TriCoS: A Tri-level Class-Discriminative Co-segmentation Method for Image Classification

Yuning Chai[1], Esa Rahtu[2], Victor Lempitsky[3],
Luc Van Gool[1], and Andrew Zisserman[4]

[1] Computer Vision Group, ETH Zurich, Switzerland
[2] Machine Vision Group, University of Oulu, Finland
[3] Yandex, Russia
[4] Visual Geometry Group, University of Oxford, United Kingdom

Abstract. The aim of this paper is to leverage foreground segmentation to improve classification performance on weakly annotated datasets – those with no additional annotation other than class labels. We introduce TriCoS, a new co-segmentation algorithm that looks at all training images jointly and automatically segments out the most class-discriminative foregrounds for each image. Ultimately, those foreground segmentations are used to train a classification system.

TriCoS solves the co-segmentation problem by minimizing losses at three different levels: the category level for foreground/background consistency across images belonging to the same category, the image level for spatial continuity within each image, and the dataset level for discrimination between classes.

In an extensive set of experiments, we evaluate the algorithm on three benchmark datasets: the UCSD-Caltech Birds-200-2010, the Stanford Dogs, and the Oxford Flowers 102. With the help of a modern image classifier, we show superior performance compared to previously published classification methods and other co-segmentation methods.

1 Introduction

The impact of foreground/background segmentation on the overall performance of an image classification system depends strongly on the type of image data that the system has to work with. In general, backgrounds can contain important cues to support the discrimination [22,27], but sometimes the background appearance can also be non-informative or even shared across different classes and therefore cause confusion in the system. For fine-grained visual categorization or image search, there is corroborative evidence that foreground-background segmentation can improve the accuracy considerably [2,6,17].

Here, we focus on the segmentation of the training images for the task of fine-grained image classification (e.g. discrimination between a large number of bird, dog, or flower species). To be practical, and to achieve sufficient accuracy, fine-grained classification systems need to discriminate between large number of classes (hundreds to many thousands [8]) and to process relatively large number of training images per class. This means that the traditional supervised-learning approach that requires manual segmentation of a large number of images, may be expensive or even infeasible. Even if possible, supervised learning-based segmentation of training images [16] assumes that the

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part I, LNCS 7572, pp. 794–807, 2012.

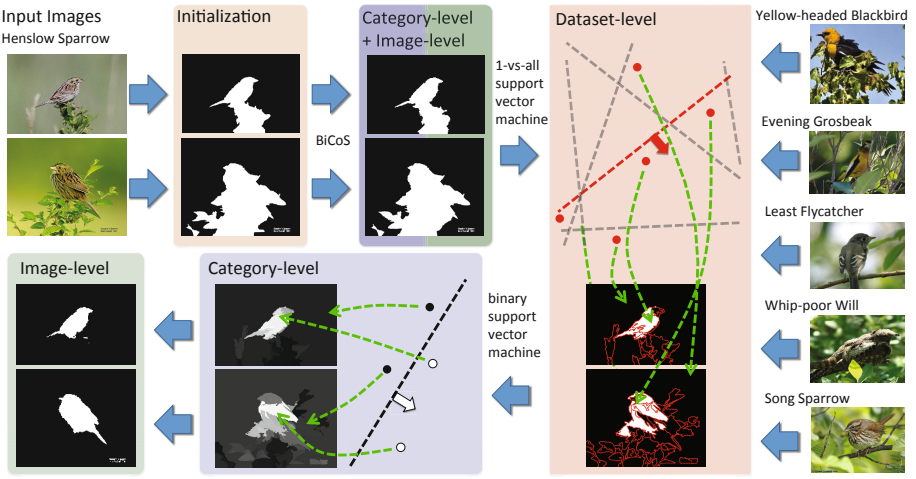

Fig. 1. An overview of the flow of the discriminative co-segmentation pipeline TriCoS. We start with the input images and their initialization segmentation masks on the top left. They are then processed by the BiCoS [6]. Although BiCoS often improves the segmentation quality, in this example, it preserves both the birds and the tree branches as foreground because they appear multiple times across the images from the same category. The new dataset-level step finds leaves to be less discriminative since most birds fly around trees and therefore discards the leaves in the segmentation.

foreground-background segmentation that is most natural for a human is also the most useful for fine-grained classification, which might not be the case for some datasets. In the remainder of the paper, the word "foreground" can mean both the human-seen foreground or the most class-discriminative region depending on the context.

Co-segmentation [21] provides an unsupervised alternative for the segmentation of training image sets. In this paper, we introduce a new scalable co-segmentation method that directly augments the co-segmentation objective with the discriminability of the categories for image classification. In more detail, the method operates on three levels: the category level, the image level and the dataset level. The category level reflects the similarity of foreground and background appearances across the images from the same category. The image level promotes the consistency of color distributions between the foreground and the background regions within each image separately. And the dataset level improves the discriminability of the foreground object for the fine-grained classification.

A component-wise scheme within our method, which we call TriCoS for Tri-level Co-Segmentation, has a complexity linear in the total number of images and does not assume any global geometric shape or foreground color distribution throughout the set. Thus, the method can be applied to rather large training sets and we are able to evaluate it through a series of experiments on the UCSD-Caltech bird dataset 200-2010 [25], the recently introduced Stanford dogs [15], and the Oxford-102 Flowers [17]. In our experiments, we observe that a modern image representation (Fisher Vectors [18]) trained on the datasets pre-segmented with TriCoS achieves considerably better than state-of-the-art classification accuracies.

2 Related Work

Depending on what information is assumed to be shared across images, the difficulty of the co-segmentation problem can vary greatly. While some early works only consider the co-segmentation of two images belonging to the same rigid object instance [21], later ones co-segment a set of dozens to hundreds of images taken from different object instances.

Our method builds on our earlier algorithm BiCoS [6], which is a co-segmentation algorithm that works on each image category separately. It operates under the mild assumption that many foreground objects are centered in the image, and initializes the alternation-based algorithm with a rectangle at the center of each image. During the iteration, it alternates between a GrabCut-like step that propagates information within each image and a superpixel-based SVM-like step that propagates information across all images in the category. In addition to these two levels of BiCoS, TriCoS embeds class-discriminative information into the co-segmentation process.

The variant of BiCoS, called BiCoS-MT [6] indirectly promotes the discriminability of the foreground regions by sharing the background appearance distributions across classes in the training dataset. It however has a limited scalability, and more importantly does not optimize the discriminability of the foreground in a direct way as TriCoS does. The experiments demonstrate that TriCoS is more efficient than BiCoS-MT at improving the final classification accuracy using the dataset-level information.

Related to the idea of training-set foreground-background segmentation are methods that generate real-valued saliency masks for training images. For example, Kanan and Cottrell [13] used natural image statistics to create saliency maps emphasizing regions that are likely to have more discriminative information. The saliency map is then treated as a probability density map of the image, according to which features are extracted and classification is done using a generative model in an iterative way. TriCoS is different from this approach as it generates segmentations using a segmentation model that is self-trained for a particular dataset with the goal of better discrimination between classes.

Yao et al. [26] proposed a random forest-based algorithm for fine-grained classification. Unlike conventional random forests, which use the same feature vector throughout entire trees, the authors let each tree node learn the location of the most discriminative features and use those combined with some inherited information from the parent node to make the decision. During the traversal of the tree, information of different locations is pulled together which results in the final decision. One limitation of this approach is the requirement of a bounding box during test time in order to have a normalized image.

Another work aiming at benefiting image classification with proper segmentation is [12]. There, the authors split each image into segment proposals and assume that one of the proposals contains the desired foreground object. By applying multiple-instance learning, the object location and the image classification can be achieved at the same time and evidence of improvement in classification accuracy was observed.

As our purpose is to evaluate the segmentation stage as a part of the joint system for fine-grained categorization, our work also relates to the vast body of work on image classification. To achieve the highest possible accuracy, for the classification stage we use Fisher Vectors [18], which is the state-of-the-art image encoding method, and

has achieved the best performance in a number of classification challenges [10] and comparisons [7].

3 Segmentation Methods

In the following sub-sections, we explain each of the modules shown in Fig. 1 in more detail. We start with the initialization step of the segmentation masks in Sec. 3.1, and then move on with a brief description of BiCoS [6], which consists of the category-level (Sec. 3.2) and the image-level steps (Sec. 3.3). On top of BiCoS, we introduce the new dataset level in Sec. 3.4. The pipeline terminates in our experiments after another BiCoS iteration appended to the dataset level. Further iterations between levels are possible, but in our experience, do not result in a significant improvement in the classification accuracy. Sec. 3.5 contains implementation details, including the choice of features for each of the levels.

Information is carried from one level to another in the form of masks, which, depending on the level, can be either hard segmentation masks (binary, either foreground or background) or soft saliency maps (real valued).

3.1 Initialization

BiCoS initialized the co-segmentation process using the central σ percent of pixels as a GrabCut initialization, with the hope that the category level can correct any error made by this simple assumption. While it performs well on many datasets where people tend to put the object of interest at the center of the images, we propose here a more principled way to initialize the foreground masks in each image that is based on a recent interest region detector. Other methods, e.g. [5,9], could equally well be used, but are more expensive than the method we propose.

The method proceeds in two stages. First, initial proposals are generated in two ways: the first approach follows [23], which starts from superpixels [11], and iteratively merges the two most similar regions (based on size and gradient orientation histogram) together until the whole image is a single region. All superpixels generated in the process are included as separate proposals. The second approach applies a saliency map [19] as a cue. For each superpixel, we compute the mean saliency value as well as the 70th, 80th and 90th percentiles of saliency values. These four values are then assigned to the pixels inside the superpixel, one at a time, and the resulting image is thresholded to get a binary image. All the connected components from the resulting four binary images are considered as separate proposals.

In the second stage, we compute a simple saliency based score for each proposal in the initial set. The definition is similar to the multi-scale saliency score in [1]:

$$score = \frac{\sum_{i \in P} \delta(s(i) > 0.7)}{area(P)} \cdot \sum_{i \in P} s(i), \tag{1}$$

where $\delta(a) = 1$ if a is true and 0 otherwise, P refers to pixels inside the proposal, $s(i)$ is a saliency value at pixel i, and $area(P)$ is the area of the proposal. The score (1)

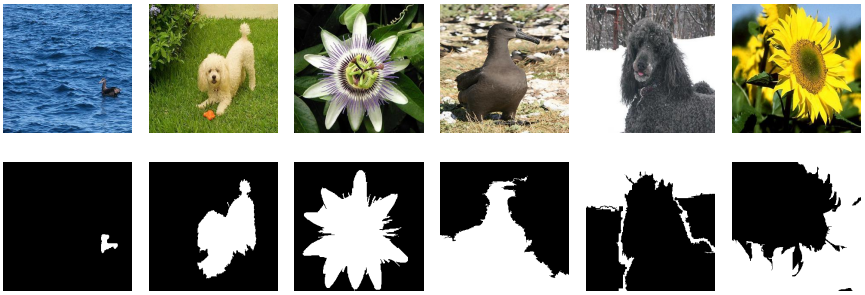

Fig. 2. Examples of initializations. The initializations themselves can already be very good, as seen in the first 3 columns. However, if the foreground object is flat or smooth, the background can become more interesting. TriCoS is able to remedy this problem.

favors large proposals with a small number of non salient pixels for which $s(i) < 0.7$. Finally, the best scoring proposal is selected for the foreground initialization. Examples of initializations can be found in Fig. 2.

3.2 Category-Level Co-Segmentation: BiCoS [6]

Using the foreground initialization from Sec. 3.1, we perform the discriminative learning part of BiCoS, which builds the core of the co-segmentation algorithm. Here, each category is treated independently, hence the name category level.

The algorithm operates on the same superpixels as mentioned in the last sub-section. A set of high dimensional descriptors is extracted for each of the superpixels. Superpixels are then classified into either foreground or background using a standard linear support-vector-machine (SVM) learned from themselves with labels according to the initializations or the output of a previous iteration. Given a set of $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ of descriptors belonging to all N superpixels of a single category, and let $\{y_1, y_2, \ldots, y_N\}$ to be hard labels (either 1 or -1). The separating hyperplane $\mathbf{w}$ in the descriptor space is defined using a standard $\ell 2$-SVM:

$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N} \ell(y_i \cdot \mathbf{w}^T\mathbf{x}_i) \longrightarrow \min_{\mathbf{w}}, \tag{2}$$

where $\ell(t) = \max(0, 1-t)$ is the hinge loss function, and C is the regularization constant set to 2 in all our experiments.

By applying the trained $\mathbf{w}$ back to each $\mathbf{x}_i$, we update each input superpixel labels y_i with:

$$y_i = \mathbf{w}^T\mathbf{x} \tag{3}$$

and reconstruct a soft saliency mask using these labels.

Using such a simple self-training scheme in the high-dimensional superpixel descriptor space, BiCoS enforces that superpixels with similar appearance across different images tend to be assigned similar values in the resulting saliency maps.

3.3 Image-Level Segmentation: GrabCut [20]

The second part of BiCoS, which is taken from the very popular GrabCut algorithm, handles each image independently. The initial Gaussian mixture model (GMM) for GrabCut is trained based on the saliency maps from Sec. 3.2, where pixels are weighted by their saliency. GrabCut ensures that within each image, pixels with similar appearance (color) tend to be assigned to the same class (either the foreground or the background). It alternates between: (1) (re)estimating foreground and background probability densities via Gaussian mixture model given the binary labels for each pixel of the entire image [3], and (2) the graphcut inference of labels in a random field with unary terms equal to the probabilities estimated from the GMM and binary terms according to local image gradients [4].

GrabCut strives for high local smoothness within the image while maintaining global color consistency. Unlike other levels that work with superpixels and high-dimensional descriptors, this procedure operates on pixels and uses simple RGB color as the feature describing each pixel. The result of this GrabCut step is a binary segmentation mask for each image.

3.4 Dataset-Level Class-Discriminative Segmentation

The idea behind class-discriminative segmentation is to figure out which parts of the images actually contribute toward differentiation of the categories and which parts do not. To do so, we first train a multi-class classifier for the different categories using features summed over the foreground given by the preceding level (GrabCut). We then apply the learned classifier back to individual superpixels in each image of the class, thus obtaining a score for the discriminating power of each superpixel.

Before we introduce the class-discriminative information into the process, we apply a pre-processing data balancing step: where the process discussed in Sec. 3.2 is applied to the whole dataset jointly, ignoring class labels, and with slightly richer features (see Sec. 3.5). We found that such process balances the sizes of the foreground segments across classes, which decreases the amount of bias towards classes during the subsequent procedure. After such balancing, we introduce the class-discriminative information into the co-segmentation using the following approach.

For each image, we then extract a very high-dimensional feature vector (e.g. Fisher vector) from the given foreground area only for the multi-category classification task. As in the standard multi-class SVM, the goal is to train K separating hyperplanes in the feature space. Given a set of $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ of descriptors for each of the N images and their category labels $\{y_1, y_2, \ldots, y_N\}$, we train a 1-vs-all SVM via quadratic optimization:

$$\frac{1}{2}||\mathbf{w}^k||^2 + C\sum_{i=1}^{N} \ell(y_i^k \cdot (\mathbf{w}^k)^T \mathbf{x}_i) \longrightarrow \min_{\mathbf{w}^k}, \forall k \in 1, 2, \ldots, K \tag{4}$$

where $\ell(t) = \max(0, 1-t)$ is the hinge loss function, and C is the regularization constant set to 10 in all our experiments. y_i^k is 1 if it belongs to the class k and -1 otherwise.

Once the classification model is trained, the question is how to use it to update the segmentation masks. To our best knowledge, there is no common solution to this type of propagation, and we propose the following solution.

Let us first consider the case of a dataset that has just two classes. As we use a linear SVM for classification, we have a single weight vector $\mathbf{w}$. Furthermore, we again extract superpixels as described in Sec. 3.2. For each superpixel, we extract the same feature vector $\mathbf{x}_j$ in the same way as in classification (note the change of the subscript as each feature vector $\mathbf{x}$ is composed in the same way as for classification, but is extracted from individual superpixels rather than images):

$$m_j = \text{sign}(y\mathbf{w}^T\mathbf{x}_j + \epsilon) \tag{5}$$

where ϵ is the bias term, y is the binary class label of the image the superpixel belongs to, and m_j is a binary foreground/background label for the superpixel $\mathbf{x}_j$.

In principle, superpixels with a positive score from $\mathbf{w}^T\mathbf{x}_j$ increase the margin of the right class and therefore should belong to foreground, while superpixels with a negative score should belong to background. However, as a result of overfitting in a high-dimensional space, the foreground area from where the classification model was trained will typically remain foreground after this process, while the background which was discarded during classification will be split into a foreground and a background part. To avoid such a uniform growth of a foreground superpixel, we introduce the bias ϵ which should have a small negative value. Intuitively, the model discards image superpixels that lie too close to the decision boundary as they can easily flip the side. As a result, the signal to noise ratio is increased. A high value of ϵ causes every superpixel to turn to foreground. And a low ϵ results in taking only a small but most discriminative part for classification, which may result in poor generalization.

Moving from the binary case to multi-class image classification, the propagation rule can be modified, in a way similar to the 1-vs-all classification rule. We thus compute the label of each superpixel as:

$$m_j = \text{sign}\left[(\mathbf{w}^k)^T\mathbf{x}_j - \max_{k'=\{1,2,\ldots,K\setminus k\}}((\mathbf{w}^{k'})^T\mathbf{x}_j) + \epsilon\right] \tag{6}$$

The superpixel label is now determined by the difference between the response of the true class and the maximum response of all other classes. In principle this may be sub-optimal as a superpixel with positive score surely contributes to classification, an image superpixel with negative score may not be necessarily confusing, as it can contribute to discriminating from other classes. However, the category level that follows after the dataset level is designed to prefer consistency across the category and typically compensates for this sub-optimality.

By mapping all m_j onto the superpixels, we obtain a new set of binary segmentation masks which can be fed back to the category level.

3.5 Implementation Details

All superpixels we use in the entire pipeline are extracted using the graph-based method [11], with $k = 200$, σ and minsize set proportional to the image size. Typically, around 40-100 superpixels are generated for each image.

The descriptors for the superpixels in Sec. 3.2, are stacked from five fairly standard sub-descriptors covering information over color, shape, size and location, as described in [6]. The color and SIFT histograms are of length 20 and 80, respectively. However for the global category-level step in Sec. 3.4, the histogram sizes are increased to 200 and 800, respectively.

For the multi-class SVM in Sec. 3.4, we sample SIFT features at multiple scales and Lab color features at dense grid locations within the category-independent segmentation masks for each image. Feature points outside the masks are discarded. Lab features are then encoded using LLC [24] with 1000 dictionary entries, while PCA is applied to dense SIFT, and the 80 most principal dimensions are encoded using the Fisher vectors [18] with a Gaussian mixture of 256 components. The total feature dimension is therefore 41960 after stacking the two descriptors.

4 Experimental Results

As we aim at class-discriminative co-segmentation, which may well differ from human-annotated foreground segmentations, we skip any segmentation accuracy measure and only measure the class-average classification accuracy. Note that our method only considers the segmentation of the training images. At test time, we apply a pipeline consisting of two steps: class-independent segmentation and image classification.

Class-Independent Segmenter. We propagate co-segmented masks of training images onto test images, which cannot be co-segmented as these are to be classified individually. Based on the co-segmented images, we train a class-independent segmenter in the same way as described at the beginning of Sec. 3.4. This class-independent segmenter can then be used to give binary segmentation masks on test image superpixels and therefore accurate foreground segmentations during testing.

Final Image Classifier. We train a 1-vs-all linear SVM classifier in the same way as described in Sec. 3.4. Unlike [6], the classifier is not trained on the output of the co-segmentation directly, but on the segmentations produced by the newly trained class-independent segmenter, with the rationale to make the data pre-processing for the final classifier training as similar to the data pre-processing at test-time as possible. During test time, we simply first apply the class-independent segmenter to each test image, and classify the image using the 1-vs-all linear SVM on the features from the foreground area.

The overall pipeline can be seen as an improved version of [6]. A direct comparison can be found in Sec. 4.1.

The BiCoS results in the second row in each comparison table use the new initialization described in Sec. 3.1. In our experiments with the particular datasets, the proposed initialization scheme did not give significant advantage over a simpler initialization with a central box (as in [6]). We, however, stick to the new initialization as it does not rely on the assumption that most objects are large and centered in images, and is, therefore, more general.

In the experiments we compare our system to previous methods from three different domains: (1) co-segmentation or, if available, ground truth segmentations/bounding

boxes, (2) methods that segment each image individually at training and at test time and (3) classification without any segmentation.

During all evaluations, TriCoS starts at the category level, moves to image level and iterates one whole loop before stopping at the image level.

4.1 Caltech-UCSD Birds 200-2010

Caltech-UCSD Birds 200-2010 [25] contains 200 classes with each class having exactly 15 training images and 15+ test images, resulting in 6033 images in total. The training/test split is fixed by the dataset. The dataset is very challenging, as it is designed for a classifier with human in the loop during test time. There are both coarse foreground segmentation and bounding box available for each image. Some of the previously published results make use of those hand annotations both during training and testing time [25,26].

We show our evaluation in Tab. 1 while qualitative examples can be found in Fig. 3 and Fig. 4. For the experiment with GrabCut, GrabCut is directly applied on both training and test images using the initialization introduced in Sec. 3.1. For the experiments using ground truth bounding box or foreground/background segmentations, we do not use the hand annotated data during test time. Instead, we use ground truth data from the training images to train the class-independent segmenter and apply it to the test images for the classification.

Most importantly, TriCoS outperform BiCoS, justifying our class-discriminative cosegmentation approach. We observe a huge decrease if all extracted features are used for classification ("no segmentation"), which once again shows the importance of object localization in classification in general. Moreover, we observe that only bounding box is not sufficient as using accurate ground truth foreground segmentation has a clear gain over using ground truth bounding box.

The comparison between the BiCoS in combination with our class-independent segmentation and the classification, and the previously published BiCoS results from [6] clearly shows our new pipelines being significantly superior than the one proposed in [6]. The majority of the gain is achieved using Fisher encoding [18] instead of LLC [24].

While TriCoS has a significant gain over hand annotated bounding boxes, it is still not as good as hand annotated pixel-level foreground segmentation. This is most likely caused by the insufficient amount of training data, as discriminative information can be discarded by TriCoS if it does not occur often enough in the training set.

The baseline accuracies provided by Welinder et al. [25] uses ground truth foreground segmentation both during training and testing, in combination with a multiple kernel learning approach for color and shape features, whereas Khan et al. [14] mainly focus on the way to better combine those features.

4.2 Stanford Dogs

The Stanford Dogs dataset originally consists of 120 dog species and 20580 images. There are 100 images per class for training and about 80 images for testing. Bounding boxes are also provided. In order to get a similar number of training images per class

Fig. 3. Bird segmentations: (top) original images, (middle) BiCoS outputs, (bottom) TriCoS outputs. The three columns on the left show that TriCoS finds the yellow head of the bird to be not discriminative and cuts it off during the co-segmentation. The right columns show that several images of one bird species contain red plastics, which misleads BiCoS to classify the red plastics as foreground. TriCoS finds the red plastics to be confusing for the classification and classifies them as background.

Table 1. Comparison on the Caltech-UCSD Birds 200-2010 dataset (15 training images per class). The last column shows whether ground truth (GT) information is used during either training or testing. seg: accurate foreground/background segmentation, bb: bounding box. [14] did not mention the usage of GT during test time, however directly compares their results to methods that make use of the GT.

Coseg	Class-indep. seg.	Final classifier	Acc.	GT used?
TriCoS (this work)	ours	ours	**25.5**	no
BiCoS [6]	ours	ours	23.7	no
BiCoS-MT	ours	ours	24.2	no
-	GrabCut	ours	19.2	no
-	no segmentation	ours	14.1	no
BiCoS	Chai [6]	Chai	15.7	no
BiCoS-MT	Chai	Chai	16.1	no
GT	ours	ours	**26.7**	seg/trn
GT	ours	ours	22.7	bb/trn
-	-	Welinder et al. [25]	19.0	seg/trn+tst
-	-	Yao et al. CVPR'11 [26]	19.2	bb/trn+tst
-	-	Khan et al. NIPS'11 [14]	22.4	unsure

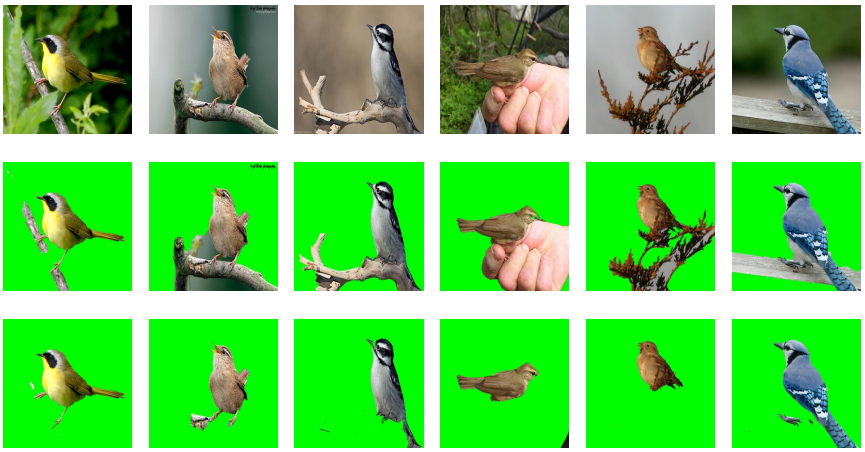

Fig. 4. More Bird segmentations: (top) original images, (middle) BiCoS outputs, (bottom) TriCoS outputs. The most noticeable difference between BiCoS and TriCoS is the removal of tree branches of leaves. These often appear in many images in one category, so that BiCoS segments them as foreground. But they appear across categories as well and are therefore not class-discriminative and are discarded in TriCoS.

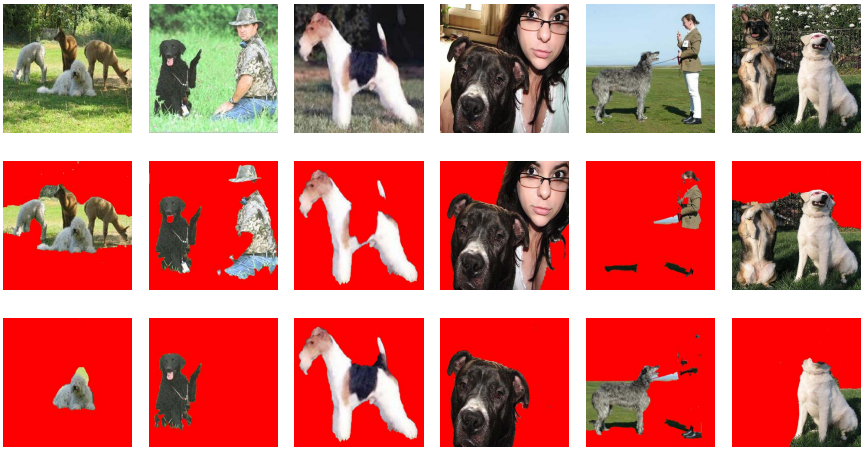

Fig. 5. Dogs segmentations: (top) original images, (middle) BiCoS outputs, (bottom) TriCoS outputs. If an image patch only appears once in the co-segmentation set, BiCoS is unable to flip its binary label from its initialization. However, using discriminative information across the entire dataset, TriCoS is sometimes able to correct this.

Table 2. Comparison on the Stanford Dogs dataset. Note that TriCoS performs better than BiCoS and hand-annotated bounding boxes.

Coseg	Class-indep. seg.	Final classifier	Acc.	GT used?
TriCoS	ours	ours	**26.9**	no
BiCoS	ours	ours	25.7	no
-	GrabCut	ours	17.3	no
-	no segmentation	ours	21.0	no
-	-	Khosla et al. [15]	14.5	no
GT	ours	ours	26.0	bb/trn

Table 3. Comparison on the Oxford Flowers 102 dataset. TriCoS, BiCoS and Nilsback's supervised segmentation method give more or less the same accuracy.

Coseg	Class-indep. seg.	Final classifier	Acc.	GT used?
TriCoS	ours	ours	85.2	no
BiCoS	ours	ours	85.5	no
-	GrabCut	ours	81.7	no
-	no segmentation	ours	81.5	no
BiCoS	Chai	Chai	79.1	no
BiCoS-MT	Chai	Chai	80.0	no
-	-	Kanan and Cottrell et al. [13]	71.4	no
-	-	Khan et al. [14]	73.3	no
GT	Nilsback [17]	ours	85.6	seg/trn
GT	Nilsback	Chai	76.8	seg/trn
GT	Nilsback	Nilsback	76.3	seg/trn

as the bird and flower datasets, we randomly selected 30 images per class for training, while keeping the test set as it is.

The dog dataset was introduced very recently, along with a baseline classification accuracy using SIFT and LLC. Tab. 2 leads to similar conclusions as in Tab. 1. GrabCut alone seems to make a lot of mistakes, and therefore achieves lower accuracy than not using any segmentation. However, both TriCoS and BiCoS remedy these errors. Examples can be found in Fig. 5.

4.3 Oxford Flowers 102

The Oxford Flower dataset is the earliest published dataset on fine-grained classification. It contains 102 flower species and 8189 images in total, with 20 images per class fixed for training and the rest for test.

The evaluation is shown in Tab. 3. Nilsback et al. [17] use hand annotated pixel-level segmentations to train a geometry-based segmentation model that is optimized for flowers and a multiple kernel learning (MKL) in conjunction with a non-linear SVM for classification.

Unfortunately, TriCoS performs worse than BiCoS and Nilsback's supervised segmentation method, but only by a small margin. We argue that the Fisher vector encoding of the densely sampled and SIFT features already saturates the performance so that no more gain can be achieved with better segmentation. Our argument is supported by the fact that Nilsback's segmentations, which were much worse than BiCoS in [6], now performs equally as well as BiCoS.

5 Discussion

We have demonstrated the importance of class-discriminative co-segmentation by comparing TriCoS to its predecessor BiCoS on the challenging Caltech-UCSD Birds, Stanford Dogs and Oxford Flowers 102 datasets, where TriCoS outperforms BiCoS in two of the three cases and is only worse by a very small margin on the flowers.

Moreover, we use an interest region detection-based method to initialize the system. In this way, it does not require any guess of the location of the foreground object as is done in [6].

TriCoS outperforms hand-annotated bounding boxes on both birds and dogs, despite only requiring image class labels and no additional annotation. Our experimental results agree with previous findings that accurate foreground segmentation is of great importance in fine-grained classification systems.

Last but not least, using our state-of-the-art classification system in conjunction with TriCoS, we achieve classification accuracy improvement over previously published results by a margin of 3.1%, 12.4% and 5.2% on the birds, dogs and flowers, respectively.

Acknowledgements. This work is supported by EC STREP project IURO and ERC grant VisRec no. 228180. Esa Rahtu is also supported by the Academy of Finland (Grant no 128975).

References

1. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: CVPR (2010)
2. Arandjelović, R., Zisserman, A.: Smooth object retrieval using a bag of boundaries. In: ICCV (2011)
3. Blake, A., Rother, C., Brown, M., Pérez, P., Torr, P.: Interactive Image Segmentation Using an Adaptive GMMRF Model. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004, Part I. LNCS, vol. 3021, pp. 428–441. Springer, Heidelberg (2004)
4. Boykov, Y., Jolly, M.-P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: ICCV (2001)
5. Carreira, J., Sminchisescu, C.: Constrained parametric min-cuts for automatic object segmentation. In: CVPR (2010)
6. Chai, Y., Lempitsky, V., Zisserman, A.: Bicos: A bi-level co-segmentation method for image classification. In: ICCV (2011)
7. Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods. In: British Machine Vision Conference (2011)
8. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)

9. Endres, I., Hoiem, D.: Category Independent Object Proposals. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 575–588. Springer, Heidelberg (2010)
10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2011 (VOC 2011) Results (2011), http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html
11. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV, 59(2) (2004)
12. Galleguillos, C., Babenko, B., Rabinovich, A., Belongie, S.: Weakly Supervised Object Localization with Stable Segmentations. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 193–207. Springer, Heidelberg (2008)
13. Kanan, C., Cottrell, G.W.: Robust classification of objects, faces, and flowers using natural image statistics. In: CVPR (2010)
14. Khan, F.S., van de Weijer, J., Badganov, A.D., Vanrell, M.: Portmanteau vocabularies for multi-cue image representation. In: NIPS (2011)
15. Khosla, A., Jayadevaprakash, N., Yao, B., Fei-Fei, L.: Novel dataset for fine-grained image categorization. In: First Workshop on Fine-Grained Visual Categorization, CVPR (2011)
16. Nilsback, M.-E., Zisserman, A.: Delving into the whorl of flower segmentation. In: BMVC (2007)
17. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: ICVGIP (2008)
18. Perronnin, F., Sánchez, J., Mensink, T.: Improving the Fisher Kernel for Large-Scale Image Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
19. Rahtu, E., Kannala, J., Salo, M., Heikkilä, J.: Segmenting Salient Objects from Images and Videos. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 366–379. Springer, Heidelberg (2010)
20. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph, 23(3) (2004)
21. Rother, C., Minka, T.P., Blake, A., Kolmogorov, V.: Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In: CVPR (2006)
22. Uijlings, J.R.R., Smeulders, A.W.M., Scha, R.J.H.: What is the spatial extent of an object? In: CVPR, pp. 770–777 (2009)
23. van de Sande, K., Uijlings, J., Gevers, T., Smeulders, A.: Segmentation as selective search for object recognition. In: ICCV (2011)
24. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T.S., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR (2010)
25. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology (2010)
26. Yao, B., Khosla, A., Li, F.-F.: Combining randomization and discrimination for fine-grained image categorization. In: CVPR (2011)
27. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. International Journal of Computer Vision 73(2), 213–238 (2007)