As Time Goes By — Anytime Semantic Segmentation with Iterative Context Forests

Björn Fröhlich and Erik Rodner and Joachim Denzler

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

Abstract. We present a new approach for contextual semantic segmentation and introduce a new tree-based framework, which combines local information and context knowledge in a single model. The method itself is also suitable for anytime classification scenarios, where the challenge is to estimate a label for each pixel in an image while allowing an interruption of the estimation at any time. This offers the application of the introduced method in time-critical tasks, like automotive applications, with limited computational resources unknown in advance. Label estimation is done in an iterative manner and includes spatial context right from the beginning. Our approach is evaluated in extensive experiments showing its state-of-the-art performance on challenging street scene datasets with anytime classification abilities.

1 Introduction

Semantic labeling or classification is an important task for localizing objects or to perform scene understanding. In a large set of applications, such as road detection [3], street scene analysis [10], and robotics [11], one is often faced with constraints on classification time. Even more severe, those constraints are sometimes a priori unknown and depend on external conditions. For example, the time in which we require road and lane detection depends on the current speed of the car. Machine learning methods that allow for tackling these requirements by providing outputs at different time steps are referred to as anytime classification approaches [5]. The main idea is that output quality increases if more time is provided, while proper results are also available after a short initialization time.

In our paper, we present an anytime semantic segmentation approach, which is able to use contextual cues immediately after the first iteration. The approach is built on a technique, which we call *Iterative Context Forests* (ICF) (Fig. 1). It performs efficient semantic segmentation without explicit need for inference with conditional random field models and without time consuming feature extraction or post-processing steps. Instead of subsequently traversing a decision tree for each pixel until a leaf node is reached, we walk through a tree in a level-based manner for each pixel jointly.

Related work on anytime classification Anytime classification has been mostly considered for standard machine learning and data mining tasks instead

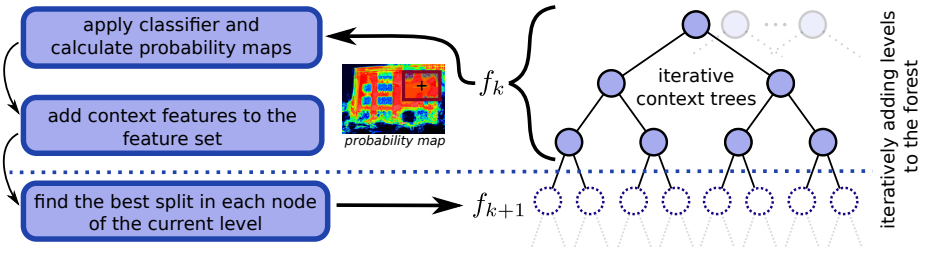

Fig. 1: Learning an Iterative Context Forest: we learn a random decision forest level by level and integrate context cues by always computing features using the previously estimated probability maps. This generates a series of classifiers f_k, which can be used for anytime classification

of semantic segmentation and visual recognition. In [5], a decision tree classifier is presented, which is able to perform anytime classification and learning. Their paper also gives an introduction into the topic and discriminates between interruptible and contract anytime classifiers. In contrast to interruptible anytime classifiers, which are considered in our paper, contract classifiers are provided with time and memory requirements in advance. The work of [12] considers anytime classification for density estimation. The main idea is to incrementally refine the density estimate by traversing a tree, in which inner nodes store a rough Gaussian approximation of the density and each leaf node is related to a Parzen density estimator. Anytime classification with SVM is studied by [4] using geometric considerations. The authors of [13] use an anytime nearest neighbor classifier and propose methods for scheduling multiple object classification similar to [8]. In contrast to those works, we study anytime classification for visual recognition and show how to perform joint classification of pixels, which incorporates contextual knowledge.

Related work on context modeling 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 [6] for their mutual boosting approach. They train a set of object detectors simultaneously. In each round of the Boosting method, they add features derived from the results of the current classifier. Our work is also related to the approach of [14], 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 is essential to allow for anytime classification, since the procedure can be stopped at any time and still provides a proper result.

Outline We first give an informal definition of anytime classification and its requirements in Sect. 2. This is followed by describing our approach in Sect. 3.

Experiments in Sect. 4 evaluate our method on street scene analysis tasks and show their advantages as well as anytime properties. A summary of our findings and a discussion of future research directions conclude the paper.

2 Anytime Classification

The goal of standard machine learning methods is to estimate the latent relationship between inputs (feature vectors) and labels from available training data. In most cases, this can be expressed with a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ mapping from the space $\mathcal{X}$ of all inputs to a defined label space $\mathcal{Y}$, such as $\mathcal{Y} \in \{1, \ldots, M\}$ for multi-class classification.

Anytime classification involves some time requirements posed during classification, *i.e.*, the evaluation of the function f. At several time steps t_k, we would like to have a result $f_k(\mathbf{x}_*)$ for a test example $\mathbf{x}_*$. Thus, we have a series $\mathcal{F} = (f_k)_{k=1}^{\infty}$ of functions at certain time steps $(t_k)_{k=1}^{\infty}$. Allowing the system to spend more time during classification, the classification result should be more accurate. In contrast to state estimation in dynamic systems, anytime classification in our case considers the input to be static without any change over time. Thus, we do not get any additional sensor information during classification. The main requirements of anytime classification are given as follows:

1. **Decreasing error rate:** The excepted error ε of the decision functions in $\mathcal{F}$ should be monotonically decreasing, *i.e.*, $\varepsilon(f_k) \geq \varepsilon(f_{k'})$ for $t_k < t_{k'}$.
2. **Flexibility:** The time differences $\triangle_k = t_{k+1} - t_k$ should be small to allow for high flexibility during classification.
3. **Direct availability:** The classification result is directly available in time step t_k and there is no additional time-consuming post-processing required after interrupting the algorithm.

To learn anytime classifiers, it is beneficial to build the series $\mathcal{F}$ of classifiers in an iterative manner, *i.e.*, f_{k+1} is an extension and adaptation of f_k.

For evaluation of anytime classification systems, time/error curves defined by $(t_k, \varepsilon(f_k))_{k=1}^{\infty}$ are an essential tool. The limit of this curve gives us the performance of the classifier disregarding any time constraints. However, in anytime classification scenarios the rate in which the error decreases during the first time steps is often more important than the limit. In the following sections, we show how to develop an anytime classification system using random decision forests, which matches the requirements stated above.

3 Iterative Context Forests (ICF)

Due to the high amount of ambiguities present in recognition tasks, incorporating context knowledge is necessary. A common approach is to utilize a CRF model to combine independent local decisions with global or relative location context. However, those techniques require a large amount of the available classification

time, since they perform optimization with a large number of variables. Iterative Context Forests (ICF) allows for incorporating context knowledge directly during learning of the RDF without CRF modeling. The idea is to train a classifier f_{k+1} for the next time step, *i.e.*, the new level of the random forest, by using the output of the previous classifier f_k to compute additional features introduced in Section 3.3.

3.1 Random Decision Forest

A random decision forest (RDF) is an extension of the well known decision trees. The main disadvantage of decision trees without pruning is the high risk of over-fitting, which [1] try to prevent by different kinds of randomization. RDFs use multiple decision trees in which each tree is trained with a different random subset of the training data. Furthermore, in an inner node of a tree, only a random subset $\mathcal{S}$ 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.

Typically each new example traverses the tree and is classified by using the empirical distribution in the reached leaf node. In contrast, we propose a breadth-first method for classification which enables anytime classification. All new examples traverse the tree jointly in a level-wise manner. In each node, the empirical class distribution estimated during learning can be used as a rough classification result. This offers the possibility to obtain a classification result for all examples at different levels k and time steps t_k. The accuracy depends on the current level reached in the tree and care has to be taken to prevent over-fitting. In our case, over-fitting due to an increasing model complexity is reduced by utilizing the randomization techniques of an RDF classifier. Given multiple trees, all trees are traversed level-wise in a parallel manner. The series of classifiers f_k, as defined in Sect. 2, is thus the learned random forest reduced to a maximum depth of k. Another idea to extend random decision forests towards anytime capability is to traverse one tree after another. However, our approach is more flexible and allows contextual information.

3.2 Color Features

An important requirement for the features is a fast extraction. Therefore, we use basically the same operations as in [14] as an initial feature set with minor modifications:

1. **pixel pair features:** the output of simple operations A, $A - B$, $|A - B|$, $A + B$, with randomly selected pixels A and B in the neighborhood of the current position (Fig. 2a).
2. **Haar like features** [15]: horizontal, vertical, and diagonal differences (Fig. 2b)
3. further **rectangular area features** using integral images to compute the mean values in these area (Fig. 2c-2e)

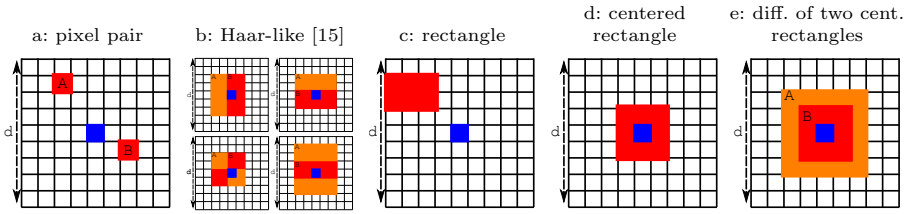

Fig. 2: Features used in our approach for both context and color cues similar to [14]. A window of size d is surrounding the current pixel position (blue pixel). Depending on the type of a feature one or two pixels (a) or one (c and d) or two areas (b and e) are randomly selected. Every parameter is selected randomly (the size of an area, the position of the area etc.) under some constraints, *e.g.*, for image (d) the rectangle is centered. For features utilizing areas, the mean values of the areas is used.

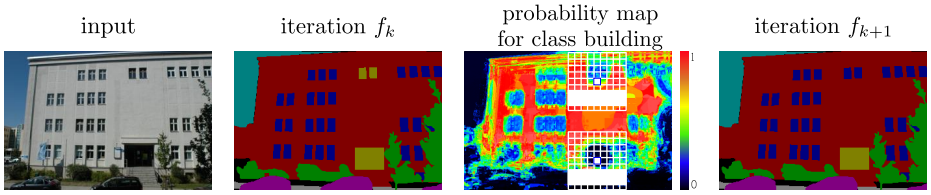

Fig. 3: An exemplary scenario: Some windows are wrongly classified as door in iteration f_k. Using the probability map for class building with the rectangle features (see Fig.2c) shown in the third image the wrongly classified windows will be classified correctly in iteration f_{k+1}

4. **relative position** in the image (normalized coordinates)

This leads to a large number of possible features computed on RGB or CIELAB color channels. Due to the reason, that only some features are randomly selected as potential split features for a node during training, those feature do not have to be computed in advance. Therefore, we only have to compute a small set of features instead of full feature representation as used in [7].

3.3 Iteratively Extending the Feature Set with Context Features

A standard RDF uses a fixed feature set $\mathcal{S}$ during the whole training process. But how is it possible to model important context cues like "window is surrounded by building" and "car stands on road or pavement"? The basic idea of the ICF is to adapt the feature set $\mathcal{S}$ during training. For classifying a local image patch, an important context cue is the relative position of other objects. This can be modeled by using the probability for specific classes in an image region with a learned offset (compare Fig. 2c). If we know the probability of each class and each pixel we can utilize the same features used before for the raw image data calculated with these so called probability maps.

This is a typical chicken-egg problem, since an already learned classification model is required. In our case we can use the output of the previous iteration as a rough estimate of the probabilities which automatically improves over time. In the first iteration $k = 1$, $\mathcal{S}$ includes only simple features, *e.g.*, RGB color features as introduced in Sect. 3.2. However, in iteration $k > 1$ the pixel-wise probabilities for each training image estimated by f_{k-1} can be considered to obtain a rough estimate of the position and alignment of other classes. To give a simple example, in the first level of the forest each tree makes a decision based only on the color features. With this the image is very roughly separated in at least two main classes, *e.g.*, "road" and "building". For the next level of the forest we use these rough segmentation as context features. For example, if we want to decide whether the red area is a roof of a building or a car, it might be important if there are some pixel below that area already labeled as building or not. Therefore, we use those pixel-wise probabilities from the previous step to compute semantic context features, which are added to $\mathcal{S}$. In contrast to Shotton [14], we model contextual information with only a single forest, which allows for anytime classification. The training step of an ICF is illustrated in Fig. 1 and an example how context is modeled is shown in Fig. 3.

3.4 Anytime Capability

In Sect. 2, we defined the requirements for anytime classification. Now we show that all of the three points are practically valid for our ICF method. The computational effort for each step is very low, since only one simple decision stump has to be evaluated for each pixel and tree in each level. Consequently, our method is very flexible and allows for decisions in equidistant time steps. Furthermore, we do not need any post-processing steps like an unsupervised segmentation used in previous work [2, 7, 16, 17]. The final labeling is done by assigning the class with the highest probability to each pixel. In our experiments, we show empirically that the first property of decreasing error rates is also satisfied. This is mainly due to the incorporated randomization during learning, which reduces overfitting effects normally appearing when increasing the model complexity of a classifier [1]. However, we are not able to provide a theoretical proof since the characteristics of the test data are not known in advance.

4 Experiments

In the following, we evaluate our method on some datasets related to facade recognition and street scene analysis.

Settings For feature extraction, we use a window with a size of $d = 50$ pixels. 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. Computational times are evaluated on a computer with 2.8GHz and four cores. We differentiate between the average recognition rate over all classes and pixel-wise accuracy, which we refer to as overall recognition rate.

Table 1: Recognition rates of our experiments with different classifiers (our approach with (*ICF*) and without context features (*ICFwoC*)) in comparison to previous work (Random Decision Forest (RDF), Sparse Logistic Regression (SLR), Conditional Random Field (CRF) and Hierarchical Conditional Random Field (HCRF)). In contrast to [7], we used random splits of training and testing for the eTRIMS dataset to allow for fair comparison with [16, 17]

dataset	approach	average recognition rate	overall recognition rate
eTRIMS	CRF [17]	49.75%	65.80%
	HCRF [16]	61.63%	69.00%
	RDF [7]	62.81% (±1.58)	64.00% (±3.28)
	SLR [7]	65.57% (±2.47)	**71.18**% (±2.69)
	ICFwoC	64.07% (±1.72)	61.11% (±1.59)
	ICF	**68.61%** (±1.71)	70.81% (±1.32)
LabelMeF	RDF [7]	44.08% (±0.45)	49.06% (±0.52)
	SLR [7]	42.81% (±0.89)	48.46% (±1.58)
	ICF	**49.39**% (±0.48)	**60.68**% (±0.72)

Facade recognition For our experiments, we use the eTRIMS dataset originally introduced by Korč and Förstner [9]. We use ten different random splits of the data into 40 images for training and 20 images for testing similar to [16, 17]. Furthermore, the LabelMeFacade dataset introduced in [7], which contains 100 images for training and 845 images for testing, is used as a second dataset being more challenging. Both datasets consists of the eight classes as 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). The results of our method in comparison to other state-of-the-art methods are shown in Table 1. On the eTRIMS dataset, our proposed approach significantly outperform all other methods with respect to average recognition rates. The overall recognition is as good as those of the SLR method introduced in [2, 7]. However, the benefits of our approach are more prominent for the challenging LabelMeFacade dataset. ICF outperforms all previous approaches clearly on this dataset. Some sample results are presented in Fig. 4. Please note that rounded corners are somehow characteristic for our approach due to the reason that we do not use an unsupervised segmentation and the usage of rectangle features smooths the result. Furthermore we do not need a time consuming feature extraction step as in all other methods, we are significantly faster. ICF needs $\sim 3s$ for testing compared to $\sim 30s$ for RDF with SIFT feature extraction and unsupervised segmentation, which achieved the fastest classification speed in the evaluation of [7]. Please note that these times include I/O operations. There are also ways to further speed-up our method. The classification step can be highly parallelized using a CUDA implementation. Furthermore, we could apply our method in a coarse to fine manner using image pyramids. The results for anytime classification for eTRIMS are shown in the left image of Fig. 6 and one sample result for

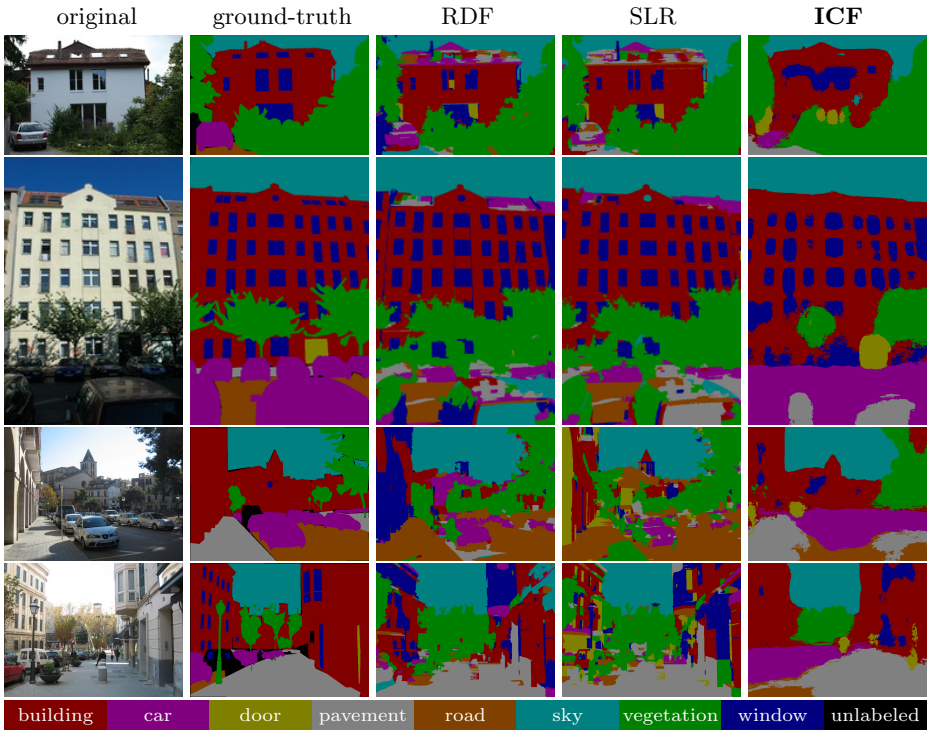

Fig. 4: Example images from eTRIMS (first two rows) and LabelMeFacade database (last two rows). The corresponding results obtained by random decision forest (RDF) [7], sparse logistic regression (SLR) [7], and Iterative Context Forests (ICF) are shown on the right side

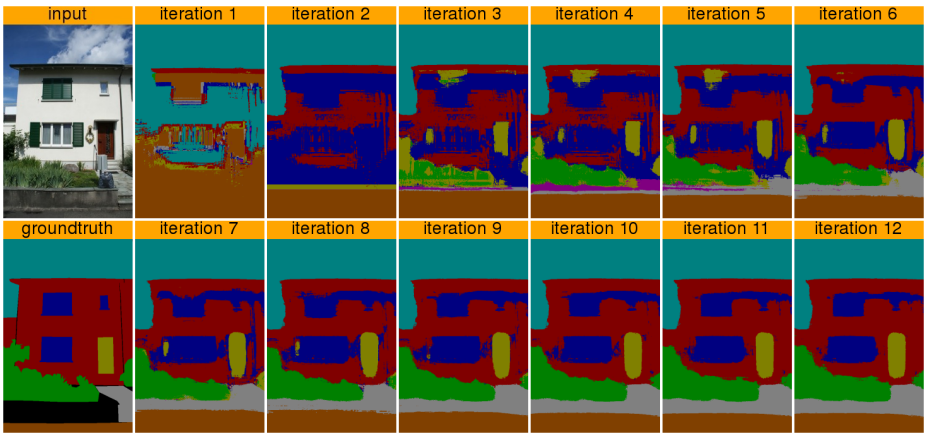

Fig. 5: Sample result for anytime semantic segmentation, input image, ground-truth and segmentation results for 12 time steps

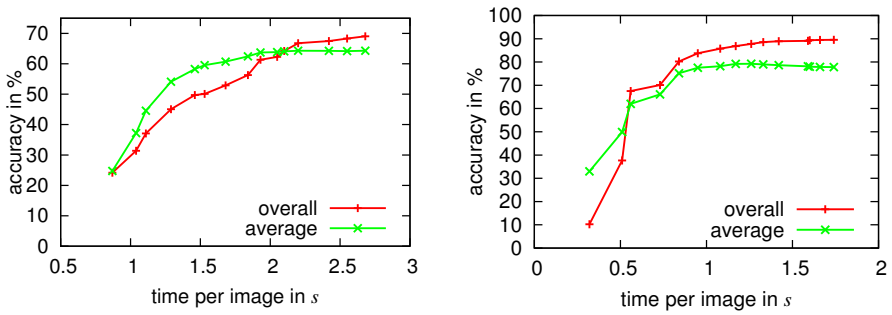

Fig. 6: Recognition performance for different time steps corresponding to the characteristically time/error curve of anytime classifiers for eTRIMS (left) and Leuven street scenes (right). Computational times include I/O operations

different iterations is presented in Fig. 5. It is obvious that the quality of the result increases with the number of iterations and consequently with time.

Street scenes for autonomous cars Additionally, we performed experiments on the Leuven street scene database introduced in [10]. This dataset represents a scenario which highly benefits from anytime applications. In the dataset, a car is steered through an urban area. For an autonomous car, it is important to know the exact position of the road and the location of objects and obstacles (like walls or persons). In contrast to [10], we do not use depth information extracted from the stereo images provided in the dataset. Furthermore, neither time context from previous images nor additional adaptations of the settings for this scenario are done. However, our method achieves an overall accuracy of 89.55%. The CRF approach of [10] resulted in 95.7% correctly labeled pixels. The benefit of our method is its speed (1.74s for each image on average) and the ability for interruption. Stopping in a prior iteration speeds up the algorithm and results in near real time capabilities (compare right Fig. 6).

5 Conclusion and Further Work

In this work, we presented a new approach for anytime semantic segmentation, which can be applied in time-critical applications with unknown resource limits. We defined the requirements in those scenarios and showed how to perform semantic segmentation by traversing random decision trees in a level-based manner. This allows for an interruptibility of the algorithm and for including context features iteratively. Context cues are integrated right from the beginning of the algorithm and meaningful classification results are available already after a short time. Evaluation was done on multiple datasets for facade recognition and street scene analysis. For very difficult tasks, our method achieved a superior performance compared to previous approaches in this area and with less computational effort. Furthermore, we have shown that our approach can be used for anytime semantic segmentation with results at several time steps in less than a second.

For future work, we plan to add complex shape features and higher-order context cues. In general, we expect that there is a large set of features benefiting from previously estimated probability maps. An additional cue might be the uncertainty of the probability maps calculated using the empirical entropy. Context features in regions of high uncertainty are unlikely to be a robust cue and their use should be limited during learning. Furthermore, we want to integrate unsupervised segmentation techniques to align the resulting segmentation to edges and object boundaries.

References

1. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
2. Csurka, G., Perronnin, F.: An efficient approach to semantic segmentation. IJCV 95(2), 198–212 (2011)
3. Dahlkamp, H., Kaehler, A., Stavens, D., Thrun, S., Bradski, G.: Self-supervised monocular road detection in desert terrain. In: Robotics: Science and Systems (2006)
4. DeCoste, D.: Anytime interval-valued outputs for kernel machines: Fast support vector machine classification via distance geometry. In: Proceedings of the International Conference on Machine Learning (ICML'02). pp. 99–106 (2002)
5. Esmeir, S., Markovitch, S.: Anytime learning of anycost classifiers. Machine Learning 82(3), 445–473 (2011)
6. Fink, M., Perona, P.: Mutual boosting for contextual inference. In: Advances in Neural Information Processing Systems (NIPS'03). 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. Hui, B., Yang, Y., Webb, G.: Anytime classification for a pool of instances. Machine learning 77(1), 61–102 (2009)
9. 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)
10. Ladický, Ľ., Sturgess, P., Russell, C., Sengupta, S., Bastanlar, Y., Clocksin, W., Torr, P.: Joint optimisation for object class segmentation and dense stereo reconstruction. In: BMVC. pp. 104.1–11 (2010)
11. Rusu, R.B., Holzbach, A., Bradski, G., Beetz, M.: Detecting and segmenting objects for mobile manipulation. In: Proceedings of IEEE Workshop on Search in 3D and Video (S3DV). pp. 47–54 (2009)
12. Seidl, T., Assent, I., Kranen, P., Krieger, R., Herrmann, J.: Indexing density models for incremental learning and anytime classification on data streams. In: Proceedings of the 12th Int. Conference on Extending Database Technology. pp. 311–322 (2009)
13. Shieh, J., Keogh, E.J.: Polishing the right apple: Anytime classification also benefits data streams with constant arrival times. In: Proceedings of the International Conference on Data Mining. pp. 461–470 (2010)
14. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR. pp. 1–8 (2008)
15. Viola, P., Jones, M.: Robust real-time object detection. IJCV 57, 137–154 (2002)
16. Yang, M.Y., Förstner, W.: A hierarchical conditional random field model for labeling and classifying images of man-made scenes. In: Proceedings of the IEEE Computer Vision Workshops (ICCV Workshops). pp. 196–203 (2011)
17. Yang, M.Y., Förstner, W.: Regionwise classification of building facade images. In: Photogrammetric Image Analysis, pp. 209–220. Springer (2011)