Un environnement efficace pour la classification d'images à grande échelle

Thanh-Nghi Doan*, François Poulet*,**

**Université de Rennes I, *IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France,
{thanh-nghi.doan | francois.poulet}@irisa.fr

Résumé. La plupart des processus de classification d'images comportent trois principales étapes : l'extraction de descripteurs de bas niveaux, la création d'un vocabulaire visuel par quantification et l'apprentissage à l'aide d'un algorithme de classification (eg.SVM). De nombreux problèmes se posent pour le passage à l'échelle comme avec l'ensemble de données ImageNet contenant 14 millions d'images et 21,841 classes. La complexité concerne le temps d'exécution de chaque tâche et les besoins en mémoire et disque (eg. le stockage des SIFTs nécessite 11To). Nous présentons une version parallèle de LibSVM pour traiter de grands ensembles de données dans un temps raisonnable. De plus, il y a beaucoup de perte d'information lors de la phase de quantification et les mots visuels obtenus ne sont pas assez discriminants pour de grands ensembles d'images. Nous proposons d'utiliser plusieurs descripteurs simultanément pour améliorer la précision de la classification sur de grands ensembles d'images. Nous présentons nos premiers résultats sur les 10 plus grandes classes (24,817 images) d'ImageNet.

1 Introduction

La classification d'images est une tâche importante dans le domaine de la vision par ordinateur, la reconnaissance d'objets et l'apprentissage automatique. L'utilisation de descripteurs de bas niveau de l'image et le modèle de sac de mots sont au coeur des systèmes de classification d'images actuels. La plupart des environnements de classification d'images comportent trois étapes : 1) l'extraction de descripteurs de bas niveau dans les images, 2) la création d'un vocabulaire de mots-visuels et 3) l'apprentissage du modèle des classes d'images. Dans la première étape d'extraction des descripteurs de bas niveau, les choix les plus courants dans les méthodes récentes sont les SIFTs (Lowe (2004)), les SURFs (Bay et al. (2006)) ou les DSIFTs (Bosch et al. (2007)). L'étape 2 est la création du vocabulaire visuel, le choix habituel pour cette étape est l'utilisation d'un algorithme de k-means et la création d'un sac de mots. La troisième étape est l'apprentissage du classifieur, beaucoup de systèmes choisissent souvent des Support Vector Machines avec des noyaux linéaires ou non-linéaires. La plupart de ces systèmes sont ensuite évalués sur de petits ensembles de données qui tiennent sans problème en mémoire centrale, comme Caltech-101 (Fei-Fei et al. (2004)), Caltech-256 (Griffin et al. (2007)) ou PASCAL VOC (Everingham et al. (2010)). Cependant l'apparition notamment de l'ensemble

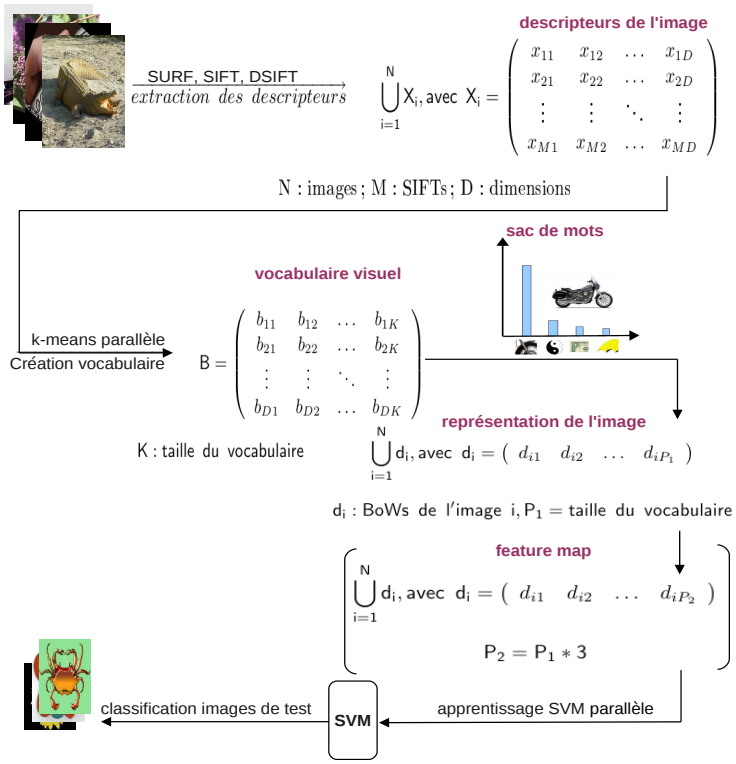

FIG. 1 – *Description de notre environnement pour la classification d'images à grande échelle*

de données ImageNet (Deng et al. (2009)), qui contient plus de 14 millions d'images et 21000 classes rend la tâche de classification d'images beaucoup plus complexe et difficile à effectuer. Ce challenge nous a motivé pour développer un système qui soit à la fois performant en temps de calcul et en précision de la classification. Nous présentons notre solution pour le traitement efficace de tels ensembles de données et les premiers résultats prometteurs par rapport aux approches de l'état de l'art sur l'ensemble ImageNet. Nous proposons un système rapide et efficace pour la classification d'images à grande échelle. Nous avons développé dans ce système une version parallèle de l'algorithme libSVM pour traiter de grands ensembles de données dans un temps raisonnable. De plus, nous proposons une nouvelle approche utilisant plusieurs descripteurs de bas niveaux en les combinant ensembles dans une approche multi-descripteurs et multi-vocabulaires.

2 Etat de l'art

Classification à grande échelle : Les travaux récents en classification d'images ont convergés vers le modèle de sac de mots (Csurka et al. (2004)) basé sur une quantification des descripteurs locaux et des Support Vector Machine comme techniques de base. Ces modèles peuvent

être améliorés par des pyramides spatiales multi-échelles sur les sacs de mots (Lazebnik et al. (2006)) ou des histogrammes de gradients orientés (HoG) (Dalal et al. (2005)). (Fergus et al. (2009)) utilisent un apprentissage semi-supervisé sur Tiny Images (126 classes d'images étiquetées manuellement) et (Wang et al. (2009)) présentent la classification d'un maximum de 315 classes. Li et al. (2009) étudient la classification de paysages dans une collection de 500 classes et 2 millions d'images. Sur un petit sous-ensemble de 10 classes, ils ont amélioré les résultats de la classification en augmentant la taille du vocabulaire visuel jusqu'à 80000 mots visuels. Pour permettre l'apprentissage de grands ensembles de données, plusieurs approches commencent par transformer les données en utilisant une fonction de noyau non linéaire puis en utilisant un classifieur linéaire (rapide) dans l'espace résultant (Deng et al. (2010)), (Perronnin et al. (2010)) Les travaux récents utilisant des hiérarchies pour la reconnaissance ou la catégorisation d'images ont permis des gains importants en précision et efficacité, mais sans étude du surcoût de ces hiérarchies. Lié à la classification, le problème de détection est souvent traité comme une répétition de classifications un contre le reste dans des fenêtres coulissantes. Dans beaucoup de cas cette localisation des objets peut être utile pour améliorer la classification même si les approches les plus performantes (Vedaldi et al. (2009)), (Everingham et al. (2010)) nécessitent des temps de calculs très longs qui les rendent difficiles à utiliser pour des ensembles de données tels qu'ImageNet. La différence entre notre approche et ces approches récentes est la mise en oeuvre d'algorithmes parallèles pour accélérer trois processus : l'extraction des descripteurs de bas niveau dans les images, la quantification et l'apprentissage du classifieur. Nos expérimentations montrent des premiers résultats prometteurs et confirment que les algorithmes parallèles sont essentiels pour la classification d'images à grande échelle en terme de temps d'exécution.

Représentation d'images : Les SIFTs et le modèle de sac de mots sont au coeur des systèmes de classification actuels. La représentation d'une image par le modèle de sac de mots requiert trois étapes non triviales : 1) la détection de points d'intérêt, 2) la description de ces points et 3) la quantification. Ces trois étapes ont fait l'objet de progrès significatifs ces dernières années. Cependant lors de chacune des étapes, une part significative d'information est perdue et les résultats obtenus par les approches utilisant le sac de mots ne sont souvent pas assez discriminantes pour la classification d'images à grande échelle. Différentes approches ont été proposées pour améliorer le pouvoir discriminant lors des différentes étapes. Dans l'étape de détection de points d'intérêts, plusieurs détecteurs sont combinés pour une meilleure discrimination. Pour le codage des descripteurs, on utilise des descripteurs de plus grande dimension ou encodant plus d'informations ((Winder et al. (2007)). Enfin dans l'étape de quantification, différentes approches ont été utilisées pour réduire l'erreur de quantification ou mieux préserver l'information des descripteurs (Moosmann et al. (2006)), (Philbin et al. (2008)). Nous prenons une vision plus globale de ces trois étapes et nous proposons une nouvelle approche qui combine à la fois de multiples descripteurs et de multiples vocabulaires visuels pour la représentation des images. Notre but est d'améliorer le pouvoir discriminant de la représentation des images en encodant plus d'information que les simples descripteurs. Dans l'approche multiples descripteurs et multiples vocabulaires visuels, tous les descripteurs et leurs vocabulaires sont utilisés pour construire les sacs de mots. On obtient donc un sac de mots pour chaque descripteur utilisé et on appelle l'ensemble un sac de paquets. Les différents sacs de mots obtenus sont alors concaténés pour obtenir la représentation finale de l'image, comme montré sur la fig.3. Cette nouvelle approche utilise donc simultanément de multiples descripteurs et

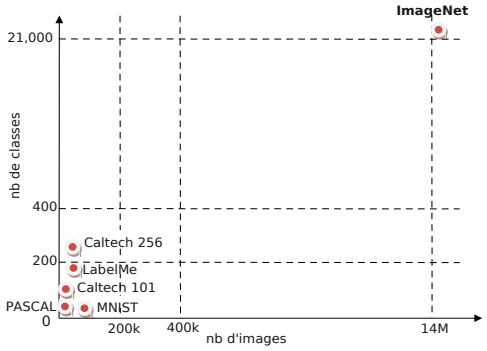

FIG. 2 – *Comparaison d'ImageNet et des autres ensembles de données d'images*

multiples vocabulaires visuels pour améliorer le pouvoir discriminant de cette représentation d'images pour la classification de grands ensembles de données d'images.

3 Ensembles de données d'images

Il existe de nombreux ensembles de données d'images tels que Caltech-101, Caltech-256, Pascal-VOC, etc. Cependant, il y en a beaucoup moins de grandes tailles avec un nombre important d'images et de classes. On peut citer Tiny-Images (Torralba et al. (2008)) qui est un ensemble d'images de 32x32 pixels obtenu par des requêtes sur les noms de WordNet (Fellbaum (1998)), sans vérification a posteriori du contenu ou ImageNet obtenu de la même manière et dont les étiquettes ont été vérifiées manuellement. Il est nécessaire de construire des outils de classification qui soient capables de traiter un grand nombre de classes. Caltech-101 et Caltech-256 ont été les premiers ensembles de données dans cette direction. La dernière version d'ImageNet va beaucoup plus loin avec 21841 classes et plus de 14 millions d'images. Il est bien entendu nécessaire d'avoir un nombre suffisant d'images par classe pour couvrir les variations d'illumination, de points de vues ou les différences d'apparences au sein d'une même classe.

4 Caractéristiques de bas niveaux des images

Comme montré sur la fig.1, à partir d'un ensemble de données d'images, notre système extrait les SIFTs, SURFs et DSIFTs dans ces images. Ces caractéristiques de bas niveau ont été utilisées avec succès dans différentes applications en reconnaissance d'objets, analyse de textures ou classification de scènes.

4.1 SIFT

David Lowe a présenté un descripteur robuste appelé SIFT (Scale Invariant Feature Transform), qui approxime le laplacien de gaussiennes en utilisant un filtre de différence de gaus-

siennes. Le SIFT consiste en quatre étapes : la détection d'extrema, la localisation des points d'intérêts, le calcul de l'orientation et du descripteur du point d'intérêt. La premiére étape utilise une fonction de différences de gaussiennes à la place du laplacien pour améliorer la vitesse de calcul. Dans l'étape de localisation du point d'intérêt, les points ne présentant que peu de contraste sont rejetés. Une matrice hessienne est utilisée pour calculer la courbure principale et éliminer les points qui ont un ratio par rapport à la courbure principale supérieur à un seuil. Un histogramme d'orientation est calculé à partir de l'orientation des gradients d'un échantillon de points autour du point d'intérêt. D'après les expériences décrites par l'auteur, les meilleurs résultats sont obtenus avec une grille 4x4 et une orientation discrétisée en 8 valeurs. Les descripteurs SIFTs sont donc des vecteurs de 4 x 4 x 8 = 128 dimensions.

4.2 SURF

Herbert Bay a présenté un nouveau détecteur de points d'intérêts robustes aux changements d'échelle et rotations appelé SURFs (Speeded Up Robust Features). Il approxime et obtient même de meilleurs résultats que les approches précédentes et il peut être calculé et comparé beaucoup plus rapidement. Le calcul des SURFs utilise des détecteurs hessiens rapides (approximation des ondelettes de Haar).

4.3 SIFT denses

Les SIFTs denses sont une variante des SIFTs avec des descripteurs extraits à de multiples échelles. C'est à peu près équivalent au calcul des SIFTs sur une grille avec une échelle et une orientation fixées. Ce type de descripteur est souvent utilisé pour la catégorisation d'objets.

- **taille d'intervalle / échelle du point d'intérêt**. Le DSIFT spécifie la taille par un seul paramètre qui contrôle la taille d'un intervalle SIFT en nombre de pixels. Dans le descripteur SIFT standard, la taille de l'intervalle SIFT est liée à l'échelle du point d'intérêt par un facteur multiplicateur qui par défaut vaut 3. En conséquence un descripteur DSIFT de taille 5 correspond à un SIFT d'échelle 5/3=1,66.
- **lissage**. Le descripteur SIFT effectue un lissage de l'image en fonction de l'échelle du point d'intérêt. Par défaut, le lissage est équivalent à une convolution par une gaussienne de variance s^2, s étant l'échelle du point d'intérêt.

5 Classification

Dans de nombreuses applications, les fonctions de noyaux, comme les Support Vector Machines (SVMs) ont été utilisées avec succès, donnant souvent les meilleurs résultats. Les méthodes de noyaux sont utilisées dans de nombreux domaines et permettent même la fusion de données en concaténant les espaces de noyau ou en utilisant de multiples apprentissages de noyaux. Avant d'effectuer la classification d'images, nous appliquons une approche multi-descripteurs et multi-vocabulaires visuels pour construire la représentation finale des images de l'ensemble de données. Pour bénéficier de l'efficacité du classifieur linéaire, on utilise la transformation explicite de (Vedaldi et al. (2010)) pour améliorer la précision de la classification d'images.

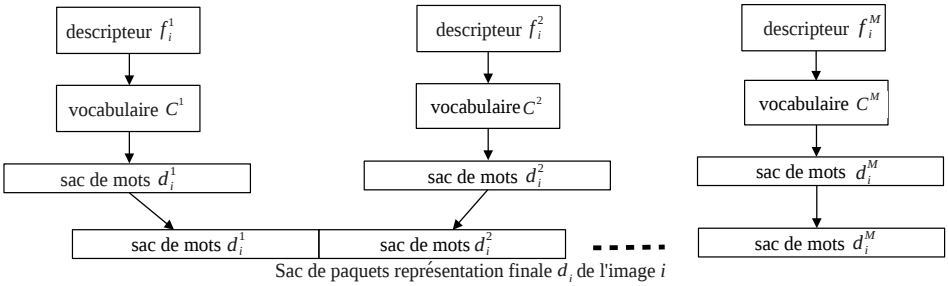

FIG. 3 – *Construction du sac de paquets avec multiples descripteurs et vocabulaires visuels*

5.1 Multiples descripteurs et multiples vocabulaires visuels

Soit $F = \{f_i^j\}$ l'ensemble des descripteurs extraits de l'image i, avec f_i^j les descripteurs de type j extraits de l'image i , M le nombre de descripteurs différents, et $j = 1, .., M$. Notre approche consiste à construire autant de sacs de mots qu'il y a de types de descripteurs utilisés comme montré sur la fig.3. Au lieu d'utiliser un seul vocabulaire visuel pour construire les mots visuels finaux, on utilise de multiples vocabulaires $\{C^1, C^2, ..., C^M\}$ construits à partir de différents descripteurs. Plus précisément le vocabulaire C^j est utilisé pour construire le sac de mots d_i^j correspondant au descripteur $f_i^j \in F$. Ensuite l'ensemble des sacs de mots d_i^j sont concaténés pour former les mots visuels finaux d_i. On a donc pour chaque image i, le mot visuel d_i composé de M éléments $d_i = \{d_i^1, d_i^2, ..., d_i^M\}$. Pour simplifier on appellera d_i un sac de paquets (BoPs), il est la représentation finale de l'image composée des différents vocabulaires visuels de l'image i en entrée. Un sac de paquets est plus discriminant qu'un sac de mots car deux sacs de paquets d_i et d_j sont considérés comme identiques si et seulement si l'ensemble des sacs de mots qui les constituent sont identiques :

$$(d_i = d_j) \equiv (d_i^1 = d_j^1) \wedge (d_i^2 = d_j^2) \wedge ... \wedge (d_i^M = d_j^M) \quad (1)$$

Cette approche améliore le pouvoir discriminant des mots visuels finaux par rapport à une approche classique avec un unique sac de mots.

5.2 LibSVM parallèle

LibSVM est une bibliothèque d'algorithmes de Support Vector Machines pour la classification (C-SVC, nu-SVC), la régression (epsilon-SVR, nu-SVR) et l'estimation de distribution (one-class SVM). Elle supporte également la classification multi-classes. Depuis la version 2.8, elle implémente un algorithme de type SMO (Fan et al. (2005)). LibSVM peut facilement s'interfacer avec les programmes utilisateurs. Parmi ses principaux composants, on peut citer : différents algorithmes de SVM, la gestion du multi-classes, la validation croisée, ...

LibSVM donne souvent de meilleurs résultats que LibLINEAR (Fan et al. (2008)), mais le coût de l'apprentissage est beaucoup plus important : plusieurs jours (voire semaines) de calcul sont nécessaires pour traiter des ensembles de données tels que ImageNet. Pour réduire ce temps d'apprentissage, nous avons développé une version parallèle utilisant OpenMP

(www.openmp.org) que nous appelons pLibSVM. Dans l'implémentation originale, l'étape la plus coûteuse en temps de calcul est le calcul des matrices de noyaux notamment lorsque l'on traite des ensembles de données de grandes tailles. Les calculs des valeurs des éléments de ces matrices sont indépendants les uns des autres, nous les avons donc parallélisés. Pour ce faire nous avons utilisé OpenMP pour pouvoir bénéficier des architectures multi-coeurs disponibles sur la plupart des machines actuelles. Cette nouvelle implémentation de l'algorithme permet de réduire de manière significative le temps de calcul de LibSVM. Pour évaluer les performances de pLibSVM, nous l'avons comparé avec LibLINEAR et OCAS (Franc et al. (2008)) dans nos expérimentations. (Franc et al. (2008)) ont montrés que leur algorithme avait une convergence plus rapide que les autres méthodes, c'est pour cette raison que l'on a choisi OCAS pour comparaison avec notre algorithme pLibSVM au lieu de Pegasos (Primal Estimated sub-GrAdient SOlver for SVM) ou Stochastic Gradient Descent SVM.

6 Expérimentations et résultats

6.1 Ensembles de données

La totalité de l'ensemble de données ImageNet représente environ 1 téra-octet de données. Dans nos premières expérimentations, nous avons évalué notre système sur les dix plus grandes classes d'ImageNet ce qui représente 24817 images et 2,4 giga-octets de données. Ces classes sont : n00483313, n01882714, n02086240, n02087394, n02094433, n02100583, n02100735, n02138441, n02279972, n09428293. Chaque classe contient plus de 2000 images.

6.2 Extraction paralléle des caractéristiques des images

Pour nos expérimentations, nous avons utilisé un PC Intel Xeon E5345 à 2,33GHz. Le temps de calcul pour l'extraction des caractéristiques (eg. SIFT) d'une image varie de 0,5 à 1 seconde en fonction des paramètres. Pour traiter les dix plus grandes classes (24817 images) il faut donc entre 3 et 7 heures de calcul. Il est donc difficile de passer à l'échelle, si l'on veut traiter la totalité de l'ensemble de données ImageNet (14 197 122 images) il faudrait 14 197 122 secondes soit environ 141 jours de calcul. Nous avons donc parallélisé ce processus pour permettre un calcul dans un temps raisonnable.

Il y a beaucoup d'images de tailles très variables dans l'ensemble de données ImageNet. Nous avons choisi de réduire les grandes images à une taille maximale de 500 pixels de côté. Cette normalisation permet de réduire le temps de calcul et permet aussi une représentation plus robuste aux variations d'échelle.

SIFT/DSIFT : Pour extraire les SIFTs et les DSIFTs nous avons utilisé la version Matlab de VL_FEAT (www.vlfeat.org) développée par Andrea Vedaldi (Vision Lab, University of California). L'implémentation originale des descripteurs SIFTs fournit des vecteurs d'entiers en 128 dimensions. Pour pouvoir les combiner avec les SURFs nous avons transformés ces vecteurs d'entiers en vecteurs de flottants. La Parallel Computing Toolbox (PCT) de Matlab permet la programmation en parallèle (en plus du multi-thread implicite de Matlab). Avec la PCT, il est possible de créer un ensemble de processus permettant d'exécuter des tâches indépendantes en parallèle. La seule restriction est que l'ensemble de ces tâches doit s'exécuter sur la même machine. Nous avons donc utilisé la PCT pour extraire les SIFTs et les DSIFTs

en parallèle sur une machine dotée de 8 coeurs. Comme montré dans le tableau 1, il nous a fallu 1 heure et 12 minutes pour extraire 30 107 662 SIFTs dans les 10 plus grandes classes d'ImageNet (soit en moyenne 0,17 seconde pour extraire les SIFTs d'une image). Pour traiter la totalité de l'ensemble de données, il faudrait donc 24 jours de calcul (ou 8 jours sur une machine à 24 coeurs).

P-SURF : P-SURF est une version parallèle des SURFs (Gossow et al. (2010)). Les descripteurs locaux calculés par l'implémentation originale des SURFs sont des vecteurs de flottants en 64 dimensions. Pour pouvoir les combiner avec les SIFTs, nous avons étendu leur dimension à 128. Toujours en utilisant une machine dotée de 8 coeurs, il nous a fallu 1 heure pour extraire 23 782 533 SURFs sur les 10 plus grandes classes d'ImageNet, comme montré dans le tableau 1 (soit en moyenne 0,14 seconde pour extraire les SURFs d'une image). Pour traiter la totalité des images de l'ensemble de données ImageNet, il faudrait donc 20 jours de calculs sur une machine à 8 coeurs (ou 6,5 jours sur une machine à 24 coeurs). Bien entendu ces temps de calcul sont donnés à titre indicatif et peuvent être facilement réduits en utilisant par exemple plusieurs machines.

6.3 Construction parallèle des sacs de paquets

Le temps de calcul de chaque vocabulaire visuel est similaire au cas d'une approche classique. Si on utilise n vocabulaires visuels il faut donc n fois plus de calculs. Pour permettre d'obtenir un résultat dans le même temps de calcul que les approches standards, nous avons donc effectuer le calcul en parallèle des sacs de paquets. Le temps de calcul global est donc le plus long temps de calcul pour les différents sacs de mots. L'ensemble des descripteurs calculés représentent 100 Go de données.

Descripteur	Temps	nb points	Taille (Go)
SIFT	1 h 12 mn	30,107,662	15
SURF	59 mn	23,782,533	11.9
DSIFT	1 h 19 mn	540,616,138	72.5

TAB. 1 – *Extraction des descripteurs sur les 10 plus grandes classes d'ImageNet.*

6.4 Construction des vocabulaires visuels

Dans les approches usuelles de sacs de mots, l'une des étapes les plus coûteuses en temps est la construction du vocabulaire visuel. Avec des ensembles de données de grandes tailles, il faut un grand nombre de points pour construire un vocabulaire visuel discriminant, la tâche devient de plus en plus coûteuse en temps de calcul. L'une des solutions la plus utilisée est l'algorithme de k-means. Cependant, l'implémentation originale de l'algorithme de k-means nécessite plusieurs jours de calcul sur des ensembles de données tels qu'ImageNet. Réduire le temps d'exécution de l'algorithme est donc un problème majeur dans un système de classification d'images à grande échelle. Nous avons donc utilisé la version parallèle de l'algorithme de k-means de Wei Dong (University of Michigan). Cette implémentation a les caractéristiques suivantes : 1) elle permet de traiter des ensembles de données qui ne peuvent tenir en mémoire

vive, 2) elle supporte la lecture en parallèle de plusieurs fichiers d'entrée si ceux-ci sont sur des disques distincts, 3) elle accélère le calcul de la distance L2 grâce à l'utilisation de la librairie BLAS ou d'une mise en æuvre de KD-tree.

Nous avons utilisé une machine à 24 coeurs pour exécuter l'algorithme de k-means. La taille du vocabulaire visuel a été fixé à 5000 mots, le nombre d'images utilisées par classe est de 400 et le nombre maximum d'itérations a été fixé à 100.

Descripteur	nb points	Temps
SIFT	4,888,576	6 h
SURF	3,878,113	5 h
DSIFT	84,566,109	6 jours

TAB. 2 – *Exécution du k-means parallèle sur les 10 plus grandes classes d'ImageNet.*

6.5 Précision de la classification

Le noyau linéaire appliqué directement sur l'histogramme classiquement utilisé ne donne pas une très bonne précision. Dans de nombreuses approches, on utilise une transformation de la représentation originale des images dans un espace intermédiaire de dimension augmentée. L'idée est de rechercher une séparatrice linéaire dans l'espace de dimension augmentée au lieu d'une séparatrice non linéaire dans l'espace initial (le coût de calcul de la séparatrice linéaire étant beaucoup moins important). Nous avons expérimenté les 2 cas, les résultats sans transformation figurent dans le tableau 3 et ceux avec transformation dans le tableau 4 (nous avons utilisé le noyau homogène de Vedaldi). Ce dernier permet une amélioration sensible de la précision (de +8,97% à +55,89%).

Pour évaluer la performance de notre algorithme pLibSVM, nous l'avons comparé avec LIBLINEAR, OCAS et la version originale de LibSVM à la fois en ce qui concerne la précision et le temps de calcul. Pour chaque classe, nous avons utilisé 90% des images pour l'ensemble d'apprentissage et les 10% restant pour l'ensemble de test. pLibSVM utilise un noyau linéaire et est exécuté sur une machine à 8 coeurs.

Descripteur	LIBLINEAR SVM	OCAS	LIBSVM	pLIBSVM
SIFT	41.69% (0mn14s)	**43.75%** (0mn17s)	12.39% (2h50mn)	12.39% (19mn55s)
SURF	46.64% (0mn16s)	**49.74%** (0mn14s)	12.27% (1h46mn)	12.27% (13mn48s)
DSIFT	47.56% (0mn3s)	**52.07%** (1mn21s)	20.64% (5h15mn)	20.64% (47mn55s)
DSIFT+SURF	59.44% (1mn22s)	**62.01%** (1mn35s)	24.23% (6h22mn)	24.23% (59mn05s)

TAB. 3 – *Précision et temps d'exécution de la classification des 10 plus grandes classes d'ImageNet*

Descripteur	LIBLINEAR SVM	OCAS	LIBSVM	pLIBSVM
SIFT	54.93%	56.18%	**58.35%**	**58.35%**
	(1mn30s)	(2mn19s)	(3h12mn)	(33mn34s)
SURF	55.85%	57.46%	**58.71%**	**58.71%**
	(2mn82s)	(1mn47s)	(2h12mn)	(34mn01s)
DSIFT	77.30%	**77.34%**	75.53%	75.53%
	(2mn20s)	(6mn11s)	(3h47mn)	(49mn56s)
DSIFT+SURF	79.53%	78.71%	**80.12%**	**80.12%**
	(3mn04s)	(1h14mn)	(5h50mn)	(1h00mn)

TAB. 4 – *Précision et temps d'exécution de la classification des 10 plus grandes classes d'ImageNet avec le noyau homogène de Vedaldi.*

Comme nous l'avons précédemment mentionné, le challenge majeur dans la classification de grandes bases de données d'images est l'étape d'apprentissage. Nous présentons une version parallèle de l'algorithme LibSVM (pLibSVM) qui est très efficace sur la classification des 10 plus grandes classes de l'ensemble de données ImageNet. Les résultats du tableau 4 montrent qu'à l'exception du cas des DSIFTs, LibSVM et pLibSVM sont ceux qui obtiennent les meilleurs résultats. De plus pLibSVM obtient ces résultats dans un temps beaucoup plus réduit que l'original LibSVM. Evidemment, ces résultats pourraient être encore meilleurs si l'on utilisait plus de ressources.

Pour évaluer la performance de l'approche multiples vocabulaires visuels sur les 10 plus grandes classes d'ImageNet, nous avons expérimenté la combinaison des descripteurs DSIFT et SURF en conservant une taille de vocabulaire visuel identique (5000). Comme le montrent les résultats des tableaux 3 et 4, cette approche permet d'améliorer la précision avec des gains qui peuvent aller jusqu'à 12% dans le cas des données originales et 4,59% dans le cas des données transformées par le noyau homogène.

7 Conclusion et travaux futurs

Nous avons présenté un système efficace pour la classification d'images à grande échelle et illustré sa performance sur les dix plus grandes classes de l'ensemble de données ImageNet. Dans ce système, nous avons développé une version parallèle de l'algorithme LibSVM pour permettre la classification de grands ensembles de données dans un temps raisonnable. Pour accélérer le processus de calcul des descripteurs des images, nous avons montré comment l'utilisation de la boîte à outil PCT de Matlab permettait de réduire les temps de calcul sur des architectures multi-coeurs, courantes à l'heure actuelle. Nous avons aussi présenté une nouvelle approche utilisant simultanément plusieurs descripteurs des images pour permettre d'améliorer la précision des algorithmes de classification sur les grands ensembles de données d'images. Une extension évidente de ces travaux sera d'utiliser simultanément des descripteurs de bas niveaux (comme les SIFTs ou les SURFs) et des descripteurs globaux (comme les GISTs ou les textures) avec le mécanisme décrit. De plus la version courante d'OCAS ne permet la parallélisation que dans le cas de classification binaire, nous comptons l'étendre au cas de la

classification multi-classes. Toutes ces pistes nous semblent prometteuses pour améliorer la classification de grands ensembles de données d'images.

Remerciement. Ces travaux bénéficient d'un sountien financier partiel de la région Bretagne.

Références

Bay H., T. Tuytelaars, et L.Van Gool (2006). SURF : Speeded Up Robust Features. In *proc. of European Conference on Computer Vision*, pp. 404-417.

Bosch A., A.Zisserman, et X.Munoz (2007). Image classifcation using random forests and ferns. In *proc. of International Conference on Computer Vision*, pp.1-8.

Csurka G., C. Dance, L. Fan, J. Willamowski, et C. Bray (2004). Visual categorization with bags of keypoints. In *proc. of European Conference on Computer Vision*, pp.1-22.

Dalal N., et B. Triggs (2005). Histograms of oriented gradients for human detection. In *proc. of Conference on Computer Vision and Pattern Recognition*, pp. 886-893.

David G.Lowe (2004). Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 60, 2, pp. 91-110.

Deng J., W. Dong, R. Socher, L. Li, et L. Fei-Fei (2009). ImageNet : A large-scale hierarchical image database. In *proc. of Conference on Computer Vision and Pattern Recognition*, pp. 248-255.

Deng J., A. Berg, K. Li, et L. Fei-Fei (2010). What does classifying more than 10,000 image categories tell us ? In *proc. of European Conference on Computer Vision*, pp. 71-84.

Everingham M., L. Van Gool , C.K.I. Williams, J. Winn, et A. Zisserman (2010). *The PASCAL Visual Object Classes Challenge 2010 Results*.

Fan R. E., P. H. Chen, et C. J. Lin (2005). Working set selection using second order information for training SVM. *Journal of Machine Learning Research*, pp. 1889-1918.

Fan R. E., K. W. Chang, C. J. Hsieh, X. R. Wang, et C. J. Lin (2008). LIBLINEAR : A library for large linear classification. *Journal of Machine Learning Research*, pp. 1871-1874.

Fergus R., Y. Weiss, et A. Torralba (2009). Semi-supervised learning in gigantic image collections. In *proc. of Advances in Neural Information Processing Systems*, pp.522-530.

Fei-Fei L., R. Fergus, et P. Peron (2004). Learning generative visual models from few training examples : an incremental bayesian approach tested on 101 object categories. In *proc. of Conference on Computer Vision and Pattern Recognition*, pp. 59-70.

Fellbaum C. (1998). WordNet : An Electronic Lexical Database. *MIT Press, Cambridge*.

Franc V., et S. Sonnenburg(2008). Optimized Cutting Plane Algorithm for Support Vector Machines. In *proc. of International Conference on Machine Learning*, pp. 320-327.

Gossow D., P. Decker, et D. Paulus (2010). An Evaluation of Open Source SURF Implementations. In *RoboCup 2010 : Robot Soccer World Cup XIV*, pp. 169-179.

Griffin G., A. Holub, et P. Peron (2007). Caltech-256 object category dataset. *Technical Report 7694, California Institute of Technology*, pp. 1-20.

Lazebnik S., C. Schmid, et J. Ponce (2006). Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories. *Conference on Computer Vision and Pattern Recognition*, Vol.2, pp. 2169-2178.

Li Y., D. Crandall, et D. Huttenlocher (2009). Landmark classification in large-scale image collections. In *proc. of International Conference on Computer Vision*, pp. 1957-1964.

Moosmann F., B. Triggs, et F. Jurie (2006). Fast Discriminative Visual Codebooks using Randomized Clustering Forests. *Advances in Neural Information Processing Systems*, pp. 985-992.

Perronnin F., J. Sanchez, et Y. Liu (2010). Large-scale image categorization with explicit data embedding. In *proc. of Conference on Computer Vision and Pattern Recognition*, pp. 2297-2304.

Philbin J., O. Chum, M. Isard, J. Sivic, et A. Zisserman (2008). Lost in quantization : Improving particular object retrieval in large scale image databases. In *proc. of Conference on Computer Vision and Pattern Recognition*, pp.1-8.

Torralba A., R. Fergus, et W. Freeman (2008). 80 million tiny images : A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence*, Vol.30, pp. 1958-1970.

Vedaldi A., V. Gulshan, M. Varma, et A. Zisserman (2009). Multiple kernels for object detection. In *proc. of International Conference on Computer Vision*, pp. 606-613.

Vedaldi A., et A. Zisserman (2010). Efficient Additive Kernels via Explicit Feature Map. In *proc. of Conference on Computer Vision and Pattern Recognition*, pp. 3539-3546.

Wang C., S. Yan, et H.J. Zhang (2009). Large scale natural image classification by sparsity exploration. In *proc. of International Conference on Acoustics, Speech, and Signal Processing*, pp. 3709-3712.

Winder S., et M. Brown (2007). Learning local image descriptors. In *proc. of Conference on Computer Vision and Pattern Recognition*, pp. 1-8.

Summary

The usual frameworks for image classification involve three steps : feature extraction, building codebook by feature quantization and training the classifier with a standard classification algorithm (eg. SVM). However, task complexity becomes very large when applying this approach on large scale datasets like the ImageNet dataset containing more than 14 million images and 21,000 classes. The complexity is both about the time needed to perform each task and the memory and disk usage (eg. 11TB are needed to store the SIFT descriptors computed on the full datasets). We have developed a parallel version of LIBSVM to deal with very large datasets in reasonable Temps. Furthermore a lot of information is lost when performing the quantization step and the obtained bag-of-words (or visual-words) are often not enough discriminative for large scale image classification. We present a novel approach using several local descriptors simultaneously to try to improve the classification accuracy on large scale image datasets. We show our first results on dataset made of the ten largest classes (24,817 images) from ImageNet.