Reconnaissance de visages : une méthode originale combinant analyse discriminante logistique et distance sur graphe

Alexis Mignon Frédéric Jurie

GREYC – CNRS / ENSICAEN / Université de Caen

alexis.mignon@info.unicaen.fr

Résumé

Nous nous intéressons dans cet article au problème passionnant qu'est la reconnaissance de visages dans des cas non contraints, c'est-à-dire dans des situations où l'éclairage, la pose, la taille du visage dans l'image ne sont pas contrôlés. Nous proposons ici deux contributions originales : une méthode d'apprentissage de distance par analyse en composantes logistiques discriminantes (LDCA), combinée à une méthode d'apprentissage semi-supervisé au moyen de graphes.
Nous avons validé cette approche sur la base d'image Labeled Faces in the Wild*, base sur laquelle nos résultats sont au-dessus de l'état de l'art.*

Mots clefs

Apprentissage de distances, reconnaissance de visages, apprentissage semi-supervisé, séparateurs à vaste marge.

Abstract

In this paper, we are interested in the challenging problem of face recognition in unconstrained cases, i.e. situations in which illumination, pose and size of the face in the picture are uncontrolled. We propose here two original contributions : a Logistic Discriminant Component Analysis (LDCA) metric learning method combined with a semi-supervised learning method based on graphs. We tested this method on the Labeled Faces in the Wild *database on which our results are above the state of the art.*

Keywords

Distance metric learning, face recognition, semi-supervised learning, SVM.

1 Introduction

La reconnaissance de personnes à partir des visages présents dans des images suscite un vif attrait [34, 23, 22, 16, 25] dans la communauté scientifique ; cela s'explique d'une part en raison des intérêts applicatifs mais aussi du défi que cela représente pour les algorithmes de vision artificielle ; ils doivent être capables de faire face à la grande variabilité des aspects des visages eux-mêmes tout autant qu'aux variations des paramètres de prise de vue (pose, éclairage, coupe de cheveux, expression, arrière-plan, etc.).

FIG. 1 – Exemples d'images de la base LFW.

En réalité, la notion de *reconnaissance de visage* recouvre plusieurs problèmes différents, comme (a) déterminer si les deux sujets photographiés sont une seule et même personne ; (b) étant donné l'image d'un visage, vérifier l'identité de la personne (tâche d'authentification) ; (c) étant donné l'image d'un visage, déterminer s'il s'agit de l'une des personnes contenues dans une liste, et si oui laquelle.
Cet article se consacre à la première de ces tâches que nous appellerons par la suite *problème des paires correspondantes*.
D'une manière générale, nous posons le problème comme celui de l'apprentissage d'une distance entre visages. Nous supposons disposer d'un ensemble de paires d'images de visages, certaines de ces paires représentant des visages de personnes différentes, d'autres des paires de visages provenant de la même personne mais avec des variations d'expression, de pose ou d'illumination. Pour chacune de ces paires nous connaissons la vérité terrain, c'est-à-dire que nous savons s'il s'agit de la même personne ou non.
Notre calcul de similarité s'appuie sur quatre grandes étapes :

1. Chaque visage est représenté par un vecteur d'attributs.
2. Nous effectuons ensuite une transformation linéaire des données de départ en utilisant une méthode inspirée de [29], dont l'intérêt est, en plus de réduire la

dimensionnalité, de calculer un espace de représentation qui sépare au mieux les données positives des négatives (paires de visages identiques ou différents).

3. Une phase d'apprentissage semi-supervisé, où les données de test (dont les labels ne sont pas connus) sont utilisées pour déterminer avec plus de précision la structure des données dans l'espace de représentation. Cette phase repose sur la construction d'un graphe où les noeuds représentent les paires de visages et les arêtes les relations entre ces paires.
4. L'apprentissage d'un classifieur qui combine les informations extraites à partir des deux méthodes précédentes pour mesurer la similarité de deux visages inconnus.

Le plan de l'article est le suivant : après avoir donné un aperçu des travaux antérieurs et présenté la base que nous utilisons pour les tests, nous présentons successivement les deux contributions majeures de cet article, puis nous en donnons une validation expérimentale sur la base de visage *Labeled Faces in the Wild*. Enfin, nous concluons et donnons des perspectives.

1.1 Travaux antérieurs

Comme nous l'avons évoqué dans l'introduction, la reconnaissance de visages est un sujet qui a été la source de très nombreux travaux dans la dernière décennie. Nous nous intéressons ici au cas particulier posé par le *problème des paires correspondantes*, en supposant que les images de visages ont été alignées au préalable. Nous ne reviendrons pas sur la méthode d'alignement (décrite dans [14]), mais de nombreuses autres méthodes pourraient être utilisées, comme [17] par exemple.
Si l'on suppose la question de l'alignement résolue, la reconnaissance de visages nécessite de prendre en compte de manière efficace (a) les variations d'apparence provoquées par des changements de pose, (b) celles provoquées par des changement d'illumination, (c) celles provoquées par les changements d'expression.

Prise en compte des variations de pose. Une approche possible consiste à utiliser des classifieurs qui sont spécialisés pour une pose particulière [26]. Seuls les visages de poses similaires sont comparés. Cette approche nécessite toutefois plusieurs images d'un même visage dans plusieurs poses différentes ainsi qu'une estimation précise de la pose.
D'autres approches telles que les modèles déformables [17] ou les modèles d'apparence active [7] utilisent des modèles 2D du visage, mais ces techniques ne sont généralement capables de traiter qu'une palette restreinte de variations. Elles sont, en particulier, sensibles aux problèmes d'auto-occlusion dus à la nature tridimensionnelle des visages.
Plus récemment, des techniques basées sur des modèles 3D de visages ont été développées. Elles consistent à reconstruire un modèle 3D du visage à partir de sa représentation 2D. Les visages sont ensuite placés dans une pose normalisée [4] pour la comparaison. Les modèles 3D peuvent également être comparés directement entre eux. Un inventaire des méthodes de comparaison de visages en 3D peut-être trouvée dans [5]. L'ajustement d'un modèle 3D sur une image 2D et la comparaison de modèles 3D restent cependant des problèmes non triviaux.

Prise en compte des variations d'illumination. A l'échelle du pixel, les variations dues aux différences d'illumination peuvent être beaucoup plus importantes que les différences entre deux personnes distinctes dans les mêmes conditions d'éclairage [1]. Les travaux théoriques de Belhumeur et Kriegman [3] ont montré que l'ensemble des variations dans l'espace des pixels dues aux différences d'illumination reposent sur une variété de faible dimension qu'ils ont appelée *le cône d'illumination*. Basri et Jacobs ont, par exemple, montré comment calculer un sous-espace de faible dimension pour obtenir une approximation du cône d'illumination [2]. Le passage de la théorie aux applications pratiques reste un vaste sujet de recherche.

Prise en compte des variations d'expression. Les modèles déformables 2D ou 3D [17, 4] ou des modèles d'apparence active [7] peuvent être utilisés pour capturer l'expression des visages. Comme pour la pose, les visages peuvent être ramenés à une expression normalisée ou des caractéristiques indépendantes de l'expression peuvent être extraites. Les calculs impliqués dans l'ajustement de tels modèles sont néanmoins lourds et peu robustes, en particulier en présence de variations de pose et d'éclairage.
Les modèles multilinéaires ou les *tensorfaces* utilisent, quant à eux, une généralisation de la décomposition en valeurs singulières pour les tenseurs multidimensionnels [20, 30]. Ces méthodes sont capables d'apprendre des sous-espaces différents pour chaque type de variations, mais nécessitent une quantité importante d'images pour chaque personne avec les variations d'expression désirées.
Certains systèmes ignorent simplement la zone de la bouche qui est la plus sujette à des modifications lors des variations d'expression.
Un tour d'horizon plus complet des différents problèmes rencontrés en reconnaissance de visages et de leur traitement peut-être trouvée dans [18].

1.2 La base *Labeled Faces in the Wild*

Les campagnes d'évaluation jouent un rôle important dans les progrès obtenus récemment. Elles permettent à la fois de dynamiser la recherche autour d'une problématique mais surtout de comparer les performances de différents algorithmes sur les mêmes données et dans de mêmes conditions, et ainsi de mieux comprendre où sont les axes de progrès.
La base *Labeled Faces in the Wild*[1] (LFW) est constituée de photographies de visages collectées sur le site d'information *Yahoo ! News*. Des images types sont présentées fi-

[1] http ://vis-www.cs.umass.edu/lfw/

gure 1. Aucune contrainte sur les paramètres de prise de vue n'a donc été imposée, si ce n'est le biais dû à la méthode de détection automatique de visages utilisée [31]. La base LFW contient 13233 images de 5479 personnes différentes. Parmi ces personnes, 1680 sont représentées par au moins deux images. Toutes les annotations ont été contrôlées par un expert humain.

Les protocoles d'expérimentation. En plus des images, les concepteurs de la base ont mis au point un protocole expérimental permettant de définir précisément les mesures de performance. A cette fin, deux *vues* sont proposées. Les deux vues prennent la forme de listes de paires d'images avec à chaque fois autant de paires d'images positives (les deux images représentent la même personne) que de paires d'images négatives (les deux images représentent des personnes différentes). La `vue1` est utilisée pour la mise au point des algorithmes et la sélection de modèle. Elle se compose d'un jeu de données d'entrainement de 2200 paires et d'un jeu de test de 1000 paires. La `vue2` n'est utilisée que pour le rapport final de performance et contient 10 séries de 600 paires, soit 6000 paires en tout. Ces 10 séries constituent les 10 étapes de la validation croisée utilisée pour la mesure de performance, laquelle est donnée par le taux moyen de classification exacte et l'erreur standard sur cette moyenne[2].

Paradigmes restreints ou non aux images. Deux manières différentes d'utiliser les vues sont proposées. En effet, celles-ci référencent les images par le nom de la personne représentée et par un numéro. La première approche consiste à simplement utiliser les paires d'images indiquées sans tenir compte de l'identité des personnes ; il s'agit de ce que les concepteurs de la base appellent le *paradigme restreint aux images*. Pour les expériences nécessitant plus de données d'entrainement, il est toutefois possible d'utiliser l'information donnée par le nom de la personne pour générer d'autres paires d'images ; il s'agit du *paradigme non restreint aux images*.
Les travaux présentés dans cet article concernent le paradigme restreint aux images, qui est le plus difficile.

2 Analyse en composantes logistiques discriminantes (LDCA)

2.1 Représentation des visages

La représentation des visages joue un rôle essentiel dans leur reconnaissance. Nous nous sommes inspirés de la méthode proposée dans [13], motivés par le fait que c'est cette représentation qui donne à l'heure actuelle les meilleurs résultats sur la base LFW.
Il s'agit de descripteurs SIFT [21] calculés en 9 points caractéristiques, et ce à 3 échelles différentes.
Les 9 points caractéristiques sont détectés par la méthode de Everingham *et. al.* [9] et correspondent aux coins des yeux, de la bouche, aux bords des narines et au bout du nez. Cela permet de rendre la représentation invariante pour de petits changements de pose du visage.
Les descripteurs SIFT ont 128 composantes, ce qui conduit pour chaque visage à un ensemble de 3456 ($= 128 \times 3 \times 9$) attributs visuels.
Nous suivons également les conclusions des auteurs de [13] et prenons pour chaque composante sa racine carrée. La distance euclidienne calculée sur ces descripteurs modifiés correspond à la *distance de Hellinger*.

2.2 Mesure de distance entre visages par LDCA

Comme nous venons de le voir, les visages correspondent à des points dans un espace à 3456 dimensions. Nous proposons de munir cet espace d'une mesure de distance ayant la propriété de rapprocher les visages identiques et d'éloigner ceux qui ne le sont pas.
L'apprentissage de distances a fait l'objet de nombreux travaux [8, 11, 12, 32, 35, 27].
Nous adoptons ici une approche classique en exprimant la distance via une matrice de Mahalanobis :

$$\mathcal{D}_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j)}$$

$\mathbf{M}$ doit être semi-définie positive afin que la propriété $\mathcal{D}_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq 0$ soit toujours vérifiée.
Il existe différentes manières d'estimer $\mathbf{M}$, en fonction du critère que l'on cherche à minimiser. Ce critère peut être basé sur la maximisation d'une marge comme dans les travaux sur les k-plus proches voisins à vaste marge [32], ou encore sur des entropies relatives de distributions [8].
Notre approche s'inspire des travaux de Guillaumin *et. al* [13] qui ont défini un critère basé sur *l'analyse logistique discriminante*. Le critère repose sur la probabilité que les éléments de la paire $n = (i, j)$ appartiennent à la même classe, autrement dit, que l'étiquette t_n de la paire soit 1. Cette probabilité est calculée à partir d'une distance de Mahalanobis et de la fonction logistique $\sigma(x) = 1/(1+e^{-x})$:

$$p_n = p(y_i = y_j | \mathbf{x}_i, \mathbf{x}_j, \mathbf{M}) = \sigma(b - \mathcal{D}^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j))$$

Le paramètre b agit comme un seuil et est appris en même temps que la matrice $\mathbf{M}$, par maximisation de la vraisemblance de l'étiquetage des paires d'apprentissage.
La *log-vraisemblance* $\mathcal{L}$ s'écrit :

$$\mathcal{L} = \sum_n t_n \ln p_n + (1 - t_n) \ln(1 - p_n)$$

et :

$$\frac{\partial \mathcal{L}}{\partial \mathbf{M}} = -\sum_n (t_n - p_n) \mathbf{C}_n$$

où $\mathbf{C}_n = (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$, et :

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_n (t_n - p_n)$$

[2] $S_E = \frac{\sigma}{\sqrt{10}}$ où σ est l'estimation de la déviation standard.

La fonction $\mathcal{L}$ est lisse et convexe. Cependant, lorsque l'espace d'entrée est de grande dimension D, la matrice $\mathbf{M}$ est grande également et le processus d'optimisation risque de pâtir d'un problème de sur-apprentissage et de temps de calcul prohibitifs. C'est pourquoi une réduction de dimension est nécessaire au préalable, par exemple avec une analyse en composantes principales (ACP) dans [13].
La force de la méthode que nous proposons réside dans la paramétrisation de la matrice $\mathbf{M}$ par $\mathbf{M} = \mathbf{L}^T\mathbf{L}$, où $\mathbf{L}$ est une matrice rectangulaire $d \times D$ avec $d \ll D$.
Ceci présente plusieurs avantages : (a) mise sous cette forme, la matrice $\mathbf{M}$ est toujours semi-définie positive ; (b) lorsque la valeur de d est petite, le nombre de paramètres à apprendre est également nettement plus petit ce qui évite le sur-apprentissage et réduit les temps de calcul ; (c) la transformation $\mathbf{x}' = \mathbf{L}\mathbf{x}$ définit implicitement une projection dans un espace de dimension réduite d.
La probabilité p_n devient alors :

$$p_n = \sigma(b - \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|^2)$$

et le gradient de la fonction $\mathcal{L}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{L}} = -2\mathbf{L}\sum_n (t_n - p_n)\mathbf{C}_n$$

La fonction $\mathcal{L}$ n'est plus convexe *a priori* par rapport à $\mathbf{L}$, mais dans la pratique, nous n'avons pas rencontré de problèmes dus à des minima locaux.
Notons que cette paramétrisation est similaire à celle utilisée par Torresani *et. al.* [29] pour l'analyse en composantes à vaste marges qui traite le problème de la réduction de dimension dans le contexte des k-plus proches voisins à vaste marge. Par analogie, nous appellerons notre méthode *analyse en composantes logistiques discriminantes*.
À l'issue de cette phase, nous disposons d'une mesure de distance qui peut être utilisée directement pour déterminer si deux visages sont identiques ou non, par simple seuillage.
Même si cette méthode donne de bons résultats, comme nous le montrons plus loin, il nous a paru intéressant de la faire suivre d'une phase permettant de tenir compte des non-linéarités dans l'espace de représentation et permettant également de procéder à un entrainement semi-supervisé.

3 Apprentissage semi-supervisé et graphes

Nous venons de proposer dans la section précédente une méthode d'apprentissage de distance dont l'objectif est de réduire la distance entre des visages similaires et d'éloigner des visages différents.
La méthode précédente souffre de deux limitations que nous nous proposons de pallier ici. Tout d'abord la transformation appliquée à l'espace d'origine est une transformation linéaire, ce qui ne permet pas de prendre en compte le fait que les données se trouvent éventuellement sur des variétés de formes quelconques. Une seconde limitation provient du fait que la méthode est entièrement supervisée, ce qui est un handicap pour ce genre de tâche : s'il est très facile d'obtenir une multitude de paires de visages à partir d'images collectées sur internet, il est très coûteux d'avoir des annotations disant quelles sont les paires positives et quelles sont les paires négatives. Une méthode semi-supervisée semble donc particulièrement pertinente dans ce cas.
L'apprentissage semi-supervisé repose principalement sur deux hypothèses :
– les données reposent sur un support dont la dimension intrinsèque est plus petite que le nombre de paramètres du problème (hypothèse de la variété),
– les données forment des groupes naturels (hypothèse des *clusters*)

Dans les deux cas, les données correspondent à un échantillonnage d'une distribution spécifique. Le nombre d'échantillons est donc déterminant si on veut pouvoir mettre en évidence cette structure. Les méthodes semi-supervisées utilisent précisement les données non étiquetées pour augmenter le nombre d'échantillons et en faciliter l'apprentissage.
Pour capturer au mieux la structure, il est usuel d'avoir recours à un graphe de voisinage [6], ce que nous faisons ici.

3.1 Représentation des visages

Nous nous plaçons dans l'espace de dimension réduite calculé grâce à la méthode LDCA ($\mathbf{x}' = \mathbf{L}\mathbf{x}$). Ensuite, afin de se ramener à un problème de classification binaire classique, nous calculons composante par composante la valeur absolue de la différence des descripteurs des éléments de chaque paire. Formellement, si on note v^k la k-ième composante d'un vecteur $\mathbf{v}$ quelconque, alors la différence absolue $\mathbf{d}_n$ entre les éléments de la paire $n = (i, j)$ est donnée par $d_n^k = |x_i'^k - x_j'^k|$. Nous obtenons donc un vecteur par paire. La norme de $\mathbf{d}_n$ correspond ainsi à la distance entre $\mathbf{x}'_i$ et $\mathbf{x}'_j$: $\|\mathbf{d}_n\| = \|\mathbf{x}'_i - \mathbf{x}'_j\|$ et les paires positives doivent former un cluster localisé près de l'origine du repère des coordonnées.
La technique d'apprentissage LDCA est limitée à une transformation linéaire des données. Nous souhaitons ici prendre en compte les composantes non linéaires. Pour cela nous calculons le résultat d'une classification par k-plus proche voisins (k-PPV) dans le nouvel espace.

3.2 Construction d'un graphe des paires

Le graphe de k-plus proche voisinage est ensuite construit pour les $\mathbf{d}_n$ (paires). La valeur de k retenue est celle donnant les meilleurs résultats sur la `vue1`, à savoir $k = 15$. Un poids $a_{nn'}$ est affecté à chaque arête dont la valeur est donnée par $a_{nn'} = \exp(-\|\mathbf{d}_n - \mathbf{d}_{n'}\|^2 / < d >^2)$ où $< d >$ représente la moyenne des distances calculée pour toutes les paires de vecteurs $(\mathbf{d}_n, \mathbf{d}_{n'})$.
Nous calculons ensuite une approximation de la *distance euclidienne du temps de commutation* [10] en projetant les données sur les 20 vecteurs propres correspondant aux 20 plus grandes valeurs propres de la matrice $\mathbf{L}^+$ associée au graphe (voir plus loin).

Temps de commutation sur graphes. Soit un graphe non orienté $G = (V, E)$ où V est l'ensemble des sommets et $E \in V^2$ l'ensemble des arêtes. L'arête reliant les sommets i et j sera notée ij et il est possible de lui associer un poids $a_{ij} > 0$. On définit la *matrice d'adjacence* $\mathbf{A}$ du graphe telle que $[\mathbf{A}]_{ij} = a_{ij}$ lorsqu'une arête existe entre i et j et $[\mathbf{A}]_{ij} = 0$ sinon. Le coefficient a_{ij} représente généralement une mesure de la similarité entre les sommets.
On définit la *matrice des degrés* $\mathbf{D}$ comme la matrice diagonale dont le i-ème élément correspond au *degré* du sommet i : $[\mathbf{D}]_{ii} = d_i = \sum_j a_{ij}$. Le *laplacien* $\mathbf{L}$ du graphe est alors donné par :

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

et le *volume du graphe* par $V_G = \sum_i d_i$.
Le *temps de commutation* $n(i, j)$ entre i et j est le nombre moyen de sauts (de franchissements d'arête) nécessaires, lors d'une marche aléatoire partant de i pour aller jusqu'à j et revenir en i, lorsque la probabilité de transition entre deux sommets quelconques x et y vaut $p_{xy} = a_{xy}/d_x$. Il peut s'exprimer en fonction des éléments de la matrice pseudo-inverse $\mathbf{L}^+$ du laplacien [10, 19] :

$$n(i, j) = V_G(l^+_{ii} + l^+_{jj} - 2l^+_{ij})$$

Soit $\mathbf{U}$ la matrice dont les colonnes contiennent les vecteurs propres de $\mathbf{L}^+$ et $\mathbf{\Lambda}$ la matrice diagonale dont les éléments sont les valeurs propres correspondantes. On a $\mathbf{L}^+ = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$.
En remarquant que $n(i, j)$ peut se mettre sous la forme :

$$n(i, j) = V_G(\mathbf{e}_i - \mathbf{e}_j)^T\mathbf{L}^+(\mathbf{e}_i - \mathbf{e}_j)$$

où les $\mathbf{e}_i$ sont les vecteurs unitaires de la base canonique définie sur les sommets et en effectuant le changement de variable $\mathbf{x}_i = \mathbf{\Lambda}^{1/2}\mathbf{U}^T\mathbf{e}_i$, on obtient :

$$n(i, j) = V_G(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)$$

Il existe donc un plongement du graphe dans lequel la quantité $\sqrt{n(i, j)}$ représente la distance euclidienne entre les sommets i et j. Les auteurs de [10] appelle cette distance la *distance euclidienne du temps de commutation* (DETC).

Réduction de dimension dans le domaine spectral. Il est possible de calculer une approximation de cette distance en se restreignant aux sous-espace formé par les k vecteurs propres associés aux k plus grandes valeurs propres de $\mathbf{L}^+$. Par définition de la matrice pseudo-inverse, $\mathbf{L}$ et $\mathbf{L}^+$ ont les même vecteurs propres et les valeurs propres associées de $\mathbf{L}^+$ sont simplement l'inverse des valeurs propres correspondantes de $\mathbf{L}$, sauf pour la valeur propre 0 qui reste inchangée. Il est alors possible de calculer l'approximation de la DETC en calculant les vecteurs propres associés aux k plus petites valeurs propres non-nulles de $\mathbf{L}$[3].

[3]Ce calcul peut se faire très efficacement en mettant à profit le fait que la matrice $\mathbf{L}$ est généralement creuse.

Les auteurs de [10] montrent que ceci équivaut à effectuer une ACP dans le domaine spectral. Nous utiliserons cette méthode pour calculer une approximation du temps de commutation d'une part, et pour obtenir une représentation des données dans le domaine spectral d'autre part.

3.3 Classification par k-PPV sur graphe

Il est donc ainsi possible de calculer le résultat d'une classification par k-plus proches voisins basée sur la distance dans ce domaine spectral réduit. Cette méthode est dénommée `LDCA+KNN-spec` dans les résultats.
Nous pouvons finalement combiner les résultats de la classification basée sur la distance de Mahalanobis et ceux de la classification par k-PPV en utilisant un séparateur à vaste marge (SVM) à base radiale. Les données fournies à ce dernier seront pour chaque paire, la valeur p_n donnée par la méthode LDCA et la valeur moyenne $\hat{t}_n$ des étiquettes des k-PPV dans le domaine spectral.

3.4 Combinaison des représentations LDCA et k-PPV sur graphe

Notre intuition est que la distance donnée par la méthode LDCA et la topologie du graphe contiennent des informations complémentaires qu'il est intéressant de combiner.
Nous réalisons cette combinaison en entrainant un classfieur SVM-RBF qui reçoit en entrée la probabilité de l'étiquette t_n donné par LDCA et le nombre moyen d'exemples positifs entourant une paire dans le graphe (k-plus proche voisinage). Cette méthode est dénommée `LDCA+KNN-spec+RBF` dans les résultas.

4 Validation expérimentale

Après avoir présenté différents algorithmes originaux, nous présentons ici leur validation expérimentale sur la base LFW.
Nous remercions les auteurs de [13] de nous avoir donné leur fichiers de descripteurs, ce qui nous permet de faire une comparaison rigoureuse avec leur approche, sachant que les données de départ sont strictement identiques et que les améliorations de performances ne peuvent être attribuées qu'à la méthode de reconnaissance elle-même.

4.1 Évaluation de la méthode LDCA

Nous avons tout d'abord souhaité valider la méthode LDCA seule. Dans ce cas la classification se fait par simple seuillage de la distance entre paires d'images.

Dimensionnalité de l'espace de projection. Dans un premier temps, nous avons cherché à déterminer quelle était la meilleure dimensionnalité pour l'espace de projection. Nous avons pour cela conduit des expériences sur la `vue1` (cf. section 1.2). Nous avons ainsi observé que les performances augmentaient avec la dimensionnalité jusqu'à ce qu'elle atteigne 40. Pour des valeurs supérieures les performances sont les mêmes.
En revanche, les temps de calculs deviennent prohibitifs au-delà de 100 dimensions. Nous avons donc choisi la di-

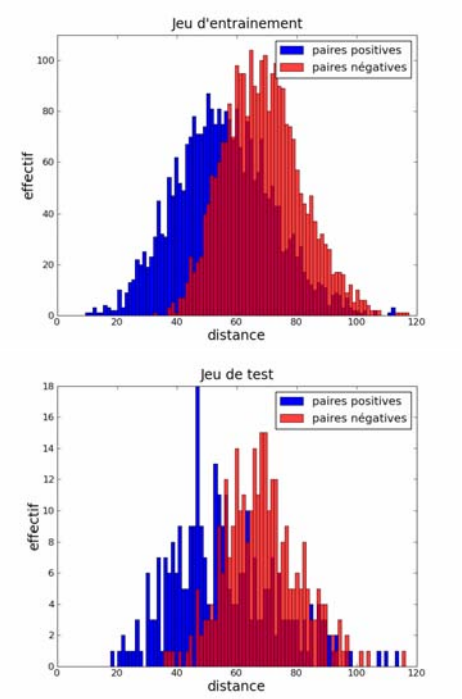

FIG. 2 – Distances entre les paires d'images positives et négatives dans l'espace d'entrée.

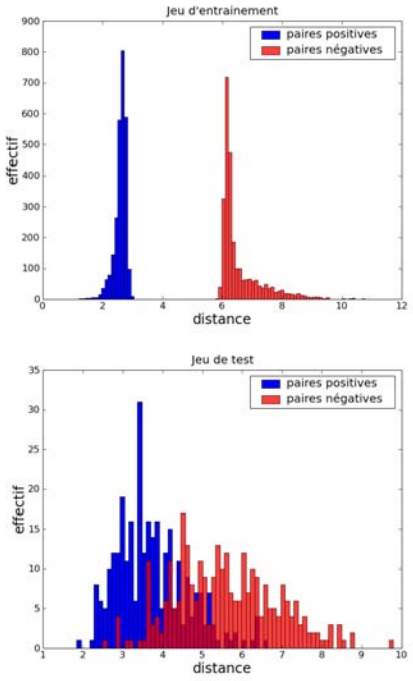

FIG. 3 – Distances entre les paires d'images du jeu d'entrainement et du jeu de test après apprentissage

mension de sortie la plus petite donnant de bons résultats, à savoir 40.
Notons que l'estimation de $\mathbf{M}$ est faite grâce à la méthode des gradients conjugués du module d'optimisation du paquetage SciPy ; cette estimation nécessite environ 2 minutes de calcul sur un PC standard.

Comparaison avec l'état de l'art. Le tableau 1 montre un aperçu des meilleurs résultats publiés pour la base LFW dans l'ordre chronologique, ainsi que les résultats que nous avons obtenus.
Nous constatons que la méthode LDCA donne un taux de bonne classification de $80.00 \pm 0.34\%$. Alors que la méthode LDML, qui donne les meilleurs résultats sur cette base, donne un score de $79.27 \pm 0.60\%$ (valeur donnée dans [13] pour une dimension de sortie de 35).

Visualisation de l'effet de l'apprentissage de distance. Nous pensons qu'il est intéressant de visualiser l'effet de la phase d'apprentissage de distance, et nous nous proposons de le faire au moyen d'histogrammes de distances.
La figure 2 montre la répartition des distances dans l'espace d'entrée pour les paires d'images respectivement positives et négatives, à la fois pour les données d'apprentissage et pour les données de test. Nous observons un fort recouvrement, ce qui explique que la distance dans l'espace d 'origine ne permette pas une bonne séparation entre les paires ; le taux de paires bien classées n'est que de $68.50 \pm 0.5\%$ (résultat tiré de [13]).
La répartition des distance après entrainement est montrée figure 3 : nous constatons que la séparation des distances sur le jeu d'entrainement est parfaite (avec une grande marge) et que sur le jeu de test la séparation des paires positives et négatives est bien plus marquée. Il n'est donc pas surprenant de constater une très forte amélioration des performances.

4.2 Évaluation de la méthode par k-PPV sur graphe

Dans cette seconde expérimentation, nous évaluons la méthode basées une classification par k-PPV sur graphe présentée section 3.
Nous supposons ici que les représentations des visages ont été projetées dans l'espace réduit défini section 2. Chaque paire est représentée ici par le vecteur différence entre les deux représentations des visages. Le problème se ramène donc dans ce cas à un problème de classification binaire.
Notons qu'une classification par k-plus proches voisins dans cet espace donne des résultats médiocres : 74.31% (voir table 1, ligne `LDCA+KNN-Spat` pour comparaison).

Évaluation quantitative. En revanche, la classification par k-plus proches voisins dans le domaine spectral (méthode `LDCA+KNN-Spec`) donne un taux de paires bien classées de 79.22 ± 0.29, ce qui montre que travailler dans un plongement spectral du graphe de voisinage permet bien de tirer parti d'une approche semi-supervisée. Toutefois, le score est légèrement inférieur à la méthode LDCA, probablement en raison du classifieur utilisé.

Visualisation des paires dans le domaine spectral. Une fois le graphe de voisinage construit, il est intéressant de re-

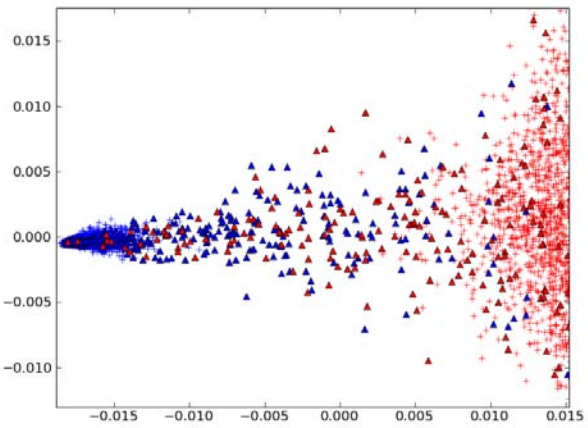

FIG. 4 – Répartition des descripteurs différence $\mathbf{d}_n$ dans le domaine spectral. Projection sur les deux premier vecteurs propres de la matrice $\mathbf{L}^+$. Les croix représentent les données d'entrainement et les triangles les données de test pour la première série de la `vue2`. En bleu les paires positives, en rouge les paires négatives.

garder comment sont répartis les $\mathbf{d}_n$ dans le domaine spectral. La figure 4 représente la projection des données sur les deux premiers axes de la matrice $\mathbf{L}^+$. Nous observons notamment que la séparation entre les paires positives et négatives est préservée pour les données d'entrainement. La figure met également en évidence le *cluster* formé par les paires positives.

Méthode	score (%)
Nowak[24]	73.93 ± 0.49
MERL+Nowak[15]	76.18 ± 0.58
Hybrid descriptor-based[33]	78.47 ± 0.51
LDML[13]	79.27 ± 0.60
LDCA	80.00 ± 0.34
LDCA+kNN-Spat	74.31 ± 1.76
LDCA+kNN-Spec	79.22 ± 0.29
LDCA+kNN-Spec+SVM-RBF	80.40 ± 0.39

TAB. 1 – Aperçu des meilleurs résultats publiés et comparaison avec nos résultats.

4.3 Évaluation de la méthode complète

La méthode complète consiste à classifier les paires au moyen d'un classifieur SVM-RBF prenant en entrée pour chaque paire la probabilité de l'étiquette $t_n = +1$ donnée par la méthode LDCA et la moyenne de ces étiquettes pour les plus proches voisins dans le graphe.
La combinaison de ces résultats donne au final, un score de $80.40 \pm 0.39\%$, significativement au dessus de l'état de l'art (voir table 1) et des résultats précédents.

5 Conclusions et perspectives

Dans cet article, nous avons proposé une méthode pour la classification de visages dans le contexte des *paires correspondantes*. L'approche proposée combine les avantages d'un apprentissage de distance par analyse en composantes logistiques discriminantes et l'apprentissage semi-supervisé au moyen d'un graphe.
Au moment où ces travaux ont été réalisés, ils ont permis d'obtenir des résultats supérieurs à l'état de l'art, sur la très difficile base *Labeled Faces in the Wild*.
Un article très récent [28] fait état de résultats encore meilleurs, par un découplage astucieux de la pose du visage et de la similarité. Une des pistes que nous allons suivre dans nos travaux futurs est justement d'intégrer cette idée intéressante dans notre approche.

Références

[1] Yael Adini, Yael Moses, and Shimon Ullman. Face recognition : the problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 :721–732, 1997.

[2] Ronen Basri and David Jacobs. Lambertian reflectance and linear subspaces. 25 :383–390, 2000.

[3] Peter Belhumeur and David Kriegman. What is the set of images of an object under all possible lighting conditions ? *IJCV*, 28 :270–277, 1996.

[4] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9) :1063–1074, 2003.

[5] Kevin W. Bowyer, Kyong Chang, and Patrick Flynn. A survey of approaches and challenges in 3d and multi-modal 3d + 2d face recognition. *Comput. Vis. Image Underst.*, 101(1) :1–15, 2006.

[6] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Proceedings of the European Conference on Computer Vision*, 2 :484–498, 1998.

[8] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *ICML '07 : Proceedings of the 24th international conference on Machine learning*, pages 209–216, New York, NY, USA, 2007. ACM.

[9] M. Everingham, J. Sivic, and A. Zisserman. Hello ! my name is... buffy – automatic naming of characters in tv video. In *Proceedings of the British Machine Vision Conference*, 2006.

[10] François Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation. *IEEE Transactions*

on Knowledge and Data Engineering, 19(3) :355–369, March 2007.

[11] Andrea Frome, Yoram Singer, Fei Sha, and Jitendra Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Proceedings of IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[12] A. Globerson and S. Roweis. Metric learning by collapsing classes. *Advances in Neural Information Processing Systems*, 18 :451–458, 2006.

[13] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Is that you ? metric learning approaches for face identification. In *International Conference on Computer Vision*, sep 2009.

[14] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild : A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, October 2007.

[15] G.B. Huang, M.J. Jones, and E. Learned Miller. LFW results using a combined Nowak plus MERL recognizer. In *Faces in Real-Life Images Workshop in European Conference on Computer Vision (ECCV)*, 2008.

[16] Vidit Jain, Erik G. Learned-Miller, and Andrew McCallum. People-lda : Anchoring topics to people using face recognition. In *ICCV*, pages 1–8. IEEE, 2007.

[17] Michael J. Jones and Tomaso Poggio. Multidimensional morphable models. In *ICCV*, pages 683–688, 1998.

[18] Micheal J. Jones. Face recognition : Where we are and where we go from here. *IEEJ Transactions on Electronic, Information and Systems*, 129 :770–777, 2009.

[19] J. D. Klein and M. Randić. Resistance distance. *Journal of Mathematical Chemistry*, 12(1) :81–95, December 1993.

[20] Jinho Lee, Baback Moghaddam, Hanspeter Pfister, and Raghu Machiraju. A bilinear illumination model for robust face recognition. In *ICCV '05 : Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1177–1184, Washington, DC, USA, 2005. IEEE Computer Society.

[21] David G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, pages 1150–1157, September 1999.

[22] Jianming Lu, Xue Yuan, and Takashi Yahagi. A method of face recognition based on fuzzy clustering and parallel neural networks. *Signal Process.*, 86(8) :2026–2039, 2006.

[23] Ajmal Mian, Mohammed Bennamoun, and Robyn Owens. An efficient multimodal 2d-3d hybrid approach to automatic face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11) :1927–1943, 2007.

[24] Eric Nowak and Frederic Jurie. Learning visual similarity measures for comparing never seen objects. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[25] Alice J. O'Toole, P. Jonathon Phillips, Fang Jiang, Janet Ayyad, Nils Penard, and Herv ? Abdi. Face recognition algorithms surpass humans matching faces over changes in illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9) :1642–1646, 2007.

[26] Alex Pentland, Baback Moghaddam, and Thad Starner. View-based and modular eigenspaces for face recognition.

[27] Shai Shalev-shwartz, Yoram Singer, and Andrew Y. Ng. Online and batch learning of pseudo-metrics. In *In International Conference on Machine Learning (ICML*, pages 743–750. ACM Press, 2004.

[28] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *The British Machine Vision Conference (BMVC)*, Sept. 2009.

[29] Lorenzo Torresani and Kuang C. Lee. Large margin component analysis. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1385–1392. MIT Press, Cambridge, MA, 2007.

[30] Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles : Tensorfaces. In *In Proceedings of the European Conference on Computer Vision*, volume 1, pages 447–460, 2002.

[31] Paul Viola and Michael Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57 :137–154, 2004.

[32] Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*. MIT Press, 2006.

[33] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Real-Life Images workshop at the European Conference on Computer Vision (ECCV)*, October 2008.

[34] John Wright, Allen Y. Yang, Arvind Ganesh, Shankar S. Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2) :210–227, 2009.

[35] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, volume 15, pages 505–512, 2002.