

To Boost or Not to Boost? On the Limits of Boosted Trees for Object Detection

Eshed Ohn-Bar and Mohan M. Trivedi
 Computer Vision and Robotics Research Laboratory
 University of California San Diego
 La Jolla, CA 92092
 {eohnbar, mtrivedi}@ucsd.edu

Abstract—We aim to study the modeling limitations of the commonly employed boosted decision trees classifier. Inspired by the success of large, data-hungry visual recognition models (e.g. deep convolutional neural networks), this paper focuses on the relationship between modeling capacity of the weak learners, dataset size, and dataset properties. A set of novel experiments on the Caltech Pedestrian Detection benchmark results in the best known performance among non-CNN techniques while operating at fast run-time speed. Furthermore, the performance is on par with deep architectures (9.71% log-average miss rate), while using only HOG+LUV channels as features. The conclusions from this study are shown to generalize over different object detection domains as demonstrated on the Fddb face detection benchmark (93.37% accuracy). Despite the impressive performance, this study reveals the limited modeling capacity of the common boosted trees model, motivating a need for architectural changes in order to compete with multi-level and very deep architectures.

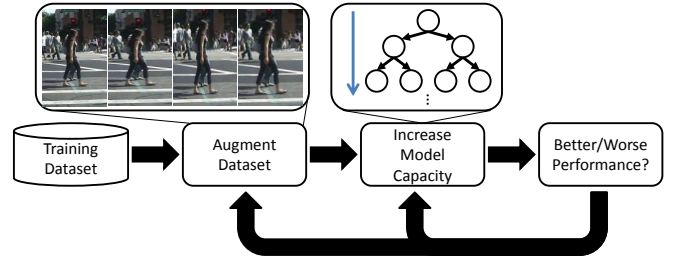


Fig. 1: By varying dataset properties with different augmentation techniques (a scaling example is shown) and model capacity (tree depth), we study limitations of the boosted decision tree classifier. Consequently, the insights revealed are employed in order to train state-of-the-art pedestrian and face detectors.

I. INTRODUCTION

The increase in data availability and computational power have led to a revolution in object recognition. Modern object recognition algorithms combine deep multi-layer architectures with large datasets in order to achieve state-of-the-art recognition performance [1]–[3]. A main advantage of such learning architectures stems from the architectural ability to increase modeling capacity (e.g. network parameters and depth) in a straightforward manner. At the same time, boosted detectors [4]–[6] remain highly successful for fast detection of objects [7]–[21]. The Viola and Jones [4] learning architecture has remained largely unchanged, with boosting [22] used for training a cascade of weak learners (e.g. decision trees). State-of-the-art pedestrian detectors often employ such models [15], while pursuing improved feature representations in order to achieve detection performance gains [9], [16]. Despite being a different learning architecture compared to the modern Convolutional Network (CNN) [1], the boosted decision tree model also allows for a straightforward increase in modeling capacity for handling large datasets (Fig. 1). Motivated by this observation, we perform an extensive set of experiments designed to better understand the limitations of the commonly employed boosted decision trees model. The large Caltech pedestrian benchmark [6] is suitable for such a task, with some preliminary work [16], [23] showing significant performance gains by increasing dataset size and modeling capacity of the weak learners.

In particular, the contributions of this work are as follows.

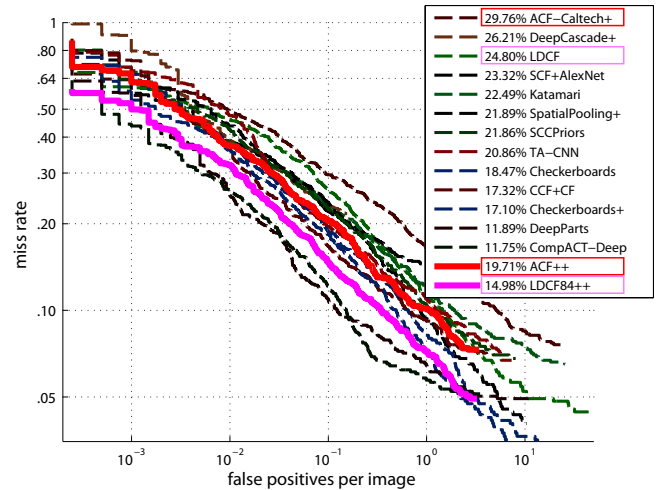


Fig. 2: Comparison of our key results (in thick lines, ACF++ and LDCF84++) with published methods on the Caltech test set. Our version of the popular ACF and LDCF models demonstrate large gains in detection performance over the baselines with no significant modifications to the model or feature extraction pipeline.

Limitations of a classifier: The relationship between increasing the modeling capacity of the decision trees jointly

with dataset size has been established in previous literature [16], [22], [24], yet its limits have not been fully explored. Surveying current literature, we found inconsistent or limited usage of this relationship in boosting for object detection. In particular, it is not known whether the modeling capacity has saturated or whether it can be further increased with additional data for the detection task. Therefore, we sought to perform a rigorous study with the relationship between dataset size and model capacity as the main research question. In the process of studying this problem, insights regarding the limitations of the model arise. The conclusion of this study is mostly negative, as increasing the modeling capacity of the weak learners in the current boosted decision tree detector provides limited generalization power. Saturation of performance occurs quickly, with no further gains shown with additional data augmentation. Therefore, architectural changes are needed in order to resolve the issue of efficiently increasing modeling capacity in the boosted tree classifier.

Performance gains: The experiments in this paper result in consistent gains in object detection performance, insights into optimal dataset properties, and best training practices on Caltech. For instance, we demonstrate that the commonly employed video-based augmentation [16], [23] on the Caltech training set is limited when compared to other augmentation options. With no significant modifications to the learning or feature extraction pipelines, we improve over the Aggregate Channel Feature (ACF) detector [15] baseline by $\sim 10\%$ points while still employing the 10 HOG+LUV feature channels. A simple filtering of the channels with 4 filter banks (LDCF [16]) results in the best performing non-CNN detector on Caltech to date. The results of the study are shown to generalize to face detection as well. Note that we intentionally avoid using deep CNN features here, as these often involve training on a large external data. On the other hand, simple channel features [15] allow for isolation of the strengths, limitations, and the role of the dataset on the boosting classifier. Furthermore, the careful examination of data augmentation and model capacity allows for a more appropriate comparison between CNN (where extensive data augmentation is commonly employed) and non-CNN detectors on datasets such as Caltech. We hope the more meaningful comparison will motivate future developments in object detection. We also note that several recent CNN-based models on Caltech [25]–[27] make use of the boosted trees classifier.

II. EXPERIMENTAL SETTINGS

A. Boosting Framework

This section presents the tools that will be used for the remainder of the study. Boosting [22] involves greedily minimizing a loss function for the following classification rule

$$\mathbf{H}(\mathbf{x}) = \sum_t \alpha_t \mathbf{h}_t(\mathbf{x}) \quad (1)$$

where $\mathbf{h}_t(\mathbf{x})$ are weak learners, α_t are scalars, and $\mathbf{x} \in \mathbb{R}^K$ is a feature vector. In this work, we employ the commonly

used soft cascade with decision trees as the weak learners [28], [29]. These are composed of decision stumps such that each non-leaf node j produces a binary decision using a feature index k , a threshold $\tau \in \mathbb{R}$, and a polarity $p \in \{\pm 1\}$, $h_j(\mathbf{x}) \equiv p_j \text{sign}(\mathbf{x}[k_j] - \tau_j)$. An important parameter is the maximum tree depth, which was generally taken to be 2 for object detection, yet recent studies [16], [23], [30] propose depths of 3–4 as suitable to accommodate an increase in dataset size. Hence, our first issue is efficient training of boosted models on large datasets. The need for quick training is key to algorithmic development and competitive performance, especially when considering current state-of-the-art CNN object detections are trained on millions of samples.

Training with randomness: In this work, we adopt the quick boosting approach from [31]. Because stump training is costly, $\mathcal{O}(K \times N)$ for K features and N samples, a common practice is to quantize feature values into bins (256 in this work) and sample a random subset of the features when searching for the optimal feature index. Most approaches employ the ACF detector [15], which samples 1/16 of the total feature set in each iteration. We observed significant variation in performance over multiple runs with this random sampling strategy, by up to 3–4%. This is an issue, yet an appropriate discussion was not found in related literature. Therefore, we modify the last stage out of four of generating hard negatives with increasing number of weak classifiers, [64, 256, 1024, 4096]. In the last stage, the feature set is exhaustively searched over for the *initial 512 weak learners*, after which random sampling of 1/16 is followed. This procedure was found to be necessary for careful analysis, and it balances reproducibility with speed (unlike the very slow exhaustive search in all iterations [23]) reducing variability to within $\sim 1\%$ in performance. In experiments where significance is uncertain, multiple random seeds are used for training initialization, with the best one shown. Batch training is another possible direction which we leave for future work.

B. Augmentation Techniques

As previously mentioned in Section I, large gains in boosting-based pedestrian detection came from dataset size increase and increased depth of the decision tree (introduced in Nam *et al.* [16]), although the design choices for these have not been well justified or studied. Consequently, existing state-of-the-art detectors employ dense frame sampling (e.g. sampling every 3^{rd} frame resulting in a ten-fold increase in dataset size, Caltech $\times 10$), yet this type of video-based augmentation will be shown to present several issues leading to sub-optimal detector performance.

Identifying the most successful augmentation techniques is essential, as the increased dataset size regularizes training of models with increased capacity. We note that others have studied data augmentation, specifically for pedestrian detection, but mostly with a goal of lowering annotation effort (i.e. render pedestrian images into real backgrounds [32], [33]) and not improved performance.

In addition to sampling more frames from the Caltech videos, the following augmentations are studied:

Color: Color alteration has been proposed in [1] using Principal Component Analysis (PCA) on the RGB pixel values in the training set. Specifically, a random Gaussian variable, $\alpha \in \mathbb{R}^3$ is drawn and the quantity $[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3][\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]$, where \mathbf{p}_i and λ_i are eigenvector-eigenvalue pairs, is added to the RGB image pixels.

Flipping: Most current trained detectors employ horizontal mirroring of images in order to double the number of positive samples.

Translation (T): Two main parameters, namely the maximum amount of pixel shift allowed (m) and the sampling intervals (n), determine how many additional positive samples are generated. We note these in the experiments as Tnm . For instance, T31 generates 4 additional samples at 1 pixel shift in each direction (west/east/north/south).

Scale+Crop (S): As experiments began, we discovered high sensitivity of the classifier to even small noise injected in the ground truth location. As an alternative to translation, we propose to scale each positive sample by a factor (either in the horizontal, vertical, or both directions), re-center, and crop it (shown in Fig. 1). Hence, the procedure is slightly different from the common cropping employed in training CNN detectors (corner/center cropping). This augmentation is similar to slight ground truth width/height jitter without changes to the center of the box.

Occlusion: Occlusion handling is an important challenge for pedestrian detection [34], [35]. Incorporation of samples with higher levels of occlusion is another technique for increasing dataset size and studying generalization capability.

FG/BG Transfer: The visibility masks of pedestrians in Caltech can be employed in order to transfer foreground/background between images in the dataset.

C. Model Settings

Our training settings specify a model size of 41×100 pixels. With padding, the sliding window becomes 64 by 128 pixels. In order to deal with smaller pedestrians (‘reasonable’ test settings involve pedestrians of height 50 pixels and up), we upsample the images by one octave. Most of the augmentation experiments will be performed on Caltech $\times 3$ (sampling every 10^{th} frame from video), but higher samplings of Caltech $\times 7.5$ (every 4^{th} frame), Caltech $\times 15$, and Caltech $\times 30$ (every frame) sampling will also be studied.

III. EXPERIMENTAL ANALYSIS

This section demonstrates the importance of data augmentation, impact of augmentation type, and significance of increased weak classifier model capacity in training a generalizable pedestrian detector. All results are measured in log-average miss rate (a lower value corresponds to better detection performance) as described in [6].

Baseline and notation: Our experiments begin with Fig. 3(a), where we first verify the impact of randomness with

multiple random seeds in training (see Section II-A) on performance variation using Caltech $\times 3$. The $L3N50$ model is the commonly trained maximum depth 3 model using the available implementation of [6] with 50k total negative samples (in each hard negative mining round we collect $N/2$ samples and replace these hard samples so that no more than N samples are trained over in a given round). We note that the only form of augmentation from Section II-B commonly employed in training boosting-based pedestrian detectors besides higher frame sampling from video is flip augmentation. Hence, flip augmentation is included in the $L3N50$ baseline. It achieves comparable results to the best known ACF results on Caltech of 29.76% log-average miss rate (Fig. 2).

Flip augmentation: Surprisingly, Fig. 3(a) depicts how the removal of the flip augmentation in the $L3N50$ baseline results in a significant reduction in log-average miss rate from 29.28% to 26.67%! The reason for the performance reduction becomes apparent immediately, when aspect ratio (ar) standardization (used in training/testing all current object detectors on Caltech [6]) is removed during training in $L3N50/ar$. Specifically, the standardization is useful in training since it is enforced in evaluation, yet it leads to non-aligned bounding boxes. Flipping these boxes introduces additional poorly-aligned samples which the boosting model does not handle well, even with increased modeling capacity in later stages of the experiments (depth 5 or more models). The solution is simple, either train $L3N50/ar$ models (without aspect ratio standardization and with flip augmentation) or remove flipped sampling ($L3N50/flip$). Since this is a one-time initial modification, we slightly rename our ‘new baselines’, so that $L3N50$ refers to $L3N50/flip$ (without flip, with aspect ratio standardization of training samples) and $L3N50*$ refers to $L3N50/ar$ (with flip, no aspect standardization). The two are shown to perform similarly in Figs. 3(a) and 3(b) (yet $L3N50/flip$ is faster to train as it contains half the samples). We note that this observation was not made in any of the related research studies, and consequently not employed by top performing boosted detectors.

Additional augmentation: Depicted in Fig. 3(b), the $L3N50$ and the $L3N50*$ models do not benefit from color augmentation, but do benefit from the scaling+crop scheme (best with a scaling factor of $S1.1$ in horizontal, vertical, and both directions) and translation augmentation (by 1 pixel shifts, T31). These experiments were run multiple times with different random seeds to ensure statistical significance. The improvements will be shown to be more dramatic with increased modeling capacity (deeper trees). We make two final notes, increasing the number of negatives to $L3N100$ does not further improve performance, and overall performance has reached a plateau. Therefore, the modeling capacity is increased in $L5N50$.

Increased model capacity: Still on Caltech $\times 3$, the analysis of Fig. 3(c) demonstrates a consistent trend with the previous experiments. Note how flip augmentation in $L5N50 + flip$ is still not beneficial, but it is much better handled. The best results are reported in Fig. 3(c). We observe a significant jump

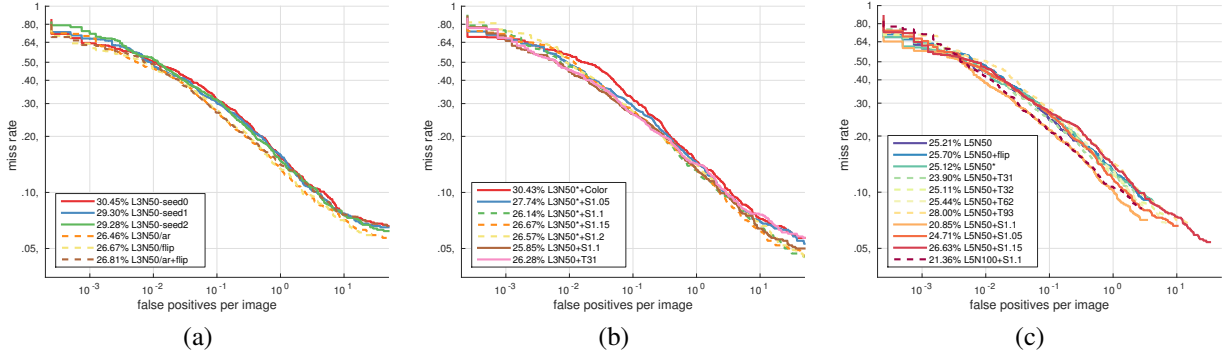


Fig. 3: Experimental analysis of different training settings on Caltech pedestrians using Caltech×3.

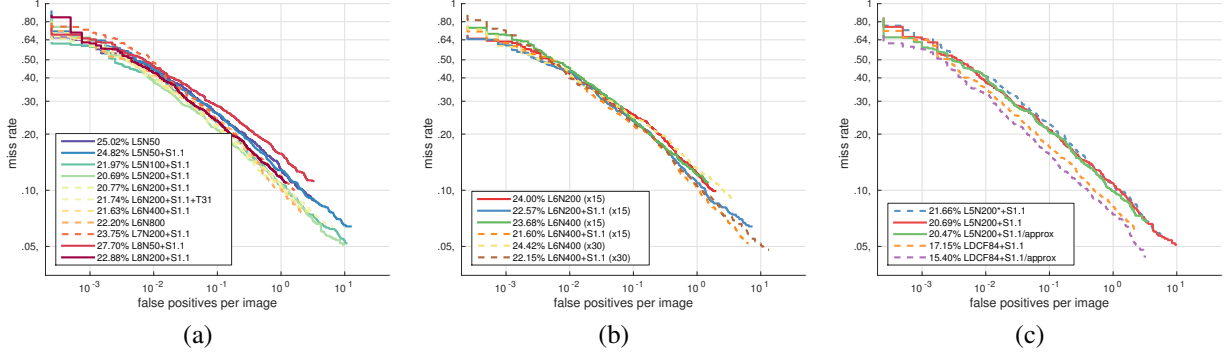


Fig. 4: Experimental analysis of different training settings on Caltech pedestrians using additional video augmentation (Caltech×7.5 unless stated).

in performance reaching 20.85% with the fast ACF model. These are the best reported results to date (nearly optimal from our experiments). Scaling is shown to be more useful than translation, as even slight off-center shifts hinder performance and are not handled well by the classifier. This is encouraging, yet we begin to understand how limited the boosting model is in sensitivity to dataset properties. Samples with even minimal occlusion and FG/BG transfer were also not shown to be beneficial for increasing the dataset size. Furthermore, we experimented with addition of external pedestrian datasets (e.g. ETH [36]) and increasing tree depth, but no benefit was shown.

Video augmentation: A main conclusion is that very little video augmentation (Caltech×3 vs. Caltech×10 in current state-of-the-art detectors [16], [19], [23]) was needed to reach best performance. Denser sampling of every 4th frame is shown in Fig. 4(a), demonstrating an interesting trend. First, we must increase the number of negatives to 200k (N_{200}) to reach comparable performance. This fact exposes another known limitation of currently used boosting models which is not reported in related research, one of dataset imbalance. This is necessary for implicitly regularizing deeper models (shown up to depth 8 in Fig. 4(a)). Second, we observe that the $S1.1$ scheme provides consistent improvements across different video sampling strategies and model choices (Fig. 4(b)). This demonstrates the sub-optimal yet commonly employed procedure of video augmentation. Third, additional

augmentation or increased model capacity does not improve over the 20.69% of $L5N_{200}$. To ensure no further gains can be made, we further increase tree depth to 6 to handle additional data and further increase dataset in Fig. 4(b), with no visible improvement. Increasing tree depth beyond 6 leads to overfitting behavior, and additional data must be sampled (although our experiments show no final gains).

Better features: The achieved 20.69% miss rate is the best known ACF results to date, nearly matching the more computationally intensive feature-rich Checkerboards (CB) detector [37] (18.47% miss rate). CB employs 61 filters for each of the 10 core HOG+LUV channels, resulting in significant increase to computational requirements. To further study the proposed modifications, we extract features using 4×8 LDCF filters [16] computed using PCA eigenvectors of feature patches (referred to as the **LDCF84** model). The method still runs 5 times faster than the CB comparison, while reaching a 17.15% miss rate. This demonstrates the generalization of the proposed best practices to other, feature-rich approaches as well. As a final experiment, we report results without the approximation of features in the multi-scale feature pyramid [15]. This allows for a more clear performance comparison against other methods [26], [37] which do not employ approximation. The gains are even more impressive, reaching 15.40% with a light-weight LDCF84 model, significantly outperforming CB.

Summary: Despite impressive gains in performance stem-

ming from simple, intuitive, and theory-driven considerations in training, the boosted detector can only handle mild deformations as augmentation. For instance, color jitter is not helpful, flipping is problematic (and removed in most of the experiments without reduction in performance), and slight translation shifts produce a more difficult learning task that is not modeled well even with deeper trees. In order to address these limitations, we proposed to use an augmentation technique which maintains centered ground truth boxes. Dataset size is key in training deeper trees, yet performance saturates at depth 5/6 even with extensive augmentation. Hence, unlike additional layers in CNN architectures, further increase of the tree depth provides a limited benefit in terms of classification power. The experiments motivate future work for the effective increase of the modeling capacity of tree models.

Run-time analysis: Although not the main focus of the study, we report run-time for the interested reader. The ACF-L5 and LDCF84 models run on a CPU at 6.7 and 2.5 frames per second (fps), respectively. These models are significantly faster than all of the top performing feature-rich models, which range in 0.1-1 fps (e.g. CB [23] runs at 0.5 fps).

A. Context Analysis

Context in the form of scene representation is essential for robust pedestrian detection. The deeper tree models are able to better capture relationships between features in the image. In particular, we have observed how deeper models are more likely to select features in the padding area of the model around the pedestrian. As a final limitation experiment on Caltech, we propose to use a location prior model in order to study to limitations of the model in capturing spatial context.

First, we studied increasing model padding further with tree depth as means of better representing contextual information, yet this did not impact performance. On the other hand, due to the application domain, it is reasonable to study context in terms of perspective and position of objects. A second approach was shown to improve performance further, hinting that there is still room for future improvements in context modeling.

Given a detection, $\mathbf{p} = [x \ y \ w \ h \ s]^T$, we construct $\phi_{\mathbf{p}} = \mathbf{p}\mathbf{p}^T$ for each training sample. Consequently, entries in $\phi_{\mathbf{p}}$ are employed as features for capturing relationship between position, size, confidence features, and their products. The object score, s , is recomputed using an SVM [38] on $\phi_{\mathbf{p}}$. We note that this generalizes (and outperforms) the approach in [39], which only employs a subset of the proposed $\phi_{\mathbf{p}}$ feature set (setting $\phi_{\mathbf{p}} = [h^2 \ y^2 \ hy \ h \ y]$).

As shown in Fig. 2, context integration consistently improve performance of both the ACF and LDCF84 models by a further 0.5-1% reduction in miss rate. Hence, there is still room for improvement in order to better capture contextual cues within the models. We term our models, ACF++ and LDCF84++, where the first ‘+’ stands for the final models trained with data augmentation (Fig. 4), and the second ‘+’ for the contextual reasoning. We observe how the proposed modifications produce consistent improvements over both the

TABLE I: Results on the new annotations on Caltech [26] using log average miss rate over $[10^{-2}, 10^0]$ and $[10^{-4}, 10^0]$ (MR_2^N and MR_4^N , respectively). Our ACF+ (with data augmentation) and ACF++ (with contextual reasoning) is on par with state-of-the-art detectors while enjoying low computational complexity. Our LDCF84++ employs just 4 filters on top of the ACF features and outperforms the VGG baseline by a large margin on the more challenging metric MR_4^N .

Method	MR_2^N (MR_4^N)
RotatedFilters [26]	12.87 (24.10)
RotatedFilters+VGG [26]	9.32 (21.72)
ACF+	15.17 (27.28)
ACF++	13.27 (25.26)
LDCF84+	11.76 (21.08)
LDCF84++	9.71 (18.29)

ACF and LDCF baselines in Fig. 2 by about 10%. The proposed high-performing models are new baselines in a way, and further gains may be obtained with combined with insights from other studies listed in Fig. 2.

B. Results on Caltech-New

The proposed improvements are consistent on the improved annotation dataset presented in [26], as shown in Table I. Consequently, we report best reported results to date among both CNN and non-CNN methods on the challenging MR_4^N metric, with a miss rate of 18.29%.

C. Generalization to Face Detection

The proposed training procedure is applied to face detection on the recently introduced large WIDER face dataset [40] and the commonly employed Fddb dataset [41]. Specifically, we do not standardize aspect ratio and further augment the dataset as discussed in Section III. As shown in Table II, the detection performance improve when up to depth 9 decision trees are employed. This is a result of the new challenges in face detection benchmarks (e.g. rotation) not commonly found in pedestrian detection. Fig. 5 demonstrates consistent gains in performance, resulting in a state-of-the-art face detector (on Fddb, at 2000 false positives, 92.67% true positive rate is achieved).

IV. CONCLUDING REMARKS

This study presented a set of novel experiments aimed at better understanding the boosted decision tree model currently employed for many vision tasks. As CNN-based models employ extensive augmentation, we sought to investigate modifying the boosted detector to handle such augmentation as well. Careful analysis regarding the generalization power and modeling capacity, dataset size and balance, and overfitting handling produced insights as to the performance limits of such detectors, as well as state-of-the-art pedestrian and face detectors. A main takeaway of this work is in the limited representation ability of the boosted model. Unlike CNN models, which consistently benefit from additional modeling capacity (e.g. depth) and data, increasing weak learner complexity saturates early, not allowing for further gains with deeper

TABLE II: Improvement due to incorporation of the insights in this paper when training face detection models.

Results on WIDER-validation	
Method	AP (%)
ACF- $L3N50*$	27.97
ACF- $L9N200*$	34.64
ACF- $L9N200*+S1.1$ (ACF+)	46.80
LDCF84- $L9N200*+S1.1$ (LDCF+)	47.66

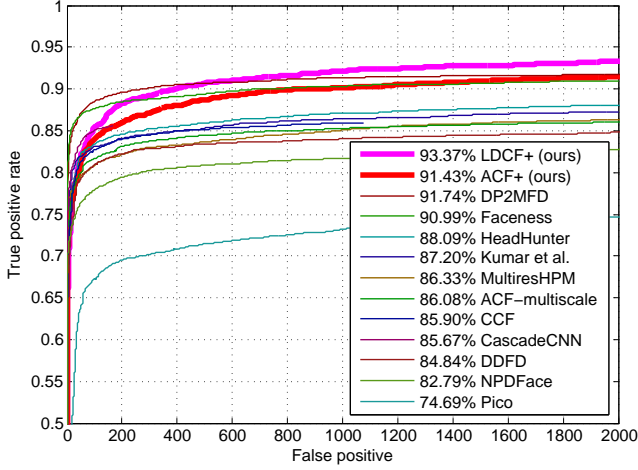


Fig. 5: Results on the Fddb [41] dataset (discrete score evaluation) compared with the state-of-the-art. True positive rate at 2000 false positives is shown for each method.

decision trees even with extensive augmentation. Although careful data augmentation was beneficial, architectural changes for handling large datasets are required on the decision-tree training level for effective increase of modeling capacity.

V. ACKNOWLEDGMENTS

We acknowledge support of associated industry partners and thank our UCSD CVRR colleagues, especially Rakesh Rajaram, for helpful discussions. We also thank the reviewers for their constructive comments.

REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.
- [2] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multistage feature learning," in *CVPR*, 2013.
- [3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, 2014.
- [4] P. Viola and M. Jones, "Robust real-time face detection," *IJCV*, vol. 57, no. 2, pp. 137–154, 2004.
- [5] F. Bartoli, G. Lisanti, S. Karaman, A. D. Bagdanov, and A. D. Bimbo, "Unsupervised scene adaptation for faster multi-scale pedestrian detection," in *ICPR*, 2014.
- [6] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *PAMI*, 2012.
- [7] S. Zhang, C. Bauckhage, and A. B. Cremers, "Efficient pedestrian detection via rectangular features based on a statistical shape model," *T-ITS*, 2015.
- [8] F. De Smedt, K. Van Beeck, T. Tuytelaars, and T. Goedemé, "The combinator: Optimal combination of multiple pedestrian detectors," in *ICPR*, 2014.
- [9] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?" in *ECCV-CVRSUAD*, 2014.

- [10] R. Benenson, M. Mathias, R. Timofte, and L. V. Gool, "Pedestrian detection at 100 frames per second," in *CVPR*, 2012.
- [11] S. Zhang, D. A. Klein, C. Bauckhage, and A. B. Cremers, "Center-surround contrast features for pedestrian detection," in *ICPR*, 2014.
- [12] M. J. Jones and D. Snow, "Pedestrian detection using boosted features over many frames," in *ICPR*, 2008.
- [13] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *BMVC*, 2009.
- [14] A. D. Costea and S. Nedevschi, "Word channel based multiscale pedestrian detection without image resizing and using only one classifier," in *CVPR*, 2014.
- [15] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *PAMI*, 2014.
- [16] W. Nam, P. Dollár, and J. H. Han, "Local decorrelation for improved pedestrian detection," in *NIPS*, 2014.
- [17] E. Ohn-Bar and M. M. Trivedi, "Learning to detect vehicles by clustering appearance patterns," *T-ITS*, 2015.
- [18] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Strengthening the effectiveness of pedestrian detection with spatially pooled features," in *ECCV*, 2014.
- [19] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Convolutional channel features," in *ICCV*, 2015.
- [20] Z. Cai, M. Saberian, and N. Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection," in *ICCV*, 2015.
- [21] A. D. Costea, A. V. Vesa, and S. Nedevschi, "Fast pedestrian detection for mobile devices," in *ITSC*, 2015.
- [22] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *The Annals of Statistics*, 1998.
- [23] S. Zhang, R. Benenson, and B. Schiele, "Filtered channel features for pedestrian detection," in *CVPR*, 2015.
- [24] N. Das, E. Ohn-Bar, and M. M. Trivedi, "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics," in *ITSC*, 2015.
- [25] J. Hosang, M. Omran, R. Benenson, and B. Schiele, "Taking a deeper look at pedestrians," in *CVPR*, 2015.
- [26] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" in *CVPR*, 2016.
- [27] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *ICCV*, 2015.
- [28] L. Bourdev and J. Brandt, "Robust object detection via soft cascade," in *CVPR*, 2005.
- [29] R. J. Quinlan, "Bagging, boosting, and c4.5," in *National Conference on Artificial Intelligence*, 1996.
- [30] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Pedestrian detection with spatially pooled features and structured ensemble learning," *arXiv*, 2014.
- [31] R. Appel, T. Fuchs, P. Dollár, and P. Perona, "Quickly boosting decision trees - pruning underachieving features early," in *ICML*, 2013.
- [32] J. Nilsson, P. Andersson, I. Y. H. Gu, and J. Fredriksson, "Pedestrian detection using augmented training data," in *ICPR*, 2014.
- [33] Vázquez, David, López, A. M., and D. Ponsa, "Unsupervised domain adaptation of virtual and real worlds for pedestrian detection," in *ICPR*, 2012.
- [34] R. N. Rajaram, E. Ohn-Bar, and M. M. Trivedi, "An exploration of why and when pedestrian detection fails," in *ITSC*, 2015.
- [35] E. Ohn-Bar and M. M. Trivedi, "Can appearance patterns improve pedestrian detection?" in *IV*, 2015.
- [36] A. Ess, B. Leibe, K. Schindler, and L. V. Gool, "A mobile vision system for robust multi-person tracking," in *CVPR*, 2008.
- [37] S. Zhang, R. Benenson, and B. Schiele, "Filtered channel features for pedestrian detection," *CVPR*, 2015.
- [38] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Intell. Sys. and Tech.*, 2011.
- [39] D. Park, D. Ramanan, and C. Fowlkes, "Multiresolution models for object detection," in *ECCV*, 2010.
- [40] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," in *CVPR*, 2016.
- [41] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009, 2010.