RAPPORT DU PROJET
Classification de documents

Rédigé par :
Nguyen Van Tho, Khong Minh Thanh

Sous la supervision de :
Rémy Mullot

1 Janvier 2014, mise à jour 10 Janvier 2014

1 Introduction

Dans le cadre du module indexation des contenus multimédias, il nous a été demandé de développer un programme qui permet de classifier les documents. Les entrées du programme, ce sont des images scannées à partir des documents réels. En effet, il y a deux premiers types de documents que le système doit reconnaître : *Demande de prime de déménagement* et *demande d'aide* Puis, pour les autres documents, le programme peut les classifier comme le type 3, type 4, ...pour que nous puissions les traiter des informations.

Pour reconnaître les documents, il existe des techniques tels que OCR, SVM, naïve Bayes, extraction de caractéristiques et les comparer .... Dans ce travail, nous essayerons de réaliser deux méthodes :

- Extraire des caractéristiques endivisant l'image en blocs
- Extraire des caractéistiques principales par le PCA

Ensuite, nous faisons la reconnaissance des documents par le méthode 1-NN, pour les classifier, nous utilisons Kmeans

Nous utilisons les données dans le cours pour la base d'apprentissage et de test.

Afin de rendre compte du travail effectué dans ce projet, nous avons rédigé ce rapport qui est structuré en des grandes parties :

- Méthodes proposées
- Expérimentation et analyse des résultats

2 Méthodes proposées

2.1 Prétraitement d'images

En voyant que les documents donnés sont grands (la taille originale est environ $2480x3500$, 5 Mo), et la classification des document n'a pas besoins d'une très bonne qualité de l'image d'entrée, nous proposons de réduire la taille de l'image entrée (dans ce travail nous avons réduit 4 fois la taille des images). Nous convertissons aussi les images en couleur aux images en gris car les documents ne tienne pas compte de couleur.

De plus, en examinant des images, nous avons vu que l'entête des documents est la partie qui représente assez informations pour la classification. Nous proposons donc de traiter l'entête des documents. L'avantage de cette approche est que nous n'avons pas besoins beaucoup de calcule, et aussi le programme est invariant avec le contenus du document (parce que le contenu est coupé).

2.2 Extraction de caractéristiques

Pour faire reconnaître les documents, il a besoins d'avoir des descripteurs. Nous proposons d'utiliser deux méthodes : extraire des descripteurs par diviser en blocs et la méthode PCA (principal component analysis)

Extraction de caractéristiques par diviser en blocs

Après avoir trouvé l'entête du document, pour obtenir les caractéristiques, nous avons implémenté comme suit :

- Nous découpons cette entête en $m \times n$ morceaux, m et n ce sont des fois que nous découpons en Y axis et X axis. Dans l'implémentation nous avons utilisé $m = 3$ et $n = 10$, nous obtenons une caractéristiques de longueur 3×10 qui présente assez les informations du document. Ci-dessous c'est l'illustration de notre méthode :

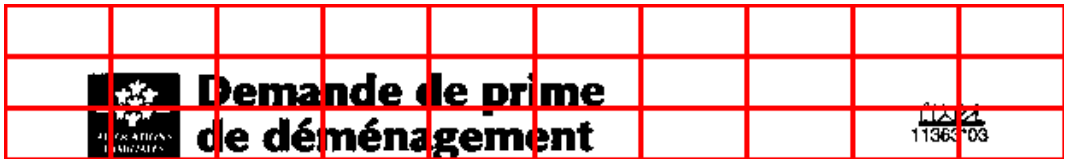

FIGURE 1 – Découpage l'entête d'un document de type 1 en 3×10 morceaux pour obtenir des caractéristiques

- Pour chaque morceau nous déterminons la moyenne normalisée des pixels (on fait la somme des pixels du morceau divisée par 255 pour la normalisation, ensuite par le nombre de pixels dans ce morceau). Ainsi pour chaque document, nous avons 3×10 valeurs permettant de le représenter (voir exemple ci-dessous)
  - *0.999461 1 1 1 1 1 1 1 1 1 1 0.847357 0.957389 1 0.501079 0.568501 1 0.631607 0.7411 0.964941 0.73301 0.970874 0.906149 0.462783 0.510248 0.925027 0.615426 0.75027 1* pour l'entête ci-dessus
  - ...

Principal component analysis (PCA)

PCA [1] est utilisée pour décomposer un ensemble de données à plusieurs variables en un ensemble de composantes orthogonales successifs qui expliquent le maximum de la variance. En effet, cette méthode permet d'extraire les caractéristiques d'images et à la fois réduire la dimension de données.

L'idée principale de cette méthode est de trouver les les composantes principales des images d'apprentissage. Ceci revient à déterminer les vecteurs propres de la matrice de covariance formée par l'ensemble des images exemples. Chaque visage exemple peut alors être décrit par une combinaison linéaire de ces vecteurs propres. Après avoir eu les vecteurs propres, on ordonne ces vecteurs par ces valeurs propres. Les vecteurs avec les petites valeurs propres sont supprimés. Ensuite, les images d'apprentissage et les images de test sont projetées sur l'espace de vecteurs propres obtenus par l'étape précédente. Le résultat de cette étape est les vecteurs caractéristiques des images de document.

2.3 Classification

L'espace de recherche dans notre problème est petite (quelques dizaines d'élément). Nous choisissons donc l'algorithme naïf de recherche le plus proche voisin (1-NN). Autrement dit, nous cherchons linéairement l'image dans la base d'apprentissage qui est plus proche de l'image de test (la distance euclidienne entre les deux images est plus petite). En fait, nous faisons deux classifications, une pour classifier l'image au type 1 ou au type 2, si le document n'est pas un de ces deux type, la deuxième classification est utilisée pour classifier document à une des classes obtenues par l'étape cluster. La figure ci-dessous est explique le processus de classification.

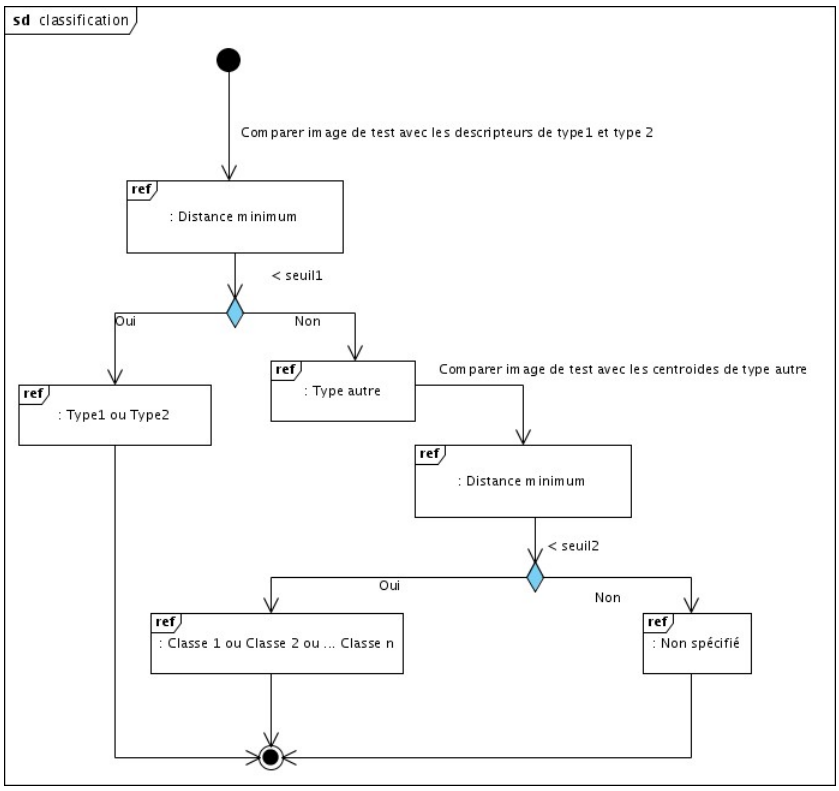

FIGURE 2 – Le diagramme interactif pour le processus des images

2.4 Cluster

Après avoir classifié les images en type 1 ou type 2, il nous reste la plupart des images qui n'appartiennent pas aux ces deux types. Nous devons utiliser une méthode d'apprentissage un-supervisé afin de regrouper ces images. Nous avons choisit la méthode Kmeans[2] pour cette tâche.

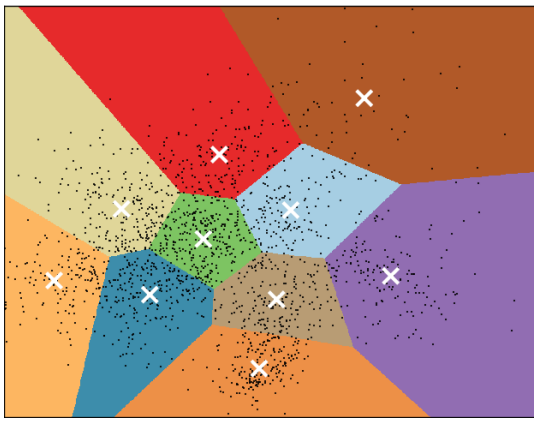

FIGURE 3 – La méthode Kmeans

Après avoir groupé les documents en cluster pour la base d'apprentissage, nous enregistrons les centroids de ces clusters et les réutiliserons dans la phase de test.

3 Expérimentation

3.1 Indicateurs pris en compte pour l'évaluation des méthodes proposées

Dans le but d'avoir une vue plus précise des performances de notre programme, nous avons implémenté les indicateurs suivants :

- Matrice de confusion (chaque colonne de la matrice représente le nombre d'occurrences d'une classe estimée, tandis que chaque ligne représente le nombre d'occurrences d'une classe réelle (ou de référence)) : chaque élément MatriceConfusion(i,j) de cette matrice correspond au nombre de fois où la je classe a été prédite alors que la vraie classe était la ie. Ainsi, cet indicateur nous permet de montrer rapidement si le système parvient à classifier correctement
- Temps d'apprentissage du programme
- Temps de reconnaissance Nombre de bonnes reconnaissances (good) Nombre de mauvaises reconnaissances (bad) Taux de reconnaissance (taux) qui est le rapport du nombre de bonnes reconnaissances par le nombre total.

3.2 Méthode d'évaluation

Nous appliquons la méthode d'évaluation k-fold cross validation afin d'assurer que notre programme va bien prédire les nouvelles données. Puisque les données proposées sont assez petites : 6 images pour le type 1, 6 images pour le type 2, ... nous choisissons 3 pour la valeur de k. Autrement, nous divisons les données en trois ensembles et chaque expérimentation

nous prenons un ensemble la validation et deux autres ensembles pour l'apprentissage. La figure ci-dessous l'explique.

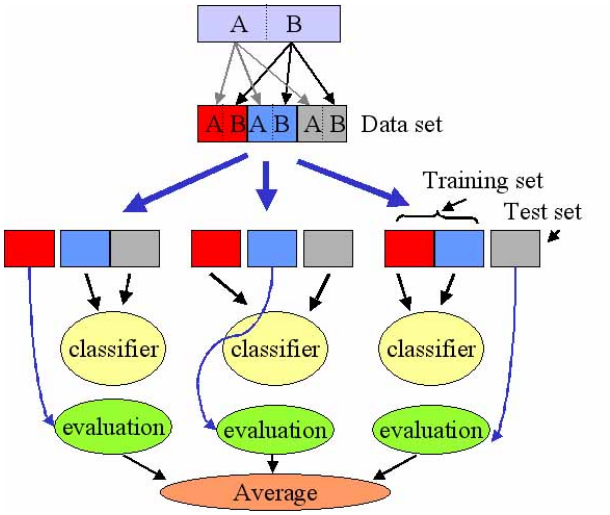

FIGURE 4 – Méthode 3-fold cross validation utilisée

Pour évaluer les documents de type 1, et type 2, nous regardons seulement si leurs document sont bien classifiés dans ces types ou non. Cependant, pour les images de type autre, il faudrait avoir une autre façon d'évaluation.

Puisque les documents de type autre ne sont pas en ordre, nous proposons d'abord de les grouper (avec l'aide de notre programme). On voit que il y a environs 11 types de l'autre document (selon nous) tel que : *UNiROSS facture, MONEY production, DELDI facture en Europe, ALLIBERT HOME France, lettre écriture, DECAYEUX,...*. Ensuite, dans la phase de test, si les documents sont regroupés dans une même classe, alors c'est une bonne regroupement. Au cas où il y un document qui n'est pas dans une même classe en comparaison avec notre classification, alors c'est une mauvais regroupement.

3.3 Expérimentations

Nous avons expérimenté deux méthode d'extraction de caractéristiques avec le Kmeans que nous avons expliqué. Un major problème de clustering avec Kmeans et de trouver le nombre de cluster optimal. Un bon choix de k est une valeur de k avec laquelle les données sont bien séparées. Nous avons observé dans les données proposées, il y a 11 classes. Nous expérimentations donc les valeurs de k de 8 à 13. Ci-dessous notre résultat :

Résultat

On voit tout de suite que le résultat de reconnaissance les documents de type 1 et 2 est toujours 100%. On constate que ces document ne varient pas beaucoup tel que la position des caractéristiques, la direction de texte, la luminance...De plus, nous avons déjà fait le prétraitement des documents. Les bruits sont presque éliminer, alors ces documents sont presque le même, donc le taux de 100% est totalement réalisable. Mais s'il existe un document de ces types, qui change la position de texte, la direction de texte, nous prédisons que le taux va diminuer beaucoup.

Méthode	Taux de reconnaissance (type 1 + type 2) (%)	Taux de reconnaissance (%)	Temps d'apprentissage (s)	Temps de reconnaissance (s)
blocs + K-means, k=8	100	89.6	0.0014	0.00040
blocs + K-means, k=9	100	85.4	0.0015	0.00042
blocs + K-means, k=10	100	97.9	0.0016	0.00043
blocs + K-means, k=11	100	95.8	0.0017	0.00045
blocs + K-means, k=12	100	97.9	0.0019	0.00046
blocs + K-means, k=13	100	97.9	0.0020	0.00048
PCA + K-means k=8	100	75.0	0.2	0.04
PCA + K-means k=9	100	83.3	0.2	0.04
PCA + K-means k=10	100	89.6	0.2	0.04
PCA + K-means k=11	100	93.8	0.2	0.04
PCA + K-means k=12	100	93.8	0.2	0.04
PCA + K-means k=13	100	93.8	0.2	0.04

TABLE 1 – 3-fold cross validation avec les valeurs de K entre 8 et 13

Le taux de reconnaissance est bien dépendu à la valeur de K. En voyant le tableau ci-dessus, le résultat avec k=11 est une bonne valeur pour cluster les données, parce qu'il est le nombre de cluster de tous les ensembles de documents. La méthode de diviser en blocs ressemble mieux la méthode PCA, la raison c'est que le PCA est sensible avec les variants du document. Les documents dans une classe peut avoir la distance un peu grande, il marche donc pas très bien avec K-means.

Un autre constat est que la méthode de diviser en blocs donne un très bon résultat, ça veut dire que le découpage en blocs de l'entête du document peut représenter plusieurs informations du document. Cela correspond bien avec la réalité où on devait écrire clairement le document dès son début.

En fin, nous voyons que le temps de calcule de méthode de diviser en blocs en plus vite que le PCA. Cela est expliqué par la projection de PCA. On doit projeter l'image entrée sur les dimensions du PCA avant de le reconnaître. Le temps de projections est en peu coûteux dans ce cas.

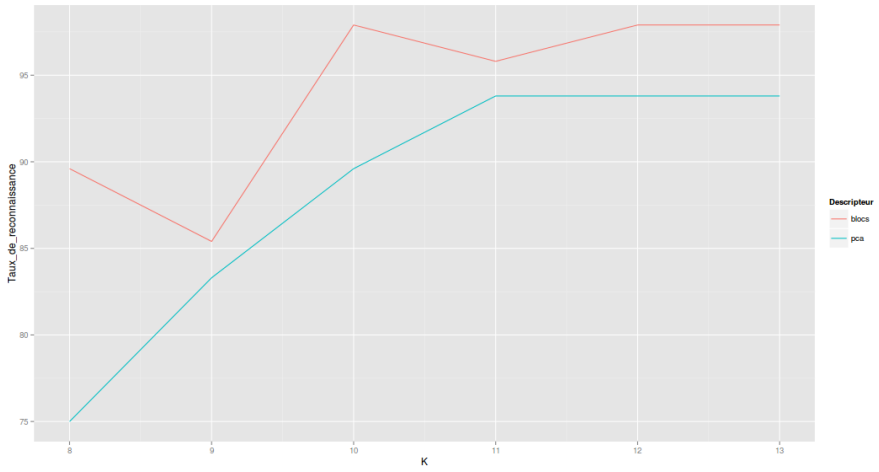

FIGURE 5 – 3-fold cross validation avec les valeurs de K entre 8 et 13

	t1	t2	1	2	3	4	5	6	7	8	9	10	11	-1
t1	6	0	0	0	0	0	0	0	0	0	0	0	0	0
t2	0	6	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	8	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	5	0	0	0	0	0	0	0	0	0	1
3	0	0	0	0	6	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	6	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	5	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	5	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	4	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	2	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	1	0
11	0	0	0	0	0	0	0	0	0	0	0	0	1	0

TABLE 2 – Matrice de confusion avec la méthode divisé en blocs + K-means, k = 11

	t1	t2	1	2	3	4	5	6	7	8	9	10	11	-1
t1	6	0	0	0	0	0	0	0	0	0	0	0	0	0
t2	0	6	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	8	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	6	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	6	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	6	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	5	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	4	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	4	0	0	0	0
9	0	0	0	0	2	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	1	0	0
11	0	0	0	0	1	0	0	0	0	0	0	0	0	0

TABLE 3 – Matrice de confusion PCA, Kmeans avec k = 11

Analyses

Au vu des matrices de confusion, nous constatons qu'il y a la confusion entre classe 10 et classe 11. Cela pourrait se justifier par le fait que la classe 10 et classe 11 n'a qu'une seul instance, la confusion est donc inévitable pour ces deux classe. De plus, la qualité des images, la différence de luminance, des bruits peuvent avoir des influence sur la précision de la classification et le regroupement.

4 Conclusion et Perspective

Dans ce projet nous avons implémenté un programme qui permet de classifier les images de document. Afin de classifier les documents qui ne sont pas étiquetés, notre programme a un module d'apprentissage un-supervisé basant sur la méthode Kmeans qui regroupe ces documents.

Nous avons aussi utilisé deux méthodes différentes pour décrire les caractéristiques d'images et pour réduire la dimension de données : division d'image en blocs, PCA. À partir des résultats de notre expérimentation, nous constatons que les documents de classe 1 et 2, les algorithmes proposées marche très bien avec un taux de reconnaissance 100%. Par contre, pour la deuxième partie, le regroupement des images, le descripteur basant sur la division d'image en blocs est supérieur à la descripteur PCA. La méthode de découpage en blocs ne marche pas bien quand on donne les documents qui sont coupés la borne, ou si les documents à reconnaître change la direction.

Nous constatons que la taille de données est petite, pour améliorer la précision du programme, on peut générer des images à partir des images originales en changeant la luminance, le contraste ou le floue gaussien. Faute du temps, nous laissons cela comme un travail dans le futur.

Références

[1] Wold, Svante, Kim Esbensen, and Paul Geladi. "Principal component analysis." Chemometrics and intelligent laboratory systems 2.1 (1987) : 37-52.

[2]Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136 : A k-means clustering algorithm." Journal of the Royal Statistical Society. Series C (Applied Statistics) 28.1 (1979) : 100-108.