SEMANTIC SEGMENTATION USING GRABCUT*

Christoph Göring, Björn Fröhlich and Joachim Denzler
Department of Mathematics and Computer Science, Friedrich Schiller University, Jena, Germany

Keywords: Semantic Segmentation, GrabCut, Shape, Texture.

Abstract: This work analyzes how to utilize the power of the popular GrabCut algorithm for the task of pixel-wise labeling of images, which is also known as semantic segmentation and an important step for scene understanding in various application domains. In contrast to the original GrabCut, the aim of the presented methods is to segment objects in images in a completely automatic manner and label them as one of the previously learned object categories. In this paper, we introduce and analyze two different approaches that extend GrabCut to make use of training images. *C-GrabCut* generates multiple class-specific segmentations and classifies them by using shape and color information. *L-GrabCut* uses as a first step an object localization algorithm, which returns a classified bounding box as a hypothesis of an object in the image. Afterwards, this hypothesis is used as an initialization for the GrabCut algorithm. In our experiments, we show that both methods lead to similar results and demonstrate their benefits compared to semantic segmentation methods only based on local features.

1 INTRODUCTION

Finding objects in images is a challenging task in computer vision. A much more complex challenge is to locate objects in a pixel-wise manner without any human interaction. Previous works usually use local features, which are classified. The results are often smoothed by utilizing an unsupervised segmentation method. A huge problem of these methods is that they operate on highly over-segmented images. Objects composed of different parts (*e.g.* black and white spots of a cow) are not seen as one object, but they are seen independently. It is difficult to incorporate shape information in such methods and they lead to slivered segments.

A famous approach for a globally optimized segmentation is the GrabCut algorithm introduced in (Rother et al., 2004). In their work, a human has to place a rectangle around an object which is segmented afterwards using an iterative algorithm. This semi-automatic segmentation method can handle objects which are composed of different homogeneous areas.

In the present paper, we propose two methods which integrate this powerful segmentation technique into a semantic segmentation framework. The first method starts with learning models for each class from a training set. We use these models as an initialization for the GrabCut framework, so that we have one segmentation per class. The segmentation with the minimum distance to the training data and the corresponding class is the final result. Because different segmentations computed by GrabCut are classified, we call it *Classification-GrabCut* (*C-GrabCut*). In the second approach an object localization algorithm determines the object class and a bounding box which encloses the object. The GrabCut algorithm is initialized with this bounding box to refine this rough segmentation. Because the object is localized before GrabCut is applied, we call it *Localized-GrabCut* (*L-GrabCut*). A flowchart of both approaches can be seen in Figure 1.

(Jahangiri and Heesch, 2009) present an unsupervised GrabCut algorithm that is initialized with a coarse segmentation obtained by active contours. However, they are only able to segment the foreground objects from a plain background and do not use any class specific information. ClassCut (Alexe et al., 2010) operates on a set of images which all contain a foreground object of the same class. The goal is to simultaneously segment this set of images and learn a class model. The model and the segmentations are computed iteratively until convergence. ClassCut bears some resemblance to *C-GrabCut* which is introduced here. In contrast the algorithm presented in this paper, ClassCut assumes the object class is already known.

The outline of this paper is organized as follows. First we introduce our two methods in Section 2. Our experiments in Section 3 show that both methods lead to comparable and satisfying results. A summary of

*Supported by the TMBWK ProExzellenz initiative.

Göring C., Fröhlich B. and Denzler J..
SEMANTIC SEGMENTATION USING GRABCUT.
DOI: 10.5220/0003829905970602
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2012), pages 597-602
ISBN: 978-989-8565-03-7

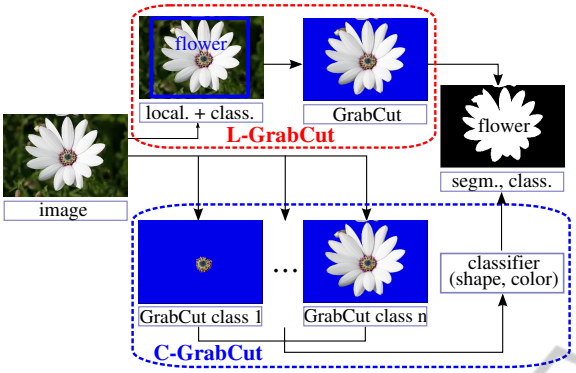


Figure 1: This flowchart shows both approaches. *C-GrabCut*: First an image is segmented using different models to get an initial segmentation. In the second step a classifier determines which possible segmentation is more likely to be the correct one. *L-GrabCut*: The class of the object and the bounding box is determined. Thereafter, GrabCut is started using the bounding box as initialization. The result of both methods is the class label and the segmentation of the foreground object.

our findings and a discussion of future work conclude this paper.

2 METHODS

We consider two ways to achieve a semantic segmentation of a given image. First we do a segmentation of the image and try to classify the foreground object or second we try to locate a specific object in an image and after this we segment it pixel-wise. For both methods, we need an already labeled training set, which is used to train the parameters of our models. The annotation can be a bounding box around the main object or a pixel-wise labeling of the objects in the image.

For our first idea, we learn for each class a background and a foreground model which we use as an initialization to segment a new input image. Therefore, we have a segmentation of an image for each class. In the following step, we want to find out which of these segmentations is the most probable one by using shape and color information for classification. We call this idea *C-GrabCut* because we first utilize the GrabCut method from (Rother et al., 2004), which we introduce in Section 2.1 followed by the mentioned classification step. A huge disadvantage of this method is obvious: the complexity of *C-GrabCut* is linear in the number of classes taken into account. For this reason, we found another method, which we call *L-GrabCut*: In a first step, we can use any object localization method which gives us a bounding box of a potential object and a corresponding class label. This bounding box is used as an initialization for the semi-automatic GrabCut segmentation.

In this section, we first give a brief introduction to the GrabCut segmentation algorithm as a basic method for both of our approaches. In the following sections, we describe *C-GrabCut* and *L-GrabCut* as a way to utilize GrabCut in a semantic segmentation framework.

2.1 GrabCut

GrabCut (Rother et al., 2004) is a state of the art unsupervised semi-automatic segmentation. A user is drawing a rectangle around the main object which is used as an initial rough segmentation. In an iterative algorithm the segmentation is improved step by step. The framework introduced in the original paper only considers color information, but texture information is also very important for some classes. (Han et al., 2009) introduced a method integrating texture information into the GrabCut framework by utilizing nonlinear multiscale structure tensors. Instead of the nonlinear diffusion by a simple Gaussian smoothing we use a multiscale structure tensor as also described in (Han et al., 2009).

2.2 *C-GrabCut*: Classification of Segmented Images

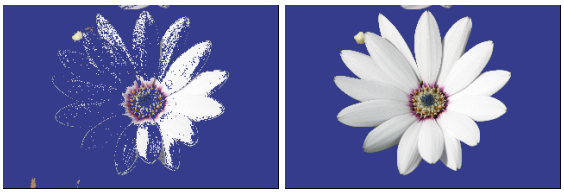
The first approach presented in this paper tries to replace the manual segmentation of the original GrabCut by learning GMMs with class specific information. In this section, we present our first idea of modifying GrabCut for training and testing on different sets of images. Thereafter, we show how to classify the located segments using shape and color similarity measures.

2.2.1 Segmenting with Prior Knowledge GMMs

Instead of using only one image as in the original GrabCut, a training set for each class is used to create a background and foreground model. Let $\boldsymbol{Z}_c = \{\boldsymbol{z}^{1,c}, \cdots, \boldsymbol{z}^{N_c,c}\}$ be the set of training images of class c and $\boldsymbol{A} = \{\boldsymbol{\alpha}^{1,c}, \cdots, \boldsymbol{\alpha}^{N_c,c}\}$ be the corresponding ground-truth data. To train the GMMs for foreground and background separately, the data is divided into a set of foreground pixels $\boldsymbol{D}_{c,\text{fgd}}$ and a set of all background pixels $\boldsymbol{D}_{c,\text{bgd}}$ according to the ground-truth data:

$$\boldsymbol{D}_{c,\kappa} = \{z_j^{i,c} | \alpha_j^{i,c} = \kappa\} \quad , \forall \kappa \in \{\text{fgd}, \text{bgd}\} \quad . \qquad (1)$$

For these two sets the corresponding GMMs are computed. The result is $\boldsymbol{\theta}$, containing both the parameters of the foreground and the background GMM of the



(a) Initial segmentation. (b) Result after GrabCut.

Figure 2: (a) initial segmentation using the learned GMMs for class "flower"; (b) result after applying GrabCut.

training images. We determine the number of components by optimization on the validation dataset.

Let $\boldsymbol{z} = \{z_1, \cdots, z_N\}$ be an image that is to be segmented. The initial segmentation $\boldsymbol{\alpha}^* = (\alpha_1^*, \cdots, \alpha_N^*)$ is computed using maximum likelihood estimation for each pixel:

$$\alpha_i^* = \underset{\alpha_i \in \{\text{fgd},\text{bgd}\}}{\operatorname{argmax}} \; p(z_i|\alpha_i, k_i, \boldsymbol{\theta}), \forall i \in \{1, \cdots, N\}. \quad (2)$$

This initial segmentation $\boldsymbol{\alpha}^*$ is used as an initialization to the GrabCut algorithm. An example of the result of such an initial segmentation after applying the GrabCut algorithm can be seen in Figure 2.

2.2.2 Classification

We explained in the previous section how to obtain a segmentation without user interaction if the class of the foreground object is known. Now, we address the problem of determining the class of the object.

Let $\boldsymbol{C}$ be the set of foreground classes and $\boldsymbol{Z}_c$ the set of training pictures of class c. By applying the algorithm described in the previous section on a new test image, we can obtain a segmentation $\boldsymbol{\alpha}_c$ with $c \in \boldsymbol{C}$ for each of the classes. The results of the classification of an example image is shown in 3.

We consider several different measures which are evaluated in the experimental section of this paper.

Color Information. The first type of distances is based on similarity of color. First, we consider measuring the distance to the foreground color GMM of the whole class:

$$\text{dist}_m(\boldsymbol{\alpha}_c, \boldsymbol{Z}_c) = \text{KL}(GMM_{\boldsymbol{\alpha}_c}, \boldsymbol{\theta}_{c,\text{fgd}}) \;, \quad (3)$$

where $GMM_{\boldsymbol{\alpha}_c}$ is the GMM computed with the foreground pixels of the test image and $\boldsymbol{\theta}_{c,\text{fgd}}$ is the foreground GMM of the model for class c. The function KL returns the symmetric Kullback-Leibler divergence between two GMMs. For this, we use a matching based approximation algorithm from (Goldberger et al., 2003).

Second, we use the distance to the nearest neighbor of the training dataset:

$$\text{dist}_f(\boldsymbol{\alpha}_c, \boldsymbol{Z}_c) = \min_{i=1,\cdots,N_c} \text{KL}(GMM_{\boldsymbol{\alpha}_c}, GMM_{\boldsymbol{\alpha}^{i,c}}). \quad (4)$$

Shape Information. A different kind of distances relates to the shape of the segmentation. As a simple measure of shape the popular Hu set of seven invariant image moments is used. The distance between two sets of Hu moments $\boldsymbol{H}$ and $\boldsymbol{H'}$ is computed in the following way (Gonzalez and Woods, 2008, p. 841):

$$M(\boldsymbol{H}, \boldsymbol{H'}) = \sum_{i=1,\ldots,7} \left| \frac{\text{sign}(H_i)}{\log|H_i|} - \frac{\text{sign}(H_i')}{\log|H_i'|} \right| \;. \quad (5)$$

To compute a distance between a segmentation $\boldsymbol{\alpha}_c$ and a class c, we use the following function:

$$\text{dist}_h(\boldsymbol{\alpha}_c, \boldsymbol{Z}_c) = \min_{i=1,\cdots,N_c} M(\text{h}(\boldsymbol{\alpha}_c), \text{h}(\boldsymbol{\alpha}^{i,c})) \;, \quad (6)$$

where the function h computes a set of Hu moments for a given segmentation $\boldsymbol{\alpha}$.

We also use the shape context algorithm described in (Belongie et al., 2002) to compute a distance between a segmentation $\boldsymbol{\alpha}_c$ and a class c:

$$\text{dist}_b(\boldsymbol{\alpha}_c, \boldsymbol{Z}_c) = \min_{i=1,\cdots,N_c} \text{dist}_{sc}(\boldsymbol{\alpha}_c, \boldsymbol{\alpha}^{i,c}) \;, \quad (7)$$

where dist_{sc} is a function that computes the distances between two shapes using the shape context algorithm.

To integrate the different distances, we compute a weighted sum of all distances:

$$\text{dist}(\boldsymbol{\alpha}_c, \boldsymbol{Z}_c) = \sum_{j \in \{f,h,m,b\}} w_j \, \text{dist}_j(\boldsymbol{\alpha}_c, \boldsymbol{Z}_c) \;. \quad (8)$$

The weights w_j are computed on a validation dataset.

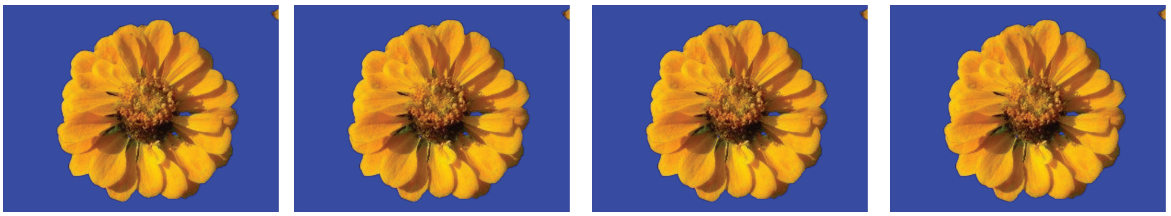
The final labeling $\hat{c}$ and the corresponding segmentation $\boldsymbol{\alpha}_{\hat{c}}$ is given by the lowest distance measure:

$$\hat{c} = \underset{c}{\operatorname{argmin}} \; dist(\boldsymbol{\alpha}_c, \boldsymbol{Z}_c) \quad (9)$$

The flowchart of the presented algorithm is illustrated in Figure 1.

2.3 *L-GrabCut*: Segmentation of Classified Rectangles

One obvious drawback of *C-GrabCut* is the running time which is linear in the number of classes. For this reason, we will now present an approach that classifies the object in an image before it is segmented using GrabCut. We use an object localization algorithm that returns a bounding box and a class to obtain the initial segmentation. The bounding box segmentation is then optimized using GrabCut. A flowchart of *L-GrabCut* can be seen in Figure 1.



(a) Model flower. dist_f=4.135, dist_h=0.003 dist_b=0.813, dist_m=38.83. (b) Model car. dist_f=46.88 dist_h=0.192 dist_b=0.457, dist_m=37.48. (c) Model airplane. dist_f=74.85 dist_h=1.180 dist_b=0.843, dist_m=65.38. (d) Model cow. dist_f=36.83 dist_h=0.246 dist_b=0.437, dist_m=39.63.

Figure 3: The resulting segmentation using different models to get the initial segmentation. Furthermore, the distance of the foreground color GMM dist_f, the distance of the hu moments dist_h, the shape context distance dist_b and the distance of the class GMM dist_m is shown.

In literature there are different approaches to object detection. Some utilize local features, like (Marszalek and Schmid, 2007) who try to combine local features with shape masks. Another class of object detectors uses a sliding window and evaluates each window with a binary classifier. A popular object detector that uses the sliding window approach is the histogram of gradients (hog) detector (Dalal and Triggs, 2005) which was successfully used for human detection.

For our experiments, we chose the algorithm from (Felzenszwalb et al., 2010). It uses an extension of the hog features. A set of parts is added which can change their position to adapt to small changes in pose. It delivers state of the art results on the challenging Pascal dataset and was awarded a "lifetime achievement" prize from the organizers of the Pascal Visual Object Class Challenge (Everingham et al., 2010).

Due to the reason that in some cases the bounding box does not enclose the whole object, we considered a modification of the GrabCut algorithm that also allows foreground pixels outside of the initial bounding box. But in our experiments we have shown that some segmentations improved, but the overall recognition rate stayed the same.

3 EXPERIMENTAL RESULTS

In this section, we concentrate on the evaluation and precise analysis of our introduced methods, *C-GrabCut* and *L-GrabCut*. Finally, we give a discussion of our results.

For our evaluation, we are using our own dataset composed of images obtained from various image sources: MSRC (Winn et al., 2004), LabelMe (Russell et al., 2008) and image search engines[2]. The final number of 90 images per category is divided into 30 images for training, validation and testing each. Some examples of the used dataset can be seen in Figure 6.

To compare results, we use the following metric:

$$r_c = (r_{ob} + r_{bg})/2 \tag{10}$$

where r_{ob} and r_{bg} are the ratios of correctly classified object and background pixels. Furthermore, we use the average recognition rate of all classes:

$$r_{av} = (\Sigma_{c \in \boldsymbol{C}} \; r_c)/|\boldsymbol{C}| \quad . \tag{11}$$

3.1 *C-GrabCut*

In this section, we want to analyze the results of *C-GrabCut* introduced in Section 2.3. For each of the four classes a model is learned over all training images. These models are used as initialization of GrabCut for each test image. The final segmentation and label is selected out of these segmentations by a classification step as introduced in Section 2.2.2.

To evaluate the classification step, we computed the different distance measures using the ground truth segmentation and computed the percentage of correctly classified foreground objects. The best result using only a single distance was achieved using dist_f with 84%. dist_m achieved 56%, dist_b 24% and dist_h 68%. This experiment showed that a weighted combination of dist_f and dist_h gives the best classification result with 85%. Incorporating the other measures did not improve the result. The weights are learned on the validation dataset. A weighted combination of all proposed distances does not improve the results.

A modified Version of *C-GrabCut* where the classification step is bypassed and the ground truth classification is used instead was evaluated. The recognition rate was $r_{av} = 0.84$ using only color and $r_{av} = 0.73$ using only texture information. By combining texture and color a recognition rate of $r_{av} = 0.88$ was reached.

In Figure 4, the results of both of our approaches can be seen. The performance varies between classes.

[2]Dataset available: www.inf-cv.uni-jena.de/ssg.

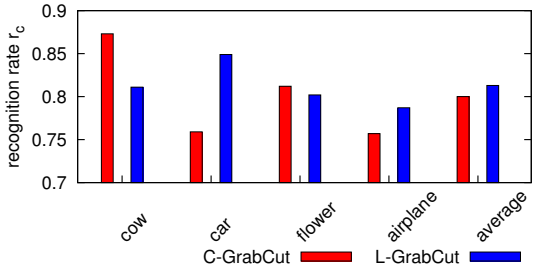


Figure 4: Recognition rates for all classes for *C-GrabCut* and *L-GrabCut*.

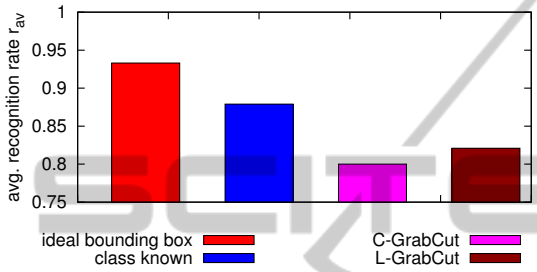


Figure 5: Average recognition rates of both of our methods in comparison to a modified version of *L-GrabCut* using an ideal bounding box and a modified version of *C-GrabCut* that skips the classification step and uses the ground truth classification instead.

The results are particularly good for the class cow. On average the achieved recognition rate was $r_{av} = 0.80$. The gap between *C-GrabCut* and the presented idealized method is about 8 percentage points.

3.2 *L-GrabCut*

In Section 2.3, we introduced *L-GrabCut* as another method for semantic segmentation of an image. First we locate the object in a test image by utilizing the localization method from (Felzenszwalb et al., 2010) which returns a bounding box and a class name. In the second step this bounding box is used as initialization of the GrabCut algorithm.

The results for *L-GrabCut* are shown in Figure 4. The performance is particularly good for the class car, which has a recognition rate of $r_c = 0.85$. On average the achieved recognition rate was $r_{av} = 0.81$.

In Figure 5, we also show results of a modified version of *L-GrabCut* where we use the ground truth bounding box as initialization. This means that with a better localization method than (Felzenszwalb et al., 2010) this algorithm could achieve results up to 10 percentage points better. We also showed that in the ideal case the addition of texture information only improved the result by 0.4%.

3.3 Discussion of Results

In this section, we have demonstrated that each component of our methods lead to satisfying segmentation and classification results. Furthermore, we have shown that both of our introduced methods yield to results close to their practical upper bounds. The outcomes of both of our methods are comparable. However, the findings of *L-GrabCut*, where we segment the previous classified bounding boxes, are slightly better than *C-GrabCut*. The preconditions for *L-GrabCut* are better compared to *C-GrabCut*. This can be seen in Figure 5, where the outcomes of the ideal bounding box for *L-GrabCut* are better than the outcomes of the perfect classification for *C-GrabCut*. Some segmentation results for both methods are presented in Figure 6.

Furthermore, we have shown that our methods do not benefit from shape context proposed in (Belongie et al., 2002), but Hu moments and nearest neighbor distance of the Gaussian mixture models lead to an improved performance. We could also show that texture information is not as important as color information, but for some classes it might be beneficial. The usage of texture information improves the average results slightly but the main disadvantage of texture features is the increased running time.

4 CONCLUSIONS

In this paper, we described two methods to use the semi supervised segmentation algorithm GrabCut in an unsupervised manner semantic segmentation.

Both methods have their advantages and disadvantages. The segmentations are less slivered than the results of a previously introduced semantic segmentation approach. But *L-GrabCut* depends very much on the performance of the localization algorithm. If the bounding box is to small, parts of objects outside of the bounding box will be ignored. This is also a problem if there are multiple instances of an object in an image, where only one is located by the object location method (*cf.* Figure 6 first line). For *C-GrabCut* the main disadvantage is that we need a segmentation for each class. As a result, the complexity is strongly controlled by the number of classes.

The presented methods only work for images with a single foreground object. It could be useful to integrate these methods into a larger semantic segmentation framework as a refining step to improve results on image parts containing only one object.

It might also be very interesting to find ways to extend our ideas to a multiclass solution. With these

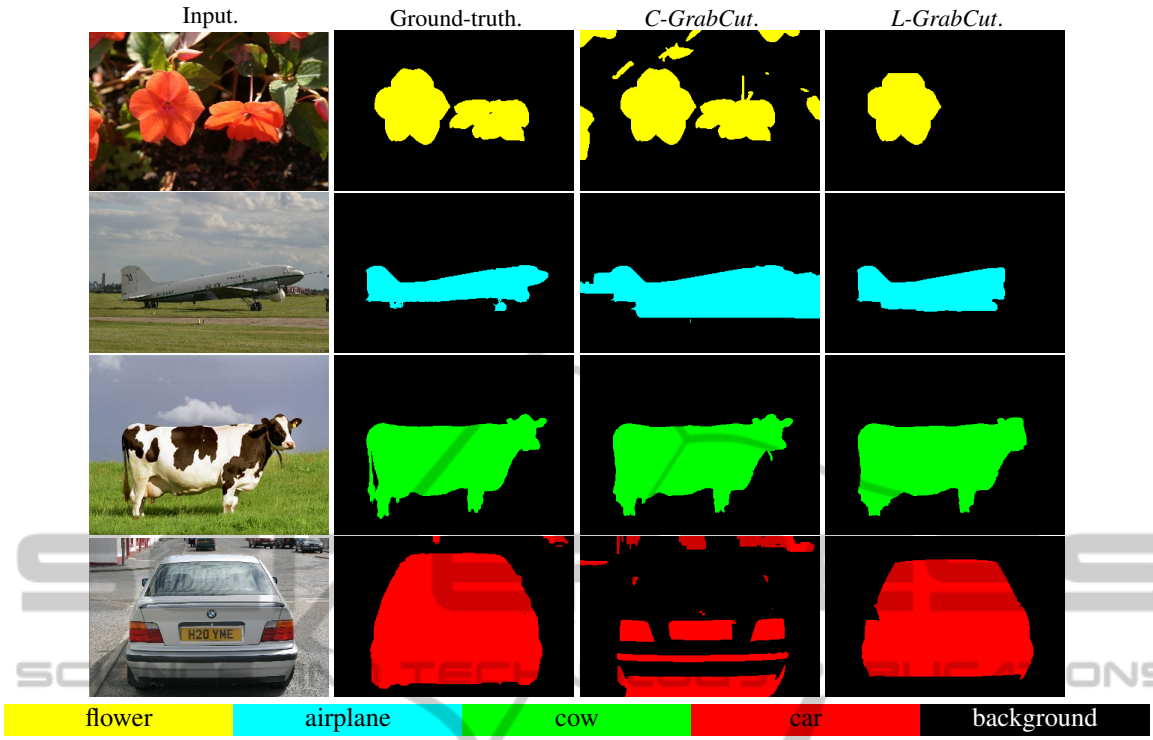


Figure 6: Example input images, ground-truth data and results of both introduced methods.

methods it is possible to evaluate the algorithms on more common datasets like PASCAL VOC (Everingham et al., 2010) or the MSRC21 dataset (Winn et al., 2004). Possible approaches are α-expansion or multi-way cuts (Boykov et al., 2001).

The way to incorporate shape information described in this paper is not very flexible and only rates the final segmentation. Hence, it might be interesting to analyze ways to directly integrate a type of shape energy into the graph cut algorithm similar to an EM method to improve the segmentation result.

REFERENCES

Alexe, B., Deselaers, T., and Ferrari, V. (2010). Classcut for unsupervised class segmentation. In *ECCV*, pages 380–393.

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *PAMI*, 23:1222–1239.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *PAMI*, 32:1627–1645.

Goldberger, J., Gordon, S., and Gordon, S. (2003). An efficient image similarity measure based on approximations of KL-divergence between two gaussian mixtures. In *ICCV*, pages 487–493.

Gonzalez, R. C. and Woods, R. E. (2008). *Digital image processing*. Prentice Hall, Upper Saddle River, N.J.

Han, S., Tao, W., Wang, D., Tai, X.-C., and Wu, X. (2009). Image segmentation based on GrabCut framework integrating multiscale nonlinear structure tensor. *IEEE Trans. on Image Processing*, 18(10):2289–2302.

Jahangiri, M. and Heesch, D. (2009). Modified grabcut for unsupervised object segmentation. In *ICIP*, pages 2389–2392.

Marszalek, M. and Schmid, C. (2007). Accurate object localization with shape masks. In *CVPR*.

Rother, C., Kolmogorov, V., and Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graphics (TOG)*, 23(3):309–314.

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). Labelme: A database and web-based tool for image annotation. *IJCV*, 77:157–173.

Winn, J., Criminsi, A., and Minka, T. (2004). Microsoft research cambridge object recognition image database.