People Detection with Heterogeneous Features and Explicit Optimization on Computation Time

A. A. Mekonnen, F. Lerasle, A. Herbulot, and C. Briand
CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France
Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France
Email: {alhayat-ali.mekonnen, cyril.briand, frederic.lerasle, ariane.herbulot}@laas.fr

***Abstract*—In this paper we present a novel people detector that employs discrete optimization for feature selection. Specifically, we use binary integer programming to mine heterogeneous features taking both detection performance and computation time explicitly into consideration. The final trained detector exhibits low Miss Rates with significant boost in frame rate. For example, it achieves a 2.6% less Miss Rate at 10^{-4} FPPW compared to Dalal and Triggs HOG detector with a 9.22x speed improvement.**

I. Introduction

In modern era, computer vision is playing a significant role in automated object perception; one such thriving role is automated people detection. Visual people detection, *i.e.*, people detection using visual cameras, is the most prominent mode employed in the literature as cameras are cheap, versatile, and provide rich color and texture information. It is indispensable primarily in surveillance systems, human-machine interaction, robotics, automotive industry, image/video indexing, *etc*. Evidently, it is also one of the challenging tasks in computer vision due to variations in peoples' appearance, background clutter, illumination, sensor motion, and so forth. In recent years astounding progress have been made by the scientific community [1], [2], but there is still room for improvement.

One important discipline where applications of visual people detection is highly proliferating is robotics. In robotic systems that entail people perception, the aforementioned challenges are further exacerbated by real-time requirements, limited computational resources, and sensor motion. A mobile robot needs to be reactive during navigation/interaction in human occupied environments. Thus, its people detection module–which is one component of an entire functioning system–should be fast. The advent of powerful camera systems in the robotic community that provide high resolution omnidirectional images, *e.g.*, the Ladybug series [3] from Point Grey, stresses this point further urging the need to give extra focus on computation time during detector design.

In this work, we try to give explicit consideration to computation time during detector design. Generally speaking, balancing computation time and detection performance is challenging; best detection results are obtained using complex features and descriptors which are computationally expensive. As an example, Histogram of Oriented Gradients (HOG) [4] is the most discriminant feature thus far, but it is also computationally expensive compared to simple features like Haar variants [5]. Furthermore, most detectors that improve over HOG either use complex human models, *e.g.*, parts based models [6], or consider various heterogeneous pool of features, *e.g.*, [7], [8], both of which contribute to added computation time unless explicit computation considerations are made. In line with this, we present a person detector that uses heterogeneous pool of features and makes explicit computation time vs detection trade-off optimization to build a performant detector that leads to a significant gain in computation time while maintaining competitive detection performance.

Related Works: The entire literature in visual people detection is overwhelming and a discussion on the different techniques is beyond the scope of this paper (please refer to [1], [2] for extensive surveys). We will focus on approaches that use heterogeneous pool of features with sliding-window detection paradigm. The best results in visual people detection are obtained using heterogeneous pool of features [1], [2]. Heterogeneous features help capture complementary information useful to handle various detection challenges. For example: Wojek *et al.* [8] used Haar, HOG, and shape context features. They presented a comparative result obtained using boosting techniques and SVMs as classifiers and demonstrated that the combination of different features successfully outperformed individual variants and even the state-of-the-art at the time. Walk *et al.* [7] also clearly showed they obtained the best detection results when concatenating HOG, Histogram Of Flow [9], and Color Self Similarity (CSS) features all together, rather than individual features or a subset of them. Similar conclusions were made by Schwartz *et al.* [10] and Hussain and Triggs [11] using–HOG, color frequency, and co-occurrence features–and–HOG, Local Binary (LBP) and Ternay (LTP) Pattern features–respectively.

Given heterogeneous pool of features, different ways can be used to build the final detector. Four main trends can be observed in the literature: (1) Direct concatenation [7], [8] in which the different features are concatenated to make one high dimensional feature vector and an SVM used afterwards for classification. This is computationally costly owing to the complex feature and SVM weights applied in sliding window detection. [11], [10] used dimensionality reduction techniques after concatenation which improved detection performance but not detection speed. (2) Direct boosting [12], [8], [13] where an ensemble classifier is learned using the entire heterogeneous pool of features. The problem here is in boosting, on each iteration, the feature with the least weighted classification error is added to the ensemble irrespective of its computation time. This favors complex features resulting in computationally costly detector. (3) Coarse-to-fine hierarchical arrangement [14], [15] where a cascade is constructed using cheap features at the initial stages and using complex features

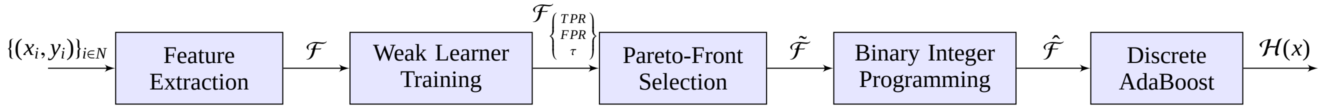


Fig. 1: Feature selection and classifier learning framework used at each node of a cascade.

at later stages. This approach is quite advantageous and tries to find a balance between detection performance and speed. The concern is, how to decide which features to use at the different stages systematically? Both [14], [15] adopt a heuristic based rule and use homogeneous family of features they deemed cheap at the initial stages, and homogeneous complex features at the latter. Finally, (4) via a computation time vs detection trade-off. This notion has been considered by the works of Wu and Nevatia [16], Jourdheuil *et al.* [17], and Mekonnen *et al.* [18]. In all cases, they defined a criterion composed of feature detection performance and computation time in a multiplicative manner. But, considering a multiplicative factor masks the contributions from the corresponding objectives and is not guaranteed to be optimal.

Our proposed framework falls in the 4^{th} category; but, it can also be considered as a variant of coarse-to-fine hierarchy in which the exact features to use at each cascade node are selected automatically via an optimization step. We use five frequently used heterogeneous features, namely: Haar-like features [5], Edge Orientation Histogram (EOH) [13], CSS [7], Center Surround Local Binary Patterns (CS-LBP) [19], and HOG [4] in a classical cascaded boosting configuration [5] with an added explicit optimization step based on Binary Integer Programming (BIP) to select a subset of features that have the least combined computation time and achieve a stipulated detection performance.

Contributions: This paper claims to make two important contributions. First, it presents a BIP formulation to mine heterogeneous features taking both detection performance and computation time into consideration. The authors assert this optimization applied to heterogeneous features is unique in the literature and marks a key contribution. Second, the paper presents a thorough evaluation of the proposed person detector–using both proprietary and public datasets–with detailed analysis of its performance compared to alternative approaches and the state-of-the-art.

II. Framework

The objective in this work is to develop a people detection framework based on heterogeneous features that capture different facets of persons in an image. Our proposed detector training framework takes discriminative power of each feature and its associated computation time into consideration explicitly to select, and subsequently use, a subset of features that fulfill the required detection performance and have the minimum cumulative computation time.

As detection speed is one of our design focus, we adopt the acclaimed Viola and Jones [5] attentional cascade detector configuration in a sliding window paradigm. To train a strong classifier at each node of the cascade, the framework depicted in figure 1 is employed. For a given set of positive and negative training samples, a total of n samples denoted as $\{(x_i, y_i)\}_{i\in\{1,\ldots,n\}}$: First, the features described in § II-A are extracted resulting in the feature set $\mathcal{F}$. For each feature a unique weak learner is trained using the examples provided and is used to characterize the discriminating power of the feature in terms of True Positive Rate (TPR) and False Positive Rate (FPR). Following, pareto-front analysis is used to select a subset of features, $\tilde{\mathcal{F}}$, taking their TPR, FPR, and computation time into account. This step is necessary to reduce the overwhelming total number of features to a tractable size for discrete optimization. Next, binary integer optimization, presented in § III, is used to retain a subset of features, $\hat{\mathcal{F}}$, that have the required performance—detection plus minimum computation time. Finally, a nodal strong classifier $\mathcal{H}(x)$ is trained using the retained feature set $\hat{\mathcal{F}}$ with discrete AdaBoost. Specific design choice motivations and brief descriptions of each block are presented herein below.

A. Features

Five different feature families are considered, namely: Haar like, CS-LBP, CSS, EOH, and HOG. This choice is motivate by two aspects: (1) their frequent use in the literature for person detection, and (2) their complementary nature. EOH and HOG capture edge distributions, CSS focuses on color symmetry, Haar-like and CS-LBP on intensity and texture variations. The feature pool of each family is extracted from a 128×64 pixels human template window.

Haar like: Here, the extended set proposed by Lienhart and Maydt [20] which includes tilted variants is used. The pool is generated by extracting feature values at all positions and scales in the template window with the extended Haar set.

CS-LBP: Computes per pixel CS-LBP [19] value by taking and modulating the intensity difference of center symmetric pixels for all the neighboring pixels. For each pixel, we privilege a 3×3 pixel region which results in a scalar integer between 0 and 16. Then, considering a rectangular region within the human template, a histogram with 16 bins is computed to signify one feature of this family. For all possible positions and scales of the rectangular region a distinct feature (which is a histogram) is computed and added in to the set of CS-LBP feature pool.

CSS: Color self similarity, proposed by Walk *et al.* [7], captures pairwise similarities of spatially localized color distributions and can be used to capture the left and right symmetry of persons' clothing (upper body and lower body). The computation first starts by subdividing the given template into non-overlapping regions called blocks. For each block a $3\times3\times3$ HSV color histogram is constructed. Then, the similarity of block with the rest of the blocks is determined via histogram intersection. In stead of concatenating all computed similarities like Walk *et al.* [7], we define a single CSS feature to be a vector of scalar values that are obtained by intersecting the histogram of one block with the rest of the blocks. The CSS feature pool set is then determined by computing this vector for all blocks. By dividing the template into blocks of 8×8 pixels, a total of 128 feature vectors, each with 127 dimensions, are obtained.

EOH: This feature pool is generated exactly as described by Geronimo *et al.* [13]: edge orientation histogram followed by ratios of magnitude of two bins to get a single scalar feature value and doing this for all positions and scales of rectangular subregions for histogram computation within the template window.

HOG: The HOG feature pool set is constructed as follows: Given the template window, it is divided to overlapping blocks and a 36 dimensional histogram of oriented gradients is computed just like [4]. But, rather than concatenating all block histograms to make one high dimensional feature, we consider concatenating a subset spanning a rectangular region. The HOG feature pool is generated by considering all possible positions, width, and hight of the rectangular region. The features range from a 36 dimensional vector, a single block, to 3780 dimensional one, all blocks in the template.

Table I summarizes the total number of features, the scaled maximum and minimum feature computation time (τ_{max} and τ_{min}), and the exact weak learner used in each feature family. For CS-LBP families Linear Discriminant Analysis combined with a decision tree (which is trained after re-projection) is privileged as SVM leads to overwhelming training period (due to the high number of CS-LBP features).

TABLE I: Feature pool summary. Time is reported relatively as a multiple of the smallest feature computation time.

Feature Type	No of features	τ_{min}	τ_{max}	Weak Learner
Haar like	672,406	1.0	3.48	Decision Tree
EOH	712,960	4.83	317.75	Decision Tree
CS-LBP	59,520	15.45	393.64	LDA + Decision Tree
CSS	128	1017.94	1017.94	SVM
HOG	3,360	489.72	51420.56	SVM

B. Pareto-front extraction

Given all set of features, $\mathcal{F}$, along with their trained associated weak learners, and characterized by three parameters: TPR, FPR, and computation time (τ), pareto-front analysis is used to find the optimal solutions that make up the pareto optimal set—the solutions that cannot be improved in one objective function without deteriorating their performance in at least one of the rest. The subset of features that are pareto optimal with respect to TPR, FPR, and computation time, denoted $\tilde{\mathcal{F}}$, are extracted and passed on to be used for the discrete optimization step.

C. Feature selection and cascade classifier learning

The final and decisive feature selection step is performed by the BIP optimizer and is discussed in § III. This module provides the set $\hat{\mathcal{F}}$. Finally, the nodal strong classifier, $\mathcal{H}(\cdot)$, is built with discrete AdaBoost by using the $\hat{\mathcal{F}}$ feature set.

The complete classifier used for detection however contains multiple nodes forming a cascade. The cascade construction starts with all positive training samples and a subset of the negative training samples (equivalent to the positive ones) to learn the set of relevant features and classifier for the initial cascade node. Once this is done, all negative training samples in the dataset are tested with it. All those that get classified correctly are rejected while all those labeled as positive samples (false positives) are retained along with the positive samples for training the following nodes. This step is repeated until all negative training sample are exhausted. This data mining techniques makes it possible to use vast number of negative training samples.

III. Discrete optimization feature selection

The BIP based feature selection applied to heterogeneous features makes the core of this work's contribution. The detailed optimization formulation to select a subset of features that fulfill a stipulated nodal FPR$_k$, TPR$_k$, with the minimum combined computation time possible is provided as follows, k denotes the node index.

Definition of parameters: The following are list of parameters used in the optimization specification. $\mathbb{B} = \{0, 1\}$ denotes a binary set. The applies to cascade node k.

- $N = \{1, ..., n\}$: set of training sample indexes with $n \in \mathbb{Z}$; a total of n training samples indexed by i;
- $M = \{1, ..., m\}$: set of weak learners indexes with $m \in \mathbb{Z}$; a total of m weak learners indexed by j;
- $\mathbf{y}^+ \in \mathbb{B}^n$, $\mathbf{y}^+ = \left\{y_i^+\right\}_{i\in N}$; $\mathbf{y}^- \in \mathbb{B}^n$, $\mathbf{y}^- = \left\{y_i^-\right\}_{i\in N}$

  $$y_i^+ = \begin{cases} 1 & \text{if } i \text{ is positive} \\ 0 & \text{otherwise} \end{cases} \quad y_i^- = \begin{cases} 1 & \text{if } i \text{ is negative} \\ 0 & \text{otherwise} \end{cases}$$

- $\mathbf{H} \in \mathbb{B}^{nxm}$ where $\mathbf{H} = \left\{h_{i,j}\right\}_{\substack{i\in N \\ j\in M}}$ with $h_{i,j} \in \{0, 1\}$

  $$h_{i,j} = \begin{cases} 1 & \text{if weak learner } j \text{ detects sample } i \text{ as positive} \\ 0 & \text{otherwise} \end{cases}$$

- TPR$_k$ $\in [0, 1]$: minimum true positive rate set at the considered node (k) of the cascade;
- FPR$_k$ $\in [0, 1]$: maximum false positive rate at the node;
- $\mathcal{T} \in \mathbb{R}^m$: with $\mathcal{T} = \left\{\tau_j\right\}_{j\in M}$ computation time of weak learner j.

Decision Variables: In BIP, the decision variables are restricted to binary values, values from the set $\mathbb{B} = \{0, 1\}$. The BIP decision variables are the following.

- $\mathbf{v} \in \mathbb{B}^m$, $\mathbf{v} = \left\{v_j\right\}_{j\in M}$ $v_j \in \{0, 1\}$: $v_j = 1$ if weak learner j is selected, else $v_j = 0$;
- $\mathbf{T} \in \mathbb{B}^n$, $t_i \in \{0, 1\}$: $t_i = 1$ if a positive sample i has been detected as positive (true positive) by at least one selected weak learner, else $t_i = 0$;
- $\mathbf{F} \in \mathbb{B}^n$, $f_i \in \{0, 1\}$: $f_i = 1$ if a negative sample i has been detected as positive (false positive) by at least one selected classifier, else $f_i = 0$.

Let vector $\mathbf{p} = \{p_i\}_{i\in N} = \mathbf{Hv}$, which denotes the total number of weak learners that have labeled each training sample i as positive.

Objective Function and Constraints:

$$\begin{array}{lll} \min & \mathcal{T}^\intercal \mathbf{v} & & (1) \\ \text{s.t} & t_i \leq y_i^+ \cdot p_i & \forall i & (2) \\ & f_i \geq y_i^- \cdot h_{i,j} \cdot v_j & \forall (i, j) & (3) \\ & \|\mathbf{T}\|_1 \geq \|\mathbf{y}^+\|_1 \cdot \text{TPR}_k & & (4) \\ & \|\mathbf{F}\|_1 \leq \|\mathbf{y}^-\|_1 \cdot \text{FPR}_k & & (5) \\ & \mathbf{v} \in \mathbb{B}^n; \mathbf{T} = \{t_i\}_{i\in N}, \mathbf{F} = \{f_i\}_{i\in N}; \mathbf{T}, \mathbf{F} \in \mathbb{B}^N & & (6) \\ & \|\cdot\|_1 \text{ is } l_1 \text{ norm.} & & \end{array}$$

The objective function (2) aims at minimizing the computation time. Constraints (2)-(6) express that a given rate of detection quality has to be reached (depending on the number of true and false positives). Constraints (2) link v_j and t_i variables (via p_i) so that $t_i = 0$ if image i has not been well-recognized by at least one selected classifier. Constraints (3) link v_j and f_i variables so that $f_i = 1$ if a negative image i has been recognized as positive by at least one selected classifier. Constraint (4) expresses that the stipulated TPR_k of true positives, obtained with the selected classifiers, has to be reached. Similarly, constraint (5) expresses that the stipulated FPR_k of false positives, obtained with the selected classifiers, must not be exceeded. In this formulation, there are a total of $(n\cdot(m+1)+2)$ binary variables in the BIP, which could be huge for large n and m values. The final subset of features $\hat{\mathcal{F}}$ corresponds to only the selected features, *i.e.*, non zero $\mathbf{v}$ entry; since each feature indexed by j is associated with a unique weak learner h_j, $\hat{\mathcal{F}}$ also represents the subset of weak learners retained

IV. Experiments and Results

In this section the different experiments carried out to investigate the performance of the proposed framework and obtained results along with commentaries are presented. The evaluation is focused on the following two aspects:

(1) Feature selection strategy evaluation: Here, the aim is to analyze the pros and cons of using BIP over other simpler alternatives. The proposed BIP based feature selection and classifier learning strategy, labeled as **BIP+AdaBoost**, is compared with two other modes. First, **Pareto+AdaBoost** which discards the BIP block in the framework and directly trains a nodal strong classifier with discrete adaboost using the features retained by the parto-front extraction block. And second, **Random+AdaBoost** which directly builds a nodal classifier using randomly sampled features from the total feature pool (proportional to each feature pool family size) and AdaBoost.

(2) General comparative evaluation with the state-of-the-art: In this part, the performance of the trained BIP+AdaBoost is compared with the prominent approaches in the literature.

A. Evaluation Criteria

For detector performance evaluation, we use two approaches: (1) The per window approach, whereby a Detection Error Trade-off (DET) curve with Miss Rate versus False Positives Per Window (FPPW) is generated by using cropped positive and negative windows; and (2) the per image approach which shows Miss Rate versus False Positives per Image (FPPI). The first curve is used to compare experimental variants of the proposed framework with respect to Dalal and Triggs HOG [4] *(aspect 1)*, and the second is used to determine how our best approach plays out compared to the different techniques in the literature *(aspect 2)*. To summarize the performance, the Miss Rate at 10^{-4} FPPW and the log-average miss rate are used in the first and second approaches respectively.

Another criterion that is taken into account is the average computation time. For a cascade detector the average computation time for a given candidate window is affected by the FPR of each node. Let K be the total number of nodes in the cascade, FPR_k be the false positive rate and τ_k be the total computation time of the k^{th} node during detection. Assuming the nodal FPR characteristics hold on a generic input image, the average time spent on a test candidate window, $\mathcal{T}_{av}$, can be estimated as $\mathcal{T}_{av} = \tau_0 + \sum_{k=1}^{K-1}(\prod_{z=0}^{k-1}\text{FPR}_z)\tau_k$. Using Dalal and Triggs [4] detector, which takes ζ_{HOG} per window, as a reference, the **Average Speed Up (ASU)** over it is determined as $\text{ASU} = \frac{\zeta_{HOG}}{\mathcal{T}_{av}}$. Consequently, the ASU values reported henceforth are with respect to Dalal and Triggs detector.

B. Dataset

For evaluation, two different datasets are considered: The **Ladybug dataset**[1], which is a proprietary dataset compiled from indoor laboratory environment using the *Ladybug2* spherical camera; and the **INRIA public dataset** [4], a publicly available dataset most predominantly used for benchmarking people detectors in the literature. A detailed description is not provided here due to space considerations, but table II summarizes the actual data used for training and testing purposes. The Ladybug dataset is used for training and testing the framework using cropped windows. On the INRIA dataset cropped windows are used for training. For testing, both cropped windows and full images are used for a per window and full image evaluation respectively. In both datasets, the cropped negative windows are uniformly sampled from provided person free full images.

TABLE II: Summary of the different dataset used for training and testing.

Dataset	Training		Test		
	pos win.	neg win.	pos win.	neg win.	full images
Ladybug[1]	1990	488, 992	1, 000	319, 653	–
INRIA [4]	2, 416	2.55×10^6	1, 132	2×10^6	288

C. Training

Each cascade node training (learning) is governed by two provided parameters: the nodal TPR_k and FPR_k for node k. The training is carried out in such a way that the final trained classifier conforms to these stipulated performance requirements. Each cascade node is built using a subset of the total negative training samples and all positive samples. This set is initially divided into a 60% training and a 40% validation set. The weak learners are trained using the 60% training set. Then, TPR and FPR values corresponding to each weak learner are determined based on the validation set. All subsequent computation, *i.e.*, pareto-front analysis and feature selection via BIP are performed using the weak learners performance conferred on the validation set. Once the pertinent features are selected, the corresponding weak learners are re-trained using the combined training and validation set within the discrete AdaBoost to build the per node final strong classifier, *i.e.*, $\mathcal{H}(\cdot)$. The complete cascaded classifier is then learned as explained in § II-C. For the associated weak learners, a decision tree depths of 2, 3, and 3 are used for Haar like, EOH, and LBP features respectively after detection performance and over-fitting trade off analysis on a validation set.

D. Results and Discussions[2]

Ladybug Dataset: The main results obtained with the Ladybug dataset are depicted in figure 2 and summarized in

[1] Please see http://homepages.laas.fr/aamekonn/ladybug_dataset/

[2] All figures in this section are best viewed in color.

table III. Clearly Pareto+AdaBoost results in the best detection performance, 2.9% MR, followed by Dalal and Triggs detector trained on this dataset, 3.0%, at 10^{-4} FPPW. In terms of detection, BIP+AdaBoost trails behind Random+AdaBoost with marginal loss. But, the most important result to notice is that BIP+AdaBoost results in a drastic 42.7x speed up over Dalal and Triggs with only a 7% loss in MR at 10^{-4} FPPW. The main reason for this speed up is that BIP+AdaBoost systematically uses cheap features in the intial stages of the cascade and only starts using computationally expensive features at later stages. The trained classifier has 10 cascade nodes with CSS features initially appearing at the 6^{th} node and HOG at the final stage.

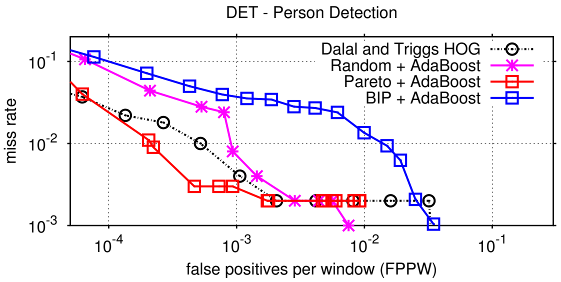


Fig. 2: DET of different detectors trained and tested on the Ladybug dataset.

Apparently, Pareto+AdaBoost and Random+AdaBoost result in worsened speeds. This is because AdaBoost always privileges the most discriminant feature, irrespective of computation cost, from the pool of features passed to it, and both pareto front extraction and random sampling are likely to pass such kind of complex features. Hence, the set of features selected in the first node result in a conglomerate that is effectively computationally demanding than Dalal and Triggs detector. These result are obtained using a fixed nodal FPR of 0.5 for all constructed nodes and the obtained results are very precise that altering the FPR is not necessary.

TABLE III: Summary of the cascade detector trained on the Ladybug dataset. Miss Rate is reported at 10^{-4} FPPW.

Detector	Feature Proportion					MR	ASU
	Haar	CSLBP	CSS	EOH	HOG		
Dalal and Triggs [4]	–	–	–	–	100%	3.0%	1.0x
Pareto + AdaBoost	10.7%	0.0%	0.0%	0.0%	83.7%	2.9%	0.7x
Random + AdaBoost	51.6%	6.2%	1.5%	36.0%	4.7%	8.0%	0.6x
BIP + AdaBoost	54.3%	8.6%	8.5%	25.7%	2.8%	10.0%	42.7x

INRIA Dataset: Similar results obtained for the INRIA dataset are shown in figure 3 and summarized in table IV. As this dataset is challenging, two variants of the BIP+AdaBoost classifier are trained. In the first case, a fixed nodal FPR of 0.5 is used for all nodes, called **BIP+AdaBoost(Fix)**. In the second case, an adaptive FPR is employed which starts at 0.3 in the initial stage and continues training nodes, whenever a solution for the BIP optimization does not exist, this constraint is relaxed/incremented by 0.1 and the procedure continues from that node likewise until all negative samples are depleted. This is called **BIP+AdaBoost(Ad)**. Again, the best detection results at 10^{-4} FPPW are obtained by the Random+AdaBoost and Pareto+AdaBoost variants. But, this time both variants of BIP+AdaBoost beat Dalal and Triggs detector at 10^{-4} by more than 2%. On top of this, the BIP+AdaBoost(Fix) achieves a 15.6x speed up while that of BIP+AdaBoost(Ad) trails with a 9.22x speed up.

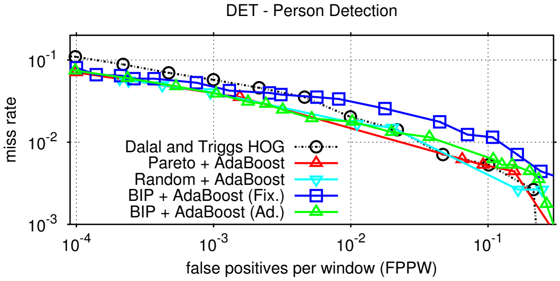


Fig. 3: DET of different detectors trained and tested on the INRIA dataset.

As the intial FPR constraints are stringent on the BIP+AdaBoost(Ad) variant, it will favor relatively discriminative features with increased computation time. But, this also contributes to its superior detection performance, over BIP+AdaBoost(Fix), throughout the FPPW range shown in figure 3. Observe in the table, there are more proportion of Haar like features (5.4% more) and less proportions of HOG features (2.0% less) in the fixed variant compared to the adaptive variant resulting in the increase speed up.

TABLE IV: Summary of the cascade detector trained on the INRIA datasets. Miss Rate is reported at 10^{-4} FPPW.

Detector	Feature Proportion					MR	ASU
	Haar	CSLBP	CSS	EOH	HOG		
Dalal and Triggs [4]	–	–	–	–	100%	11.0%	1.0x
Pareto + AdaBoost	42.8%	14.5%	7.8%	25.6%	9.3%	7.0%	0.4x
Random + AdaBoost	26.3%	10.8%	3.7%	53.5%	5.6%	6.0%	0.4x
BIP + AdaBoost (Fix)	60.4%	10.8%	8.0%	9.7%	11.0%	8.0%	15.6x
BIP + AdaBoost (Ad)	55.0%	14.6%	8.1%	9.3%	13.0%	7.4%	9.22x

Figure 5 shows histogram of the selected features, with relative proportions, for the first 9 nodes of both the fixed and adaptive variants. Clearly, the fixed variant initially uses cheaper features and increases along the cascade both in number and complexity. On the contrary, for the variable variant, complex features appear in the initial nodes and increase in number along the cascade. Figure 4 illustrates a few of the selected features overlaid on an average human gradient image for BIP+AdaBoost(Ad). Observe that all selected features capture discriminant facets of people.

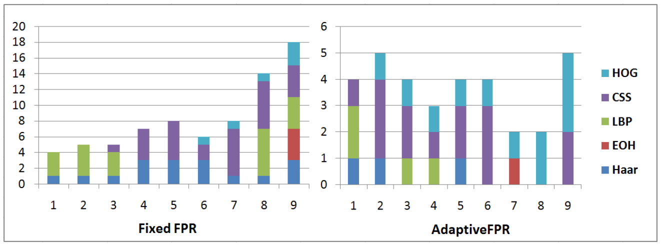
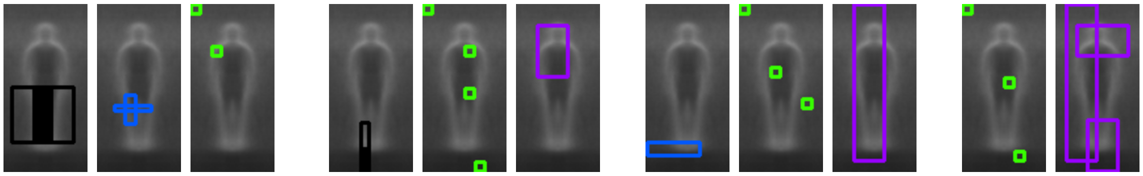


Fig. 5: Histogram of selected features in the first 9 nodes of the model trained on the INRIA dataset using both fixed FPR of 0.5 and adaptive FPR.

Finally, figure 6 shows the comparative evaluation of BIP+AdaBoost (Ad) detector on the INRIA dataset using the full image evaluation criteria. Comparative evaluations



(a) node 0 (b) node 1 (c) node 2 (d) node 8

Fig. 4: Sample depictions (overlaid on an average human gradient image) of the heterogeneous features selected at different nodes of the cascade trained on the INRIA dataset using an adaptive FPR. Black rectangular regions show Haar features, blue is for CS-LBP, green boxes represent CSS features and their position indicates the reference block, and finally, violet shows the spatial region spanned by the concatenated HOG blocks.

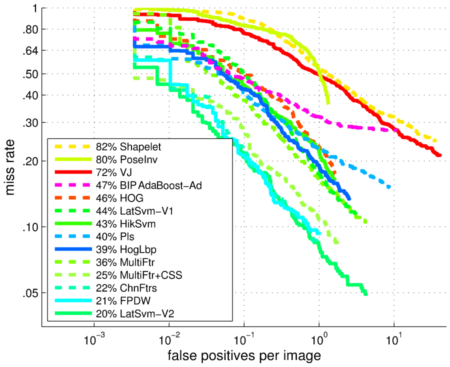


Fig. 6: Comparative full image evaluation on the INRIA test set.

are taken from [1]; the reader is referred to this survey for explanation of each detector (as space does not permit here). To generate these results, a Pairwise Max non-maximal suppression [1] with an overlap threshold of 0.65. Again, here, BIP+AdaBoost(Ad) does well achieving a log-average miss rate of 47%. At lower FPPI values, less than 0.1 FPPW, the BIP variant consistently supersedes Dalal and Triggs HOG. Using the computation speed reported in [1] for people more than 100 pixels in a 640×480 image, our detectors achieves 2.3 frames per second (fps) for the adaptive variant, and 3.9 fps for the fixed FPR variant trained on the INRIA dataset. These values are amongst the top best only exceeded by **FPDW** which achieves approximately 6.0 fps. But, actually **FPDW** uses the underlying principles of **ChnFeats** and optimizes the detection process by approximating the features over scale space. Similar techniques can be used to further improve the fps of our detector. On the other hand, the model trained on the Ladybug dataset, achieves 10.6 fps on the simpler dataset. This is an added advantage as a majority of the methods in the state-of-the-art do not have the ability to automatically change the complexity of the trained detector based on the dataset; examples include Dalal and Triggs HOG and **HogLbp** which have fixed size feature vector irrespective of dataset.

V. Conclusions

In conclusion, a novel framework based on heterogeneous pool of features and discrete optimization for developing a computation time and detection performance optimized person detector has been presented. The proposed framework has been validated thoroughly using proprietary and public datasets. The results obtained conform to our aims and result in a faster detector with competitive detection performance amongst the state-of-the-art.

In the near future, we plan to investigate ways to achieve more faster versions of the detector by focusing on implementation optimization and specialized accelerator hardwares like Graphical Processing Units (GPUs).

Acknowledgment

This work was supported by a grant from the French National Research Agency (ANR) under grant number ANR-12-CORD- 0003.

References

[1] P. Dollar, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: An evaluation of the state of the art,” *IEEE T-PAMI*, vol. 34, no. 4, pp. 743–761, 2012.

[2] D. Geronimo, A. Lopez, A. Sappa, and T. Graf, “Survey of pedestrian detection for advanced driver assistance systems,” *IEEE T-PAMI*, vol. 32, no. 7, pp. 1239–1258, 2010.

[3] Point Grey Inc., “Spherical vision catalog,” http://ww2.ptgrey.com/spherical-vision, accessed: 2013-10-14.

[4] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. CVPR*, 2005, pp. 886–893.

[5] P. A. Viola and M. J. Jones, “Robust real-time face detection,” *IJCV*, vol. 57, no. 2, pp. 137–154, 2004.

[6] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *Proc. CVPR*, 2008, pp. 1–8.

[7] S. Walk, N. Majer, K. Schindler, and B. Schiele, “New features and insights for pedestrian detection,” in *Proc. CVPR*, 2010, pp. 1030–1037.

[8] C. Wojek and B. Schiele, “A performance evaluation of single and multi-feature people detection,” in *DAGM-Symposium*, 2008, pp. 82–91.

[9] N. Dalal, B. Triggs, and C. Schmid, “Human detection using oriented histograms of flow and appearance,” in *Proc. ECCV*, 2006, pp. 428–441.

[10] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis, “Human detection using partial least squares analysis,” in *Proc. ICCV*, 2009, pp. 24–31.

[11] S.-u. Hussain and B. Triggs, “Feature sets and dimensionality reduction for visual object detection,” in *Proc. BMVC*, 2010, pp. 1–10.

[12] P. Dollar, Z. Tu, P. Perona, and S. Belongie, “Integral channel features,” in *Proc. BMVC*, 2009, pp. 1–11.

[13] D. Gerónimo, A. M. López, D. Ponsa, and A. D. Sappa, “Haar wavelets and edge orientation histograms for on-board pedestrian detection,” in *Proc. IbPRIA*, 2007, pp. 418–425.

[14] A. Mogelmose, A. Prioletti, M. Trivedi, A. Broggi, and T. Moeslund, “Two-stage part-based pedestrian detection,” in *Proc. ITSC*, 2012, pp. 73–77.

[15] H. Pan, Y. Zhu, and L. Xia, “Efficient and accurate face detection using heterogeneous feature descriptors and feature selection,” *CVIU*, vol. 117, no. 1, pp. 12 – 28, 2013.

[16] B. Wu and R. Nevatia, “Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection,” in *Proc. CVPR*, 2008, pp. 1–8.

[17] L. Jourdheuil, N. Allezard, T. Chateau, and T. Chesnais, “Heterogeneous adaboost with real-time constraints - application to the detection of pedestrians by stereovision,” in *Proc. VISAPP*, 2012, pp. 539–546.

[18] A. A. Mekonnen, F. Lerasle, and A. Herbulot, “Person detection with a computation time weighted adaboost and heterogeneous pool of features,” in *Proc. ACIVS*, 2013, pp. 632–644.

[19] M. Heikkil, M. Pietikinen, and C. Schmid, “Description of interest regions with local binary patterns,” *Pattern Recognition*, vol. 42, no. 3, pp. 425 – 436, 2009.

[20] R. Lienhart and J. Maydt, “An extended set of haar-like features for rapid object detection,” in *Proc. ICIP*, 2002, pp. 900–903.