



HAL
open science

Classification rapide de pixels par SVM dans un espace couleur hybride

G Lebrun, C Charrier, O Lezoray, C Meurie, H Cardot

► **To cite this version:**

G Lebrun, C Charrier, O Lezoray, C Meurie, H Cardot. Classification rapide de pixels par SVM dans un espace couleur hybride. *COMpression et REprésentation des Signaux Audiovisuels - CORESA*, Nov 2005, Rennes, France. pp.301-306. hal-04762114

HAL Id: hal-04762114

<https://hal.science/hal-04762114v1>

Submitted on 31 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Classification rapide de pixels par SVM dans un espace couleur hybride

G. Lebrun¹

C. Charrier¹

O. Lezoray¹

C.Meurie¹

H.Cardot²

¹LUSAC EA 2607, groupe Vision et Analyse d'Image
120 Rue de l'exode, F-50000 Saint-Lô, France

{gilles.lebrun, c.charrier, o.lezoray, cyril.meurie}@chbg.unicaen.fr

²Laboratoire Informatique (EA 2101), Université François-Rabelais de Tours
64 Avenue Jean Portalis, Tours, F-37200, France

{hubert.cardot@univ-tours.fr}

Mots clefs

SVM, Quantification Vectorielle, Methode Tabou, Espace couleur hybride, Classification de pixels

Résumé

Dans cet article nous présentons une nouvelle méthode d'apprentissage qui permet de produire des fonctions de décision binaire de complexités réduites avec des SVMs (Support Vector Machines). Cette méthode sélectionne aussi un espace couleur hybride approprié pour chaque fonction de décision produite. L'objectif est de permettre la construction d'un classificateur de pixels rapide et performant en généralisation à base de SVM. La quantification vectorielle est utilisée dans notre méthode pour simplifier la base d'apprentissage en remplaçant les pixels la constituant par un ensemble réduit de prototypes. La création de fonctions de décision performantes et de complexités réduites correspond à la recherche d'un modèle optimal qui regroupe à la fois le niveau de simplification de la base d'apprentissage et le choix d'un espace couleur hybride dédié, ainsi que les hyperparamètres des SVMs. La recherche Tabou est utilisée dans notre méthode pour la sélection d'un modèle quasi-optimal dans des temps exploitables. Des résultats expérimentaux montrent l'efficacité de notre méthode.

Mots clefs

SVM, Quantification Vectorielle, Methode Tabou, Espace couleur hybride, Classification de pixels

1 Introduction

La classification de pixels est régulièrement utilisée comme étape initiale dans des schémas de segmentation d'images couleur [1, 2]. Comme pour tout problème de classification supervisée, le choix d'un algorithme produisant par induction une Fonction de Décision (FD) performante en généralisation à partir d'une base d'apprentissage est crucial. Mais dans le cadre de la classification de pixels le taux de reconnaissance ne peut pas être le seul critère

pour évaluer la Qualité de la FD (QFD). En effet, la FD est exploitée pour chaque pixel de l'image. Dans ce cas une FD ayant un taux de reconnaissance performant, mais dont le temps d'évaluation est très important ne peut pas être utilisée dans le cadre de la classification de pixels, surtout si le nombre d'images à segmenter dans un temps restreint est important.

Les SVMs [3] sont des algorithmes très performants pour produire des FD qui ont une forte capacité en généralisation. Cependant, leur principal défaut réside dans une augmentation de la complexité des FD Binaires (FDB) produites en fonction de la taille de la base d'apprentissage [4, 5]. L'utilisation des SVMs sur des bases de pixels n'est donc pas directement exploitable pour la classification de pixels.

Nous avons alors défini une nouvelle méthode d'apprentissage qui permet d'utiliser des SVMs dans le cadre de la classification de pixels. Cette méthode utilise la Quantification Vectorielle (QV) [6] pour simplifier la base d'apprentissage et ainsi réduire la complexité des FDBs produites par les SVMs. Il est à noter que le choix d'un espace colorimétrique ou bien la création d'un espace couleur hybride à partir d'un ensemble d'espaces colorimétriques permet dans la plupart des cas d'améliorer la classification de pixels par rapport à l'utilisation de l'espace couleur initial des images [1]. Notre méthode produit aussi un espace couleur hybride approprié pour chaque FDB produite par les SVMs. La création d'un espace couleur hybride est réalisée par sélection d'un ensemble de composantes couleurs issues de divers espaces colorimétriques [1]. Le mécanisme utilisé dans notre méthode pour la sélection des composantes couleur est similaire à celui couramment utilisé dans le cadre de la sélection d'attributs [7].

La QFD pour la classification de pixels dépend du taux de reconnaissance de la FD, mais aussi de la complexité de la FD. Un nouveau critère permettant d'évaluer la QFD adapté à la classification de pixels et tenant compte des deux précédentes termes a alors été défini. Pour chaque FDB produite par les SVMs, notre méthode doit donc choisir les valeurs des hyperparamètres des SVMs, le niveau de

simplification de la base d'apprentissage et l'espace couleur hybride afin d'optimiser la QFD. Cela revient à rechercher un modèle constitué d'un ensemble de variables dont les valeurs permettent de déterminer l'ensemble des choix à réaliser pour cette optimisation. Comme la recherche d'un modèle optimal à partir d'une exploration exhaustive n'est pas pratiquement réalisable, nous avons choisi d'utiliser comme métaheuristique la Recherche Tabou (RT) pour ses bonnes performances en général [8]. La combinaison de l'utilisation des SVMs, de la QV, d'un espace couleur hybride, d'un nouveau critère de qualité et de la RT a permis de définir une méthode d'apprentissage qui produit des FDB performantes et de complexités réduites directement exploitables pour la classification de pixels.

2 Machine à Vecteurs de Support

La théorie de l'apprentissage statistique de Vapnik et de Chervonenkis [3] a conduit au développement d'une classe d'algorithmes connue sous le nom de *Support Vector Machines*. Une des originalités de l'algorithme des SVMs est de produire une FDB qui n'utilise qu'un sous-ensemble de la base d'apprentissage. Les éléments de ce sous-ensemble sont nommés Vecteurs de Support (VS). Soit une base d'apprentissage $S_a : S_a = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$ composée de m couples (vecteur d'attributs, classe) avec $x_i \in \mathcal{R}^n$ et $y_i \in \{-1, +1\}$. L'algorithme des SVMs projette les vecteurs x_i dans un espace de travail \mathbf{H} ($\phi : \mathcal{R}^n \rightarrow \mathbf{H}$). L'hyperplan optimal de séparation des deux classes dans l'espace \mathbf{H} est ensuite recherché. Cet hyperplan (w, b) matérialise la frontière de séparation entre les deux classes. La classe y d'un nouvel exemple \mathbf{x} est définie par : $y = \text{sign}(w \cdot \phi(\mathbf{x}) + b)$. L'hyperplan est optimal s'il maximise la distance qui le sépare des exemples dont il est le plus proche. Cette distance est usuellement appelée marge du classificateur. Il a été démontré [3] que maximiser cette marge correspond à maximiser le «pouvoir» généralisateur du classificateur. Vapnik [3] a proposé une version duale du problème qui revient à minimiser la fonction $\mathcal{W}(\alpha) = \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^m \alpha_i$ sous les contraintes $\forall i \in [1, \dots, m] : \sum_{i=1}^m y_i \alpha_i = 0, 0 \leq \alpha_i \leq C$. L'avantage de cette formulation porte sur l'utilisation d'une fonction noyau K respectant l'égalité $K(x_i, x_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$. Ce qui évite de travailler directement dans l'espace \mathbf{H} qui a généralement une dimension largement supérieure à n . Le compromis entre erreur de classification et complexité du modèle dépend du paramètre C [3] (un des hyperparamètres des SVMs). L'algorithme SMO [4] a été proposé par Platt pour résoudre le problème dual de manière efficace. La solution optimale (α^*, b^*) définit l'ensemble VS des vecteurs de support ($\mathbf{VS} = \{i \mid i \in [1, \dots, m], \alpha_i > 0\}$) et permet de définir la fonction [3] :

$$f(x) = \sum_{i \in \mathbf{VS}} \alpha_i^* y_i K(x_i, x) + b^* \quad (1)$$

Le choix de la classe pour un problème binaire dépend du signe de $f : h(x) = \text{sign}(f(x))$. Les SVMs étant des classificateurs binaires, la résolution d'un problème multi-classes est généralement réalisée en le transformant en une combinaison de problèmes binaires [9]. Pour cette étude, nous avons choisi la méthode *un contre tous* qui transforme un problème ayant n_c classes en n_c problèmes binaires. Chaque problème correspond alors à identifier la classe i ($i \in [1, \dots, n_c]$) par rapport aux autres classes. Pour chaque problème binaire, l'algorithme des SVMs produit une fonction f^i . Le choix de la classe est finalement décidé par : $h(x) = \arg \max_{i \in [0, \dots, n_c]} f^i(\mathbf{x})$ [9].

3 Quantification vectorielle

La QV est une technique de classification très utilisée dans le domaine de la compression [6]. Elle associe à un vecteur d'entrée \mathbf{x} un vecteur \mathbf{x}' appartenant à un dictionnaire S' contenant m' vecteurs *prototypes*. La construction du dictionnaire S' est réalisée à partir d'une base d'apprentissage S_a ($m' \ll m$). L'algorithme de construction du dictionnaire doit produire un ensemble de prototypes \mathbf{x}' qui minimisent la distorsion d' définie par : $d' = \frac{1}{m} \sum_{i=1}^m \min_{1 \leq j \leq m'} d(\mathbf{x}_i, \mathbf{x}'_j)$ ($d(\cdot, \cdot)$ étant une norme ℓ_2). La méthode la plus couramment utilisée pour la construction du dictionnaire est l'algorithme LBG [6]. Il s'agit d'un algorithme itératif qui produit 2^k prototypes après k itérations tout en minimisant la distorsion d' .

4 Espace couleur hybride

La couleur des pixels des images numérisées est généralement exprimée dans l'espace couleur (R,G,B). Cependant, cet espace couleur n'est pas nécessairement le plus approprié pour réaliser des traitements (e.g. classification de pixels) sur ces images. Un grand nombre d'espaces couleur ont alors été définis. Ils ont des propriétés physiques, colorimétriques ou physiologiques spécifiques [1]. Pour notre étude, nous avons retenu des espaces couleur¹ parmi les plus utilisés : (X, Y_1, Z) , (L^*, a^*, b^*) , (L^*, u^*, v^*) , (L_1, C, H_1) , (Y_2, Ch_1, Ch_2) , (I_1, I_2, I_3) , (H_2, S, L_2) , (Y_3, C_b, C_r) . De plus, pour certains problèmes, il a été montré [1] qu'en combinant des composantes issues de plusieurs espaces couleur, il est possible de créer un espace couleur hybride plus approprié que les espaces couleur initiaux.

Par définition un espace couleur hybride H_E^β :

$$H_E^\beta = \bigcup_{i \in [1, \dots, n_E], \beta_i = 1} C_E^i \quad (2)$$

correspond à un sous-espace comportant n_β composantes issues d'un espace E qui regroupe n_E composantes couleur provenant de e espaces couleur. C_E^i désigne la composante i de l'espace E . Le vecteur β désigne les composantes couleur de E conservées dans l'espace couleur

¹Nous ajoutons des indices à certaines composantes couleur pour différencier celles qui ont le même nom dans différents espaces couleur mais pas les mêmes formules de calcul

hybride (*i.e.* $\beta_i = 1$ si la composante i de l'espace E est conservé dans H_E^β , $\beta_i = 0$ sinon).

Pour notre étude, $e = 9$, $n_E = 25$ et

$$E = (R, G, B, X, Y_1, Z, L^*, a^*, b^*, u^*, v^*, C, H_1, Y_2, Ch_1, Ch_2, I_1, I_2, I_3, H_2, S, L_2, Y_3, C_b, C_r) \quad (3)$$

L'objectif est alors de trouver l'espace couleur hybride H_E^β (la valeur de β) qui améliore la QFD produite par les SVMs.

5 Qualité d'une Fonction de Décision

La QFD q d'une FD pour un modèle θ donné dépend du taux de reconnaissance T_R et de la complexité C_P de la FD h_θ lorsque des contraintes de temps existent. La QFD peut être modélisé par : $q(h_\theta) = T_R(h_\theta) - C_P(h_\theta)$. Dans le cas de la classification de pixels à partir d'une FDB h_θ produite par des SVMs avec une fonction noyau $K(.,.)$ donnée, la complexité C_P dépend du nombre de vecteurs de support n_{VS} et de l'espace couleur hybride utilisé H_E^β (la valeur de β est contenue dans θ). Nous avons choisi de la modéliser par :

$$C_P(h_\theta) = c_{p_1} \log(n_{VS}) + c_{p_2} \log(\text{coût}(\beta)) \quad (4)$$

Les constantes c_{p_1} et c_{p_2} représentent respectivement l'importance du nombre de vecteurs de support et du choix de l'espace couleur hybride ($\text{coût}(\beta)$) dans l'évaluation de la complexité de h_θ . Elles permettent aussi de fixer le compromis entre amélioration du taux de classification et réduction de la complexité.

Les composantes C_E^i avec $i > 3$ pour un pixel d'une image sont déterminées à partir des composantes couleur R, G et B du même pixel par des transformations linéaires ou non linéaires [1]. Le coût en temps de calcul de ces transformations est plus ou moins important suivant la nature de la transformation et le fait qu'elle soit ou non implémenté directement sur une puce électronique. Soit κ_i le coût de la transformation pour obtenir la composantes C_E^i . Le coût associé à l'espace couleur hybride H_E^β est alors défini par :

$$\text{coût}(\beta) = \sum_{i=1}^{n_E} \beta_i \kappa_i$$

6 Recherche Tabou

La recherche tabou est une métaheuristique pour l'optimisation difficile qui s'est développée dans les années 80 [10, 11]. Elle s'est révélée particulièrement efficace et a été appliquée avec succès à de nombreux problèmes [8]. Elle est basée sur une recherche itérative qui choisit dans un voisinage restreint la meilleure solution même si elle est plus mauvaise que celle de l'itération précédente. Une mémoire à court terme est utilisée pour éviter tout cycle visitant périodiquement le même optimum local. À partir des derniers mouvements mémorisés, un ensemble de solutions est considéré comme *tabou*. Un mouvement n'est donc réalisable que vers l'une des solutions voisines qui n'est pas tabou.

Le choix des solutions taboues peut être tout simplement les dernières solutions visitées, mais en général le critère déterminant les solutions taboues à un moment précis de la recherche dépend du problème à optimiser et de la représentation d'une solution. Supposons qu'une solution soit représentée par un ensemble de n' couples (θ_i, v_i) , θ_i et v_i étant respectivement la variable numéro i et une valeur possible pour cette variable. Supposons ensuite, qu'un mouvement élémentaire correspond à augmenter (ou à diminuer) la valeur d'une des variables θ_i en lui affectant la valeur v'_i la plus proche de v_i dans le domaine de θ_i . Il y a alors 2^N solutions proches (*i.e.* $\max_{1 \leq i \leq N} |\theta'_i - \theta_i| \leq 1$) d'une solution visitée. Si le nombre de variables est suffisamment important, le nombre d'itérations où la RT peut rester proche d'un même optimum local, sans passer par une solution déjà visitée, peut devenir supérieur au nombre maximum d'itérations autorisé pour la recherche. Pour éviter ce problème, la règle de decision suivante a été utilisé : une solution est taboue si une de ses variables a une valeur identique à une des variables modifiées parmi les t précédentes solutions. t représente dans ce cas la durée de l'état tabou en nombre d'itération et la mémoire à court terme enregistre les modifications des variables plutôt que les solutions visitées.

Lorsque des caractéristiques de modification définissent l'appartenance d'une solution à l'ensemble tabou, l'ensemble des solutions interdites à chaque itération peut contenir des solutions meilleures que toutes celles déjà visitées. Un mécanisme particulier, appelé l'aspiration, permet de palier cet effet. Il permet de lever le statut tabou d'une solution, sans pour autant produire de cycle dans le processus de recherche. La fonction la plus simple d'aspiration consiste à annuler le statut tabou d'un mouvement si ce dernier permet d'atteindre une solution de qualité supérieure à celle de la meilleure jusqu'alors.

Cette méthode possède donc un ensemble de mécanismes simples dans leurs principes avec une grande liberté de spécification de ces mécanismes pour l'adapter à un problème donné et donc à notre méthode.

7 Nouvelle méthode d'apprentissage

La complexité en temps de calcul de l'algorithme des SVMs pour l'apprentissage augmente rapidement en fonction de la taille de la base d'apprentissage (pour SMO elle se situe entre $O(m^{1,6})$ et $O(m^{2,1})$ [4]). De plus le nombre de vecteurs de support utilisés par la fonction de décision augmente avec la taille du problème [4, 5]. Comme l'objectif de notre méthode est d'avoir une QFD optimale, l'augmentation du nombre de VS n'est intéressante que si elle s'accompagne d'une amélioration significative du taux de reconnaissance. L'idée principale de notre méthode est d'entraîner les SVMs en utilisant des versions simplifiées de la base d'apprentissage. Ceci afin de réduire la complexité des FDB et par effet de cascade les temps d'apprentissage. La simplification de la base d'apprentissage est réalisée à partir de l'algorithme LBG. L'algorithme 1

donne les détails de cette simplification. Le niveau de simplification k ne pouvant pas être fixé facilement de manière arbitraire, une notion importante dans notre méthode est de considérer k comme une variable.

L'optimisation de la QFD en utilisant des SVMs nécessite donc pour une fonction noyau donnée K le choix du niveau de simplification k , de β pour l'espace couleur hybride, de la constante de régularisation C et des éventuels paramètres λ associés au noyau K . La recherche des valeurs de ces variables correspond à la recherche d'un modèle θ . $k_\theta, \beta_\theta, C_\theta$ et λ_θ désigne alors les valeurs des variables extraites du modèles θ . La sélection d'un modèle θ^* qui optimise la QFD est réalisée par une RT tel que défini à la section 6. Pour que le modèle θ soit facilement utilisable par la RT, il doit correspondre à un vecteur de n' valeurs entières. Nous avons utilisé la correspondance suivante : $(\theta_1, \dots, \theta_{n'}) = (\beta_1, \dots, \beta_{n_E}, k, C', \lambda'_1, \dots, \lambda'_{|\lambda|})$ avec $C_\theta = 2^{C'}$, $C' \in [-5, \dots, 15]$ [12]. L'algorithme 2 donne le synopsis de la procédure d'évaluation de la qualité du modèle θ pour une base d'apprentissage S_a . S_e, S_v désignent respectivement la base utilisée pour l'entraînement des SVMs et l'estimation du taux de reconnaissance. Cette dissociation de la base S_a ($|S_e| = \frac{2}{3}|S_a|$, $|S_v| = \frac{1}{3}|S_a|$) est indispensable pour éviter le risque de sur-apprentissage lorsque l'erreur empirique est utilisée pour l'estimation de T_R . La fonction q que doit optimiser la RT est définie par : $q(h_\theta) = \text{QFDSVM}(\theta, S_a)$ L'entraînement des SVMs est réalisé en utilisant une version de l'algorithme SMO présente dans la librairie Torch [13]. Les fonctions noyau K_β utilisées pour l'entraînement des SVMs sont définies à partir d'une distance d_β :

$$d_\beta(x_i, x_j) = \sqrt{\sum_{l=1}^{n_E} \beta_l (x_i^l - x_j^l)^2}, \quad (5)$$

ce qui correspond à utiliser une norme ℓ_2 avec l'espace couleur hybride H_E^β pour la FDB h_θ produite. Pour cette étude, seul les noyaux : $K_\beta^L = d_\beta^2$ et $K_\beta^G = \exp(-d_\beta^2/\lambda_1^2)$ sont utilisés (Pour la RT : $\lambda'_1 \in [-10, \dots, 10]$ et $\lambda_{1\theta} = 2^{\lambda'_1}$ [12]).

Algorithme 1 Simplification(S, k)

```

 $S' \leftarrow \emptyset$ 
pour  $c = 1$  à  $n_c$  faire
   $T = \{x \mid (x, c) \in S\}$ 
  si  $2^k < |T|$  alors
     $T' \leftarrow \text{LBG}(T, k)$ 
  sinon
     $T' \leftarrow T$ 
  fin si
   $S' \leftarrow S' \cup \{(x, c) \mid x \in T'\}$ 
fin pour

```

Algorithme 2 QFDSVM(θ, S_a)

```

 $(S_e, S_v) \leftarrow \text{Discocier}(S_a)$ 
 $S'_e \leftarrow \text{Simplification}(S_e, k_\theta)$ 
 $h_\theta \leftarrow \text{EntraînerSVM}(S'_e, K_{\beta_\theta}, C_\theta, \lambda_\theta)$ 
 $T_R \leftarrow 1 - \text{ErreurEmpirique}(h_\theta, S_v)$ 
 $C_P \leftarrow \text{Complexité}(h_\theta)$ 
retourner  $T_R - C_P$ 

```

8 Expérimentations

Nous avons appliqué notre méthode d'apprentissage à la classification de pixels issus d'images microscopiques [2]. Les bases d'apprentissage S_a et de test S_t sont construites à partir de 4 images couleurs (RGB, 574*752 pixels) microscopiques de tumeur bronchiques. Pour chaque image, une segmentation manuelle : fond (classe 1), cytoplasme (classe 2), noyau (classe 3) est réalisée par un expert. Comme la répartition du nombre de pixels dans chaque classe n'est pas équilibrée (1 : 89%, 2 : 7%, 3 : 4%) dans les images, seul un sous-ensemble des pixels des classes 1 et 2 a été sélectionné aléatoirement pour la constitution de S_a et S_t , ceci afin que chaque classe possède le même nombre d'exemples (≈ 60000 par classe). La sélection de 3 modèles θ^i ($i \in [1, 2, 3]$) est réalisée avec notre méthode à partir des trois bases S_a^i induites par la méthode *un contre tous* [9] appliquée à la base S_a . Pour chaque modèle θ^i une FDB h_θ^i est produite. Pour éviter tous biais lié à la sélection d'un modèle, le taux de reconnaissance est évalué pour une FDB h_θ^i à partir de la base S_t^i (induite à nouveau par la méthode *un contre tous*).

Les figures 1(a) à 1(d) illustrent l'évolution du taux de reconnaissance et du nombre de VS en fonction du niveau de simplification k pour chaque FDB h_θ^i avec un noyau K_β^L (pour chaque FDB la valeur optimale de C est recherchée) et les valeurs choisies pour β correspondant au 9 espaces couleur retenus (cf. section 4). On peut remarquer que l'amélioration de T_R n'est réalisée que pour des valeurs de k petites. De plus, le choix de l'espace couleur qui optimise la QFD est variable suivant le compromis entre la complexité et le taux de reconnaissance. Ces remarques corroborent les choix réalisés pour la définition de notre méthode d'apprentissage.

Les tableaux 2 et 3 illustrent les résultats obtenues avec notre méthode d'apprentissage en utilisant un noyau K_β^L ou un noyau K_β^G . Le tableau 1 donne les différentes configurations utilisées pour les valeurs des constantes. Dans les tableaux 2 et 3 : ECH désigne l'espace couleur hybride utilisé par la FDB, ΔT le temps d'apprentissage, et dans la colonne FD est mentionnée après θ la configuration utilisée. La colonne κ_i représente deux cas : $\kappa_i = 1$ correspond au cas où la transformation est réalisée par une puce électronique et $\kappa_i = T_i/T$ correspond au cas où la transformation est réalisé par un logiciel avec T_i égal au temps nécessaire pour calculer la composante couleur i et $T = \sum_{i \in [1, \dots, n_E]} T_i$. Ces résultats montrent que notre méthode permet de produire des FDB de complexité réduite

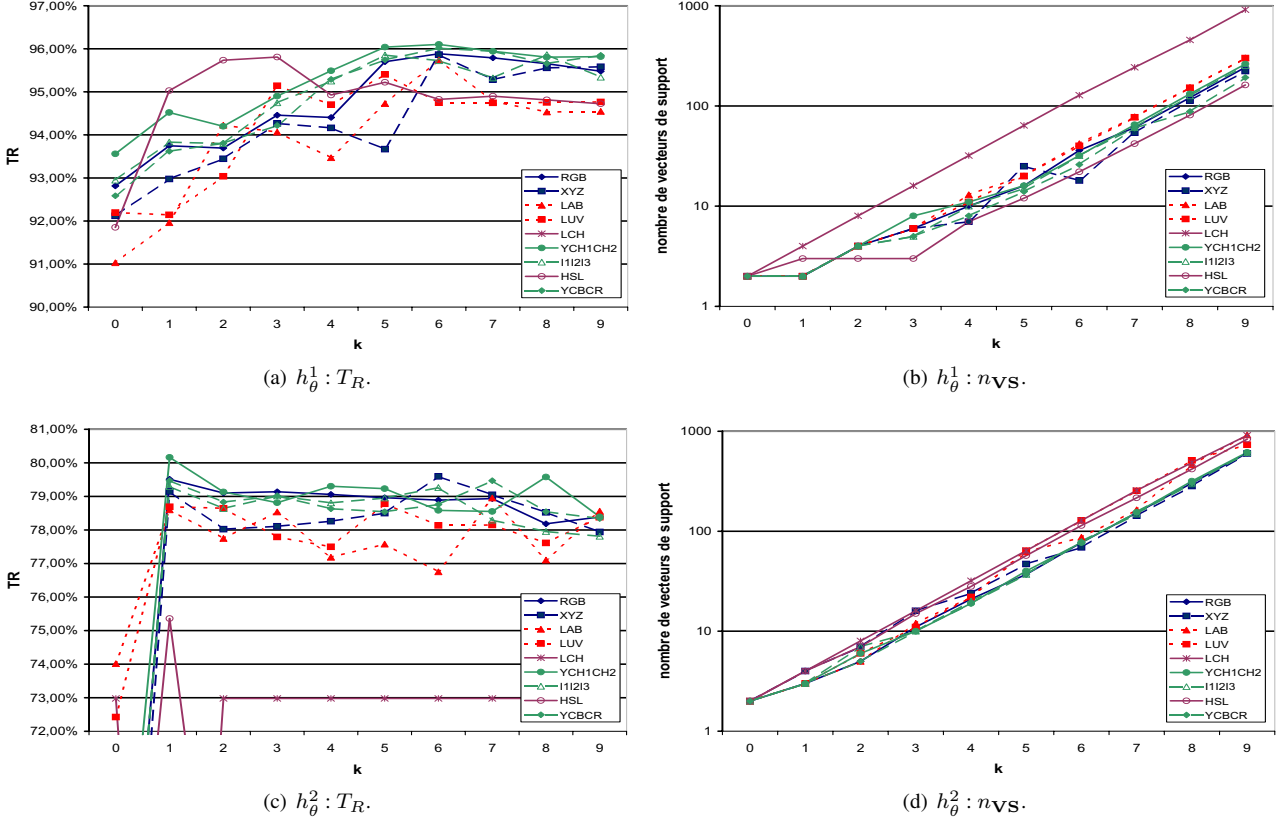


Figure 1 – Taux de reconnaissance T_R et nombre de vecteurs de support n_{VS} en fonction du niveau de simplification k

qui ont de très bonnes performances en généralisation. Le choix d'un espace couleur hybride propre à chaque FDB améliore généralement le taux de reconnaissance, si on ne pénalise pas trop la complexité de la FDB dans le calcul de la QFD. En particulier pour la FDB h_θ^2 avec un noyau K_β^L l'amélioration est très significative ($\approx 5\%$). En effet, pour ce problème, le choix d'utiliser un noyau K_β^G n'améliore pas la reconnaissance et augmente la complexité de la FDB par rapport à l'utilisation d'un noyau K_β^L . Pour un noyau K_β^L on a : $h(x) = d_\beta(x^*, x)^2 + b^*$ avec $x^* = \sum_{i \in VS} \alpha_i y_i x_i$. Le nombre de vecteurs de support ne pénalise donc pas la complexité de la fonction de décision. Bien qu'il semble logique de choisir dans ce cas une valeur nulle ou très faible pour c_{p1} , on peut cependant observer une augmentation importante du temps nécessaire à la sélection d'un modèle sans amélioration significative du taux de reconnaissance (cf. Tab 2 : configuration A et B).

On observe aussi (cf. Tab. 2 et 3 : configuration B et C) que le choix d'un espace hybride peut être optimisé en fonction du coût de sa production. En particulier, dans le cas d'une transformation logiciel, les composantes R, G, B sont plus utilisées et celles nécessitant une transformation non linéaire peu utilisées.

Comme actuellement l'ensemble du processus de segmentation des images microscopiques est entièrement réalisé par un logiciel, nous avons alors choisi d'utiliser les FDB

correspondant à la configuration D avec un noyau K_β^L . Les images 2(b) à 2(e) illustrent les résultats obtenues à partir de cette configuration et de l'image microscopique 2(a). Le temps nécessaire pour produire cette segmentation (P4 2,4 GHz) est inférieur à 1 seconde. La comparaison entre la classification de pixels et la segmentation de l'expert permet de vérifier visuellement l'efficacité de notre méthode.

configuration	c_{p1}	c_{p2}	κ_i
A	0,0001	0,01	1
B	0,01	0,01	1
C	0,03	0,03	1
D	0,01	0,01	T_i/T

Tableau 1 – les valeurs des constantes

9 Conclusion

Une nouvelle méthode d'apprentissage qui produit des FDB adaptées à la classification de pixels en utilisant des SVMs a été présentée. Elle produit des FDB qui ont à la fois l'avantage d'être performantes en généralisation, de complexités réduites et d'utiliser un espace couleur hybride adapté au problème de classification et au coût de sa production. Les travaux futurs devront tester la méthode sur d'autres problèmes de classification de pixels, vérifier l'influence de différents schémas de combinaisons des FDB, en particulier lorsque le nombre de classes est plus important. Ils devront également mesurer l'influence d'autres

FD	T _R	n _{VS}	ECH	k	ΔT
$h_{\theta,A}^1$	96,44%	2	ZC_r	2	144
$h_{\theta,A}^2$	85,99%	48	$Bb^*L_2H_1CH_2SC_b$	5	26574
$h_{\theta,A}^3$	90,48%	63	$Y_1L^*a^*u^*$	6	29540
$h_{\theta,B}^1$	96,50%	2	C_r	2	140
$h_{\theta,B}^2$	84,98%	18	$L^*b^*u^*CI_2S$	4	2352
$h_{\theta,B}^3$	89,79%	2	Gu^*C_b	1	147
$h_{\theta,C}^1$	96,50%	2	C_r	2	140
$h_{\theta,C}^2$	82,82%	3	SC_r	1	179
$h_{\theta,C}^3$	89,73%	2	v^*	1	140
$h_{\theta,D}^1$	95,66%	2	R	2	151
$h_{\theta,D}^2$	83,53%	2	$RBb^*SL_2C_r$	2	397
$h_{\theta,D}^3$	90,09%	4	GB	2	197

Tableau 2 – Résultats avec un noyau K_{β}^L

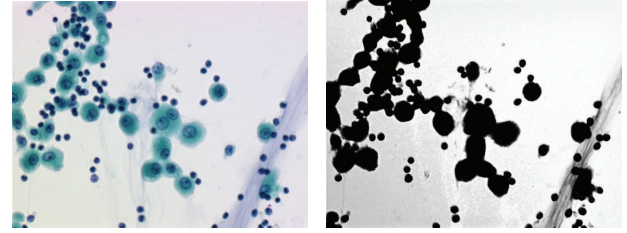
FD	T _R	n _{VS}	ECH	k	ΔT
$h_{\theta,B}^1$	96,08%	5	RH_1	3	865
$h_{\theta,B}^2$	85,90%	10	$RXa^*H_1Y_2Y_3$	3	1239
$h_{\theta,B}^3$	90,43%	4	BYu^*v^*	2	708
$h_{\theta,C}^1$	95,66%	2	R	1	482
$h_{\theta,C}^2$	85,17%	10	$RH_1Y_2Y_3$	3	1223
$h_{\theta,C}^3$	89,47%	2	b^*	0	434
$h_{\theta,D}^1$	95,66%	2	R	1	440
$h_{\theta,D}^2$	85,78%	10	$RY_1Ch_1Y_3C_r$	3	1174
$h_{\theta,D}^3$	90,45%	5	GB	2	409

Tableau 3 – Résultats avec un noyau K_{β}^G

méthodes de simplification de la base d'apprentissage et comparer d'autres métaheuristiques pour l'optimisation de la QFD.

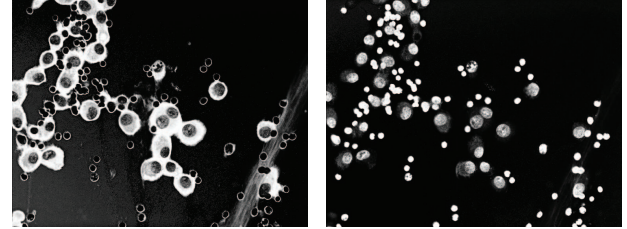
Références

- [1] N. Vandenbroucke, L. Macaire, et J.-G. Postaire. Color image segmentation by pixel classification in an adapted hybrid color space : application to soccer image analysis. *Comput. Vis. Image Underst.*, 90(2) :190–216, 2003.
- [2] C. Meurie, G. Lebrun, O. Lezoray, et A. Elmoataz. A comparison of supervised pixels-based color image segmentation methods. application in cancerology. Dans *WSEAS Transactions on Computers*, volume 2, pages 739–744, 2003.
- [3] V. N. Vapnik. *Statistical Learning Theory*. New York, wiley édition, 1998.
- [4] J. Platt. *Fast Training of Support Vector Machines using Sequential Minimal Optimization, Advances in Kernel Methods-Support Vector Learning*. MIT Press, pp. 185-208, 1999.
- [5] G. Lebrun, C. Charrier, et H. Cardot. SVM training time reduction using vector quantization. Dans *ICPR*, volume 1, pages 160–163, 2004.



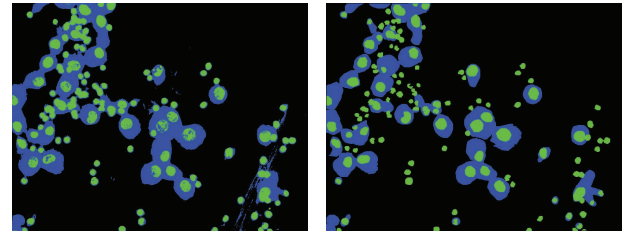
(a) image à segmenter.

(b) h_{θ}^1 : discrimination du fond.



(c) h_{θ}^2 : discrimination des cytoplasmes.

(d) h_{θ}^3 : discrimination des noyaux.



(e) Classification de pixels.

(f) Segmentation de l'expert.

Figure 2 – Classification de pixels produite par la combinaison des fonctions de décision ($h_{\theta}^1, h_{\theta}^2, h_{\theta}^3$) obtenues avec un noyau K_{β}^L et la configuration D.

- [6] A. Gersho et R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic, 1991.
- [7] M. Kudo et J. Sklansky. Comparison of algorithms that select features for pattern classifiers. Dans *Pattern Recognition*, volume 33, pages 25–41, 2000.
- [8] J.-K. Hao, P. Galinier, et M. Habib. Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. Dans *Revue d'Intelligence Artificielle*, volume 13, pages 283–324, 1999.
- [9] C.-W. Hsu et C.-J. Lin. A comparison of methods for multi-class support vector machines. Dans *IEEE Transactions in Neural Networks*, volume 13, pages 415–425, 2002.
- [10] F. Glover. Tabu search : part i. Dans *on Computing*, volume 1, pages 190–206, 1989.
- [11] F. Glover. Tabu search : part ii. Dans *on Computing*, volume 2, pages 4–32, 1989.
- [12] C.-C. Chang et C.-J. Lin. Libsvm : a library for support vector machines. Software Available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- [13] R. Collobert et S. Bengio. SVMtorch : Support vector machines for large-scale regression problems. Dans *Journal of Machine Learning Research*, volume 1, pages 143–160, 2001.