Détection de personnes par apprentissage de descripteurs hétérogènes sous des considérations CPU

Alhayat Ali Mekonnen, Frédéric Lerasle, Ariane Herbulot, Cyril Briand

Détection de personnes par apprentissage de descripteurs hétérogènes sous des considérations CPU

A. A. Mekonnen F. Lerasle A. Herbulot C. Briand

CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France
Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France
{alhayat-ali.mekonnen, frederic.lerasle, ariane.herbulot, cyril.briand}@laas.fr

Résumé

Cet article présente un nouveau détecteur de personnes utilisant une sélection de descripteurs par optimisation discrète type branch and bound. Plus précisément, nous utilisons une programmation binaire pour sélectionner un sous-ensemble de descripteurs hétérogènes qui optimisent conjointement les performances en détection et le coût CPU. La mise en oeuvre de ce détecteur puis son évaluation sur des bases publiques montre clairement que cette reformalisation offre un bon compromis entre taux de faux négatifs et temps de calcul comparativement aux détecteurs existants de la littérature.

Mots Clef

Détection de personnes, sélection de descripteurs, apprentissage.

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 their detection performance and computation time explicitly into consideration. The final implemented and trained detector on public dataset clearly demonstrates that this framework offers a good compromise between detector Miss Rates and achieved frame rate compared to other methods in the literature.

Keywords

People Detection, Feature Selection, learning.

1 Introduction

De nombreuses applications s'appuient aujourd'hui sur des techniques avancées de vision par ordinateur. La détection visuelle de personnes *i.e. via* une caméra perspective est certainement la plus usitée car ce capteur optique est bas coût, non intrusif, et délivre une information très riche sur la scène observée (couleur, texture). Citons ici les applications de vidéosurveillance, interaction homme-machine, robotique, automobile, indexation d'images, etc. Un enjeu est le coût CPU et la robustesse du détecteur à divers artefacts : variations d'apparence des personnes, du point de vue, d'illumination, voir mouvement du capteur si celui-ci est embarqué. Certes, des avancées notables [3] ont été observées dans la communauté Vision mais cet enjeu reste encore aujourd'hui d'actualité.

Notre approche vise ici à prendre en considération explicitement le coût CPU dans le processus de sélection des descripteurs sous-jacents au détecteur. Ce coût est vital dans tout système réel *e.g.* en robotique où la réactivité du système est conditionnée par les ressources CPU embarquées et les temps de traitement. Ces temps de traitement peuvent être prohibitifs notamment pour des capteurs optiques de dernière génération, *e.g.* la caméra Ladybug de Point Grey [12] qui exhibent des résolutions pixel élevées. Il est vital alors de privilégier pour le détecteur des descripteurs discriminants mais aussi peu coûteux en CPU. Ce compromis est en pratique difficile à obtenir. Ainsi, les histogrammes de gradients orientés (HOG) [1] sont des descripteurs très discriminants mais très coûteux comparativement aux descripteurs de type Haar [13]. Certes, les dernières avancées considèrent des détecteurs mixant des descripteurs hétérogènes (HOG, Haar, etc.) [14, 15] ou modélisant explicitement/implicitement [4] les parties corporelles... mais toujours au détrimant du coût CPU qui n'est pas explicité dans la formulation. Ce constat a motivé nos travaux qui visent à développer un détecteur offrant un compromis entre taux de classification et coût CPU.

Travaux antérieurs : La littérature propose de nombreux détecteurs de personne et un état de l'art détaillé serait ici superflu ; le lecteur pourra se référer ici à [3]. Notre étude se limitera ici aux investigations privilégiant un ensemble hétérogène de descripteurs et une technique de fenêtre glissante pour générer les échantillons/régions à classer. Cette démarche, en mixant des informations complémentaires, améliore les performances à l'instar de Dollar *et al.* dans [3].

Citons également Wojek *et al.* [15] qui mixent des descripteurs de type Haar, HOG, et *shape context*. Leur étude comparative à partir de classifieurs SVMs ou *boosting* montre clairement que la fusion de descripteurs hétérogènes est plus performante et donc supplante les approches se bornant à un pool de descripteurs homogènes. Walk *et al.* dans [14] ont abouti au même constat en concaté-

nant HOG, histogramme de flot optique [2], et *Color Self Similarity* (CSS).

Quatre stratégies de fusion des descripteurs hétérogènes sont alors privilégiées dans la littérature pour construire le détecteur :

1. Une concaténation directe des descripteurs [14, 15] induisant un fort coût CPU de par la complexité du descripteur final et les poids du classifieur associés dans la détection par fenêtre glissante.

2. Un *boosting* direct [15, 5] *i.e.* chaque classifieur fort apprend directement et itérativement le sous ensemble des descripteurs pertinents parmi le pool complet hétérogène. Hélas, à chaque itération, un descripteur est sélectionné par le classifieur indépendamment de son coût CPU. Cette démarche privilégie les descripteurs certes discriminants mais complexes et donc augmentant le coût CPU.

3. Un arrangement hierarchique [10, 11] *i.e.* la cascade multi-classifieurs considère des descripteurs à faible coût CPU dans ses étages initiaux et des descripteurs plus complexes dans ses étages supérieurs. Cette démarche offre un compromis entre taux de détection et vitesse. Certains travaux [10, 11] s'appuient ici sur des heuristiques et des familles homogènes de descripteurs simples et complexes respectivement pour les étages initiaux et suivants.

4. Un compromis entre vitesse et taux de détection à l'instar des travaux menés dans Wu and Nevatia [16], Jourdheuil *et al.* [7], et Mekonnen *et al.* [9]. Le principe est de combiner au sein d'un même critère, et avec des pondérations dédiées, le coût CPU et les performances de détection. Cette formulation masque les contributions de chacun des deux critères sous-jacents et n'offre aucune garantie d'optimalité.

Notre approche s'inscrit dans cette dernière stratégie mais elle sélectionne à chaque nœud de la cascade les descripteurs optimisant conjointement et distinctement les deux critères pré-cités. Nous considérons quatre familles usuelles de descripteurs : Haar [13], Histogramme orientation des contours (EOH) [5], CSS [14], *Center Surround Local Binary Patterns* (CS-LBP) [6], et HOG [1] dans un cadre de *boosting* structuré en cascade [13] avec une optimisation discrète basée sur une programmation 1/0 (BIP) sélectionnant le sous ensemble des descripteurs offrant le meilleur compromis coût CPU-performance.

Contributions : Cet article propose une reformulation du processus d'optimisation, ici BIP, et prenant en compte coût CPU et performance de détection dans le processus de sélection des descripteurs. Cette reformulation est clairement novatrice dans la littérature. Des évaluations sur la base d'images publiques INRIA [1] sont ensuite proposées afin de quantifier les gains obtenus comparativement aux détecteurs existants de la littérature.

2 Descriptif de notre approche

Pour rappel, l'objectif est de prototyper un détecteur basé sur des descripteurs capturant l'aspect visuel d'un individu et ceci indépendamment du point de vue de la caméra, de l'apparence, de l'illumination, etc. Bref, l'apprentissage hors ligne vise à sélectionner un sous-ensemble de descripteurs discriminants au mieux une silhouette humaine générique... et peu onéreux en CPU.

Nous privilégions, pour son faible coût CPU et à l'instar de [13], un mécanisme de cascade attentionnelle classant en positifs (humain) ou négatifs (autres) des sous-images générées par une technique de fenêtre glissante dans l'image entière. L'apprentissage du classifieur fort propre à chaque nœud de la cascade est schématisé par le synoptique figure 1. Soient n échantillons positifs ou négatifs d'apprentissage notés $\{(x_i, y_i)\}_{i\in\{1,\dots,n\}}$), les descripteurs listés en section § 2.1 sont extraits, l'ensemble associé est noté $\mathcal{F}$. Pour chaque descripteur, un classifieur faible est entraîné à partir de la base d'apprentissage afin de caractériser son pouvoir discriminant en termes de taux de vrais positifs (TPR) et taux de faux positifs (FPR). Puis, une analyse par front de Pareto permet de sélectionner un sous-ensemble de descripteurs notés $\tilde{\mathcal{F}}$, et prenant en considération les critères TPR, FPR, et coût CPU. Cette étape est vitale pour réduire de façon drastique le nombre de descripteurs candidats... en préambule à l'étape d'optimisation discrète. Cette étape d'optimisation, détaillée en § 3, est exécutée pour sélectionner un sous-ensemble restreint de descripteurs $\hat{\mathcal{F}}$ ayant le meilleur compromis performance-coût CPU. Au final, un classifieur fort par nœud $\mathcal{H}(\cdot)$ est entrainé à partir de ce sous ensemble de descripteurs $\hat{\mathcal{F}}$ par une technique de *boosting* discret. Chaque bloc du synoptique est détaillé ci-après.

2.1 Les descripteurs

Au total, cinq familles de descripteurs sont considérées, qui sont : Haar, CS-LBP, CSS, EOH, et HOG. Ce choix est motiver par deux aspects : (1) leur usage fréquent dans la littérature pour la détection de personne, et (2) leur complémentarité. EOH et HOG capturer distributions de bord, CSS se concentre sur la couleur symétrie, Haar et CS-LBP sur l'intensité et les variations de texture. Les descripteurs entières de chaque famille sont extraites en utilisant un 128×64 pixels de la fenêtre de modèle humain.

Haar : Ici, l'ensemble étendu proposé par Lienhart et Maydt [8] qui comprend les variantes inclinés est utilisé. L'ensemble complet est recueilli par extraction des valeurs de descripteurs à tous les postes et les échelles de la fenêtre de modèle.

CS-LBP : Calcule par pixel CS-LBP [6] valeur en prenant et en modulant la différence d'intensité de pixels centraux symétriques pour tous les pixels voisins. Pour chaque pixel, on privilège d'une région de pixels de 3×3 ce qui conduit à un nombre entier scalaire entre 0 et 16. Ensuite , un histogramme de bacs 16 est calculé compte tenu d'une zone

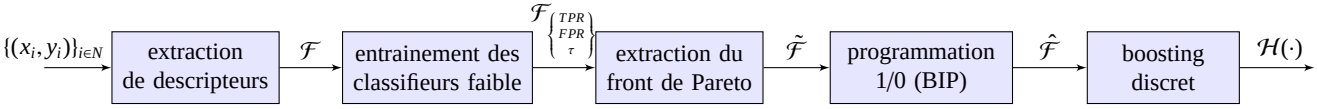


FIGURE 1 – Schéma du synoptique de l'apprentissage du classifieur fort propre à chaque nœud de la cascade.

rectangulaire. Cela signifie une descripteur de cette famille. Pour toutes les positions et les échelles possibles de la région rectangulaire, une descripteur distincte (qui est un histogramme) est calculé et ajouté à l'ensemble CS-LBP .

CSS : Le calcul commence d'abord par la subdivision de la fenêtre de modèle donné en blocs. Pour chaque bloc, une couleur histogramme HSV de 3x3x3 est construit. Ensuite, la similarité de bloc avec le reste des blocs est déterminé par l'intersection d'histogramme. Au lieu de la concaténation de toutes les similitudes calculés comme Walk *et al.* [14], nous définissons un seul CSS descripteur comme un vecteur de valeurs scalaires qui sont obtenus par l'intersection de l'histogramme d'un bloc avec les autres blocs. L'ensemble de CSS desciprteur est alors déterminé par le calcul de ce vecteur pour tous les blocs. En divisant le modèle en blocs de 8×8 pixels, un total de 128 descripteurs, chacun avec 127 dimensions, sont obtenus.

EOH : Ce pool de descripteur est générée exactement comme décrit par Geronimo *et al.* [5] : histogramme de l'orientation du contour suivi par les ratios de magnitude de deux bacs pour obtenir une valeur scalaire unique et le faire pour toutes les positions et les échelles de sous-régions rectangulaires dans la fenêtre de modèle.

HOG : L'ensemble HOG est construit comme suit : Soit la fenêtre de modèle, il est divisé en blocs et un histogramme des gradients orientés de 36 dimensions est calculée comme [1]. Mais, plutôt que la concaténation de tous les histogrammes de blocs pour faire une descripteur de grande dimension, nous considérons la concaténation un sous-ensemble couvrant une zone rectangulaire. La famille de la descripteur HOG est généré en considérant toutes les positions possibles, la largeur et hauteur de la région rectangulaire. Les desctribeurs varient d'un vecteur de 36 dimensions, qui contient un seul bloc, à un vecteur de 3780 dimensions, qui contient tous les blocs dans le modèle.

Table 1 présente les nombres total de descripteurs, maximale et minimale temps de calcul (τ_{max} et τ_{min}), et le classifieur faible utilisée dans chaque famille de descripteur. Pour le famille CS-LBP, analyse discriminante linéaire (LDA) associé à un arbre de décision (qui est construit après reprojection) est privilégié comme SVM mène à la période d'entraînement immense (en raison du nombre élevé de discripteurs CS-LBP).

TABLE 1 – Récapitulatif des descripteurs utilisés ; u = $0.0535\mu s$.

descripteurs	nombre total	τ_{min}	τ_{max}	classifieurs faible
Haar	672,406	1.0u	3.48u	arbre de décision
EOH	712,960	4.83u	317.75u	arbre de décision
CS-LBP	59,520	15.45u	393.64u	LDA + arbre de décision
CSS	128	1017.94u	1017.94u	SVM
HOG	3,360	489.72u	51420.56u	SVM

2.2 Extraction du front de Pareto

Soit $\mathcal{F}$ l'ensemble initial de descripteurs, leurs classifieurs faibles associés avec trois paramètres sous-jacents : TPR, FPR, et coût CPU (noté τ). L'analyse par front de Pareto exhibe les solutions optimales au sens de ces paramètres. Le sous-ensemble associé de descripteurs constitue le front de Pareto optimal *i.e.* on ne peut améliorer un paramètre sans dégrader un des deux autres : ce sous-ensemble de descripteurs, optimal au sens de Pareto pour les trois paramètres pré-cités, est noté $\tilde{\mathcal{F}}$; il est alors exploité par le processus d'optimisation discrète.

2.3 Sélection des descripteurs et apprentissage de la cascade

Le processus de sélection finale des descripteurs est piloté par optimisation discrète type BIP détaillé en § 3. Cette étape génère le sous ensemble $\hat{\mathcal{F}}$ de descripteurs. Au final, le classifieur fort propre à chaque nœud $\mathcal{H}(\cdot)$ s'appuie sur ce sous-ensemble et une technique de Adaboost discrète.

Le classifieur complet est structuré autour de plusieurs nœuds formant la cascade. Sa construction s'appuie initialement sur tous les échantillons positifs et un sous-ensemble d'échantillons négatifs (en nombre équivalent aux positifs) pour apprendre les descripteurs relatifs au premier nœud/étage. Tous les négatifs sont alors testés sur ce premier nœud, les vrais négatifs sont rejetés tandis que les faus positifs sont conservés pour les nœuds suivants. La démarche est re-itérée jusqu'à traitement de tous les négatifs. Cette technique dite de *data mining* permet l'exploitation d'un nombre flexible de négatifs.

3 Optimisation discrète

Une sélection des descripteurs basée sur un programme linéaire en variables binaires (une programmation 1/0) constitue une contribution essentielle de ce travail. La formulation proposée vise à minimiser le temps de traitement dans la cascade de détection. Elle prend en paramètre les taux de vrais et faux positifs souhaités (TPR_k, FPR_k), à chaque étage k.

Définition des paramètres : La liste suivante indique les paramètres utilisés dans la formulation. $\mathbb{B} = \{0, 1\}$ est l'ensemble binaire.

- $N = \{1, ..., n\}$ est l'ensemble des échantillons avec $n \in \mathbb{Z}$; un échantillon étant référencé par l'index i ;
- $M = \{1, ..., m\}$ est l'ensemble des classifieurs faibles avec $m \in \mathbb{Z}$; un classifieur faible étant référencé par l'index j ;
- les vecteurs $\mathbf{y}^+ \in \mathbb{B}^n$, $\mathbf{y}^+ = \left\{y_i^+\right\}_{i\in N}$ et $\mathbf{y}^- \in \mathbb{B}^n$, $\mathbf{y}^- = \left\{y_i^-\right\}_{i\in N}$ indiquent la nature des échantillons :

$$y_i^+ = \begin{cases} 1 & \text{si } i \text{ est positif} \\ 0 & \text{sinon} \end{cases} \quad y_i^- = \begin{cases} 1 & \text{si } i \text{ est négatif} \\ 0 & \text{sinon} \end{cases}$$

– $\mathbf{H} \in \mathbb{B}^{nxm}$ où $\mathbf{H} = \left\{h_{i,j}\right\}_{\substack{i\in N \\ j\in M}}$ avec $h_{i,j} \in \{0,1\}$ est la matrice de couverture des échantillons par les classifieurs faibles.

$$h_{i,j} = \begin{cases} 1 & \text{si le classifieur faible } j \text{ détecte l'échantillon } i \\ & \text{comme positif} \\ 0 & \text{sinon} \end{cases}$$

– $\text{TPR}_k \in [0,1]$ est le taux minum de vrais positifs souhaité à l'étage (k) de la cascade ;
– $\text{FPR}_k \in [0,1]$ est le taux maximum de faux positifs attendu à l'étage (k) de la cascade ;
– $\tau \in \mathbb{R}^m$, avec $\tau = \left\{\tau_j\right\}_{j\in M}$, désigne le temps de calcul associé au détecteur j.

Variables de décision : Les variables de décision sont binaires.

– $\mathbf{v} \in \mathbb{B}^m$, avec $v_j \in \{0,1\}$, définit l'ensemble des classifieurs faibles sélectionnés à l'étage k : $v_j = 1$ si le détecteur j est sélectionné, 0 sinon ;
– $\mathbf{T} \in \mathbb{B}^n$, avec $t_i \in \{0,1\}$, correspond à l'ensemble des vrais positifs détectés : $t_i = 1$ si l'échantillon positif i est détecté positif par au moins un détecteur , 0 sinon ;
– $\mathbf{F} \in \mathbb{B}^n$, avec $f_i \in \{0,1\}$, correspond à l'ensemble des faux positifs détectés : $f_i = 1$ si l'échantillon négatif i est détecté positif par au moins un détecteur , $f_i = 0$ sinon.

Nous introduisons le vecteur $\mathbf{p}$, $\mathbf{p} = \{p_i\}_{i\in N} = \mathbf{Hv}$ qui indique, pour chaque échantillon i, le taux total de détecteurs ayant détecté l'échantillon positif.

Formulation :

$$\begin{aligned}
&\min \quad \tau^\top \mathbf{v} && && (1)\\
&\text{s.t} \quad t_i \le y_i^+ \cdot p_i && \forall i && (2)\\
&\qquad f_i \ge y_i^- \cdot h_{i,j} \cdot v_j && \forall (i,j) && (3)\\
&\qquad \|\mathbf{T}\|_1 \ge \|\mathbf{y}^+\|_1 \cdot \text{TPR}_k && && (4)\\
&\qquad \|\mathbf{F}\|_1 \le \|\mathbf{y}^-\|_1 \cdot \text{FPR}_k && && (5)\\
&\qquad \mathbf{v} \in \mathbb{B}^n ; \mathbf{T} = \{t_i\}_{i\in N}, \mathbf{F} = \{f_i\}_{i\in N} ; \mathbf{T}, \mathbf{F} \in \mathbb{B}^n && && (6)\\
&\qquad \|\cdot\|_1 \text{ est la norme } l_1. && &&
\end{aligned}$$

La fonction objectif (1) a pour but de minimiser le temps de calcul total à l'étage k considéré. L'ensemble des contraintes (2)-(5) imposent qu'un certain niveau de qualité soit atteint (déterminé par les taux de vrais et faux positifs désirés). Les contraintes (2) font le lien entre les variables v_j et t_i (via p_i) : ainsi $t_i = 0$ si aucun détecteur sélectionné n'a identifié correctement l'échantillon positif i. Les contraintes (3) relient les variables v_j et f_i tel que $f_i = 1$ si l' échantillon négatif i a été reconnu positif par au moins un des classifieurs faibles sélectionnés. La contrainte (4) exprime qu'un taux de reconnaissance de TPR_k échantillons positifs doit être atteint. De façon symétrique, la contrainte (5) impose que le taux total de faux positifs ne doit pas excéder FPR_k. Le nombre total de contraintes dans cette formulation est égal à $(n \cdot (m+1) + 2)$, ce qui peut être élevé lorsque des nombres importants de détecteurs (n) et d'échantillons (m) sont considérés. Nous appelons $\hat{\mathcal{F}}$ l'ensemble final des échantillons détectés positifs par les détecteurs sélectionnés.

4 Evaluations et résultats

Les évaluations menées dans ce travail sont axées sur les deux aspects suivants :
(1) L'évaluation de la stratégie de sélection de descripteur : Ici le but est d'analyser les avantages et inconvénients de l'utilisation de l'optimisation discrète de type BIP par rapport aux alternatives plus simples. La stratégie d'utiliser une sélection de descripteur basée sur l'optimisation BIP et un apprentissage par classifieur est comparée à deux autres modes. Le premier, appelé **Pareto+AdaBoost** supprime le bloc BIP du cadre du travail et entraine directement un classifieur fort à chaque nœud avec une technique d'Adaboost discrète utilisant les descripteurs retenues par le bloc d'extraction du front de Pareto. Le second, appelé **Random+AdaBoost**, construit directement un classifieur fort à chaque nœud en utilisant des descripteurs choisies aléatoirement depuis l'ensemble total des descripteurs (proportionnellement à la taille de chaque famille de descripteurs).
(2) Une évaluation générale par-rapport à l'etat de l'art : Dans cette partie, la performance de la méthode BIP+Adaboost est comparée aux méthodes principales de la littérature.

4.1 Critères d'évaluation

Pour évaluer la performance du détecteur, nous utilisons deux approches : (1) L'approche par fenêtre, où est générée une courbe DET (Detection Error Trade-off) représentant les faux négatifs par-rapport aux faux positifs par fenêtre (FPPW) en utilisant des fenêtres de taille réduite de positifs et de négatifs. La première courbe est utilisée pour comparer des variantes dde l'algorithme proposé par-rapport au détecteur HOG de Dalal et Triggs [1] *(aspect 1)*, et la seconde est utilisée pour déterminer comment se comnporte notre meilleure variante par rapport aux méthodes de la littérature *(aspect 2)*. Un taux de faux négatifs à 10^{-4} FPPW et une log-moyenne du taux de faux négatifs sont utilisés respectivement pour la première et la seconde approche.
Une autre critère à prendre en compte et le temps moyen de calcul. Pour un détecteur en cascade, le temps moyen de calcul pour une fenêtre candidate donnée dépend du FPR à chaque nœud. Soient K le nombre total de nœuds dans la cascade, FPR_k lle taux de faux positifs et τ_k le temps total de calcul du k^{ime} nœud pendant la détection. En supposant un taux de faux positifs d'une image d'entrée générique, le temps moyen passé sur une fenêtre-test candidate, $\mathcal{T}_{av}$, peut être estimé par $\mathcal{T}_{av} = \tau_0 + \sum_{k=1}^{K-1}(\prod_{z=0}^{k-1} \text{FPR}_z)\tau_k$. En utilisant le détecteur de Dalal et Triggs [1] comme référence, qui prend un temps ζ_{HOG} par fenêtre, l'**accélération moyenne**

(ASU) est donnée par ASU = $\frac{\zeta_{HOG}}{\mathcal{T}_{av}}$. Par conséquent, les valeurs d'accélération moyennes reportées désormais sont calculées par-rapport au détecteur de Dalal et Triggs.

4.2 Jeux de données

Dans ce travail, en raison de contraintes de place, les résultats sont présentés sur un seu jeu de données public, la **base publique de données de l'INRIA** [1]. Il s'agit d'une base de données accessible au public utilisée principalement pour évaluer les performances des détecteurs de la littérature. Un total de 2416 fenêtres positives recadrées et de 2.55×10^6 fenêtres négatives uniformément réparties sont utilisées pour l'apprentissage. Pour l'évaluation par fenêtre, on utilise 1132 fenêtres positives recadrées et 2.00×10^6 fenêtres négatives uniformément réparties. Pour l'évaluation d'une image entière, la base de données fournit 288 images complètes annotées.

4.3 Apprentissage

Chaque nœud de la cascade d'apprentissage est régi par deux paramètres donnés : les TPR_k et FPR_k pour le nœud k (TPR_k vaut toujours 1.0). L'apprentissage est fait de telle sorte que le classifieur du nœud final soit conforme aux exigences de performance. Chaque nœud de la cascade est contruit en utilisant un sous-ensemble des échantillons négatifs d'apprentissage et tous les échantillons positifs. Cet ensemble est divisé initialement en deux sous-ensembles : 60% pour l'apprentissage et 40% pour la validation. Les classifieurs faibles sont entraînés en utilisant la base de données d'apprentissage. Ensuite, les valeurs de TPR et de FPR correspondand à chaque classifieur faible sont déterminées en se basant sur la base de données de validation. Tous les calculs suivants, c'est-à-dire l'analyse par front de Pareto et la sélection des descripteurs par BIP sont effectués en utilisant les performances des classifieurs faibles conférées sur la base de données de validation. Une fois que les caracteristiques pertinentes sont sélectionnées, les classifieurs faibles correspondant sont re-entraînés en utilisant à la fois la base de données d'apprentissage et celle de validation par une technique de boosting discret pour construire le classifieur fort final par nœud $\mathcal{H}(\cdot)$. Le classifieur complet en cascade est ensuite entraîné comme expliqué dans en § 2.3. Pour les classifieurs faibles associés, des arbres de décision de profondeur 2, 3 et 3 sont utilisés respectivement pour les descripteurs de Haar, EOH après compromis entre les performances de détection et le surapprentissage sur l'ensemble de validation.

4.4 Résultats et discussions

Les résultats principaux obtenus avec la base de l'INRIA sont montrés sur la figure 2 et sont présentés dans la table 2. Nous avons entraîné deux variantes du classifieur BIP+AdaBoost. Dans le premier cas appelé **BIP+AdaBoost(Fix)**, un FPR par noeud de 0.5 est utilisé pour tous les noeuds. Dans un second cas, un FPR adaptatif est utilisé, en démarrant à 0.3 à l'étape initiale et en continuant les noeuds d'apprentissage, à chaque fois qu'une solution de l'optimisation par BIP n'existe pas, cette contrainte est relâchée en augmentant le FPR de 0.1 et la procédure continue jusqu'à ce que tous les échantillons négatifs soient épuisés. Cette version est appelée **BIP+AdaBoost(Ad)**. Les meilleurs résultats de détection à un FPPW de 10^{-4} sont obtenus par les variantes Random+AdaBoost et Pareto+AdaBoost. Les deux variantes de BIP+Adaboost surpassent le détecteur de Dalal et Triggs à 10^{-4} de plus de 2%. De plus, la méthode BIP+AdaBoost(Fix) atteint une accélération de la méthode de 15.6x tandis que BIP+AdaBoost(Ad) admet une accélération de 9.22x.

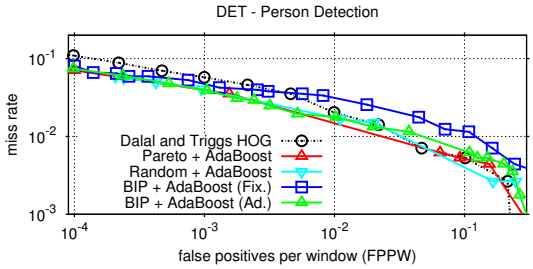


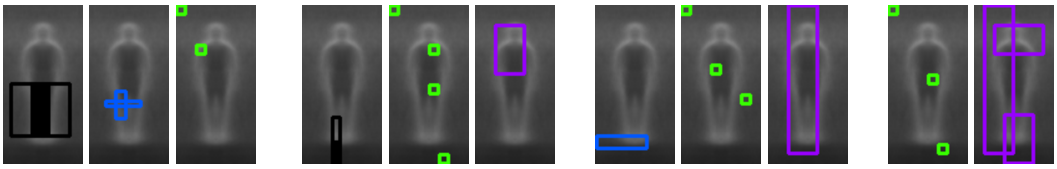
FIGURE 2 – DET des détecteurs entraînés et testés sur la base INRIA.

Comme les contraintes initiales de FPR sont strictes sur la variante BIP+AdaBoost(Ad), cela va favoriser les descripteurs discriminantes avec des temps de calcul augmentés. Mais cela va aussi contribuer à des performances de détection supérieures par rapport à BIP+AdaBoost(Fix), sur toute la gamme de FPPW présentée sur la figure 2. Sur la table 2, il y a une proportion plus importante de descripteurs de Haar (5.4% plus) et moins importantes de HOG (2.0% moins) dans la version fixe que dans la version adaptative, ce qui contribue à l'amélioration du temps de calcul.

TABLE 2 – Résumé du détecteur en cascade entrainé sur les bases de données de l'INRIA. Les taux de faux négatifs sont donnés à un FPPW de 10^{-4}.

détecteur	composition de descripteurs					MR	ASU
	Haar	CSLBP	CSS	EOH	HOG		
Dalal and Triggs [1]	–	–	–	–	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

Sur la figure 4 sont représentés les histogrammes des descripteurs sélectionnées, dans des proportions relatives, pour les premiers 9 noeuds des variantes fixe et adaptative de la méthode. Clairement, la variante fixe utilise des descripteurs moins coûteuses et augment le long de la cascade à la fois en nombre et en complexité. Au contraire, pour la variante variable, les descripteurs complexes apparaissent dans les noeuds initiaux et augmentent en nombre le long de la cascade. La figure 3 illustre quelques descripteurs sélectionnées superposées à une image de gradient d'un humain pour la version BIP+AdaBoost(Ad). Nous pouvons remarquer que toutes les descripteurs sélection-



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

FIGURE 3 – Représentations d'exemples (en superposition sur une image moyenne de gradient humain) des descripteurs hétérogènes choisis dans les différents n¡uds de la cascade formés en utilisant des données INRIA et FPR adaptative. Régions rectangulaires noires montrent descripteurs de Haar, bleu est pour CS-LBP, boîtes vertes représentent les descripteurs CSS et leur position indique le bloc de référence, et enfin, le violet montre la région de l'espace engendré par les blocs de HOG concaténés.

nées représentent des facettes discriminantes de personnes.

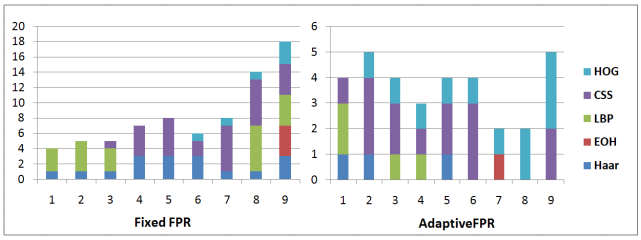


FIGURE 4 – Histogramme de descripteurs sélectionnées pour les 9 premiers noeuds des modèles entraînés sur la base INRIA avec un FPR fixe de 0.5 et avec un FPR adaptatif.

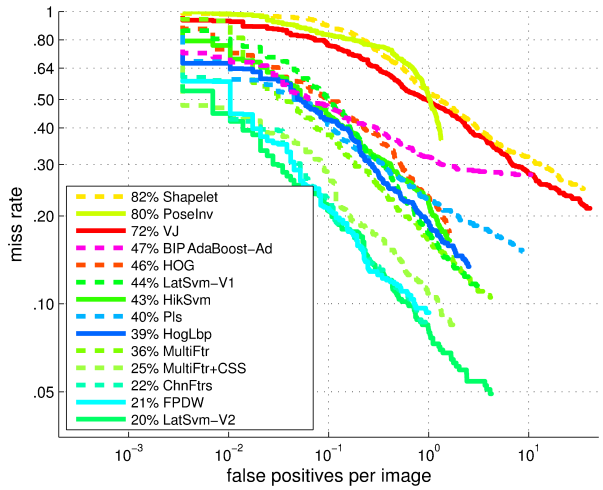


FIGURE 5 – Evaluation comparative avec images complètes sur la base test de l'INRIA.

Finalement, la figure 5 présente l'évaluation comparative du détecteur BIP+AdaBoost(Ad) (la meilleure variante qui donne un bon compromis entre performance de détection et coût calcul) sur la base INRIA en utilisant les critères d'évaluation sur image complète. Les évaluations comparatives sont issues de [3] ; le lecteur pourra se referer à cete étude pour l'explication de chaque détecteur. Pour générer ces résultats, nous utilisons une suppression des non maxima par paire [3] avec un seuil de recouvrement de 0.65. Encore une fois, ici, la variante BIP+AdaBoost(Ad) réussit à une log-moyenne de faux négatifs de 47%. At des valeurs plus basses de FPPI, à moins de 0.1 FPPW, la variante BIP surpasse les HOG de Dalal and Triggs systématiquement. En utilisant les vitesses de calcul mentionnées dans [3] pour des personnes de plus de 100 pixels sur des images de taille 640×480, nos détecteurs arrivent à 2.3 images par seconde (fps) pour la variante adaptative, et à 3.9 fps pour la vairante à FPR fixe, entraînés sur la base INRIA. Ces valeurs sont parmi les meilleures, seulement surpassées par **FPDW** qui arrive approximativement à 6.0 fps. Mais en fait **FPDW** repose sur les principes de **ChnFeats** et optimise le processus de détection en approximant les descripteurs le long d'un espace-échelle. Des techniques similaires pourraient être utilisées pour améliorer la vitesse de notre détecteur. D'un autre coté, le modèle entraîné sur la base de données Ladybug [1] atteint un fps de 10.6 sur un jeu de données plus simple. C'est un avantage supplémentaire du fait que la majorité des m"ethodes de l'état de l'art n'ont pas la possibilité de changer automatiquement la complexité du détecteur entraîné sur un jeu de données, comme par exemple le détecteur HOG et le **HogLbp** qui ont une taille fixe de vecteur de descripteur quel que soit le jeu de données.

5 Conclusions et perspectives

Cet article présente un nouveau détecteur basé sur des descripteurs hétérogènes sélectionnés *via* un processus d'optimisation discrète sur leur performance et coût CPU conjointement. Le formalisme est validé sur la base publique d'images INRIA *i.e.* les résultats sont conformes à nos attentes : le détecteur offre un excellent compromis entre performances et vitesse comparativement à la littérature.

Les travaux futurs portent sur une accélération supplémentaire du détection ainsi prototypé *via* son implémentation GPU (pour *Graphical Processing Units*) puis son intégration sur un robot mobile autonome.

REMERCIMENTS

Ce travail a été financé par une subvention de l'Agence Nationale de la Recherche (ANR) sous le numéro de subvention ANR-12-CORD-0003.

Références

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, pages 886–893, 2005.

1. Nous n'avons pas présenté l'ensemble de données de Ladybug en raison de contraintes d'espace. Mais, le lecteur pourra se référer ici à l'URL `http://homepages.laas.fr/aamekonn/ladybug_dataset/` pour plus de détails sur de le jeu de données Ladybug.

[2] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *Proc. ECCV*, pages 428–441, 2006.

[3] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection : An evaluation of the state of the art. *IEEE T-PAMI*, 34(4) :743–761, 2012.

[4] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. CVPR*, pages 1–8, 2008.

[5] D. Gerónimo, A. M. López, D. Ponsa, and A. Domingo Sappa. Haar wavelets and edge orientation histograms for on-board pedestrian detection. In *Proc. IbPRIA*, pages 418–425, 2007.

[6] M. Heikkila, M. PietikÃd'inen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern Recognition*, 42(3) :425 – 436, 2009.

[7] 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*, pages 539–546, 2012.

[8] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Proc. ICIP*, pages 900–903, 2002.

[9] A. A. Mekonnen, F. Lerasle, and A. Herbulot. Person detection with a computation time weighted adaboost. In *Proc. ACIVS*, pages 632–644, 2013.

[10] A. Mogelmose, A. Prioletti, M.M. Trivedi, A. Broggi, and T.B. Moeslund. Two-stage part-based pedestrian detection. In *Proc. ITSC*, pages 73–77, 2012.

[11] H. Pan, Y. Zhu, and L. Xia. Efficient and accurate face detection using heterogeneous feature descriptors and feature selection. *CVIU*, 117(1) :12 – 28, 2013.

[12] Point Grey Inc. Spherical vision catalog. `http://ww2.ptgrey.com/spherical-vision`. Accessed : 2013-10-14.

[13] P. A. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2) :137–154, 2004.

[14] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. In *Proc. CVPR*, pages 1030–1037, 2010.

[15] C. Wojek and B. Schiele. A performance evaluation of single and multi-feature people detection. In *DAGM-Symposium*, pages 82–91, 2008.

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