Accurate Object Detection with Joint Classification-Regression Random Forests

Samuel Schulter[†] Christian Leistner[‡] Paul Wohlhart[†] Peter M. Roth[†] Horst Bischof[†]

[†]Graz University of Technology
Institute for Computer Graphics and Vision
{schulter,wohlhart,pmroth,bischof}@icg.tugraz.at

[‡]Microsoft Photogrammetry
Austria
christian.leistner@microsoft.com

Abstract

In this paper, we present a novel object detection approach that is capable of regressing the aspect ratio of objects. This results in accurately predicted bounding boxes having high overlap with the ground truth. In contrast to most recent works, we employ a Random Forest for learning a template-based model but exploit the nature of this learning algorithm to predict arbitrary output spaces. In this way, we can simultaneously predict the object probability of a window in a sliding window approach as well as regress its aspect ratio with a single model. Furthermore, we also exploit the additional information of the aspect ratio during the training of the Joint Classification-Regression Random Forest, resulting in better detection models.

Our experiments demonstrate several benefits: (i) Our approach gives competitive results on standard detection benchmarks. (ii) The additional aspect ratio regression delivers more accurate bounding boxes than standard object detection approaches in terms of overlap with ground truth, especially when tightening the evaluation criterion. (iii) The detector itself becomes better by only including the aspect ratio information during training.

1. Introduction

Object detection is one of the most important tasks in computer vision. Modern object detectors have to be both accurate and fast as they are often employed in time-critical applications, such as gaming or robotics. Moreover, they also act as a building block in various other applications like semantic segmentation [19, 28] or scene recognition [24].

In recent years, the progress in this field has been tremendous. Popular fast and accurate detection approaches can be roughly subdivided into three different strands: First, the works based on the rigid Dalal and Triggs [10] detector using HOG features and SVMs. In particular, the Deformable Parts Model [15] extends the rigid detector with a part-based multi-component model and achieves state-of-the-art results on many benchmarks. Second, object detectors building on the work of Viola and Jones [29] using Boosting and various feature channels [5, 12, 13]. These rigid detectors are less flexible in terms of object outlines, but achieve state-of-the-art results on pedestrian detection benchmarks and run in real-time. Third, detection models operating over small patches that vote for object centers with the generalized Hough transform [16, 22]. These detectors are very flexible and powerful in detecting body parts [26] or fiducial points in faces [11] but are less successful on common object detection benchmarks.

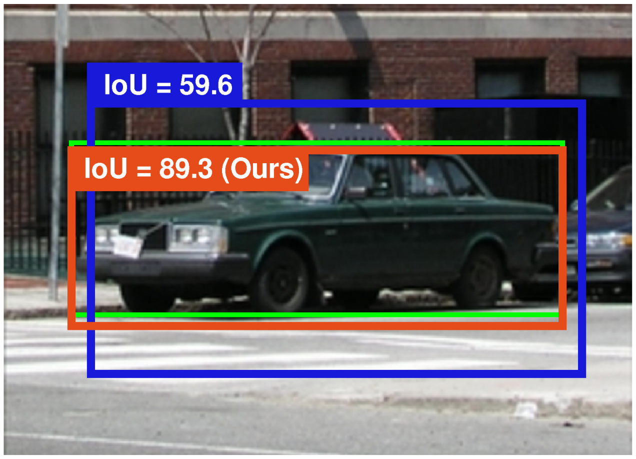


Figure 1: Illustrative output of the proposed detector (red) and a state-of-the-art method (blue). The given values show the overlap with the green ground truth bounding box, where our detector achieves a much higher overlap because it regresses the aspect ratio of the object.

The quantitative evaluation of all these detection models typically focuses on the amount of true and false positive detections in a certain data set. True positives are commonly characterized by a predefined overlap (50% in most benchmarks) of the detection with the ground truth bounding box. However, the question arises if a criterion based on such an arbitrarily chosen threshold actually reflects the quality of an object detector. Consider for instance the blue bounding box in Figure 1, a detection given by a template-based detector similar to [5, 12, 13], which is a true positive but has an overlap of only 59% with the green ground truth bounding box. Thus, tightening the overlap criterion would drastically decrease the overall performance of such a detector, as we show in our experiments.

In this work, we propose a novel object detection approach capable of predicting more accurate bounding boxes with a Joint Classification-Regression Random Forest formulation similar to [16, 17]. We exploit the fact that RFs, can predict arbitrary output spaces, *cf.* [14, 16, 18], and augment the label space for object detection with the bounding box size, which we exploit during both training and testing. In this way, we cannot only predict the foreground probability of a detection but can also regress the extent of the object with a single model, alleviating the need for learning many mixture models [15]. An illustrative result is shown in Figure 1, where the red bounding box is the output of our detection algorithm, which has a much higher overlap of 89% with the ground truth, thus identifying the extent of the object more accurately.

In our experiments, we demonstrate that our object detection approach yields state-of-the-art results on several data sets and compare it with related approaches like Hough Forests [16], DPM [15] and a Boosting-based rigid template approach [5, 12, 13]. More importantly, we also show that our approach can accurately regress the bounding box aspect ratio of objects in unseen test images, which is also reflected in the results. Namely, the detection performance of most typical detectors breaks down when increasing the overlap criterion for true positives, while our approach still gives comparably good results.

2. Related Work

The most related approaches to ours are the works of Dollár *et al.* [12, 13] and Benenson *et al.* [5], who build on a similar detection pipeline and employ the same features. The influential work on Integral Channel Features [13] computes several feature channels, including color, gradient magnitude and orientation quantized gradients, which is similar to [5]. Both works rely on an efficient Boosting framework for learning the object models. This line of work almost exclusively focuses on pedestrian detection and there are some extensions that make this framework extremely fast [4] or exploit application-specific knowledge [3]. However, only fixed size bounding boxes are predicted, which is reasonable for pedestrians, but is a limitation if dealing with arbitrary objects.

The Deformable Parts Model (DPM) [15] extends the rigid HOG template and SVM approach of [10] and includes deformable parts and multiple components. This mixture model captures the intra-class variability by separating the training data according to the aspect ratio (newer versions also include appearance), thus enabling the prediction of a discrete set of aspect ratios. Furthermore, a linear regression on the inferred part locations refines the aspect ratio prediction. Nevertheless, the DPM has some disadvantages because (i) different models have to be trained for each component and (ii) the aspect ratio prediction per component is limited on a linear model solely based on the part locations. In particular, the first issue limits the DPM in two ways: First, the model becomes slower during both training and testing, and second, the more components are employed, the less training data is available for each of them. In contrast, we have a single model that can exploit all the training data and can predict a continuous aspect ratio.

Blaschko and Lampert [7] formulate the detection as a regression problem and train a structured output SVM for learning. While yielding accurate results, a fast localization method is necessary to have reasonable runtimes during both training and detection. They use Efficient Subwindow Search, which requires computing an upper bound on the detection score, thus limiting the choice of features and learning method [20]. Our work is more flexible in the choice of features, faster during training and also has a reasonable runtime within a sliding window scheme.

We also note that RFs, in general, have rarely been employed for object detection. One exception is the Hough Forest framework [16], which describes an object as a set of small patches that are connected to a reference point, typically the center of the object. However, this patch-based approach is relatively slow compared to other detection models. To overcome this issue, [27] proposes a two-level approach for speeding up the detection process. Nevertheless, it still relies on the Hough voting scheme for the final prediction, where the non maximum suppression is a delicate task, *cf.* [30]. While Hough Forests [16] typically predict a fixed bounding box, it can also handle variable aspect ratios; either via back-projecting the voting elements, which then define the bounding box, or via voting in a third dimension in Hough space. However, employing the back-projection is rather slow or memory intensive, while increasing the Hough space dimensionality hampers the maximum search.

Recently, [23] showed a holistic RF model that trains local experts (SVMs) in each node. However, this model also builds on a fixed bounding box prediction and was only evaluated on pedestrian detection benchmarks.

3. Joint Classification-Regression Forests

To derive our Joint Classification-Regression Forest (JCRF) formulation, we first give a brief review of standard Random Forests (RF) in Section 3.1 and of the basic object detection model in Section 3.2. Then, we show how the label space is augmented for predicting variable bounding box aspect ratios in Section 3.3. Finally, the training and detection procedures of JCRF are illustrated in Sections 3.4 and 3.5, respectively.

3.1. Random Forests

Random Forests [1, 8, 9] (RF) are ensembles of T binary decision trees $f_t(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}^n$ is the n-dimensional feature space and $\mathcal{Y} = \{0, 1\}$ describes the label space[1].

During testing, each decision tree returns a class probability distribution $p_t(y|\mathbf{x})$ for a given test sample $\mathbf{x}$ and the final class label y^* is calculated via

$$y^* = \arg\max_y \frac{1}{T} \sum_{t=1}^{T} p_t(y|\mathbf{x}) \ . \tag{1}$$

During training, the decision trees are provided with training data $\mathcal{T} = (\mathbf{x}_i, y_i)_{i=1}^N$, where N is the number of training examples, and all trees are trained independently from each other. For training a single decision tree, the parameters Θ of a splitting function

$$\Phi(\mathbf{x}, \Theta) = \begin{cases} 0 & \text{if } r_\Theta(\mathbf{x}) < 0 \\ 1 & \text{otherwise} \end{cases} \ , \tag{2}$$

which separates the data into two disjoint sets, have to be estimated. In Equation (2), $r_\Theta(\mathbf{x}) \rightarrow \mathbb{R}$ calculates a response of the feature vector $\mathbf{x}$. The quality of a given splitting function Φ is typically defined as

$$\mathrm{I}(\Theta) = \frac{|L|}{|L| + |R|} H(L) + \frac{|R|}{|L| + |R|} H(R) \ , \tag{3}$$

where $L = \{\mathbf{x} : \Phi(\mathbf{x}, \Theta) = 0\}$, $R = \{\mathbf{x} : \Phi(\mathbf{x}, \Theta) = 1\}$, $|\cdot|$ denotes the size of a set, and $H(\cdot)$ measures the purity of a set of training examples in terms of class labels. The purity $H(\cdot)$ is typically calculated via the entropy or the Gini index [8].

The standard procedure in Random Forests for finding a good splitting function in a single node is to randomly sample a set of parameters $\{\Theta_j\}_{j=1}^k$ and simply choosing the best one, Θ^*, by evaluating Equation (3). This splitting function is then fixed, the training data is separated accordingly and the tree growing continues until some stopping criteria, such as a maximum tree depth or a minimum number of samples in the node, are reached.

3.2. Object Detection Model

The training data for our detection model is a set of rigid templates $\mathcal{P}$ with size $\bar{h} \times \bar{w}$, which is the mean bounding box size of all the objects in the training set, normalized to 100 pixel width or height, depending on what is larger[2]. As in [12], the size of this template also includes a padding of 20% in order to capture some context around the objects. We center such a template at each of the annotated training objects and on randomly sampled bounding boxes from negative images. For each of the cropped training examples, we calculate 10 different feature channels: We use the 3 LUV color channels, the gradient magnitude, and 6 gradient channels, quantized into equally sized orientation bins, similar to [5, 12, 13].

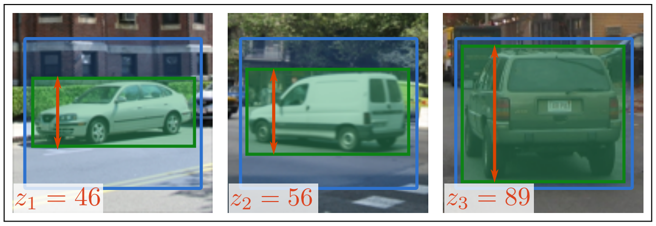


Figure 2: Training samples capturing objects with different height. Features are computed from the whole template (blue box), but the regression targets z_i are different.

The training data $\mathcal{T}$, *i.e.*, positive and negative samples, is thus given as a set of pairs $(\mathbf{x}_i, y_i)_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^{\bar{h} \times \bar{w} \times 10}$ and $y_i \in \{0, 1\}$. Given this data, we train a Random Forest $\mathcal{F}(\mathbf{x}) = \frac{1}{T} f_t(\mathbf{x})$ with the standard objective given in Equation (3). As splitting functions, we use pixel-pair test, *cf.* [16], and thus define

$$r_\Theta(\mathbf{x}) = \mathbf{x}(\Theta_l^1, \Theta_C) - \mathbf{x}(\Theta_l^2, \Theta_C) - \Theta_\sigma \ , \tag{4}$$

where Θ_l^1 and Θ_l^2 describe two 2D locations within the template $\mathcal{P}$, $\Theta_C \in \{1, \ldots, 10\}$ defines a feature channel and $\Theta_\sigma \in \mathbb{R}$ is a threshold. Please note that in the following sections, we will extend the label space and also provide a different objective function for training the model.

We also include three rounds of bootstrapping after the initial training of the Random Forest. In each round, a set of hard negative windows is identified by applying the current model on the negative images, which are then added to the pool of negative data. In each round we re-train the Random Forest from scratch.

3.3. Augmenting the Label Space

As described in the previous section, each training example is cropped and scaled such that it fits in the template $\mathcal{P}$ of size $\bar{h} \times \bar{w}$. The scaling factor is defined by the fixed model width. Thus, the actual height of the objects captured in the training images most likely varies with the viewpoint of the object. See Figure 2 for some examples. In order to give predictions about the correct height, and thus the aspect ratio of an object, we additionally store the actual height z for each of the training examples.

Therefore, we augment the label space with the ground truth height of each positive training example, which extends the label space to $\mathcal{Y} = \{0, 1\} \times \mathbb{R}$. Our training set now becomes a set of triplets $(\mathbf{x}_i, y_i, z_i)_{i=1}^N$, where

[1] Please note that we only consider the binary case here as our application is binary object detection. However, RF are inherently multi-class.

[2] Throughout the paper, we will always refer to a fixed-width model.

$y_i \in \{0, 1\}$ still corresponds to the positive and negative label, and $z_i \in \mathbb{R}$ is the correct bounding box height. Please note that for background training samples, *i.e.*, $y^i = 0$, z_i is undefined.

3.4. Training the Random Forest Model

For training our Joint Classification-Regression Forest (JCRF), we have two objectives. First, we want to separate positive and negative classes and, second, we want to regress the bounding box height for positive samples. Similar to Hough Forests [16], we use two separate split node types that optimize the different objective functions. The first node type is a (binary) classification node and the second is a regression node.

For *classification nodes*, we use the recently proposed Alternating Decision Forests (ADF) [25] that show how a well defined loss function can be optimized over all trees, while still being able to parallelize the training. For employing ADF, we assign each training sample $\mathbf{x}_i$ a corresponding weight w_i, similar to Boosting. Then, the trees are trained breadth-first and after training each level of depth, the weights are updated according to the respective loss, which was calculated over all trees. The purity measure $H(\cdot)$ in Equation (3) is still the same, *i.e.*, the entropy.

For *regression nodes*, we employ a standard reduction-in-variance approach, where the purity measure $H(\cdot)$ is thus defined as the variance of the target values z_i in the corresponding sets. Please note that the regression objective is only evaluated with the positive training data.

In general, a single splitting node decides randomly with equal probability which of the two objectives are being optimized. As in Hough Forests [16], we also assign certain levels of depth in the tree a fixed type of evaluation objective that has to be optimized. We thus introduce the steering parameter γ which has different interpretations: While setting $\gamma = 0$ ignores the regression objective at all, setting $\gamma > 0$ indicates that starting with depth γ, only regression nodes are evaluated in all trees, similar to [16]. In this work, we additionally allow setting $\gamma < 0$, where only the first levels up to depth $|\gamma|$ are optimizing the regression objective.

Tree growing stops as in standard Random Forests when either the maximum tree depth is reached or not enough training examples are available for further splitting. In contrast to standard Random Forests, where tree growing also stops if a certain node becomes pure in terms of class labels, in this setting, we continue splitting nodes containing only positives but fix the splitting objective to be regression.

The resulting leaf nodes then calculate (i) the class histogram based on the training data falling in that leaf and (ii) the mean of the regression targets z of all positive training examples. As thus each tree can return two kinds of outputs, we denote by $f_t^C(\mathbf{x})$ the classification output of tree t and by $f_t^R(\mathbf{x})$ the regression output.

3.5. Detection with Aspect Ratio Regression

For detecting objects in unseen test images, we employ a standard sliding window approach over the scale-space. The score of a window $\mathcal{W}$ at location (x, y) in the image is given by the classification output of the RF $s = \mathcal{F}^C(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} f_t^C(\mathbf{x})$. As we use an ensemble method which consists of several independent weak classifiers (randomized trees in this case), we could parallelize their evaluation to achieve a higher detection speed. However, given a certain detection threshold τ below which detections are discarded, we can also iteratively evaluate the trees and employ an early-stopping scheme. Moreover, we can still benefit from parallel processing, *e.g.*, at the scale space pyramid or for the simultaneous evaluation of multiple images.

In our early stopping approach, we examine whether or not the trees in the RF not evaluated up to now can theoretically achieve such a high foreground probability that the total score for that window $\mathcal{W}$ exceeds the detection threshold τ. Assume that we already evaluated the trees up to index $t < T$, the current unnormalized score thus is $s_{i \leq t} = \sum_{i=1}^{t} f_i^C(\mathbf{x})$. The upper bound of the score from the remaining trees is $\bar{s}_{i>t} = \sum_{i=t+1}^{T} 1.0 = T - t$. Therefore, if

$$s_{i \leq t} + \bar{s}_{i>t} < \tau \cdot T \ , \tag{5}$$

we can already stop evaluating the feature vector $\mathbf{x}$ in the current window $\mathcal{W}$. For instance, having $T = 10$ and a detection threshold $\tau = 0.95$, the evaluation can already stop if the first tree has a foreground probability $f_1^C(\mathbf{x}) < 0.5$. Using this approach, we can reject clear negative windows during the evaluation process very quickly without reducing detection performance.

For each of window with $s \geq \tau$ we evaluate $f_t^R(\mathbf{x})$ for each tree in the JCRF to return the prediction about the regression target, *i.e.*, the height of the object captured in the current window. The final estimate z of the object height is given as the average over all independent trees:

$$z = \mathcal{F}^R(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} f_t^R(\mathbf{x}) \ . \tag{6}$$

Please note that a mode seeking approach like mean shift could also be employed, but averaging turned out to be a good choice in our setting. The resulting detection window including the detection score in the original scale is then given by $\mathcal{D} = (\frac{x}{\kappa}, \frac{y}{\kappa}, \frac{w=100}{\kappa}, \frac{z}{\kappa}, s)$, where κ is the scale of the detection.

After having identified a set of potential detections for a test image, we apply a standard greedy non-maximum suppression approach that removes detections having an overlap greater than 50% with a higher scoring detection [15].

4. Experiments

In our experiments, we demonstrate the performance of our object detection approach. First, we compare with state-of-the-art methods on three different data sets with a standard object detection evaluation criterion. Second, we also evaluate the detection performance of all methods when this evaluation criterion is tightened. Finally, we analyze our trained model and the most relevant parameters.

4.1. Overall Performance Evaluation

We first evaluate the overall detection performance of the proposed approach on three standard benchmarks. We investigate three different variants of our approach: *StdRF* implements a standard Random Forest disregarding the regression information at all. *StdRF-Regr* trains a RF and includes the regression information during both training and testing. *ADF-Regr* employs the training scheme from [25] for classification nodes, see Section 3.4.

In addition, we give a comparison to state-of-the-art detection approaches. First, we evaluate Aggregate Channel Features (ACF) [12], a Boosting-based approach similar to [5, 13] that builds on the same detection pipeline (*i.e.*, the same features and bootstrapping scheme) as our detector. Second, we also compare with Hough Forests [16] (*Hough-Forest*) that also rely on a Random Forest framework for learning the object model, however, it works on the patch-level and employs the generalized Hough voting scheme for detection. Please note that both approaches only predict a single bounding box aspect ratio, which is averaged over the training data. Finally, we also compare with the Deformable Parts Model [15] (*DPMfull*), where we additionally evaluate a version that only uses the root filter *DPM-root* to have a fair comparison with the other approaches building on a rigid template. However, more important for our scenario are the multiple components included in this model, which are defined by clustering the aspect ratio of the training bounding boxes. To denote the different versions of [15] we add the number of components (1, 2 or 4) as postfix to *DPMroot* or *DPMfull*, respectively. For all approaches building on randomization steps, *i.e.*, [16] and our variants, we average the results over three independent runs.

Data sets: We use three different data sets for evaluation purpose, namely the ETHZcars [21], the TUDpedestrian [2] and the MITStreetsceneCars [6]. For the *ETHZcars* data set, we use 420 training and 175 testing images that capture cars from different viewing angles. Although the test set defines a rather easy detection task, because it shows cars prominently with little background, it exactly fits the needs for evaluating the quality of the bounding box detections. Namely, the aspect ratio of the ground truth bounding boxes in the test set strongly varies. The *TUDpedestrian* data set contains 400 training images and 250 test images. Also in this scenario, the aspect ratio varies due to different articulations. Finally, the *MITStreetsceneCars* data set is a larger data set capturing cars in different street scene scenarios and under different illumination. We split this data set in $\frac{2}{3}$ training and $\frac{1}{3}$ testing images, resulting in 2909 and 1020 images, respectively.

Evaluation Criterion: The evaluation criterion in this experiment is the commonly used Pascal overlap. For each detection $\mathcal{D}$ it calculates the overlap with a ground truth bounding box $\mathcal{G}$ as the Intersection over Union:

$$\text{IoU}(\mathcal{D}, \mathcal{G}) = \frac{\mathcal{D} \cap \mathcal{G}}{\mathcal{D} \cup \mathcal{G}} \, . \tag{7}$$

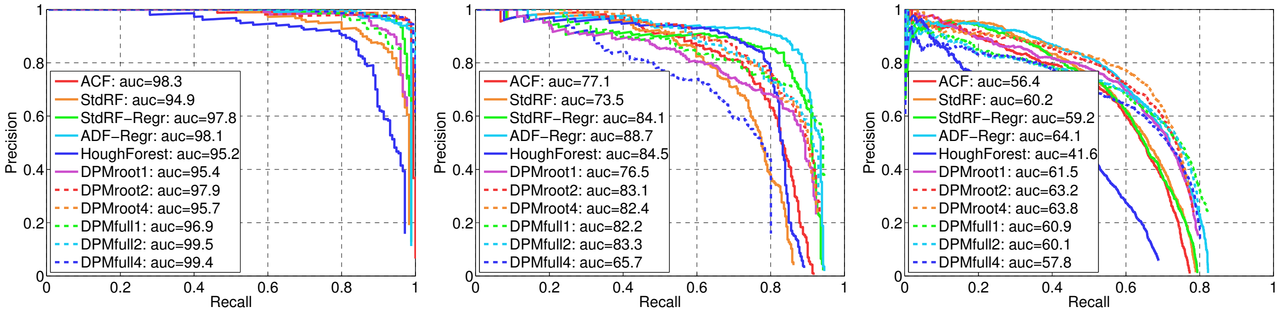
The $\text{IoU}(\mathcal{D}, \mathcal{G})$ overlap separates all detections $\mathcal{D}$ in true ($\text{IoU} \geq \xi$) or false ($\text{IoU} < \xi$) positives, which are used for drawing precision-recall curves and calculating the Area Under Curve (AUC) measure. The parameter ξ is the success threshold that is typically, and also in this experiment, set to 0.5. Note that in the following section we evaluate the detection results when setting $\xi > 0.5$.

Results: We depict our results as precision-recall curves for all data sets in Figure 3 and report the AUC values in the legend. As can be seen, the proposed *ADF-Regr* is always en par with the best performing approach and clearly wins on one of the data sets. We also note that *ADF-Regr* is typically better than *ACF*, except for the ETHZcars where the difference is insignificant, and the results are rather saturated. On the *TUDpedestrian* and the *MITStreetsceneCars*, our approach is 11.7% and 7.7% better than *ACF*, respectively. Furthermore, our approach performs significantly better than Hough Forests, the only other Random Forest based method. Finally, we also note that *ADF-Regr* outperforms all versions of DPM, except for *DPMfull2* and *DPMfull4* on the *ETHZcars* data set, and that including the parts in the DPM does not always improve the results on these data sets.

Comparing the different proposed variants, *i.e.*, *StdRF*, *StdRF-Regr* and *ADF-Regr*, we see that including the regression output typically gives significantly better performance and that the ADF learning scheme [25] further improves the results.

4.2. Tightening the Pascal Overlap Criterion

In this experiment, we directly evaluate the quality of the bounding box predictions of the different methods and investigate the performance of our joint classification-regression prediction in more detail. To do so, we use a different type of evaluation curve. Following the standard Pascal criterion, we draw precision-recall curves and measure the AUC. However, we vary the success threshold ξ for a true positive detection, see Equation 7. We then plot the AUC value over ξ in the range between 0.5 to 1.0. Except from the evaluation criterion, the experimental setup is the same as in the previous experiment.



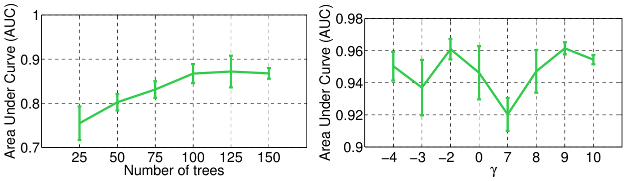
(a) (b) (c)

Figure 3: Precision recall curves for all evaluated approaches on three different data sets: (a) *ETHZcars*, (b) *TUDpedestrian* and (c) *MITStreetsceneCars*.

Results: We illustrate our results for all three data sets in Figure 4. As can be seen, the proposed *ADF-Regr* gives the best results on two benchmarks and is en par with DPM (2 or 4 components) on one benchmark. The performance difference between all methods is most pronounced on the *ETHZcars* data set (Figure 4a), which shows most aspect ratio variations. We can see that *HoughForest*, *ACF*, *StdRF*, *DPMroot1* and also *DPMfull1* drastically lose performance if the success threshold ξ gets increased. For instance, setting $\xi = 0.7$, none of these approaches achieve higher AUC than 50%. Using DPM with 2 or 4 components (regardless of the parts) increases the performance to 62% and 80%, respectively. This is intuitive as increasing the number of components also increases the number of aspect ratios that can be predicted. However, further adding components will likely decrease the performance as less training examples will be available per component (can be observed on *TUDped* and *MITStreetsceneCars*). In contrast, our Joint Classification-Regression Random Forest formulations, *StdRF-Regr* and *ADF-Regr*, can predict the bounding boxes even more accurately, improving over *DPMroot4* by 6% for $\xi = 0.7$ and by 41% for $\xi = 0.8$. Our best variant, *ADF-Regr* achieves an AUC value of 73% at the very tight success threshold of $\xi = 0.8$, while all methods predicting a fixed bounding box do not exceed an AUC value of 20%. We illustrate some qualitative results in Figure 5.

4.3. Analysis of the Random Forest Model

In this section, we analyze the most relevant parameters of our RF framework. We evaluate the number of trees T, as well as the parameter γ that regulates the amount of regression nodes evaluated during training (for *StdRF-Regr* and *ADF-Regr*). Furthermore, we also investigate the feature selection process of Random Forests when including the regression objective during the training phase.



(a) (b)

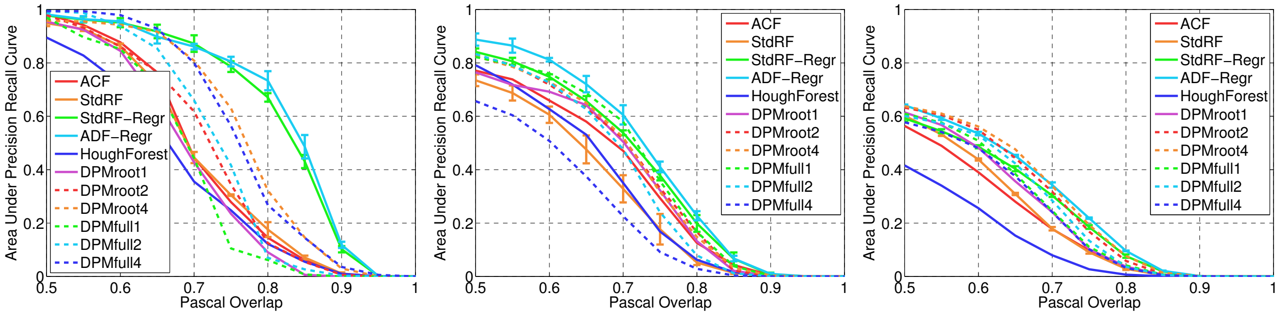
Figure 6: (a) Evaluation of the number of trees T on the TUD-pedestrian data set. (b) Evaluation of the parameter γ that influences the behavior of the regression objective in the RF. We plot the accuracy as (AUC) for different values of γ for the ETHZcars data set.

Number of trees: First, we evaluate the number of trees T when fixing the maximum tree depth to 12 on the *TUD-pedestrian* data set. In this setting, the additional regression is also turned on and the parameter γ is set to -2. The results are depicted in Figure 6a. As expected, we can see a clear trend of increasing performance with the number of trees T up to a certain limit $T = 100$. For more trees, the results are saturated.

Regression-Only Parameter: We further analyze the parameter γ. For this evaluation, we use the *ETHZcars* data set, which provides the most variation in aspect ratios. In Figure 6b, we see the mean and standard deviation of 3 independent runs for different values of γ. We can observe that using too many or too few regression nodes is unfavorable. Best performance can be observed by setting γ either to -2 or 9 (when using trees with maximum depth of 12).

Feature Selection: We also visualize the different feature selection processes for *StdRF-Regr* and *StdRF*, *i.e.*, with or without including the regression objective, on the car model of the *ETHZcars* data set. To do so, we count how often a certain location in the rigid template was selected for performing a split in the Random Forest, regardless of the feature channel. Again, we set $\gamma = -2$ for *StdRF-Regr*.

Figure 7 illustrates the behavior for both training schemes when summing up all selected feature locations; In this setting, we used 100 trees with a maximum tree depth of 12 over 2 independent runs. Interestingly, we can observe



(a) (b) (c)

Figure 4: Area under the Curve (AUC) for increasingly harder Intersection over Union success thresholds for different data sets: (a) *ETHZcars*, (b) *TUDpedestrian* and (c) *MITStreetsceneCars*.

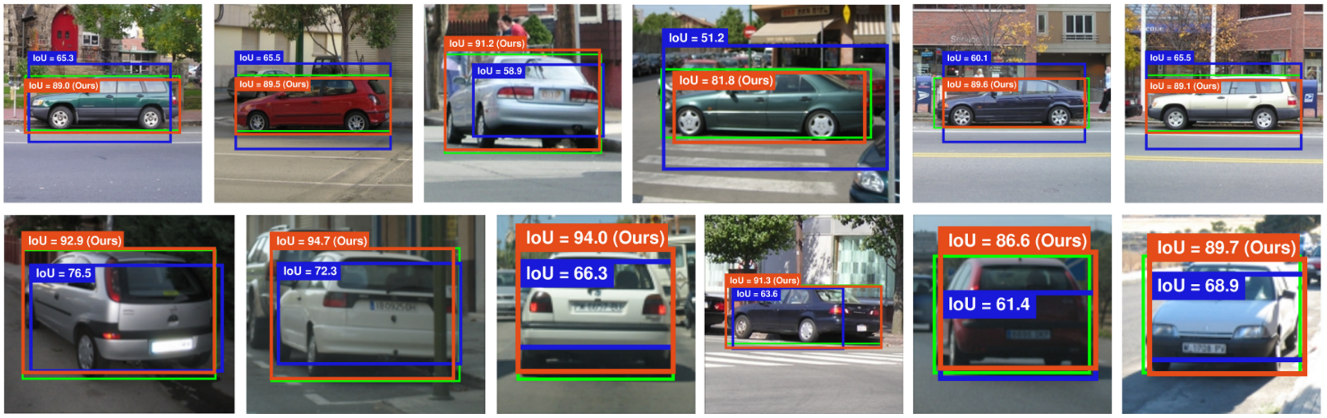


Figure 5: Example results when comparing a fixed bounding box prediction and a flexible one. The green box is the ground truth, the blue one is the fixed size bounding box, *i.e.*, the mean over the training examples, and the red one is the flexible bounding box predicted with our Joint Classification-Regression Forest.

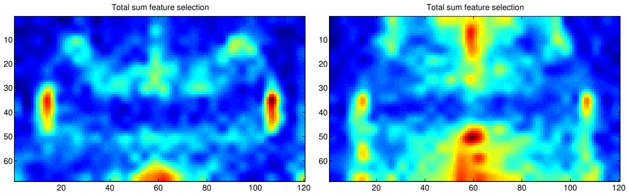
that the feature selection is very different for the two learning schemes. *StdRF* concentrates on positions on the left and right side of the cars, which is clear as the width of the model is fixed and, thus, the gradients at these locations are present in all training images. Furthermore, for this training scheme, also the bottom of the templates are frequently selected. This can be explained by the fact that cars typically stand on the street and also have gradients on the bottom, which is typically not the case in negative images.

On the other hand, *StdRF-Regr* evaluates a much more diverse set of locations as it also tries to separate the different viewing angles, *e.g.*, frontal-view from side-view cars. Thus, it selects the features at different y-coordinates in the center of the x-dimension. Of course, this training scheme also evaluates classification nodes, thus we can observe similarly selected locations from *StdRF* as well.

Classification model: Finally, we also evaluate if the regression objective during training also improves the classification performance of the RF. That is, we train two models, one including the regression objective ($\gamma = -2$) and one only evaluating classification nodes ($\gamma = 0$). During testing, however, we only use the fixed mean bounding box in order to turn off the effect of regressing the bounding box height on the final AUC performance. We observe that including regression during training improves the performance regardless if the regression output is actually used during detection. On the *TUDpedestrian* data set, we get $78.3 \pm 1.2\%$ AUC without using the regression objective ($\gamma = 0$) and $83.3 \pm 3.1\%$ when including it ($\gamma = -2$). Including the regression output during the testing phase for predicting variable bounding box aspect ratios further improves the results, *cf.*, Section 4.1.

5. Conclusion

In this work, we proposed a Random Forest based object detection model that is capable of predicting variable bounding box aspect ratios. We augmented the standard binary label space accordingly with a regression target for predicting the bounding box aspect ratio. Our Joint Classification-Regression Forest (JCRF) formulation exploits the additional label information during both train-



(a) (b)

Figure 7: Feature selection of both training schemes: (a) Only classification and (b) including regression.

ing and testing. For training JCRF, we employ the ADF training scheme and include regression nodes for optimizing the bounding box aspect ratio.

Our results on common object detection benchmarks showed that our proposed detection model is better or en par with related state-of-the-art approaches. Furthermore, our experiments revealed that most commonly used object detectors that predict a fixed bounding box size break down as soon as the evaluation criterion becomes tighter in terms of overlap with the ground truth bounding box. In contrast, our JCRF formulation can efficiently deal with variable aspect ratios in a single model and achieves good results even if the overlap criterion gets harder.

Acknowledgement: This work was supported by the Austrian Science Foundation (FWF) project Advanced Learning for Tracking and Detection in Medical Workow Analysis (I535-N23)

References

[1] Y. Amit and D. Geman. Shape Quantization and Recognition with Randomized Trees. *NECO*, 9(7):1545–1588, 1997. 3

[2] M. Andriluka, S. Roth, and B. Schiele. People-Tracking-by-Detection and People-Detection-by-Tracking. In *CVPR*, 2008. 5

[3] R. Benenson, M. Mathias, R. Timofte, and L. v. Gool. Fast stixel computation for fast pedestrian detection. In *ECCV Workshops*, 2012. 2

[4] R. Benenson, M. Mathias, R. Timofte, and L. v. Gool. Pedestrian detection at 100 frames per second. In *CVPR*, 2012. 2

[5] R. Benenson, M. Mathias, T. Tuytelaars, and L. v. Gool. Seeking the strongest rigid detector. In *CVPR*, 2013. 1, 2, 3, 5

[6] S. M. Bileschi. *StreetScenes: Towards Scene Understanding in Still Images*. PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2006. 5

[7] M. B. Blaschko and C. H. Lampert. Learning to Localize Objects with Structured Output Regression. In *ECCV*, 2008. 2

[8] L. Breiman. Random Forests. *ML*, 45(1):5–32, 2001. 3

[9] A. Criminsi and J. Shotton. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer, 2013. 3

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1, 2

[11] M. Danone, J. Gall, G. Fanelli, and L. v. Gool. Real-time Facial Feature Detection using Conditional Regression Forests. In *CVPR*, 2012. 1

[12] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast Feature Pyramids for Object Detection. *PAMI*, 2014. 1, 2, 3, 5

[13] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral Channel Features. In *BMVC*, 2009. 1, 2, 3, 5

[14] P. Dollár and L. Zitnick. Structured Forests for Fast Edge Detection. In *ICCV*, 2013. 2

[15] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2010. 1, 2, 4, 5

[16] J. Gall and V. Lempitsky. Class-Specific Hough Forests for Object Detection. In *CVPR*, 2009. 1, 2, 3, 4, 5

[17] B. Glocker, O. Pauly, E. Konukoglu, and A. Criminisi. Joint Classification-Regression Forests for Spatially Structured Multi-Object Segmentation. In *ECCV*, 2012. 2

[18] P. Kontschieder, S. Rota Bulò, H. Bischof, and M. Pelillo. Structured Class-Labels in Random Forests for Semantic Image Labelling. In *ICCV*, 2011. 2

[19] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr. What,Where & How Many? Combining Object Detectors and CRFs. In *ECCV*, 2010. 1

[20] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. In *CVPR*, 2008. 2

[21] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3D Scene Analysis from a Moving Vehicle. In *CVPR*, 2007. 5

[22] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshops*, 2004. 1

[23] J. Marín, D. Vázquez, A. M. López, J. Amores, and B. Leibe. Random Forests of Local Experts for Pedestrian Detection. In *ICCV*, 2013. 2

[24] M. Pandey and S. Lazebnik. Scene Recognition and Weakly Supervised Object Localization with Deformable Part-Based Models. In *ICCV*, 2011. 1

[25] S. Schulter, P. Wohlhart, C. Leistner, A. Saffari, P. M. Roth, and H. Bischof. Alternating Decision Forests. In *CVPR*, 2013. 4, 5

[26] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-Time Human Pose Recognition in Parts from a Single Depth Image. In *CVPR*, 2011. 1

[27] D. Tang, Y. Liu, and T.-K. Kim. Fast Pedestrian Detection by Cascaded Random Forest with Dominant Orientation Templates. In *BMVC*, 2012. 2

[28] V. Vineet, J. Warrell, L. Ladicky, and P. H. S. Torr. Human Instance Segmentation from Video using Detector-based Conditional Random Fields. In *BMVC*, 2011. 1

[29] P. Viola and M. Jones. Robust Real-time Object Detection. *IJCV*, 57(2):137–154, 2004. 1

[30] P. Wohlhart, M. Donoser, P. M. Roth, and H. Bischof. Detecting Partially Occluded Objects with an Implicit Shape Model Random Field. In *ACCV*, 2012. 2