Semantic Segmentation with Millions of Features: Integrating Multiple Cues in a Combined Random Forest Approach

Björn Fröhlich and Erik Rodner and Joachim Denzler

Computer Vision Group, Friedrich Schiller University Jena, Germany
http://www.inf-cv.uni-jena.de

Abstract. In this paper, we present a new combined approach for feature extraction, classification, and context modeling in an iterative framework based on random decision trees and a huge amount of features. A major focus of this paper is to integrate different kinds of feature types like color, geometric context, and auto context features in a joint, flexible and fast manner. Furthermore, we perform an in-depth analysis of multiple feature extraction methods and different feature types. Extensive experiments are performed on challenging facade recognition datasets, where we show that our approach significantly outperforms previous approaches with a performance gain of more than 15% on the most difficult dataset.

1 Introduction

Recognition of semantic categories in images is an important field in computer vision and especially labeling each pixel of an image is a challenging structural task. Solving this task requires to take several different cues into account, such as color, shape, and texture. Furthermore, contextual information, like probable constellations and positions of categories in an image, is essential to achieve consistent and accurate results [16].

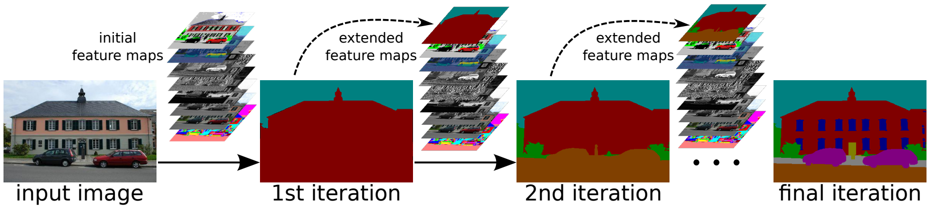


Fig. 1: Basic idea of our approach: features are added and updated incrementally to the set of available features to refine the semantic segmentation result.

sponsored by the Graduate School on Image Processing and Image Interpretation, TMBWK ProExzellenz program.

In this paper, we present a powerful semantic segmentation framework, which handles different information cues and features in a combined manner. In contrast to previous work [16, 22, 23], where contextual constraints are integrated after a local classification step, our approach allows for learning them directly and jointly together with other feature types. This is done by estimating contextual cues in an iterative manner based on the output of the classifier built in a previous step.

Features, whether or not contextual, are calculated with feature extraction methods performed on so called feature maps, which contain a value for each pixel of the original image. The number of possible combinations of all parameters leads to millions of possible features and we show how to handle them with random decision forests in an efficient manner. The approach can be easily extended and adapted to other application areas, simply by extending the set of feature maps. Another advantage is that due to the iterative and combined nature of our approach, the trade-off between accuracy and computation time for learning and testing can be controlled. With slightly modifications and a loss of accuracy the introduced method has anytime capabilities which has be shown in [8].

Related Work on Semantic Segmentation We incorporate context knowledge by using the output of previous levels of a decision tree classifier as features for a new one. This strategy is similar to the one used by Fink and Perona [6] for their mutual boosting approach, where they train a set of object detectors simultaneously. In each round of the Boosting method, features are added derived from the results of the current classifier.

Our work is also related to the approach of Shotton *et al.* [16], where a two stage segmentation technique is proposed. Their idea is to first train a random forest using basic local features and then to train a second random forest using context features calculated using the first forest. In contrast, we learn a single random forest and incrementally add context features derived from coarser levels. This allows for handling the problem in a combined manner, where dependencies between contextual features and non-contextual features are exploited directly. Considering image and context features jointly is beneficial, because it reflects more the inherent dependency between both types. For example, blue might be a typical color for a car, but only when we know that there is no building underneath, which would give a good hint for a sky region. Those situations can not be modeled by considering contextual after color features. Typically, context information is modeled by time consuming random field approaches [11, 13, 22, 23]. For a good overview of other semantic segmentation approaches, we refer the reader to Arbelaez *et al.* [1].

Related Work on Facade Recognition The application considered in this paper is semantic segmentation for facade recognition based on standard color images. The task is to estimate the position and size of various structural (*e.g.* "window", "door") and non-structural elements (*e.g.* "sky", "road", "building") in a given image of a building or street scene. This recognition task has gained interest in recent years [9], which is mainly due to the growing need to store the appearance of buildings in large 3D city models [9]. For example, an efficient representation of already labeled images with a grammar based compression scheme [15] allows for reducing each facade image to a few parameters. Furthermore, by incorporating a large amount of prior knowledge, the recognition of facade elements also allows for estimating the rough 3D structure of buildings [9].

The work of Fröhlich *et al.* [7] propose an approach, which classifies local color features with a random decision forest and further refines the result by fusing with an unsupervised segmentation. In contrast to our approach, they do not incorporate contextual information and the feature set is strictly limited. Yang and Förstner [23] use a conditional Markov random field (CRF), in which the unary potentials are computed by applying a random forest classifier. A subsequent work of the same authors [22] improves this method by considering a hierarchical CRF that exploits region segmentations on multiple image scales. Our approach takes high-order dependencies of multiple pixels into account and integrates classification and contextual inference in a combined approach. Furthermore, the approach does not incorporate any prior model-based knowledge about facades as utilized in Teboul *et al.* [18].

Outline The remainder of this paper is structured as follows. In Section 2, we introduce a new flexible framework for semantic segmentation based on random decision forests including feature extraction and classification with respect to spatial context. Several high-level cues and feature types that are integrated in our approach are presented in Section 3. Extensive experiments and an analysis of the results are done in Section 4. A summary of our findings and a discussion of future research directions conclude the paper.

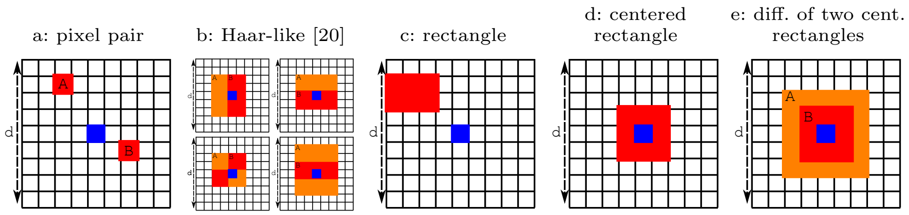
2 Semantic Segmentation with Iterative Context Forests

Our semantic segmentation approach, named Iterative Context Forests (ICF), is based on the massive use of random decision forests and the computation of several basic as well as high-level contextual features during learning.

Random Decision Forest Random decision forests (RDF) are an extension of the well known decision trees. The main disadvantage of decision trees is the high risk of over-fitting and the high computation time during learning. Breiman [2] showed how to circumvent both aspects with different kinds of randomization. RDFs use multiple decision trees in which each tree is trained with a different random and balanced subset of the training data. Furthermore, in an inner node of a tree, only a random subset $\mathcal{S} \subseteq \mathcal{U}$ with τ features is used to find the best binary split of the training data, which is done by maximizing the information gain. A huge benefit of this idea is that not all available features have to be computed in each inner node, which is an essential property for our approach. To treat every feature equally, independent of its number of possible parameter

Table 1: List and description of feature extraction methods.

abbr.	description	abbr.	description
PP1	diff. of two random pixel	RA2	centered rectangle
PP2	absolute diff. of two random pixel	RA3	diff. of two centered rectangles
PP3	sum of two random pixel	HL1	horizontal Haar features
PP4	value of a single random pixel	HL2	vertical Haar features
RP1	relative x-position of the pixel	HL3	diagonal Haar features
RP2	relative y-position of the pixel	HL4	3 rows of horizontal Haar features
RA1	random rectangle	HL5	3 columns of vertical Haar features


a: pixel pair

b: Haar-like [20]

c: rectangle

d: centered rectangle

e: diff. of two cent. rectangles

Fig. 2: Feature extraction methods applied to feature maps: features are computed in a window of size d around the current pixel position (blue pixel). Depending on the type of a feature one or two pixels [16] (a) or one (c and d) or two areas (b and e) are randomly selected. Every parameter θ is selected randomly (the size of an area, the position of the area, etc.) under some constraints, *e.g.*, for (d) the rectangle is centered. For features utilizing areas, the mean values of the areas are used.

values, we first sample the feature type uniformly and then sample the parameter vector (*e.g.* position and size of the region) in a second step.

For classification, a new example finds its path through each decision tree and the average of the empirical distribution in the reached leaves is used as an estimate for class-wise probabilities.

Generating Millions of Features The question remains how the set $\mathcal{U}$ of all available features is defined. Our approach is based on extracting large sets of features of very different characteristics from an input image $\mathcal{I}$. This is done by first computing several feature maps $(\mathcal{M}_i)_{i=1}^{m}$, which are matrices that store a value for each pixel of the input image. For example, one very simple feature map is the red channel of the image. After computing these maps, we apply several feature extraction methods g_θ to the feature maps to actually compute feature values. Those feature extraction methods are parameterized by a vector θ including the index of the feature map used and parameters for the exact position. A list of the feature extraction methods used in this paper is given in Table 1 and illustrated in Fig. 2.

Due to the large number of possible locations and feature maps, the set of available features $\mathcal{U}$ goes up to several million features. For example, for only one feature extraction method on a window of a size of $d = 50$ pixel, the number of possible feature pairs is $6.25 \cdot 10^6$. Due to the reason that we have many different feature extraction methods for many channels the real number of possible features increases dramatically. However, the randomization techniques of the RDF classifier allow us to handle these sets in an efficient manner by only computing a small random selection of them.

Auto Context Features Estimating semantic labels for each image pixel is a structured task requiring the usage of context knowledge to exploit the intrinsic dependencies between different parts of the image. A common approach is to use conditional Markov random fields with a pairwise potential modeling dependencies between two pixels. However, often high-order contextual cues are required to capture important context information. For example, if we like to model the relative locations of object categories, *e.g.*, "building" is above "road" but below "sky". This sort of prior knowledge can not be captured by a plain pairwise CRF.

Therefore, we use a concept known as *auto-context* [19], where features are computed based on previous classification results. Shotton *et al.* [16] used this technique in a two stage manner, where a first RDF was built on color features only and a second RDF used the results of the first one as auxiliary features. Our approach was inspired by this technique but extends it by applying auto context in an iterative manner. We built and traverse the trees always in a breadth-first manner, which allows us to use the results of a previous level as a source for additional features. In our case, we compute probability maps for the whole image in each level of a decision tree and use them as additional feature maps. This allows for extracting high-level contextual features. For example, the rectangle feature extraction method RA1 (see Table 1) evaluated in a region above a pixel yields a feature giving a cue whether a certain class is present on top of the current one.

3 High-level Cues for Semantic Segmentation

In this section, we present how to compute feature maps and how to incorporate them in our framework. Besides simple operations like the conversion of the image from the RGB color space to the HSI color space and the computation of gradient images we are using an unsupervised segmentation [3], 3D geometric context features [10], and a high-level color transformation [21]. We call the set of all used feature maps the feature pool. The feature extraction methods from the previous section are applied on these feature maps to extract features for each pixel.

Unsupervised Segmentation In previous approaches an unsupervised segmentation is used to smooth the results to get one label for homogeneous regions [4, 7, 22, 23]. There are two common ways in literature to incorporate the

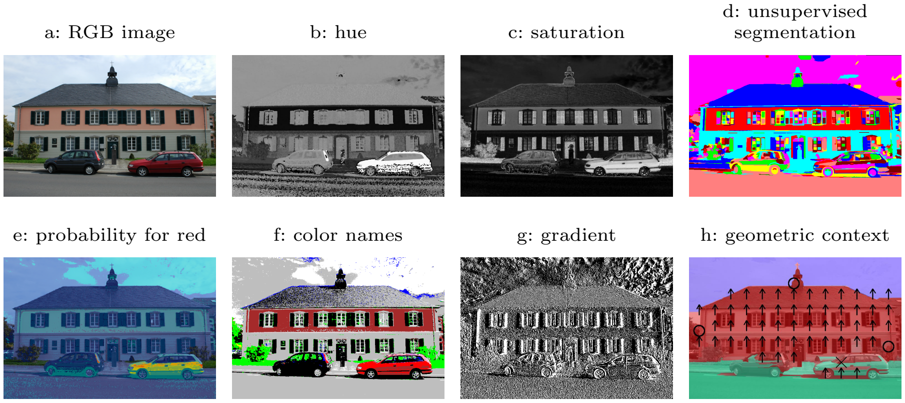


Fig. 3: Overview of the algorithms used to compute feature maps: a: input RGB image, b: color hue, c: color saturation, d: unsupervised segmentation result, where each area is encoded by a different color, e: probability map for class red, the right car is highlighted in this map, f: most probable color for each pixel based on the probability maps from image e, g: gradient image of the RGB image, h: geometric context features, for details we refer to [10].

regions. The first way is to annotate every pixel with the most probable class of one region after all other steps are finished [4, 7]. Furthermore, the second way is to utilize the regions in an early step to initialize a graph for a CRF were each region is a node and the neighborhood of two regions is modeled as an edge [22, 23]. In our framework, we found a third way to integrate region information. We propose to use the segmentation result as an additional feature added to the feature pool. After each iteration we compute the mean probability for each class in each region. Consequently, we have for each pixel the information about the previous classification result of all classes in the region where the pixel belongs to. We decided to utilize one of the most used unsupervised segmentation method, which is the mean-shift segmentation introduced by [3]. Please note that there are many alternative unsupervised segmentation methods like the very fast graph based segmentation introduced by Felzenszwalb and Huttenlocher [5].

3D Geometric Context Features A human is using 3D context features not only for classifying objects in the real world, but also for objects in 2D images. Of course, there is no direct 3D information in a 2D image without any additional knowledge. An important aspect is the Manhattan world assumption which says that most of the man-made environments are based on objects perpendicular to each other. Hoiem *et al.* [10] tries to learn such information from some hand labeled images, where they differ between the three main classes: “ground plane”, “surfaces at roughly right angles to the ground plane” and “sky”. These surfaces are split into “planar” and “non planar” surfaces. Furthermore, the “non planar” surfaces are split up into “solid” and “porous‘” and “planar” is subdivided into planar surfaces facing "left", "right", or "centered" towards the camera. Following this method, we can extract seven probability maps, one for each of these 3D geometric context classes, which are added directly as features to our feature pool.

Color Names An interesting idea to transform RGB color features to another feature space is introduced by van de Weijer *et al.* [21]. They describe these color names as linguistic labels that humans attach to colors. Therefore, they use eleven main colors which are not describable through a combination of two or more of the other colors. For example, nobody would say "reddish yellow" instead of "orange". To learn a transformation between the L*a*b*-color space and the color names the authors use a set of annotated images, where in each image one object of a specific color is masked. The color space is partitioned into $10 \times 20 \times 20$ bins for each channel of the L*a*b*-space. The distribution of each bin is calculated by counting pixels of each color ending up in a specific bin. With this it is possible to transform each RGB value into pseudo probabilities for each color name.

4 Experiments

In the following, we evaluate our methods on some facade datasets. The analysis concentrates on recognition performance as well as time needed for labeling a single image.

Experimental Setup For feature extraction, we use a window with a size of $d = 50$ pixels for the non-context features and $d = 200$ pixels for the auto context features. The random forest contains five trees with a maximum depth of 15 levels and a random subset of $\tau = 400$ features is used in each node during learning. Computation times are evaluated on an Intel®Core$^{\text{TM}}$i7 CPU 930 with 2.8GHz with four cores. We differentiate between the average recognition rate over all classes and pixel-wise accuracy, which we refer to as overall recognition rate. The different modifications of the ICF are illustrated by additional letters. H represents the use of the geometric context features, G the usage of the gradient image and W the color representation of van de Weijer [21]. Furthermore, mean-shift [3] is used as an unsupervised segmentation method providing optional feature for the feature pool (S+), or for post processing (S-). For example, ICFHG represents an Iterative Context Forest using the geometric context features and gradient images besides the HSI channel and the auto context features.

Facade recognition For our experiments, we use the eTRIMS dataset originally introduced by Korč and Förstner [12]. We use ten different random splits of the data into 40 images for training and 20 images for testing similar to [22, 23]. Furthermore, the LabelMeFacade dataset introduced in [7], which contains 100 images for training and 845 images for testing, is used as a second more

Table 2: Recognition rates of our experiments with different classifiers in comparison to previous work. In contrast to [7], we used random splits of training and testing for the eTRIMS dataset to allow for fair comparison with [22, 23]. ICF represents our proposed approach including auto-context and the HSI color channels. An additional letter shows the usage of the feature channels: *H* in the name represents the usage of the geometric context features of Hoiem *et al.* [10], *W* the usage of the color names from van de Weijer *et al.* [21], *S+* the direct incorporation of the mean-shift segmentation [3] and *S-* the usage of mean-shift as a post-processing step.

dataset	approach	average recognition rate	overall recognition rate
eTRIMS	CRF [23]	49.75%	65.80%
	HCRF [22]	61.63%	69.00%
	3-Layer [14]	63.25%	**81.94%**
	SIFT/RDF [7]	62.81% (±1.58)	64.00% (±3.28)
	SIFT/SLR [7]	65.57% (±2.47)	71.18% (±2.69)
	ICF	68.61% (±1.71)	70.81% (±1.32)
	ICFwoC	64.07% (±1.72)	61.11% (±1.59)
	ICFHGW	71.47% (±1.25)	72.59% (±1.06)
	ICFHGWS+	68.94% (±1.48)	73.65% (±1.07)
	ICFHGWS-	72.22% (±2.17)	75.09% (±1.60)
	ICFHS-	**72.26%** (±3.25)	76.10% (±1.24)
	ICFHGS-	72.23% (±1.76)	77.22% (±1.22)
LabelMeF	SIFT/RDF [7]	44.08% (±0.45)	49.06% (±0.52)
	SIFT/SLR [7]	42.81% (±0.89)	48.46% (±1.58)
	ICF	49.39% (±0.48)	60.68% (±0.72)
	ICFwoC	47.66% (±0.06)	43.97% (±0.03)
	ICFHGW	56.95% (±0.28)	61.93% (±0.65)
	ICFHGWS+	56.61% (±0.32)	**67.33%** (±0.67)
	ICFHGWS-	57.11% (±0.20)	66.08% (±1.68)
	ICFHS-	**57.82%** (±0.19)	64.76% (±0.78)
	ICFHGS-	57.35% (±0.51)	66.86% (±0.41)

challenging dataset. Both datasets consists of the eight classes shown in Fig. 4 and an additional background class named “unlabeled”. For trivial decision rules or random guessing the average recognition rate for both datasets is 12.5% and the overall recognition rate is less than 35% (all pixels labeled as building).

Table 2 and Fig. 4 shows some results on the eTRIMS and the LabelMe-Facade datasets using different methods for semantic segmentation. First of all, one can see that we outperform all previous state-of-the-art approaches on these datasets significantly. All previous approaches from [7, 22, 23] are based on SIFT features, which need to be fully computed in advance. The Iterative Context Forest (ICF) only using simple color features (HSI) and the auto context features is as good as previous state-of-the-art results. Furthermore, incorporating additional features improves the results significantly. The usage of the gradients, the unsupervised segmentation as an post-processing step and the geometric context from Section 3 improves the recognition rate obviously. Unfortunately,

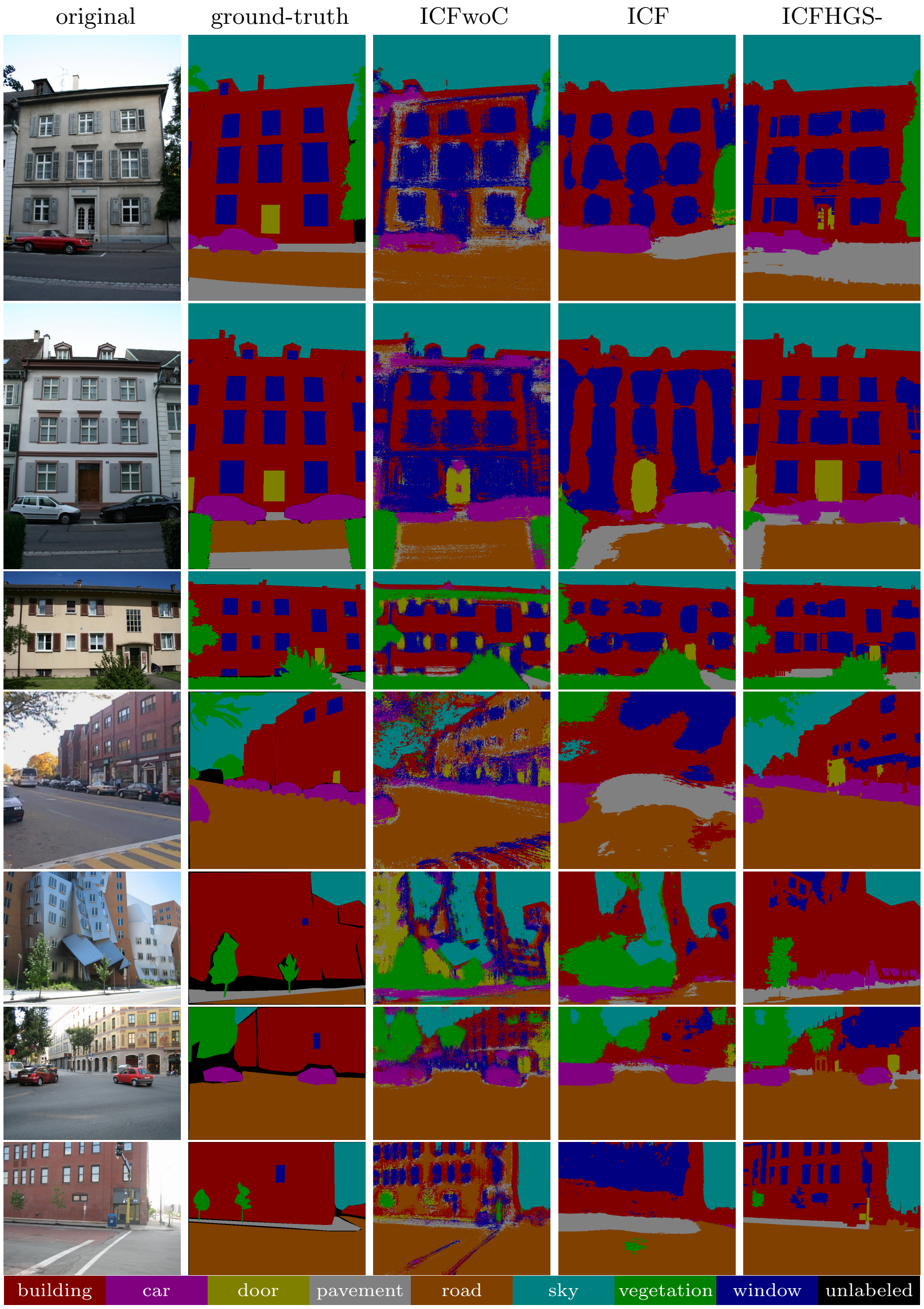


Fig. 4: Example images from eTRIMS (first three rows) and LabelMeFacade database (last four rows). The corresponding results obtained by a decision tree without any auto context (ICFwoC), Iterative Context Forests using only color features (ICF), and Iterative Context Forests using color, gradients, 3D geometric context and an unsupervised segmentation (ICFHGS-) are shown.

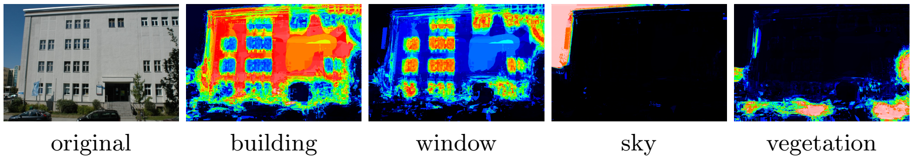
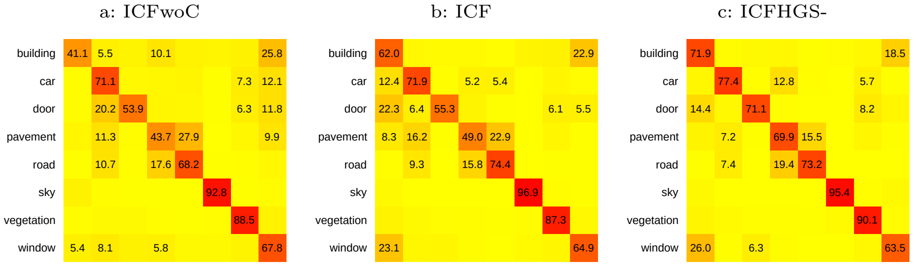


Fig. 5: Probability maps of some classes for a specific image. Warm colors correspond to areas with high probability.


a: ICFwoC

b: ICF

c: ICFHGS-

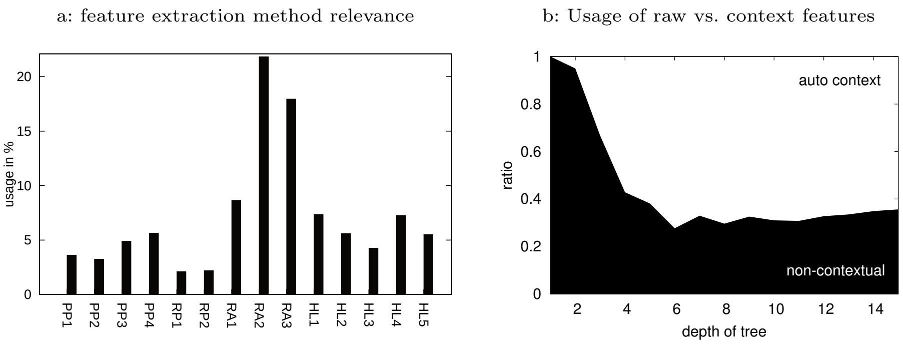
Fig. 6: Confusion matrices for one run in the eTRIMs dataset using the same settings as shown in Fig. 4.

the utilization of the color names does not bring any improvement of the results, but a decrease in performance. We also did experiments with the MSRC21 dataset achieving an overall accuracy of 67.2% using ICFHGWS-. The accuracy increases with each iteration until it converges. We point to [8] for further results of the different iterations.

The computation time depends on the usage of the feature channels. The ICF using auto-context and the HSI color channels needs $\approx 3s$ per image. Computing the 3D geometric context features increases the time by $\approx 10s$, the segmentation $\approx 3s$ and the color names $\ll 1s$. Therefore, it is possible to adjust between a fast computation of the result or a high accuracy by choosing different types of features and by selecting parameters like the amount of trees and the depth of each tree.

Some samples for the auto context feature maps are shown in Fig. 5. Each of these channels is computed after each iteration and used for computing the splits in the next level of the forest (see Section 2).

The confusion matrices for three different settings of our framework are presented in Fig. 6. Incorporating context features increases the recognition rate for the classes “building”, “road”, and “sky” significantly. Furthermore, using the additional features increases the recognition rates for “building”,“car”, “door”, “pavement” and “vegetation” clearly. All proposed methods still have problems with the confusion of “window” and “building” as well as “pavement” and “road”.


a: feature extraction method relevance

b: Usage of raw vs. context features

Fig. 7: Statistical analysis of used features types and feature extraction methods. a: Usage of each feature extraction method in a learned decision forest, b: Usage of context features and non-contextual features for each level of the decision trees.

Evaluation of Feature Usage As mentioned above, we use a huge amount of different feature extraction methods. In this section, we want to analyze the usage of each of the methods presented in Table 1. As shown in Fig. 7a by far the most important features are different kinds of rectangle features (*cf.* [17]), led by the single window centered at the current pixel position, followed by the difference of two windows centered in the middle and a rectangle with a random position relative to the current position. More than 48% of all decisions over all trees are done using these three feature extraction types. Another 30% of the decisions are done using some of the Haar features. The pixel-pair respectively the single pixel features are chosen in about 17% of all cases and only 4% of all decisions are based on the relative position of a pixel in an image. This is more or less what we have expected. Relative positions should not be that important, due to a high risk of over-fitting, but it is still an useful information for some classes like "sky". Pixel-pair features are much more sensitive against image noise compared to rectangle features.

In Fig. 7b the usage of non-contextual versus auto context features is plotted. In the first level of a tree it is not possible to use any auto context features, but from the second level of a tree the influence of the contextual features is slowly increasing. Beginning at a depth of about six levels the ratio converges at about 60% auto context features and about 40% non-contextual features. This shows that both feature types are important to train an ICF and that the importance of the auto context features increases with the quality of previous outputs.

Looking Beyond the Current Horizon As we have seen in the previous sections, our method is able to remarkably outperform state of the art approaches for semantic segmentation. However, we discovered several special cases, where our proposed method failed during evaluations. We visualized exemplary images in Fig. 8 showing problematical details of the classification results from Fig. 4.

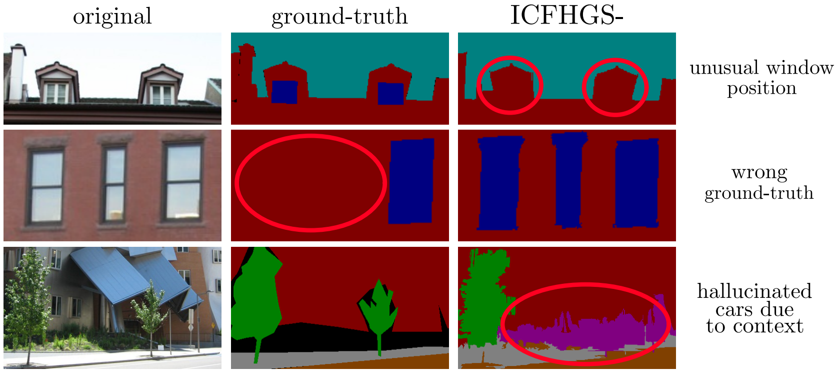


Fig. 8: Detailed analysis of the results of our approach.

(1) Untypical positions like the windows on the roof are underrepresented in the training data. Obviously, these problems can be solved by adding more images to the training data. (2) Another problem are badly labeled images as shown in the second row of Fig. 8. This is a twofold problem. First of all, during training the classifier learns the class "building" based on noisy data leading to disturbed classification models. On top of that, evaluations are negatively skewed since test images are counted as wrongly classified in these regions due to the missing ground truth data. (3) As shown in the last row of Fig. 8 in some special cases objects are identified which are not in the image but would have been expected to be based on contextual assumptions. In this example the ICF tries to fit the class "car" between "pavement" and "building" instead of "vegetation". Apparently the classifier was not sure how to classify these regions. Although this may be due to missing training data for this constellation of classes, it would be interesting to regularize the influence of context in such scenarios.

5 Conclusion and Further Work

In this work, we presented a new approach for incorporating context features and multiple other features in a single framework for semantic segmentation. We have shown that our approach is very flexible and can simply be adjusted between fast evaluation and high accuracy. In extensive experiments, we have shown that our approach significantly outperforms other state of the art methods including time consuming conditional random field techniques. Furthermore, we have shown that extending the set of available features can increase the recognition rate. Especially 3D geometric context features lead to a high performance gain for the challenging LabelMeFacade dataset.

For future work, we plan to adapt the random sampling to allow for integration of prior knowledge about feature relevance. The current approach samples uniformly from the set of available feature types and does not differentiate between them. Furthermore, the sampling could be also tuned towards sampling easy-computable features with high probability. This strategy would lead to trees with a lower average computation time for classifying a new test input.

References

1. Arbelaez, P., Hariharan, B., Gu, C., Gupta, S., Bourdev, L., Malik, J.: Semantic segmentation using regions and parts. In: CVPR (2012)
2. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
3. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. PAMI 24(5), 603–619 (2002)
4. Csurka, G., Perronnin, F.: An efficient approach to semantic segmentation. IJCV 95(2), 198–212 (2011)
5. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV 59(2), 167–181 (2004)
6. Fink, M., Perona, P.: Mutual boosting for contextual inference. In: NIPS. vol. 16, pp. 1515–1522 (2003)
7. Fröhlich, B., Rodner, E., Denzler, J.: A fast approach for pixelwise labeling of facade images. In: ICPR. pp. 3029–3032 (2010)
8. Fröhlich, B., Rodner, E., Denzler, J.: As time goes by — Anytime semantic segmentation with iterative context forests. In: DAGM. pp. 1–10 (2012)
9. Gool, L.J.V., Zeng, G., den Borre, F.V., Müller, P.: Towards mass-produced building models. In: Photogrammetric Image Analysis. pp. 209–220 (2007)
10. Hoiem, D., Efros, A.A., Hebert, M.: Geometric context from a single image. In: ICCV. vol. 1, pp. 654–661. IEEE (October 2005)
11. Kohli, P., Ladicky, L., Torr, P.: Robust higher order potentials for enforcing label consistency. In: CVPR. pp. 1–8 (2008)
12. Korč, F., Förstner, W.: eTRIMS image database for interpreting images of man-made scenes. Tech. Rep. TR-IGG-P-2009-01, University of Bonn (2009)
13. Ladický, Ľ., Russell, C., Kohli, P., Torr, P.H.S.: Associative hierarchical crfs for object class image segmentation. In: ICCV. pp. 739–746 (2009)
14. Martinovic, A., Mathias, M., Weissenberg, J., van Gool, L.: A three-layered approach to facade parsing. In: ECCV (2012)
15. Ripperda, N., Brenner, C.: Evaluation of structure recognition using labelled facade images. In: DAGM. pp. 532–541 (2009)
16. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR. pp. 1–8 (2008)
17. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: ECCV. pp. 1–15 (2006)
18. Teboul, O., Simon, L., Koutsourakis, P., Paragios, N.: Segmentation of building facades using procedural shape priors. In: CVPR. pp. 3105–3112 (2010)
19. Tu, Z., Bai, X.: Auto-context and its application to high-level vision tasks and 3d brain image segmentation. PAMI 32(10), 1744–1757 (2010)
20. Viola, P., Jones, M.: Robust real-time object detection. IJCV 57, 137–154 (2002)
21. van de Weijer, J., Schmid, C.: Applying color names to image description. In: ICIP (3). pp. 493–496 (2007)
22. Yang, M.Y., Förstner, W.: A hierarchical conditional random field model for labeling and classifying images of man-made scenes. In: ICCV Workshops. pp. 196–203 (2011)
23. Yang, M.Y., Förstner, W.: Regionwise classification of building facade images. In: Photogrammetric Image Analysis, pp. 209–220. Springer (2011)