Boosting actif pour la recherche interactive d'images

Alexis Lechervy[1] Philippe-Henri Gosselin[1] Frédéric Precioso[1]

[1] ETIS CNRS

ENSEA, Univ Cergy-Pontoise, F-95000 Cergy-Pontoise
alexis.lechervy@ensea.fr , philippe-henri.gosselin@ensea.fr , frederic.precioso@ensea.fr

Résumé

Cet article présente un nouvel algorithme de recherche interactive d'objets dans les images basé sur le principe du boosting. De récents travaux ont proposé des algorithmes de boosting "en ligne" dans lesquels des ensembles de classifieurs faibles sont itérativement construit à partir des données. Dédiées au suivi d'objets dans les vidéos, ces méthodes ne s'appliquent pas efficacement au contexte de la recherche interactive. Nous proposons ici de sélectionner les classifieurs faibles itérativement à partir des images annotées par l'utilisateur dans un processus de recherche interactive. Le choix des images à annoter intervient à deux niveaux. Tout d'abord il est utilisé comme stratégie de sélection active afin d'optimiser le classifieur (fort) courant. Il participe aussi à l'élaboration des classifieurs faibles à considérer dans le processus de boosting. Des expériences ont été menées pour comparer cette nouvelle approche à une méthode de référence basée sur des SVMs.

Mots Clef

Indexation, Bases d'images, Boosting, Recherche interactive, Apprentissage actif

Abstract

This paper presents a new algorithm based on boosting for interactive object retrieval in images. Recent works propose "online" boosting algorithms where weak classifier sets are iteratively trained from data. These algorithms are proposed for visual tracking in videos, and are not well adapted to interactive retrieval. We propose in this paper to iteratively build weak classifiers from images labelled as positive by the user during a retrieval session. A novel active learning strategy for the selection of images for user annotation is also proposed. This strategy is used to enhance the strong classifier, but also to build new weak classifiers. Experiments have been carried out on a generalist database in order to compare the proposed method to a SVM based reference method.

Keywords

Indexation, Images Databases, Boosting, Interactive retrieval, Active learning,

1 Introduction

Face à l'augmentation toujours croissante des documents multimédia, il est de plus en plus nécessaire de faire appel à des outils informatisés pour rechercher des informations. L'un des principaux enjeux de la recherche d'information est de faire le lien entre le contenu sémantique recherché par l'utilisateur et les données numériques.
Dans le cadre de cet article on s'intéresse plus particulièrement à la détection d'objet dans des images par une approche supervisée. On a d'une part un utilisateur qui souhaite créer un détecteur d'une classe d'objet sémantique et d'autre part des données numériques que l'on peut représenter à l'aide de primitives visuelles (couleurs, textures, ...).
La construction d'un ensemble d'apprentissage par un expert est une tâche difficile et coûteuse. De plus les exemples choisis ne sont pas nécessairement les images les mieux adaptées aux systèmes de recherche.
C'est pourquoi il est intéressant de limiter la collecte des exemples d'apprentissage et d'aider l'utilisateur dans son travail. Différent travaux ont abordés les problématiques liée à un nombre d'exemple d'apprentissage limité, on peut citer par exemples [5] et [20].
La recherche interactive permet d'exprimer et d'affiner la requête de l'utilisateur. Elle instaure un dialogue entre l'utilisateur et le système en vu de construire un ensemble d'apprentissage avec un nombre limité d'exemple, en fonction des besoins du système.
L'une des principales approches pour la recherche interactive est le bouclage de pertinence (relevance feedback). L'utilisateur fournit, à un système de recherche, des exemples qu'il considère appartenir à la classe recherchée ainsi que des contre-exemples. À partir de ces données, le système est mis à jour pour converger vers la "pensée" de l'utilisateur.
Une "stratégie active" permet de sélectionner les données à présenter à l'utilisateur pour être annoter dans le but d'opti-

miser le classifieur final. L'utilisation de méthodes actives pour un apprentissage actif à fait l'objet de plusieurs travaux dont [18] pour les SVM. Des méthodes ultérieures ont étendu les travaux précédemment cité, aux algorithmes de boosting ([3],[13],[21]).
Contrairement aux approches par SVM nécessitant des réglages mathématiques afin de pouvoir combiner tous types d'attribut, nous étudions ici un algorithme reposant sur le boosting qui est ainsi adapté à n'importe quel type d'attribut.
Nous commençons cet article en présentant le principe général du boosting et certaines méthodes récentes qui en découlent. Nous proposons par la suite, un nouvel algorithme de boosting appliqué au contexte de la recherche interactive. Nous y abordons notamment une gestion évolutive de l'ensemble des classifieurs faibles, ainsi qu'une méthode active de sélection d'image à annoter. Nous présentons également les classifieurs utilisés. Nous décrivons ensuite le protocole expérimental utilisé et nous comparons notre méthode à une approche similaire utilisant des SVM.

2 Boosting

2.1 Principe général

Le boosting est une classe de méthodes d'apprentissage basée sur la sélection de classifieurs. Ce sont des techniques utilisées dans de nombreuses activités d'apprentissage, aussi bien pour de la catégorisation [16] et le filtrage de texte [17], que pour des problèmes linguistiques [1], de bioinformatique [4] ou de détection [12] et classement d'objet dans des images.
Le boosting repose sur l'observation suivante :
– il est facile de trouver des règles simples, généralement justes
– il est très difficile de trouver une règle "fortes" qui est vérifiée dans tous les cas.

Comment combiner alors ces règles simples ayant des résultats légèrement meilleurs qu'un tirage aléatoire, dans le but de construire une règle "forte" qui donne des résultats optimaux ?
L'idée de construire un classifieur fort à partir de classifieurs faibles fut introduit par Schapire en 1990 [15]. L'une des premières méthodes mettant en place cette idée est l'algorithme AdaBoost [9]. Il fut adapté à la détection d'objet dans des images par [19].
Dans le contexte de la détection d'objet, le boosting est utilisé pour sélectionner des classifieurs faibles d'objet dans des images. En effet, le résultat de la recherche d'une catégorie d'objets correspond à la classification de la base en deux ensembles d'images : positif X_p (les images qui contiennent les objets) et négatif X_n (celles qui ne les contiennent pas).
Les images d'apprentissage pré-annotées (positivement ou négativement) permettent de sélectionner des classifieurs faibles h_t dans un ensemble W. A chaque itération t de sélection, un classifieur est choisi en fonction d'un critère reposant sur la capacité du nouveau classifieur faible à classer correctement les images qui ont été mal classées par les classifieurs faibles précédemment choisis. Le choix de ces classifieurs se fait itérativement en utilisant l'ensemble de la base.
Le classifieur fort final H effectue la somme des résultats de chacun des classifieurs faibles sélectionnés en les pondérant en fonction de la qualité de leur classement. H est alors meilleur que chacun des h_t pris séparément.
Ce processus d'apprentissage peut être formalisé par la production d'un classifieur fort H connaissant un ensemble d'exemples X et un ensemble de classifieurs faibles W

$$App : (X, W) \rightarrow H \tag{1}$$

Les classifieurs faibles utilisés peuvent être de nature quelconque, il faut juste qu'ils soient légèrement meilleurs que le hasard, c'est à dire que leur probabilité de bonne réponse soit supérieure à 0,5.
Le boosting permet ainsi d'utiliser des classifieurs de "moindre qualité" mais qui peuvent avoir de bonnes propriétés notamment en termes de vitesse de calcul.
Cependant, le boosting ne permet pas intrinsèquement de trier l'ensemble des images de la base par leur pertinence à la catégorie recherchée. Or, dans le contexte de la recherche interactive, la plupart des stratégies de sélection des images à annoter sont basées sur le tri des données par rapport à cette pertinence. L'adaptation du boosting au tri des images d'une base par leur pertinence par rapport à la recherche effectuée, a été proposée par l'algorithme RankBoost que nous détaillons dans la section suivante.

2.2 RankBoost

Le RankBoost est une méthode de Boosting introduite par Freund et Schapire [8]. Contrairement aux approches classiques qui, appliqués à l'image, permettent la détection d'objet, cette méthode offre la possibilité de classer des images les unes par rapport aux autres.
Le problème de classement s'inscrit dans un contexte différent de la détection d'objet. L'objectif n'est plus d'attribuer une classe à une zone mais de classer les zones étudiées les unes par rapport aux autres, sans se soucier de la frontière entre objet et non-objet. On ne cherche donc pas à répondre à la question : ceci est-il l'objet ($H(x) \stackrel{?}{=} 1$) ?, on cherche à savoir si cette zone ressemble plus à la classe recherchée qu'une autre ($H(x_1) \stackrel{?}{>} H(x_2)$). Ce qui revient à transformer la fonction de décision par une fonction de rang :

$$\begin{array}{rccc} h : & I & \rightarrow & [0,1] \\ & \text{zone x} & \mapsto & \text{score de la zone x} \end{array}$$

Les classifieurs faibles renvoient un score de classification pour chacune des images. On sélectionne les classifieurs qui attribuent un meilleur score aux exemples positifs.
Pour cela, on s'intéresse à des couples d'exemples y, un positif contre un négatif ($y =$

(x_p, x_n) avec $x_p \in X_p$ et $x_n \in X_n$). On considére que le classifieur classe correctement le couple y si l'image positive a un score plus fort que l'image négative, soit : $h(x_p) > h(x_n)$.
Le critère de sélection des classifieurs faibles prend en compte la qualité du classement des couples y en attribuant plus d'importance aux exemples mal classés par les classifieurs déjà sélectionnés.
Le RankBoost classique est un processus d'apprentissage produisant un classifieur fort à partir de classifieurs faibles et d'images annotées. Il ne prévoit pas directement l'évolution de l'ensemble des images d'apprentissage et la mise à jour du classifieur produit.

2.3 Boosting en ligne

Le contexte de recherche interactive diffère du batch learning, composé d'une première phase d'optimisation du classifieur sur une base d'apprentissage et d'une seconde phase de classification des données (hors données d'apprentissage).
Dans le cadre du boosting, on cherche à faire évoluer un classifieur fort H_j en ajoutant successivement les exemples d'apprentissages x_j. Cette problématique peut être exprimée à l'aide d'un fonction d'apprentissage de la forme :

$$App_{online} : ((x_i, y_i), X_j, W, H_j) \rightarrow (X_{j+1}, H_{j+1}) \quad (2)$$

À l'itération $j+1$, la fonction d'apprentissage prend en argument une image x_i et son annotation y_i, l'ensemble des exemples X_j d'apprentissage établis aux itérations précédantes, ainsi que W l'ensemble des classifieurs faibles et H_j la sélection de classifieurs faibles réalisée à l'étape j.
Le OnlineBoost est une méthode de boosting introduite par Oza [14] puis reprise par Grabner [11]. Contrairement aux approches traditionnelles qui nécessitent de parcourir tous les exemples d'apprentissage avant de pouvoir définir le classifieur fort, l'approche "online" cherche à optimiser un classifieur fort par des mises à jour successives. Cette méthode cherche à éviter de relancer tout le processus d'apprentissage à chaque ajout d'un nouvel exemple. Elle consiste à mettre à jour les classifieurs et leurs sélections.
Grabner [11] optimise le processus de mise à jour en introduisant un concept d'ensemble de classifieurs faibles. On cherche, dans cette méthode, à optimiser N ensembles de classifieurs faibles dans lesquels on va effectuer la sélection de chaque nouveau classifieur. On peut résumer l'algorithme comme suit : à chaque exemple on sélectionne un classifieur faible dans chacun des N ensembles de classifieurs faibles. Le critère de choix est une approximation de celui utilisé dans Adaboost. L'idée est de s'en rapprocher de plus en plus à chaque ajout d'un nouvel exemple. On utilise pour cela des variables $\lambda_{correct}$, λ_{wrong} pour estimer l'erreur du classifieurs faible étudié et une variable λ pour prendre en compte la difficulté de l'exemple en cours d'analyse.
Cette approche est utilisée dans le domaine du suivi d'objet car elle permet non seulement de prendre en compte des changements et des mises à jour des objets, mais aussi et surtout d'obtenir un nombre important d'exemples d'apprentissage successifs, des objets à détecter, de par la redondance intrinsèque de l'information dans le contexte de suivi en vidéo.
Cet usage est différent de l'approche étudiée dans notre contexte. En effet, les résultats avec peu d'images dépendent de la difficulté de la classe recherchée et il est souvent préférable d'effectuer un pré-apprentissage [14]. Cela est dû, entre autres, à la difficulté d'établir une frontière de décision des classifieurs faibles avec un nombre d'exemples restreint.
Nous proposons, dans cette article, une méthode innovante permettant de tirer partie de ces approches pour définir un algorithme de recherche interactive par boosting.

3 Algorithme proposé

3.1 Apprentissage

Nous proposons un nouvel algorithme capable de gérer les contraintes de la recherche interactive, notamment les impératifs de rapidité d'exécution et la capacité à travailler sur un ensemble d'apprentissage réduit et évolutif.
L'idée introduite par notre méthode est de construire itérativement l'ensemble des classifieurs faibles à partir des exemples positifs de l'utilisateur (cf Fig. 1).
L'algorithme débute avec un ensemble vide W_0 de classifieurs et à chaque ajout d'exemple positif, nous construisons de nouveaux classifieurs faibles produit à partir des caractéristiques visuelles de l'image annotée positivement par l'utilisateur.
On fait ainsi évoluer l'ensemble W_j des classifieurs faibles à chaque ajout d'exemple positif selon la formule :

$$\begin{cases} W_0 & = \varnothing \\ W_{j+1} & = W_j \cup \mathcal{H}_{x_i} \end{cases} \quad (4)$$

Avec $\mathcal{H}_{x_i}$ l'ensemble des classifieurs faibles générés par l'image x_i.

$$\mathcal{H}_{x_i} = \{h_{k,x_i}\}_k \quad (5)$$

On lance ensuite un RankBoost classique sur l'ensemble d'apprentissage formé par les annotations successives X_j de l'utilisateur et l'ensemble évolutif W_j de classifieurs faibles.
L'algorithme que nous proposons est présenté plus en détail dans **Algorithme 1**.
L'usage de classifieurs faibles directement liés aux données permet de restreindre l'ensemble des classifieurs et d'ignorer tout ceux n'ayant pas de liens directs avec les images apprises. Cela permet de réduire le nombre de classifieurs à étudier et contribue à diminuer le temps dédié à l'apprentissage.

Algorithm 1 après j itération

Require: un exemple d'apprentissage $x_i \in X_p \cup X_n$

Require: un ensemble de classifieurs faibles W_{j-1} intialisé à $\varnothing$

Initialisation de la distribution des exemples

$$\nu_0(x_i) = \begin{cases} \dfrac{1}{|X_p|} \text{ si } x_i \text{ est un exemple positif } (x_i \in X_p) \\ \dfrac{1}{|X_n|} \text{ si } x_i \text{ est un exemple négatif} (x_i \in X_n) \end{cases}$$

if $x_i \in X_p$ **then**

  $W_j = W_{j-1} \cup \mathcal{H}_{x_i}$

else

  $W_j = W_{j-1}$

end if

for $t = 0$ to T **do**

  Trouver le classifieur $h_t \in W_j$ maximisant le score des classifieurs faibles :

$$r_t(h) = \sum_{x_p \in X_p} \nu_t(x_p) h(x_p) - \sum_{x_n \in X_n} \nu_t(x_n) h(x_n) \tag{3}$$

  Calculer le poids du classifieur sélectionné :

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 + r_t}{1 - r_t} \right)$$

  Mettre à jour la distribution des exemples :

$$\nu_{t+1}(x_i) = \begin{cases} \dfrac{\nu_t(x_i) \mathrm{e}^{-\alpha_t h_t(x_i)}}{\sum_{x_p \in X_p} \nu_t(x_p) \mathrm{e}^{-\alpha_t h_t(x_p)}} \text{ si } x_i \in X_p \\ \dfrac{\nu_t(x_i) \mathrm{e}^{\alpha_t h_t(x_i)}}{\sum_{x_n \in X_n} \nu_t(x_n) \mathrm{e}^{\alpha_t h_t(x_n)}} \text{ si } x_i \in X_n \end{cases}$$

end for

return la somme pondérée des classifieurs sélectionnés :

$$H(x) = \sum_{t=0}^{T-1} \alpha_t h_t(x)$$

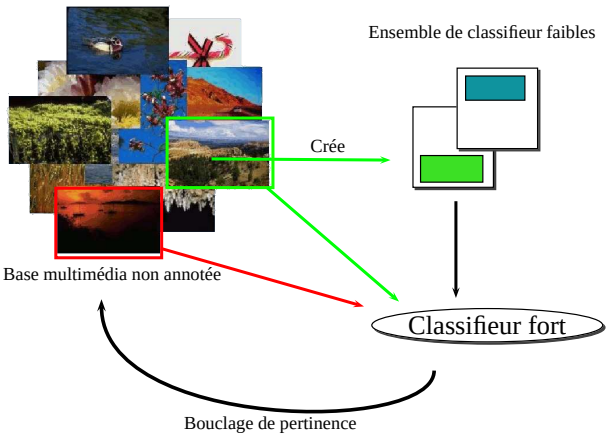

FIG. 1 – Principe de l'algorithme

Les choix des exemples d'apprentissage influe donc directement sur l'évolution du classifieur fort, puisqu'il conditionne la présence des classifieurs faibles dans l'ensemble des classifieurs sélectionnables. On cherche donc à annoter les exemples les plus pertinents afin de faire évoluer le système le plus rapidement possible.

3.2 Apprentissage actif

L'idée de notre stratégie active est de prendre les exemples d'apprentissage non annotés et de suggérer à l'utilisateur, les exemples qui, s'ils étaient annotés positivement, apporteraient les classifieurs faibles qui classeraient au mieux les exemples connus du système.

Nous proposons de choisir les images qui produisent les classifieurs faibles maximisant le critère de sélection r_t de l'algorithme (cf. Eq. 3) dans le cas où les exemples ont la même importance ($t = 0$).

On cherche ainsi les images $x_{i^*} \notin X_p \cup X_n$ correspondant à :

$$i^* = \arg\max_i \left(\max_{h \in \mathcal{H}_{x_i}} r_0(h) \right) \tag{6}$$

La sélection des classifieurs faibles, qui formeront le classifieur fort final, dépend entièrement du critère r_t. En cherchant les images contenant des classifieurs maximisant ce critère, on cherche à introduire les classifieurs au plus grand potentiel dans notre ensemble de classifieurs sélectionnables. Ce critère permet également de trouver des images négatives ayant des caractéristiques très proche des images positives. On peut donc s'attendre à améliorer à la fois notre ensemble de classifieurs faibles et notre ensemble d'exemples avec des images négatives difficiles à classer.

4 Les classifieurs faibles proposés

On a introduit précédemment un ensemble de classifieurs faibles $\mathcal{H}_{x_i}$ pour chaque image. Nous proposons dans cette partie deux types de classifieurs faibles possibles. L'en-

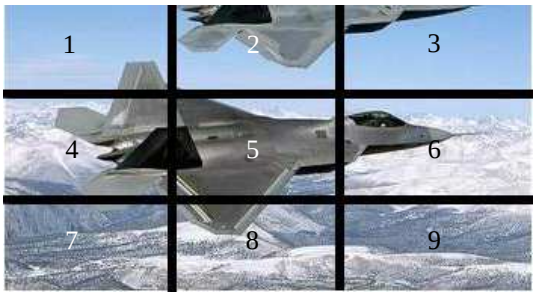

FIG. 2 – Exemples de régions pour un classifieur faible

semble $\mathcal{H}_{x_i}$ peut être découpé selon ces deux types de classifieurs.

$$\mathcal{H}_{x_i} = \mathcal{H}^1_{x_i} \cup \mathcal{H}^2_{x_i} \tag{7}$$

Les classifieurs utilisés dans notre contexte, doivent répondre à certaines conditions de vitesse et de qualité.
L'ensemble du processus d'apprentissage s'appuie sur une comparaison d'un grand nombre de classifieurs faibles. Même si le nombre est plus réduit par notre processus itératif, il reste suffisamment important pour ne pas négliger le temps d'évaluation de chaque classifieur faible. La rapidité de ces derniers conditionne l'usage interactif.
Lors d'un apprentissage interactif, l'utilisateur s'attend généralement à trouver en tête de classement les images qu'il a annotés positivement. Les classifieurs doivent donc être suffisamment discriminant pour retrouver ces images.
On introduit donc par la suite deux types de classifieurs faibles, répondant à ces conditions. Ils utilisent un mécanisme simple permettant de garantir un temps d'exécution restreint et classent de façon optimale les images dont ils sont issus.

4.1 Extraction de caractéristique de l'image

Chaque image est découpée en 9 zones z_l selon une grille 3x3. Dans chacune de ces zones, on calcule un histogramme d'une caractéristique (ex : un histogramme de couleur).
Nous considérons aussi les régions $\rho_m = \bigcup_l z_l$ correspondant à des combinaisons de zone z_l.
Le calcul de l'histogramme de la région correspond alors à la combinaison des histogrammes de chacune des zones la composant.
On définit également Z, l'ensemble des zones et R l'ensemble des régions possibles.
La figure (Fig. 2) illustre ces notions. Chaque case numérotée correspond à une zone. Tandis que la combinaison des zones 2,5,7 forme une région ($\rho_{257} = z_2 \cup z_5 \cup z_7$).

4.2 Les classifieurs de type 1

$\mathcal{H}^1_{x_i}$ est l'ensemble de tous les classifieurs de type 1 que l'on peut générer avec l'image x_i.

$$\mathcal{H}^1_{x_i} = \{h^1_{k,x_i}\}_k \tag{8}$$

Chaque classifieur de type 1 est composé d'une région ρ_k et d'une image génératrice x_i.
Le classifieur correspond à la fonction de classement :

$$h^1_{k,x_i} : x_j \longmapsto 1 - d(histo_{\rho_k}(x_i), histo_{\rho_k}(x_j)) \tag{9}$$

Il effectue la comparaison entre l'histogramme de la région ρ_k dans l'image de référence x_i et dans l'image à tester x_j.
On utilise pour cela une distance du χ^1 définit par :

$$d(g,x) = \frac{1}{M} \sum_{p=1}^{M} \frac{|g_p - x_p|}{g_p + x_p} \tag{10}$$

M est la taille des histogrammes g et x. g_p et x_p correspondent quant à eux, à chaque composante de ces histogrammes.
On peut remarquer que l'image de référence a un score maximal pour tous les classifieurs qu'elle génère.

4.3 Les classifieurs de type 2

$\mathcal{H}^2_{x_i}$ est l'ensemble de tous les classifieurs de type 2 que l'on peut générer avec l'image x_i.

$$\mathcal{H}^2_{x_i} = \{h^2_{k,x_i}\}_k \tag{11}$$

Un classifieur de type 2 est composé uniquement des données d'une zone d'une image de référence. Il correspond à la fonction de classement :

$$h^2_{k,x_i}(x_j) = 1 - \min_{z_{k'} \in Z} d(histo_{z_k}(x_i), histo_{z_{k'}}(x_j)) \tag{12}$$

Ces classifieurs comparent chaque zone de l'image de test x_j à une zone donnée z_k de l'image de référence x_i.
La distance d utilisée est la même que précédemment (Eq. 10).

5 Expérimentations et résultats

5.1 Protocole expérimental

Nous avons utilisé pour nos tests les images de la base Voc2006 [6]. Cette base est constituée de 5304 images provenant principalement des sites Microsoft Research Cambridge et Flickr. Elle contient 9507 annotations décomposées en 10 classes d'objets, certaines images pouvant contenir plusieurs objets éventuellement de classes différentes. Les images sont réparties en deux ensembles disjoints, les images d'apprentissage et les images de test.
Le protocole mis en place, pour cette article, donne une estimation des performances auxquelles un utilisateur peut s'attendre en démarrant une recherche à partir d'une image quelconque de la catégorie qu'il recherche. Il ne s'inscrit pas dans le même contexte que les algorithmes habituellement ([7]) utilisé avec cette base.
Cette estimation est calculée en simulant un grand nombre de sessions de recherche, chacune étant démarrée à l'aide d'une image pris au hasard dans la catégorie. A chaque

	Vélo	Bus	Voiture	Chat	Vache	Chien	Cheval	Moto	Personne	Mouton	Tous
Boost Lab	29,6	37,4	55,4	25,5	36,5	21,0	14,0	31,2	36,4	44,1	33,1
Boost Qw	37,6	38,1	**68,5**	27,1	24,1	21,2	18,8	40,6	36,9	23,9	33,7
Boost Lab|Qw	**44,3**	**44,9**	67,9	**30,0**	**39,5**	**24,7**	**19,0**	**42,1**	**38,3**	**52,8**	**40,3**
SVM Lab	19	22,1	39,4	22,6	29,2	17,1	12,4	19,6	32,2	44,8	25,8
SVM Qw	20,7	24,3	50,7	19,5	12,4	17,6	13,1	25,6	29,8	24,6	23,8
SVM Lab|Qw	34,1	40	57	25,7	31,8	20,4	15,7	34,6	34,5	46,4	34

TAB. 1 – Précision moyenne sur la base VOC2006 en %

bouclage de pertinence, le système annote les images dans la base d'entraînement qui ont été sélectionnées par la méthode active, lance la mise à jour de la classification de la base de test, et enfin mesure la qualité du résultat sur cette dernière. Pour chaque session simulée, 5 images ont été annotés par bouclage de pertinence, et 10 bouclages ont été effectués. A l'issue de chacune des sessions simulées, un total de 51 annotations ont été fournies au système.
Le critère de comparaison utilisé est la précision moyenne de TrecVid (AP).Elle correspond à l'aire sous la courbe précision/rappel et donne un aperçu de la qualité des premiers résultats.
La précision mesure le nombre d'images pertinentes retournées par le système de recherche par rapport au nombre d'images de la base étudiée. Tandis que le rappel mesure le nombre d'images pertinentes retournées par rapport au nombre d'images pertinentes de la base.
L'algorithme présenté est comparé à une méthode à base de SVM [10]. Dans le but d'avoir des résultats comparables, les classifieurs faibles et les noyaux des SVM utilisent les même histogrammes : couleurs CIE L*a*b* et ondelettes quaternioniques [2] (Qw), ainsi que la même distance de comparaison (Eq. 10).
Les histogrammes utilisés pour les deux méthodes sont réalisés avec des dictionnaires de 32 mots visuels.

5.2 Résultats des expériences

Les expériences réalisées (Tab. 1) attribuent de meilleurs résultats à l'algorithme introduit dans cet article. Les résultats sont meilleurs lorsque l'on combine les différents types d'attributs aussi bien en SVM que pour l'algorithme proposé.
On peut également constater que les résultats varient en fonction de la nature de la classe recherchée. Les voitures et les moutons ont de bien meilleures performances de classification que les chevaux ou les chiens.
Ceci peut s'expliquer par la nature des images de la base. La classe mouton contient principalement des images ayant une grande quantité de vert et de blanc, les voitures quant à elles sont essentiellement composées d'images à fort gradient dans des paysages urbains. A contrario les chevaux et les chiens sont des classes très variées, situées aussi bien en intérieur qu'à l'extérieur, dans des villes ou à la campagne… Ce sont des images très différentes qui ont donc peu d'attributs communs. Ces classes sont donc plus difficilement cernables avec peu d'exemples.

6 Conclusion

Nous avons présenté dans cette article, un nouvel algorithme de boosting appliqué au contexte de la recherche interactive. Cette méthode permet de traiter les problèmes liés à ce contexte d'apprentissage, dont le faible nombre d'exemples. De plus, elle assure une mise à jour très rapide de la classification, quasi imperceptible pour l'utilisateur sur une base de 10 000 images.
Pour ce faire, nous proposons de construire l'ensemble des classifieurs faibles de manière itérative et d'appliquer une méthode active pour sélectionner les exemples à annoter. Cette approche s'appuie sur une représentation de chaque image d'apprentissage par un ensemble de classifieurs faibles. Les types de classifieurs faibles proposés possèdent une bonne capacité discrimante, ce qui contribu à produire de bons résultats même avec très peu d'exemples.
Un algorithme de boosting sélectionne et pondère les classifieurs faibles les plus intéressants parmis ceux extraits des images annotées positivement.
Enfin, une méthode active innovante est proposée pour sélectionner les images qui, une fois annotées, contriburont à la fois à enrichir l'ensemble des classifieurs faible et à améliorer la qualité de la classification.
Nous avons comparé cette approche à une autre méthode de recherche interactive utilisant des SVM. Nous pouvons en déduire un net avantage en faveur de cette nouvelle approche.

Références

[1] Steven Abney, Robert E. Schapire, and Yoram Singer. Boosting applied to tagging and pp attachment. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 38–45, 1999.

[2] E. Chang, B. T. Li, G. Wu, and K.S. Goh. Statistical learning for effective visual information retrieval. In *IEEE International Conference on Image Processing*, pages 609–612, Barcelona, Spain, September 2003.

[3] B. Collins, J. Deng, K. Li, and L. Fei-Fei. Towards scalable dataset construction : An active learning approach. In *European Conference on Computer Vision (ECCV)*, 2008.

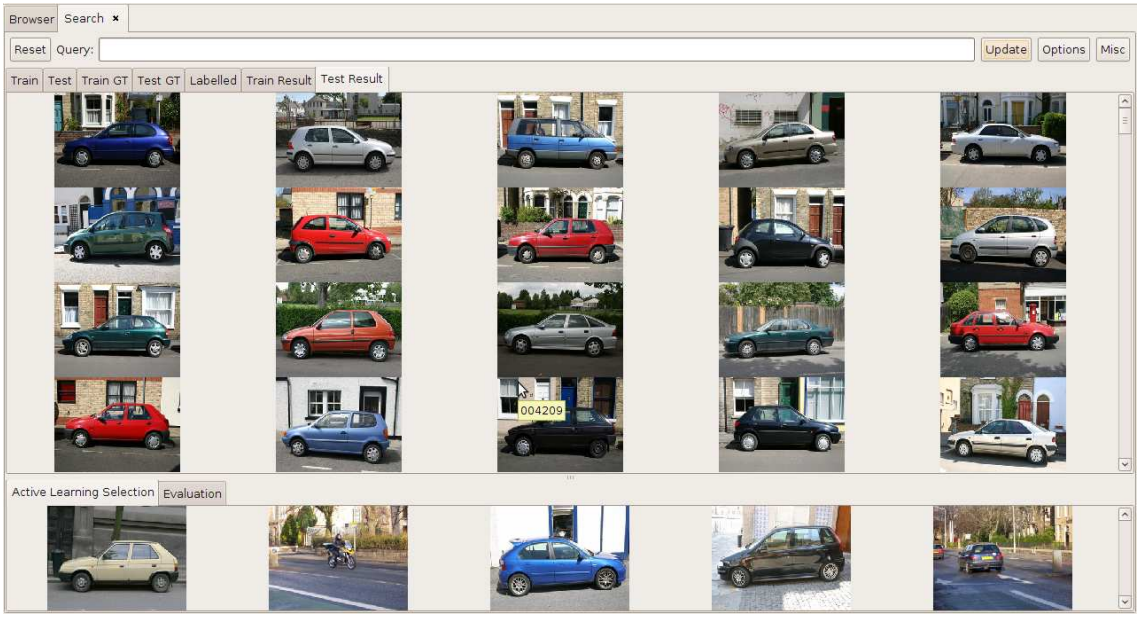

FIG. 3 – Résultat pour la détection de voiture. La dernière ligne présente les propositions d'image par la méthode active.

[4] Marcel Dettling and Peter Buhlmann. Boosting for tumor classification with gene expression data. *Bioinformatics*, 19(9) :1061–1069, June 2003.

[5] Lili Diao, Keyun Hu, Yuchang Lu, and Chunyi Shi. A method to boost support vector machines. In *PAKDD '02 : Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 463–468, London, UK, 2002. Springer-Verlag.

[6] Mark Everingham. The pascal visual object classes challenge 2006 development kit, 2006.

[7] Mark Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. http ://www.pascal-network.org/challenges/VOC/voc2006/results.pdf.

[8] Yoav Freund, Raj Iyer, Robert E. Schapire, Yoram Singer, and G. Dietterich. An efficient boosting algorithm for combining preferences. In *Journal of Machine Learning Research*, pages 170–178, 2003.

[9] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *In Proceedings of the thirteenth international conference on machine learning*, pages 148–156. Morgan Kaufmann, 1996.

[10] P.H. Gosselin, M. Cord, and Sylvie Philipp-Foliguet. Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval. *CVIU, special issue on Similarity Matching in Computer Vision and Multimedia*, (110/3) :403–417, 2008.

[11] Helmut Grabner and Horst Bischof. On-line boosting and vision. In *CVPR '06 : Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 260–267, Washington, DC, USA, 2006. IEEE Computer Society.

[12] I. Laptev. Improving object detection with boosted histograms. *Image and Vision Computing*, 2008.

[13] Y.J. Lu, Q. Tian, and T.S. Huan. Interactive boosting for image classification. pages 315–324, 2007.

[14] Nikunj C. Oza. *Online Ensemble Learning*. PhD thesis, The University of California, Berkeley, CA, Sep 2001.

[15] Robert E. Schapire. The strength of weak learnability. *Journal of Machine Learning*, 5 :197–227, 1990.

[16] Robert E. Schapire and Yoram Singer. Boostexter : A boosting-based system for text categorization. *Machine Learning*, 39(2/3) :135–168, 2000.

[17] Robert E. Schapire, Yoram Singer, and Amit Singhal. Boosting and rocchio applied to text filtering. In *In Proceedings of ACM SIGIR*, pages 215–223. ACM Press, 1998.

[18] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. In *Journal of Machine Learning Research*, volume 2, pages 999–1006, 2000.

[19] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. 2001.

[20] L. Wolf and I. Martin. Robust boosting for learning from few examples. pages I : 359–364, 2005.

[21] Lei Wang Xuchun Li and Eric Sung. Improving adaboost for classification on small training sample sets with active learning. Korea, 2004. The Sixth Asian Conference on Computer Vision (ACCV).