative analysis and finally we conclude in section 6.



(a) (b)

Figure 1. (a) Full segmentation of the horse in one image, consisting of the silhouette of the horse manually delineated in red. (b) Coarse segmentation of the same image using a bounding box, also in red.

2. Components of the system

In this section we briefly describe the components of our system. They comprise the extraction of visual descriptors from the images, for describing the appearance of the object to be learned; and the learning algorithm, for obtaining a model of the typical appearance of the object. Let us describe each of module in turn.

2.1. Extraction of visual descriptors

Given a training set of images, the user delineates a bounding box for each example of the target object category. Each bounding box defines a rectangular region that is a sub-image I_i containing the object. We call such sub-image a positive example. The system then samples counter-examples (also called negative examples) which are rectangular regions of similar size and which are located close to each positive image without overlapping it.

For each of these examples I_i (both positive and negative), the system extracts a visual representation. In our case we used a similar representation to the one employed in [12]. First, the system performs a basic region segmentation that permits to obtain so-called *super-pixels*. Each super-pixel is a group of neighboring pixels with similar color and texture. In our case, we use the algorithm described in [13], which is based on K-means clustering on top of low-level features such as LUV color and Daubechies wavelets, and we set the parameters so as to obtain over-segmentation, in order to avoid regions crossing the boundaries of the object. Once these regions (or super-pixels) are obtained, we extract a feature vector representing each region. For this purpose, Pantofaru et al. [12] propose to use a Bag-of-Words (BoW) histogram of the region. In our case we use Distance-based Bag-of-Words (D-BoW) described in [14], which is similar descriptor to BoW but was seen to provide slightly better results[1].

At the end of this process we have, for each sub-image I_i, a set of feature vectors $B_i = \{\vec{x}_1^{(i)}, \vec{x}_2^{(i)}, \ldots, \vec{x}_N^{(i)}\}$, where the j-th feature vector $\vec{x}_j^{(i)}$ is a D-BoW descriptor that describes the j-th super-pixel of the image I_i. The set of feature vectors B_i associated with the sub-image I_i is called a *bag* in the MIL literature. In the rest of the discussion, we will use the words *sub-image* and *bag* interchangeably. A sub-image or bag is composed of a set of super-pixels described by the feature vectors $\vec{x}_j^{(i)}$.

1. In order to obtain both BoW and D-BoW descriptors we need first to quantize the low-level features of each pixel. In this preliminary work we use the same LUV color and wavelets used for region segmentation. We can expect to obtain better results if more complex features such as SIFT [15] are used instead, but we let this for future work.

2.2. Learning module

From the previous step, we have a training set of positive and negative sub-images. Positive sub-images (i.e., those containing an example of the object class) will contain positive super-pixels, which are the ones located on the object. Negative sub-images, on the other side, will contain negative super-pixels belonging to the background. Based on this training set we can learn a model $\Theta = \{\theta_1, \ldots, \theta_M\}$ of the positive super-pixels, where M is the number of parameters of the model. This model is then used by a classification function $f_\Theta(\vec{x}) \in [0, 1]$ that provides the likelihood that the feature vector $\vec{x}$ corresponds to a positive super-pixel belonging to the object of interest. This function can then be used in order to obtain a likelihood map of the object given a new image, and we can segment the object out of the image by using some threshold on the likelihood.

Together with the classification function $f_\Theta(\vec{x})$, there is usually a post-processing spatial regularization process that forces the likelihood of neighboring super-pixels to be similar. This is usually carried out by different types of graphical models, mostly Markov Random Fields (MRF) and Conditional Random Fields (CRF), see [10] for an example, as they use a framework similar to ours. In this preliminary work we have not applied any spatial regularization post-processing, and we let this for future work. Note, however, that this post-processing just refines the quality of the classification function f_Θ. In other words, if we have two classification functions and the first one is better than the second, then the regularization step will not change this fact, so that the first classification function will continue to be better after regularization. In this work we concentrate on studying the performance of different classification functions $f_\Theta(\vec{x})$, some obtained with low human effort (coarse manual segmentation) and some with high effort (full manual segmentation).

Note that, if we apply full manual segmentation to the examples of the training set, then we are sure that all the super-pixels of every positive example will be positive, i.e., they will belong to the object of interest, while all the super-pixels of the negative examples will be negative and will belong to parts of background stuff. In this scenario we can apply standard machine learning algorithms such as Boosting [16] and SVM [17] in order to obtain an appearance model Θ of the positive super-pixels.

On the contrary, when we apply coarser segmentations such as bounding box delineation, then a positive sub-image will contain both positive super-pixels belonging to the object and negative super-pixels belonging to the background (see fig. 2), and we do not know what are the positive ones. The only thing we know is that the negative sub-images, i.e., those sub-images containing background, contain only negative super-pixels. In order to learn with this partially-labelled data, we can use Multiple Instance Learning techniques [6]–[8], which first try to identify the positive super-pixels in the training set, and then learn a classification function $f_\Theta(\vec{x})$ that distinguishes between positive and negative super-pixels.

3. Multiple-Instance Learning algorithms

In this work we have evaluated three different MIL algorithms: the generative approaches used by Pantofaru et al. in [10] and Verbeek and Triggs in [11], and the discriminative approach proposed in [18]. The first two approaches have been used for object-class segmentation by their respective authors, while the third one has not been used for object-class segmentation until now. Let us explain the idea of each of them, we do not enter into technical details due to lack of space.

In [10], the idea is to group the super-pixels into several clusters C_i, and identify those clusters with a higher proportion of positive super-pixels. For this purpose, all the super-pixels from all the sub-images are pooled together into one big set $\mathcal{S}$, and an unsupervised clustering algorithm such as K-means is used to partition $\mathcal{S}$ into M clusters $C_1, \ldots, C_M$, where M is a parameter of the system. Then, for the i-th cluster C_i we count the number N_p of super-pixels that originally come from a positive sub-image and the number N_n of super-pixels that come from negative sub-images. The higher the ratio of N_p versus N_n, the higher the likelihood that a super-pixel in C_i will be positive. Based on this idea, the authors define a so-called *relevance* function $\mathcal{R}(C_i)$ that provides the likelihood that the super-pixels in C_i are positive (see [10] for technical details). Given a new image containing the object of interest, this object can be segmented by taking those super-pixels of the image that have a high relevance, i.e., those super-pixels that fall into a cluster C_i where $\mathcal{R}(C_i)$ is high.

In [11], the authors propose to use the Probabilistic Latent Semantic Analysis (PLSA) technique introduced in [19]. This technique originally comes from the information retrieval field, where it is applied for classifying documents according to their content. The idea is to discover the probability $Pr(z|d)$ that the topic z describes the content of the document d. This probability is determined according to the words existing in the document, where some words will be more related to the topic z than others. In

order to obtain this probability, Hofmann [19] proposes to use an Expectation-Maximization algorithm that iteratively computes the following probabilities: $Pr(z_k|d_i, w_j)$, $Pr(z_k|d_i)$ and $Pr(w_j|z_k)$. Here, z_k is the $k-th$ topic, d_i is the i-th document of the training set and w_j is the j-th word of the vocabulary. In the Expectation step, the algorithm computes $Pr(z_k|d_i, w_j)$, i.e., the probability of having the topic z_k given that we have the document d_i and there is the word w_j in this document. This probability is determined according to the probabilities $Pr(z_k|d_i)$ and $Pr(w_j|z_k)$ which are computed in the Maximization step of the algorithm.

When applying the PLSA technique for object class segmentation, the images play the role of documents, the super-pixels of the images play the role of words, and the topics play the role of object-classes that might be found inside the image (here the background is considered another class of object). Given a super-pixel (word) w_j found in the current image (document) d_j, we want to determine the probability that this super-pixel belongs to the object-class (topic) z_k, i.e., we want to determine $Pr(z_k|d_i, w_j)$, which is obtained in the Maximization step of the PLSA algorithm as explained above. Although the original algorithm was completely unsupervised, Verbeek and Triggs [11] propose to introduce some sort of supervision by forcing the probabilities $Pr(z_k|d_i)$ to be zero for those object-classes z_k that are not present in the image d_i of the training set. A final detail is that the super-pixels S_j of the image are described by real-valued feature vectors $\vec{x}_j$, while the words w_j of the PLSA algorithm take discrete values $\{1, \ldots, W\}$. Therefore, we must first quantize the feature vectors into discrete values $\{1, \ldots, W\}$. This is usually done by clustering algorithms such as K-means, as in [11].

As a third MIL technique, we studied the use of the Wrapper method proposed by Frank and Xu in [18], which has not been used in object-class segmentation until now. Contrary to the other two methods, the Wrapper is based on standard discriminative classifiers such as Boosting [16] and SVM [17], which have been shown to be among the most powerful classifiers. Traditionally, standard classifiers have not been applied in Multiple-Instance Learning problems because they require that every feature vector of the training set has an associated label (positive or negative in the case of binary object-class segmentation). In the case of MIL problems, however, we only know the label of each bag B_i (corresponding to sub-image I_i, see section 2.1). We know that bags B_i corresponding to sub-images that contain the object of interest have a positive label while bags corresponding to sub-images that do not contain the object have a negative label. However, we ignore the particular label (positive or negative) of the feature vectors $\vec{x}_j^{(i)} \in B_i$ belonging to the bags. In order to solve this problem, Frank and Xu propose to simply assign the label of the bag to each feature vector contained in it. As a result, we obtain a fully labelled training set of feature vectors, and we can use standard and powerful classifiers such as Boosting and SVM.

Frank and Xu [18] argue that this strategy works well if we weight every feature vector appropriately. The idea is that the sum of the weights of the feature vectors inside one bag is constant for all the bags. This is done by assigning to the feature vector $\vec{x}_k^{(i)}$, belonging to bag B_i, the weight $\frac{1}{|B_i|}$, where $|B_i|$ denotes the number of feature vectors belonging to the bag B_i. Using this weighting scheme, we are giving the same importance to every bag of the training set, i.e., to every object example of the category we want to learn. In this work we apply the Wrapper method together with the Boosting classifier [16] (in particular, we use the Gentle Boost version proposed in [20]). Boosting is, together with SVM, one of the most powerful classifiers existing in the current literature, and it is more appropriate than SVM in cases where the training set is very large (in the order of hundreds of thousands of feature vectors), as applying SVM is infeasible in these cases due to its computational cost.

4. Active Learning

The idea of Active Learning is to use an initial model in order to carefully select the most interesting examples to be labelled by the human user in the learning process. In our case, we study the use of Active Learning for object-class segmentation as follows. In an initial step, the user performs a coarse manual segmentation of all the object examples by using bounding box delineation. With this coarse segmentation we can learn an initial model by means of the MIL algorithms described before. Based on this initial model, we can use techniques based on Active Learning in order to select the most interesting object examples that can undergo full manual segmentation. The user is asked to manually segment the examples selected by Active Learning, and a new, refined model is learned based on these examples together with the ones used before with coarse segmentation.

Active Learning is usually based on examining which examples are close to the decision boundary of the initial classifier. Those examples are the ones on which the belief of the classifier is weaker, i.e., where the classifier has more doubts and where we need the input of the human user to determine their real label (positive or negative).

In our case, the trained classifier $f_\Theta(\vec{x}) \in [0,1]$ is applied over feature vectors $\vec{x}$ corresponding to super-pixels (see section 2.2). Therefore, if we apply a standard Active Learning strategy, the system would select those super-pixels lying on the decision boundary of f_Θ, i.e., those super-pixels whose feature vector $\vec{x}$ receives a score $f_\Theta(\vec{x})$ close to 0.5. The system would then present these super-pixels to the user so that he or she would be able to label them as belonging or not to the object of interest. This would lead to an impractical interactive system, as there are thousands of super-pixels and we would need a high number of iterations with the user.

Instead of selecting interesting super-pixels, we would like to select interesting sub-images where the object can be manually segmented. For this purpose, we define a score $\mathcal{S}(I_i)$ for each sub-image I_i as follows:

$$\mathcal{S}(I_i) = \sum_{\vec{x} \in I_i} \exp\left(-\frac{(f_\Theta(\vec{x}) - 0.5)^2}{\sigma^2}\right),$$

where the value of the parameter σ was set heuristically to 0.5, which provided good results in practice. This score is higher for those sub-images containing a large number of super-pixels close to the boundary 0.5.

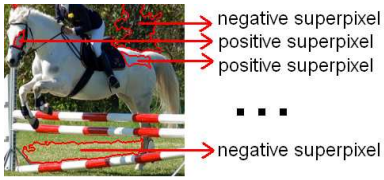


Figure 2. Positive sub-image containing both positive and negative super-pixels. One super-pixel is a region of contiguous pixels with similar color and texture.

5. Experimental analysis

For conducting our experiments, we used one of the most popular and difficult databases in the Object-class segmentation literature, the PASCAL Visual Object Classes (VOC) database [21] from the last edition. The complete database contains twenty categories, but not all of them are always used in the works. In this preliminary work, we used six categories of different nature: animals (horse, sheep, cow, cat) and man-made objects (bus, car). The VOC database provides, for each category, a small number of manual segmentations. Because this number is actually very small, we fused the VOC 2010 database [21] (which corresponds to the last edition) with the VOC 2007 database [22] (which corresponds to the first edition for the object-class segmentation task). In order to build our training sets, we took the images that were not manually segmented (only bounding boxes are provided for those images). For the test set, we took the images which were manually segmented, as we need the manual segmentation for our ground-truth. The statistics of the resulting training and test sets are listed in table 1. For each category, we took as negative examples windows randomly sampled from the background of each positive image, in such a way that the window is close to the bounding box delineated by the user but without overlapping it. The idea is to make the negative examples come from the background of the object, so that the system can be trained to differentiate between foreground (object) and background. For each positive image the number of sampled negative windows were twice the number of positive object examples contained in the image.

	Horse	Sheep	Cow	Cat	Bus	Car
Training (Positive examples, non-segmented)	936	1005	746	1336	659	3233
Test	99	146	118	162	114	197

Table 1. Number of images used for training and testing in each category

In table 2 we show the results for the different MIL methods of section 3, using the area under the precision-recall curve as a standard measure of performance [21]. These results were obtained restricting the test images to be the bounding boxes of the positive object. Note that the important thing here was to compare the performance between the different methods. Restricting the test image to be the bounding box of the object just decreases the amount of background, but does not change the relative performance of the methods. Also the differences between the methods remains very similar. We show here the performance with bounding box restriction just to focus on the relative performance of the methods. In the last row we show the performance that we would obtain with a random segmentation. This performance is around 50% for most of the categories, which means that the object occupies approximately 50% of the area of the bounding box in the test example, except for the bus and car categories, which consist of more rectangular objects, so that the object fits better the bounding box and occupies more than 50% of its area. Examples of segmentation results obtained with the Wrapper

Method / Database	Horse	Sheep	Cow	Cat	Bus	Car
Wrapper [18]	**71.1**	**72.5**	**77.1**	68.5	**86.1**	**76.6**
Relevance of clusters [10]	63.6	67.2	73.4	**70.5**	85.8	73.1
PLSA [11]	50.1	64.0	65.4	60.0	77.2	66.2
Random	46.7	57.7	58.3	56.5	76.2	65.2

Table 2. Results of the MIL methods.

method are shown in fig. 3 for some categories (horse, cow and car), just as an illustration.

For the Wrapper method, we found empirically that a uniform weighting scheme is indeed superior for our segmentation problem when we use Gentle Boost. Also, we tested other Boosting variants such as the AdaBoost version in [23], but Gentle Boost consistently provided better results. Therefore, the scores showed in table 2 for the Wrapper method are based on Gentle Boost and uniform weighting. As we can see, the Wrapper method significantly outperforms the other MIL methods in all but one category (cat), where the method of [10] is superior. Compared to the PLSA method, PLSA gets results just slightly better than random chance, and the other two MIL methods are clearly superior in all the categories. Our implementation of PLSA obtained good results on a different database, which is the MSRC database used by the original authors of this method [11], where we obtained a similar score (52%) as the one reported in [11] when the SIFT feature vector is used (random chance obtains around 10% of accuracy on that database). The MSRC database provides a very rich annotation of every image, in such a way that all the objects appearing in the image are labelled. The PLSA method considers the co-existence of the different objects in the image, in the sense that, for example, it is more probable to find a cow in an image if there is grass and it is more probable to find a car if there is a road, etc[2] The use of this contextual information (interaction between the different objects) makes it more appropriate to use the PLSA method in fully labelled databases such as the MSRC. However, indicating all the objects that appear in the image is a cumbersome task and becomes infeasible for complex and realistic images. In this work we want to reduce the amount of manual labelling required, i.e., we focus on the case where only one target object of interest (or just a few) is indicated in the image, as is the case of the VOC database.

If fig. 3 we show a few segmentation examples using the Wrapper method, i.e., with a training set that only usees bounding boxes as input, just for illustration purposes. The images were obtained by thresholding the likelihood obtained with Gentle Boost on every super-pixel. We used as threshold the one that achieves an equal error rate of the ROC curve, as done in [10].

Figure 3. Segmentation examples

In fig. 4 we compare the performance of a full segmentation approach, which uses standard supervised learning, against the one based on coarse segmentation (bounding boxes), which uses multiple instance learning (we used the Wrapper method in this comparison). We show for this purpose the precision-recall curve obtained for both approaches. For the full segmentation approach we used Gentle Boost [20] in order to obtain a fair comparison. In order to fairly compare both strategies, we measured the time spent in manually segmenting the images with full segmentation (138.4 seconds per image, in average) and with bounding boxes (3.4 seconds per image, in average). In order to spend the same total amount of time for both approaches, the coarse segmentation uses 936 images while the full segmentation only uses 23 images. In both cases the total manual segmentation cost was about 53 minutes. As we can see in fig. 4, using the Wrapper method together with a high number of coarsely segmented images is advantageous over using a expensive approach based on accurate manual segmentation of a fewer number of images together with a standard supervised algorithm. The curves shown in fig. 4 correspond to the horse category, and we obtained a similar behavior for other categories (not shown for lack of space).

An important advantage of the proposed framework

2. This is why PLSA is successful in classification of textual documents [19], where the topics arise from the combination of different words in the same document, which in the case of images translates to the combination of different types of super-pixels in the same image

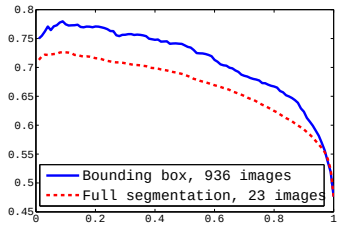


Figure 4. Precision-recall curve using bounding boxes and MIL (solid blue line) versus full segmentation and standard learning (dotted red line).

is the possibility of incorporating information from fully segmented images in a more efficient way through the use of Active Learning, as proposed in section 4. In fig. 5 we show the performance of using the proposed Active Learning strategy versus not using it. In the Active Learning case, we learned an initial model with a bounding box segmentation of 380 images, and used this initial model to iteratively select the most informative one hundred images that should be manually segmented by the user, as explained in section 4, before learning an updated model based on these fully segmented images. In the case of not using Active Learning, the one hundred images to be segmented are selected just randomly from the set of images. For lack of space we only show the graphic for the horse category, although a similar behavior is obtained for other categories.

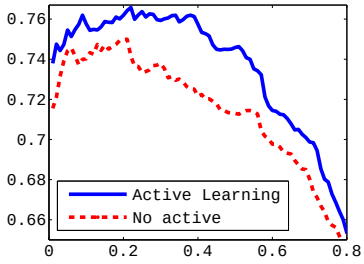


Figure 5. Precision-recall curve using the proposed active learning approach (solid blue line) versus not using it (dotted red line) for the horse category.

6. Conclusions

In this work we explored strategies that reduce and optimize the cost of manual segmentation necessary in object-class segmentation. For this purpose, we proposed to combine a coarse type of segmentation of the objects with an appropriate Multiple Instance Learning (MIL) algorithm such as the Wrapper method [18], which, as we showed here, provided the best results for our object-class segmentation problem. As we showed in the results, using the proposed strategy permits to obtain a clearly better performance than using a full manual segmentation coupled with a standard learning algorithm. This is because a full manual segmentation is very tedious and time consuming, which reduces the number of segmented objects that can be introduced to the learning system. In fact, we can see in standard databases such as VOC [21] that the number of fully segmented images is usually very small, in the order of a few dozens per training set. 0n the contrary, using a strategy based on coarse segmentation, with bounding boxes, we can segment a significantly higher amount of images spending the same amount of time. As we have seen, this extra amount of information is efficiently managed by a MIL algorithm such as the Wrapper method, leading to higher performance than the one obtained with full segmentation and a standard learning algorithm.

In addition to this analysis, we showed in the results that we can combine the proposed coarse segmentation and MIL with a posterior Active Learning algorithm. The coarse segmentation with MIL permits to learn an initial model and the Active Learning step permits to use this model in order to select the most informative objects that should be fully segmented by the user. For this purpose, we proposed in this paper a particular form of Active Learning that can be applied for object-class segmentation, and we saw in the results that it permits to augment the accuracy.

Finally, in addition to the two previous contributions, our work includes an important comparative analysis of different MIL algorithms that can be applied to the proposed framework. Two of these algorithms were already applied in the context of object-class segmentation, while the third one (the Wrapper method) has not been applied until now. As the results show, the Wrapper method provides a better performance in general, which might be explained by the fact that this method incorporates the advantages of powerful, standard discriminative learning algorithms such as SVM and Boosting, while the other two methods proposed until now are based in part on generative strategies which seems to be a less powerful approach.

References

[1] D. Larlus, J. Verbeek, and F. Jurie, “Category level object segmentation by combining bag-of-words mod-

els with dirichlet processes and random fields," *IJCV*, vol. 88, no. 2, pp. 238–253, jun 2010.

[2] J. M. Gonfaus, X. Boix, J. V. Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzalez, "Harmony potentials for joint classification and segmentation," in *IEEE Proc. CVPR*, 2010.

[3] F. Schroff, A. Criminisi, and A. Zisserman, "Object class segmentation using random forests," in *BMVC*, 2008.

[4] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes, "Layered object detection for multi-class segmentation."

[5] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Proc. ECCV*, 2006.

[6] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez, "Solving the multiple-instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, pp. 31–71, 1997.

[7] S. Ray and M. Craven, "Supervised vs multiple instance learning: an empirical comparison," in *ICML*, 2005.

[8] X. Xu, "Statistical learning in multiple instance problems," Master's thesis.

[9] Y. Abramson and Y. Freund, "Semi-automatic visual learning (seville): Tutorial on active learning for visual object recognition," in *IEEE Proc. CVPR*, 2005.

[10] C. Pantofaru and M. Hebert, "A framework for learning to recognize and segment object classes using weakly supervised training data," in *British Machine Vision Conference (BMVC)*, 2007.

[11] J. Verbeek and B. Triggs, "Region classification with markov field aspect models," in *IEEE Proc. CVPR*, 2007.

[12] C. Pantofaru, G. Dorko, C. Schmid, and M. Hebert, "Combining regions and patches for object class localization," in *The Beyond Patches Workshop, IEEE CVPR*, 2006.

[13] Y. Chen and J. Wang, "A region-based fuzzy feature matching approach to content-based image retrieval," *IEEE TPAMI*, vol. 24, no. 9, pp. 1252–1267, 2002.

[14] J. Amores, "Vocabulary-based approaches for multiple-instance data: a comparative study," in *ICPR*, 2010.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[16] Y. Freund and R. E. Schapire., "Experiments with a new boosting algorithm." in *Proc. Int'l Conference on Machine Learning.*, 1996, pp. 148–156.

[17] C. Cortes and V. Vapnik, "Support-vector networks," *Matchine Learning*, vol. 20, no. 3, pp. 272–297, 1995.

[18] E. Frank and X. Xu, "Applying propositional learning algorithms to multi-instance data," Department of Computer Science, University of Waikato, Tech. Rep., 2003.

[19] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, no. 1/2, 2001.

[20] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting." *Annals of statistics*, vol. 38, no. 2, pp. 337–374, 2000.

[21] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results," http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[22] ——, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[23] P. Viola and M. J. Jones, "Robust-real time face detection," *IJCV*, vol. 57, no. 2, pp. 137–154, 2004.