Université Mohamed Khaider de Biskra
Faculté des Sciences et de la Technologie
Département de Génie Électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Télécommunication
Réseaux et Télécommunication

Présenté et soutenu par :

LOUAM ABDELHAK BILAL

Le : Samedi 6 juillet 2019

Deep Learning basé sur les méthodes de réduction pour la reconnaissance de visage

---

Jury :

M. Abdelmalik OUAMANE	MCA	Université de Biskra	Président
M. Noura ATAMENA	MAA	Université de Biskra	Examinatrice
M. Mebarka BELAHCENE	Pr	Université de Biskra	Encadreur

Année universitaire : 2018 – 2019

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement Supérieur et de la recherche scientifique

Université Mohamed Khider Biskra

Faculté des Sciences et de la Technologie
Département de Génie Électrique
Filière : Électronique
Option : Réseaux et Télécommunication

Mémoire de Fin d'Études
En vue de l'obtention du diplôme:

MASTER

Thème

Deep Learning basé sur les méthodes de réduction pour la reconnaissance de visage

Présenté par :
LOUAM ABDELHAK BILAL

Avis favorable de l'encadreur :
BELAHCENE Mebarka

Avis favorable du Président du Jury

Abdelmalik OUAMANE
Cachet et signature

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement Supérieur et de la Recherche scientifique

Université Mohamed Khider Biskra

Faculté des Sciences et de la Technologie
Département de Génie Électrique
Filière : Électronique
Option : Réseaux et Télécommunication

Thème :

Deep Learning basé sur les méthodes de réduction pour la reconnaissance de visage

Proposé par : BELAHCENE Mebarka
Dirigé par : BELAHCENE Mebarka

Résumé

L'apprentissage en profondeur est devenu un nouveau domaine d'apprentissage automatique et s'applique à un certain nombre de domaines.
Le travail présenté dans ce travail a pour objectif principal d'appliquer le concept d'un algorithme d'apprentissage en profondeur, à savoir:
Méthodes de Réduction de Dimension (MRD) dans la reconnaissance de visage. L'algorithme est testé sur la base de données FERET (Dup II). La performance de l'algorithme est évaluée en fonction de la métrique de qualité connue sous le nom taux d'erreur égale (TEE) et de la précision (Accuracy) de la reconnaissance. La représentation graphique des résultats expérimentaux sont donnés sur la base de la TEE et Accuracy par rapport au nombre de paramètres caractéristiques. L'analyse des résultats expérimentaux (basée sur la métrique de qualité et précision) et la représentation graphique prouvent que l'algorithme (PCANet2-SVD et PCANet2-LDA) donnent une assez bonne précision de classification pour tous les ensembles de données tests et sont compétitifs à l'état de l'art.

ملخص:

أصبح التعلم المتعمق مجالًا جديدًا للتعلم الآلي وينطبق على عدد من الميادين. الغرض الرئيسي من العمل المقدم في هذا العمل هو تطبيق مفهوم خوارزمية التعلم العميق ، وهي:

أساليب التقليل من الأبعاد (MRD) في التعرف على الوجوه. تم اختبار الخوارزمية على مجموعة بيانات FERET(DupII). يتم تقييم أداء الخوارزمية وفقًا لمقياس الجودة المعروف باسم معدل الخطأ المتساوي (TEE) والدقة (التعرف). يتم تقديم التمثيل الرسومي للنتائج التجريبية على أساس TEE والدقة فيما يتعلق بعدد العينات المميزة. يثبت تحليل النتائج التجريبية (استنادًا إلى مقاييس الجودة والدقة) والتمثيل الرسومي أن الخوارزمية (PCANet2-SVD و PCANet2-LDA) تعطي دقة تصنيف جيدة إلى حد ما لجميع مجموعات بيانات الاختبار وهي تنافسية مع آخر ما توصل إليه.

Introduction Générale

La reconnaissance de visage est un des problèmes les plus étudiés de l'apprentissage automatique. Il a été bien étudié au cours des 50 dernières années. Les premières tentatives d'exploration de la reconnaissance faciale ont été faites dans les années 60, mais c'est jusqu'à ce que Turk et Pentland aient mis en œuvre l'algorithme «Eigenfaces» pour que ce champ produise des résultats vraiment intéressants et utiles.

Récemment la reconnaissance de visage attire de plus en plus d'attention. La sécurité reste le domaine d'application principal. Dans ce domaine la reconnaissance de visage est responsable de l'identification et de l'authentification.

La reconnaissance de visage peut également être utilisée pour accélérer l'identification des personnes. Nous pouvons imaginer un système reconnaissant le client dès son entrée dans une succursale (banque, assurance) et le préposé au bureau d'accueil pourra ensuite accueillir le client par son nom et préparer son dossier avant qu'il ne parvienne au comptoir.

La reconnaissance de visage a reçu une attention croissante en raison de ses nombreuses applications potentielles dans divers domaines. Les systèmes de RV les plus récents dépendent principalement des représentations de caractéristiques obtenues à l'aide de descripteurs localisés à la main, de modèles de commandes binaires locales (LBP) ou d'une approche d'apprentissage en profondeur [1].

L'identification automatisée des individus peut s'accomplir par une reconnaissance de visages qui nécessite l'application de techniques avancées et robustes de sorte à éviter les erreurs de classification des gens détectés. SRV font face à plusieurs défis majeurs liés à l'environnement de capture des images ainsi qu'aux méthodes d'évaluation des visages qui ont été détectés. Entre autres, les problèmes d'illumination, de pose et d'expression variante sur les visages à identifier font que la reconnaissance est très difficile à paramétrer. Aussi, l'acquisition des images peut s'effectuer dans divers environnements de capture plus ou moins contrôlés, ce qui apporte des difficultés supplémentaires par rapport à la résolution des images, l'ajout de mouvement des individus dans la scène, ainsi que certains cas d'occlusion des visages.

L'identification et la vérification des visages ont attirées l'attention des chercheurs depuis quelques décennies, et restent encore et toujours un sujet de recherche attractif et très ouvert. Beaucoup de connaissances dans les domaines de la reconnaissance des formes, du traitement d'images, des statistiques ont été appliquées au domaine de la reconnaissance du visage. En plus, les capacités grandissantes des moyens informatiques et l'existence de bases de

données de grandes tailles ont permis de mettre au point des algorithmes et des approches de plus en plus complexes [2]. Les données de haute dimension peuvent être converties en codes de basse dimension en formant un réseau de neurones multicouches avec une petite couche centrale afin de reconstruire des vecteurs d'entrée de haute dimension [3].

Avec le développement de Deep Learning (DL), Les réseaux de neurones profonds (DNN), en particulier les réseaux de neurones convolutifs (CNN), construits par des opérations de convolution et de mise en commun, ont fait l'objet d'une attention soutenue [4]. DL est devenue une nouvelle tendance pour extraire des fonctionnalités en raison de son efficacité à découvrir des structures complexes de données. Il est différent des modèles d'extraction de caractéristiques artisanaux traditionnels [5], et permet d'obtenir des résultats étonnants en matière de reconnaissance [4].

L'apprentissage supervisé (supervised Learning en anglais) est une tâche d'apprentissage automatique consistant à apprendre une fonction de prédiction à partir d'exemples annotés, au contraire de l'apprentissage non supervisé. Les exemples annotés constituent une base d'apprentissage, et la fonction de prédiction apprise peut aussi être appelée « hypothèse » ou « modèle ». On suppose cette base d'apprentissage représentative d'une population d'échantillons plus large et le but des méthodes d'apprentissage supervisé est de bien généraliser, c'est-à-dire d'apprendre une fonction qui fasse des prédictions correctes sur des données non présentes dans l'ensemble d'apprentissage. Ce qui nous motive à nous intéresser davantage à l'apprentissage automatique.

La complexité de la dimensionnalité est connue depuis plus de trois décennies et son impact varie d'un domaine à l'autre. Dans l'optimisation combinatoire sur plusieurs dimensions, elle est vue comme une croissance exponentielle de l'effort de calcul avec le nombre de dimensions. En statistique, cela se traduit par un problème d'estimation de paramètre ou de densité, dû au manque de données. L'effet négatif de cette rareté résulte de certaines propriétés géométriques, statistiques et asymptotiques de l'espace des fonctions de grande dimension. Ces caractéristiques présentent un comportement surprenant pour les données en dimensions supérieures [6].

La réduction de dimensionnalité facilite la classification, la visualisation, la communication et le stockage de données de grande dimension. Une méthode simple et largement utilisée est l'analyse en composantes principales (ACP), qui recherche les directions

de la plus grande variance dans l'ensemble de données et représente chaque point de données par ses coordonnées le long de chacune de ces directions [3].

Traditionnellement, la réduction de la dimensionnalité était réalisée à l'aide de techniques linéaires telles que l'analyse en composantes principales, l'analyse factorielle et la mise à l'échelle classique. Cependant, ces techniques linéaires ne peuvent pas traiter de manière adéquate des données non linéaires complexes. Un grand nombre de techniques non linéaires de réduction de dimensionnalité ont été proposées [7].

Apprentissage automatique et réduction du nombre de dimensions ? La faculté d'apprendre est essentielle à l'être humain pour reconnaître une voix, une personne, un objet... On distingue en général deux types d'apprentissage : l'apprentissage «par cœur» qui consiste à mémoriser telles quelles des informations, et l'apprentissage par généralisation où l'on apprend à partir d'exemples un modèle qui nous permettra de reconnaître de nouveaux exemples. Pour les systèmes informatiques, il est facile de mémoriser un grand nombre de données (textes, images, vidéos...), mais difficile de généraliser. Par exemple, il leur est difficile de construire un bon modèle d'un objet et d'être ensuite capable de reconnaître efficacement cet objet dans de nouvelles images. L'apprentissage automatique est une tentative de comprendre et de reproduire cette faculté d'apprentissage dans des systèmes artificiels. Il nous semble donc approprié d'utiliser des techniques issues de ce domaine pour découvrir et modéliser des connaissances liant texte et image, et pouvoir ainsi réduire le fossé sémantique.

Dans le domaine de l'indexation et la recherche d'images, l'apprentissage peut jouer les rôles suivants : sélectionner les descripteurs visuels les plus pertinents, associer des descripteurs de bas niveaux à des concepts, regrouper les images de manière hiérarchique ou non par similarité visuelle et/ou conceptuelle, apprendre au fur et à mesure des interactions avec les utilisateurs,...

Le grand volume de données multimédia et le nombre important de dimensions nécessaires pour les décrire imposent d'utiliser des techniques de réduction de dimensions afin de pouvoir, d'une part, extraire et résumer les connaissances souvent inconnues des données, et, d'autre part, indexer et retrouver rapidement les documents. On peut séparer les techniques d'analyses de données en deux grandes catégories : les classifications qui permettent de

réduire la taille de l'ensemble des individus en regroupant ceux qui se ressemblent, et les analyses factorielles qui permettent de réduire le nombre de descripteurs.

Dans ce mémoire, nous nous intéressons d'abord aux techniques de réduction de données et d'apprentissage automatique en général, et nous montrons quels sont leurs avantages et difficultés. Puis, nous présentons le problème de la malédiction de la dimension, et nous décrivons quelques techniques que l'on peut utiliser pour réduire le nombre de dimensions d'un espace.

Nous allons étudier dans notre travail trois méthodes de réduction de dimension des images basée sur le CNN (PCANet2, PCANet2-SVD, PCANet2-LDA). Ces dernières seront validées sur la base de données FERET (Dup- II).

Notre mémoire est structuré en quatre chapitres :

- Le **chapitre 1** repose sur les méthodes de réduction et Deep Learning pour RV.
- Le **chapitre 2** expose l'état de l'art sur les méthodes de réduction de dimensionnalité des images basée sur Deep Learning.
- Le **chapitre 3** nous définissons la conception de réduction en profondeur PCANet, LDANet, SVDNet.
- Le **chapitre 4** est consacré à l'implémentation de nos méthodes de réduction et aux résultats obtenus. Nous utilisons trois méthodes différentes.

Notre mémoire se termine par une conclusion générale.

Dédicaces

Je dédie ce travail

A mon père Nacerallah et ma mère Hakima en témoignage de leur affectation, leurs sacrifices et de leurs précieux conseils qui m'ont conduit à la réussite dans mes études.

A mes frère Anis et Sohaib, ma sœur Hadjer, ma tante Wassila et toute la famille pour leurs soutiens et encouragements.

A mes chères amis Iliasse, Zied et surtout mon cousin Louam Younes et tous les amis proches.

À tous les professeurs et enseignants qui m'ont suivi durant tout mon cursus scolaire et qui m'ont permis de réussir dans mes études.

Et à tous ceux que j'aime.

Abdelhak Bilal

Remerciements

Nous remercions tout d'abord, **ALLAH** qui nous a donné la force et le courage pour terminer nos études et élaborer ce modeste travail.

J'exprime ma gratitude et je tiens à remercier **Prof. Belahcen Mebarka**, qui m'a encadré. Sa grande connaissance dans le domaine, ainsi que son expérience, ont joué un rôle important dans la conception de ce travail.

Mes remerciements et mes respects vont également à **Mr. OUAMANE Abdelmalik** d'avoir accepté de présider le jury, J'adresse mes remerciements aussi à **Mme. ATAMENA Noura** qui m'ont fait l'honneur d'accepter le jugement de mon travail.

Je saisis aussi l'occasion pour remercier tous les enseignants du département de génie électrique.

Enfin, je remercie affectueusement tous ceux qui m'ont soutenu dans mes études, mes parents, ma famille et mes amis.

Liste d'abréviations

RF : Reconnaissance faciale

RV : Reconnaissance de Visage

DL : Deep Learning

PCA : Analyse en Composantes Principales

ADL : Analyse Discriminante Linéaire (LDA)

SILD : Side-Information based Linear Discriminant Analysis

KDF : Kernel Analyse Discriminante de Fisher

PCANet : réseau d'Analyse en Composantes Principales

DeepLDA : Analyse de Discrimination linéaire profonde

DCTNet : Réseau Transformations Cosinus Discretes

DCCA : Deep Canonical Correlation Analysis

LDANet : Réseau d'Analyse Discriminante Linéaire

SVDNet : Réseau de Décomposition Vecteur Singulier

FERET : Facial Recognition Technology

SVM : Support Vector Machines

DPM : modèle de pièce déformable

CNN : Convolutional Neural Network

DNN : Deep Neural Network

FAR : False Accept Rate (Taux de fausse acceptation)

FRR : False Reject Rate (Taux de faux rejet)

EER : Error Equal Rate (Taux d'égale erreur)

ROC : Receiver Operating Characteristic

TP : taux des vrais positifs

FP : taux des faux positifs

LBP : binaire local pattern

FER : reconnaissance d'expression faciale

DCNN : Deep-CNN

BDD : base de donnée

AI : Artificial Intelligence

Bibliographie

[1] Al-Waisy, Alaa & Qahwaji, Rami & Ipson, Stanley & Al-Fahdawi, Shumoos. (2017). A multimodal deep learning framework using local feature representations for face recognition. Machine Vision and Applications. 10.1007/s00138-017-0870-2.

[2] M. Belahcene A, Ouamane M, Boumehrez A, Benakcha. Comparaison des méthodes de réduction d'espace et l'application des SVMs pour la classification dans l'authentification de visages. Courrier du Savoir, [S.l.], v. 13, mai 2014. ISSN 1112-3338.

[3] Hinton, G.E. & Salakhutdinov, R.R. (2006). Reducing the Dimensionality of Data with Neural Networks. Science (New York, N.Y.). 313. 504-7. 10.1126/science.1127647.

[4] Yu, Dan & Wu, Xiao-Jun. (2017). 2DPCANet: A deep leaning network for face recognition. MultiMedia Tools and Applications. 77. 10.1007/s11042-017-4923-3.

[5] Sun, Kai & Zhang, Jiangshe & Yong, Hongwei & Liu, Junmin. (2018). FPCANet: Fisher discrimination for Principal Component Analysis Network. Knowledge-Based Systems. 166. 10.1016/j.knosys.2018.12.015.

[6] O. Jimenez, Luis & A. Landgrebe, David. (1998). Supervised classification in high-dimensional space: Geometrical, statistical, and asymptotical properties of multivariate data. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on. 28. 39 - 54. 10.1109/5326.661089.

[7] Van der Maaten, Laurens & Postma, Eric & Herik, H. (2007). Dimensionality Reduction: A Comparative Review. Journal of Machine Learning Research - JMLR. 10.

[8]A. Ouamane, M. Bengherabi, A. Hadid and M. Cheriet, "Side-Information based Exponential Discriminant Analysis for face verification in the wild," *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Ljubljana, 2015, pp. 1-6.

[9] Adjoudj, Reda & GAFOUR, Abdel-Kader & Boukelif, Aoued & Lehireche, Ahmed. (2008). La reconnaissance des visages: une comparaison entre les réseaux des neurones compétitifs et les réseaux des neurones à spike.

[10] M. Turk, A. Pentland, Eigenfaces for recognition. J. of Cognitive Neuroscience 3, 72–86, 1991.

[11] Anouar Mellakh, Reconnaissance des visages en conditions dégradées. Thèse de doctorat préparée au Département Électronique et Physique de l'Institut National des Télécommunications dans le cadre de l'École Doctorale SITEVERY en co-accrédidation avec l'université d'Evry-Val d'Essonne. 2009.

[12] Scholkopf, Bernhard, Smola, Alexander, and Muller, Klaus-Robert. Kernel principal component analysis. In Advances in Kernel Methods – Support Vector Learning, pp. 327–352. MIT Press, 1999.

[13] S. Mika, G. RQatsch, J. Weston, B. Sch Qolkopf, K.-R. Muller, Q Fisher discriminant analysis with kernels, IEEE International Workshop on Neural Networks for Signal Processing IX, Madison, USA, August, 1999, pp. 41– 48.

[14] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Comput. 12 (10) (2000) 2385–2404.

[15] K. Chellapilla, S. Puri, et P. Simard, 2006. High performance convolutional neural networks for document processing. Dans les actes de Tenth International Workshop on Frontiers in Handwriting Recognition.

[16] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, et Y. Bengio, 2010a. Theano : A CPU and GPU math expression compiler. Dans les actes de SciPy. Oral Presentation.

[17] F. Chollet. Keras: Theano-based deep learning library. Code : https ://github.com/fchollet Documentation : http ://keras.io.

[18] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, et Z. Zhang, 2015. Mxnet: A lexible and eicient machine learning library for heterogeneous distributed systems. CoRR abs/1512.01274.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, et Y. Bengio, 2014. Generative adversarial nets. Dans les actes de NIPS, 2672–2680.

[20] S. Iofe et C. Szegedy, 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Dans les actes de ICML, 448–456.

[21] D. Clevert, T. Unterthiner, et S. Hochreiter, 2015. Fast and accurate deep network learning by exponential linear units (elus). ICLR.

[22] D. P. Kingma et J. Ba, 2015. Adam: A method for stochastic optimization. ICLR.

[23] S.Song et J.Xiao, 2016. Deep sliding shapes for amodal 3d object detection in rgb-d images. Dans les actes de The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[24] C.-Y. Lee et S. Osindero, 2016. Recursive recurrent nets with attention modeling for ocr in the wild. Dans les actes de The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[25] W.Yang, W.Ouyang, H.Li,et X.Wang,2016. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. Dans les actes de The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[26] Q. Wu, H. Zhang, S. Liu, et X. Cao, 2015. Multimedia analysis with deep learning. Dans les actes de 2015 IEEE International Conference on Multimedia Big Data.

[27] Killian Janod. Neural network representations for spoken documents Understanding. Theses, Universit´e d'Avignon, November 2017.

[28] Lionel Pibre, Marc Chaumont, Dino Ienco, and Jérôme Pasquet. Etude des réseaux de neurones sur la stéganalyse. InCORESA: Compression et Représentation des Signaux Audiovisuels, Nancy, France, May 2016.

[29] https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529?fbclid=IwAR0UWoPkFYTEAqFitcR4fuuQUcNBvxV8ig1oJGZ3EhbRypu8Qf9pk9CXdy4

[30] https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1b-relu-layer/

[31] Y. LeCun, F.J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 2, pages II–97. IEEE, 2004.

[32] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesianapproachtestedon101objectcategories. Computer Vision and Image Understanding, 106(1):59–70, 2007.

[33] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. URL http://authors.library.caltech.edu/7694.

[34] A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto, 2009.

[35] D. Cires¸an, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. Arxiv preprint arXiv: 1202.2745, 2012.

[36] N. Pinto, D.D. Cox, and J.J. DiCarlo. Why is real-world visual object recognition hard? PLoS computational biology, 4(1):e27, 2008.

[37] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. Labelme: a database and web-based tool for image annotation. International journal of computer vision, 77(1):157–173, 2008.

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.

[39] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In International Conference on Computer Vision, pages 2146–2153. IEEE, 2009.

[40] A. Krizhevsky. Convolutional deep belief networks on cifar-10. Unpublished manuscript, 2010.

[41] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 609–616. ACM, 2009.

[42] Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, et al. Hand written digit recognition with a back-propagation network. In Advances in neural information processing systems, 1990.

[43] N. Pinto, D. Doukhan, J.J. DiCarlo, and D.D. Cox. A high-throughput screening approach to discovering good forms of biologically inspired visual representation. PLoS computational biology, 5(11):e1000579, 2009.

[44] S.C.Turaga, J.F.Murray, V.Jain, F.Roth, M.Helmstaedter, K.Briggman, W.Denk and H.S.Seung. Convolutional networks can learn to generate affinity graphs for image segmentation. Neural Computation, 22(2):511–538, 2010.

[45] T. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng and Y. Ma, "PCANet: A Simple Deep Learning Baseline for Image Classification?" in IEEE Transactions on Image Processing, vol. 24, no. 12, pp. 5017-5032, Dec. 2015.

[46] Meina Kan, Shiguang Shan, Dong Xu and Xilin Chen. Side-Information based Linear Discriminant Analysis for Face Recognition. In Jesse Hoey, Stephen McKenna and Emanuele Trucco, Proceedings of the British Machine Vision Conference, pages 125.1-125.0. BMVA Press, September 2011.

[47] (Krizhevsky et al.,) A.Krizhevsky, I.Sutskever, et G.E.Hinton. Imagenet classification with deep convolutional neural networks. Dans les actes de NIPS, 1097–1105.

[48] (He et al., 2016) K. He, X. Zhang, S. Ren, et J. Sun, 2016. Deep residual learning for image recognition. Dans les actes de CVPR, 770–778.

[49] (Xiong et al.,2017b) W.Xiong, J.Droppo, X.Huang, F.Seide, M.L.Seltzer, A.Stolcke, D.Yu, et G.Zweig, 2017b. Toward human parity in conversational speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing 25(12), 2410– 2423.

[50] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: a review and new perspectives,” IEEE TPAMI, vol. 35, no. 8, pp. 1798–1828, 2013.

[51] G. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” Neural Computation, vol. 18, no. 7, pp. 1527– 1554, 2006.

[52] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, “Maxout networks,” in ICML, 2013.

[53] L. Sifre and S. Mallat, “Rotation, scaling and deformation invariant scattering for texture discrimination,” in CVPR, 2013.

[54] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, and Y. LeCun, “Learning convolutional feature hierarchies for visual recognition,” in NIPS, 2010.

[55] Dorfer, Matthias & Kelz, Rainer & Widmer, Gerhard. (2015). Deep Linear Discriminant Analysis.

[56] Clevert, Djork-arné, Unterthiner, Thomas, Mayr, Andreas, Ramsauer, Hubert, andHochreiter, Sepp. Rectified factor networks. In Advances in neural information processing systems, 2015.

[57] Fisher, Ronald A. The use of multiple measurements in taxonomic problems. Annals of eugenics, 7 (2):179–188, 1936.

[58] Andrew, Galen, Arora, Raman, Bilmes, Jeff, and Livescu, Karen. Deep canonical correlation analysis. In Proceedings of the 30th International Conference on Machine Learning, pp. 1247–1255, 2013.

[59] Stuhlsatz, Andre, Lippel, Jens, and Zielke, Thomas. Feature extraction with deep neural networks by a generalized discriminant analysis. IEEE Transactions on Neural Networks and Learning Systems, 23(4):596–608, 2012.

[60] Ng, Cong Jie & Teoh, Andrew. (2015). DCTNet: A simple Learning-free Approach for Face Recognition. 10.1109/APSIPA.2015.7415375.

[61] Yu, Dan & Wu, Xiao-Jun. (2017). 2DPCANet: a deep leaning network for face recognition. Multimedia Tools and Applications. 77. 10.1007/s11042-017-4923-3.

[62] Feng Z et al (2015) DLANet: a manifold-learning-based discriminative feature learning network for scene classification. Neurocomputing 157:11–21

[63] Y. Le Cun et Y. Bengio, Convolutional Network for Speech, Image and Time-Series, AT&T Bell Laboratory, Holmdel, 1991

[64] S. NECIB, «Fusion de face 3D couleur,profondeur et profil pour srv3D,» mémoire de master, Université de Mohamed khaider , Biskra, 2013

[65] P. Buyssens et A. Elmoataz, «Réseaux de neurones convolutionnels multi-échelle pour la classification cellulaire,» RFIA 2016, Clermont-Ferrand, France, Jun 2016.

[66] M. Baccouche, « Apprentissage neuronal de caractéristiques spatio-temporelles pour la classification automatique de séquences vidéo » These de doctorat, INSA de Lyon, France, 2013.

[67] Blanc-Durand, Paul. (2018). Réseaux de neurones convolutifs en médecine nucléaire : Applications à la segmentation automatique des tumeurs gliales et à la correction d'atténuation en TEP/IRM. 10.13140/RG.2.2.23231.97441.

[68] https://medium.com/machine-learning-for-li/different-convolutional-layers-43dc146f4d0e

[69]. Mehdipour Ghazi, Mostafa & Yanikoglu, Berrin & Aptoula, Erchan & Muslu, Özlem & Ozdemir, Murat Can. (2015). Sabanci-Okan System in LifeCLEF 2015 Plant Identification Competition.

[70] https://fr.wikipedia.org/wiki/D%C3%A9composition_en_valeurs_singuli%C3%A8res

[71] Chapelle, Olivier & Haffner, Patrick & Vapnik, Vladimir. (2000). SVMs for Histogram-Based Image Classification.

[72] P. Jonathon Phillips, Harry Wechsler, Jeffery Huang, Patrick J. Rauss. The FERET database and evaluation procedure for face-recognition algorithms (1997). Promising Research, U.S. Army Research Laboratory, George Mason University.

Chapitre 1
Méthodes de réduction de dimension des images basée sur le CNN

1.1 Introduction

Dans la vision par ordinateur, la reconnaissance des visages est devenue de plus en plus importante dans la société d'aujourd'hui. L'intérêt récent pour la reconnaissance du visage peut être attribué à l'intérêt commercial croissant et au développement de technologies réalisables pour soutenir le développement de la reconnaissance du visage. Les principaux domaines d'intérêt commercial comprennent la biométrie, l'application de la loi et la surveillance, les cartes à puce et le contrôle d'accès. Contrairement à d'autres formes d'identification telles que l'analyse d'empreintes digitales et le balayage de l'iris, la reconnaissance des visages est conviviale et non intrusive. Les scénarios possibles de reconnaissance des visages comprennent : l'identification à la porte d'entrée pour la sécurité à domicile, la reconnaissance au guichet automatique ou en conjonction avec une carte à puce pour l'authentification, la surveillance vidéo pour la sécurité. Avec l'avènement des supports électroniques, en particulier des ordinateurs, la société est de plus en plus dépendante des ordinateurs pour le traitement, le stockage et la transmission d'informations. L'ordinateur joue un rôle important dans toutes les parties de la vie et de la société d'aujourd'hui dans la civilisation moderne. À mesure que la technologie progresse, l'homme s'investit dans l'informatique en tant que leader de cette ère technologique et la révolution technologique a eu lieu dans le monde entier. Il a ouvert une nouvelle ère pour que l'humanité puisse entrer dans un nouveau monde, communément appelé le monde technologique. La vision par ordinateur fait partie de la vie quotidienne. L'un des objectifs les plus importants de la vision par ordinateur consiste à atteindre une capacité de reconnaissance visuelle comparable à celle de l'humaine.

1.2 Système de reconnaissance de visage

La reconnaissance des visages fait partie du domaine de la reconnaissance des formes. Le but de la reconnaissance des formes est de classifier des objets d'intérêt dans un certain nombre de catégories ou de classes. Les objets d'intérêt sont appelés généralement les modèles ou patterns et dans notre cas ils sont des vecteurs de caractéristiques appelés les matrices de code, ces derniers sont extraits à partir des images de visage d'input en utilisant les techniques décrites dans les sections suivantes. Les classes ici représentent les différentes personnes. Puisque la procédure de classification dans notre cas sera appliquée sur des vecteurs de caractéristiques, elle peut être donc désignée par le nom de la comparaison de caractéristiques [8].

L'approche de la **figure 1.1** démontre comment un système de reconnaissance des visages peut être conçu par un réseau de neurones artificiel conventionnel et par un autre réseau de neurones plus récent, qui est appelé un réseau de neurones à spike ou un réseau de neurones à potentiel d'action asynchrone, les deux réseaux de neurones sont employés en tant que processus de comparaison ou de reconnaissance. La figure ci-dessous résume comment le système de reconnaissance proposé fonctionne [9].

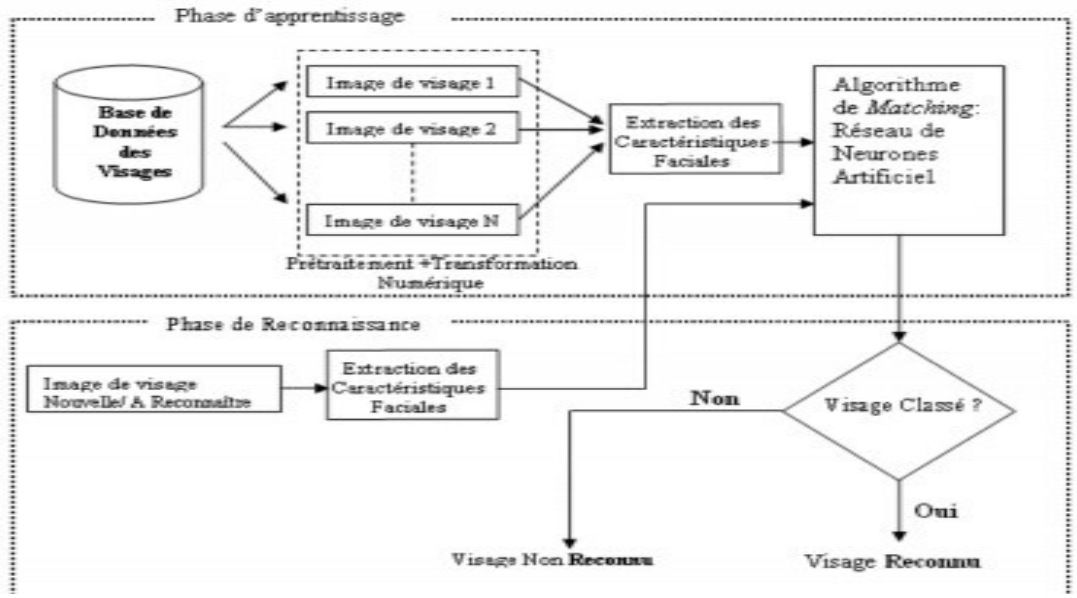


Figure 1. 1 Système de reconnaissance des visages basée sur le Réseau de Neurones Artificiel [9]

1.3 Présentation des méthodes de réduction dimensionnelle

Une image du visage est un signal à deux dimensions, acquis par un capteur digital. Ce capteur codera la couleur ou l'intensité des différents points de l'image dans une matrice de pixels à deux dimensions dans l'espace des images, nous devons spécifier une valeur pour chaque pixel de cette image. Le nombre de points constituant cet espace devient rapidement très grand, même pour les images de petite dimension. Cette dimensionnalité pose un certain nombre de problèmes pour les algorithmes de reconnaissance, qui se basent sur cette représentation de l'image, à savoir :

- Dans un contexte de la reconnaissance de visages, travailler dans un grand espace pose un problème de complexité de calcul.
- Pour les méthodes paramétriques, le nombre de paramètres à estimer peut rapidement dépasser le nombre d'échantillons d'apprentissage, ce qui pénalise l'estimation.
- Pour les méthodes non paramétriques, le nombre d'exemples nécessaires afin de représenter efficacement la distribution des données peut être insuffisant.

En 1994, Ruderman a démontré que les images naturelles possèdent une grande redondance statistique. En 1996, Penev a démontré que dans le cas précis des images normalisées des visages, cette redondance statistique est d'autant plus forte.
L'avantage de la redondance statistique est qu'elle permet une extraction d'une structure simple des caractéristiques importantes et pertinentes de l'image du visage. Cette structure permettrait de représenter le visage tout en gardant l'information la plus importante, et par conséquent, de réduire la dimensionnalité de l'espace visage. Tout l'intérêt des approches globales est la construction de cette base de projection qui permettra de comparer, de reconnaître ou d'analyser l'information essentielle des visages [2].

1.3.1 Les méthodes Linéaires

1.3.1.1 Analyse en Composantes Principales ou ACP

Le premier système de reconnaissance de visages qui a permis d'obtenir des résultats significatifs a été réalisé par Turk et Pentland [10] en utilisant la méthode dite des « Eigenfaces » est l'une des méthodes d'analyse les plus populaires. L'ACP a pour objectif de résumer les informations contenues dans des données en réduisant la dimensionnalité des données sans perdre les informations importantes.

1.3.1.2 Analyse Discriminante Linéaire (ADL)

L'ACP est d'abord utilisée pour projeter les images dans un espace de données inférieur. Le but de l'ADL est de maximiser les distances inter-classes tout en minimisant les distances intra-classe [11]. La LDA est une méthode d'analyse numérique qui permet de chercher la combinaison linéaire des variables qui représentent au mieux les données, elle est très utilisée dans le domaine de la reconnaissance des formes à savoir la reconnaissance de visage. Elle permet de maximiser l'éparpillement inter-classes (the between-class scatter) et de réduire l'éparpillement intra-classes (the within-class scatter).

1.3.1.3 Side-Information based Linear Discriminant Analysis (SILD)

La méthode SILD qui ne fonctionne bien qu'avec side-information, dans laquelle les matrices de dispersion intra-classe et inter-classes sont calculées en utilisant directement Side-Information. Il est à noter que cette méthode est différente de la FLDA à deux classes. En effet, une seule direction de projection peut être obtenue en utilisant une FLDA à deux classes, tandis que beaucoup plus de directions de projection peuvent être obtenues en utilisant la méthode. De plus, théoriquement la

méthode SILD est équivalente à FLDA multi-classes lorsque des étiquettes de classe sont prévues.

1.3.1.4 Soft Linear Discriminant Analysis (SLDA)

L'analyse discriminante linéaire (ADL) est l'une des méthodes les plus utilisées dans la reconnaissance de formes pour réduire la dimensionnalité des vecteurs de caractéristiques, augmentant généralement la robustesse des caractéristiques. Une modification de la LDA afin de pouvoir gérer des ambiguïtés étiquettes de référence de manière souple. Nous démontrons que mesure de précision douce évaluant la sortie de confiance du classifieur au moyen d'une base de données de discours émotionnels à étiquette douce, offre un degré de similarité avec les votes humains (naturellement ambigus)

1.3.2 Les méthodes Non Linéaires

1.3.2.1 Kernel Analyse en Composantes Principales (KACP)

L'ACP (PCA en anglais) standard permet uniquement la réduction de la dimensionnalité linéaire. Cependant, si les structures de données sont plus compliquées et ne peuvent pas être bien représentées dans un espace linéaire, la PCA standard (basique) ne sera pas très utile. Heureusement, le noyau PCA nous permet de généraliser PCA standard à la réduction de dimensionnalité non linéaire [12]

1.3.2.2 Kernel Analyse Discriminante de Fisher (KAFD)

L'analyse discriminante linéaire de Fisher (LDA) est une technique statistique traditionnelle de réduction de la dimensionnalité. Elle a été largement utilisée et a fait ses preuves dans de nombreuses applications du monde réel. Mais, en raison de sa limite de linéarité, LDA ne parvient pas à bien fonctionner pour tous les problèmes posés par la non linéarité. Pour pallier cette faiblesse de la LDA, des versions non linéaires de l'analyse discriminante de Fisher ont été proposées. Au cours des dernières années, Mika [13] a formulé le noyau Fisher discriminant (KFD) pour les cas à deux classes, tandis que Baudat [14] a développé l'analyse discriminante généralisée du noyau (GDA) pour les problèmes multi-classes. Depuis des problèmes à deux classes sont un cas particulier de problèmes multi-classes, nous nous concentrons sur l'analyse KFD pour les cas à classes multiples. KFD s’avère plus efficace que LDA dans divers domaines et applications. Cependant, les algorithmes KFD existants ne sont pas aussi simples et transparents que LDA. C'est la formalisation complexe des algorithmes KFD qui couvre les caractéristiques intuitives de l'analyse discriminante du noyau.

1.4 Généralités du Deep Learning dans l'Imagerie

L'introduction des neurones artificiels et leurs applications sont survenues dans les années 60. Depuis 2009, les réseaux de neurones artificiels profonds sont devenus les outils majeurs de l'apprentissage automatique, et ce en grande partie grâce à l'implémentation de réseaux de neurones sur carte graphique (Graphical Processing Unit (GPU)) [15]. L'utilisation de GPU a permis d'entrainer plus rapidement des réseaux plus profonds sur des volumes de données plus importants. De nombreuses bibliothèques logicielles [16-17-18] 2 ont vu le jour pour simplifier l'utilisation des ressources de calcul GPU. L'étude des réseaux de neurones profonds est un domaine de recherche en plein essor avec notamment les récentes introductions d'architectures [19], de méthodes de régularisation [20], des fonctions d'activation [21], des méthodes d'optimisation [22], etc. Ils sont, par ailleurs, le moteur d'une recherche intensive dans de nombreux domaines applicatifs. Ils sont devenus incontournables, en traitement d'images [23-24-25] ou encore pour le traitement de documents multimédia [26]. Ils sont aussi devenus source d'innovation dans de nombreux autres domaines applicatifs, par exemple : le traitement automatique de données médicales, l'amélioration de la sécurité routière ainsi que l'apprentissage automatique pour la reconnaissance biométrique notamment le visage.

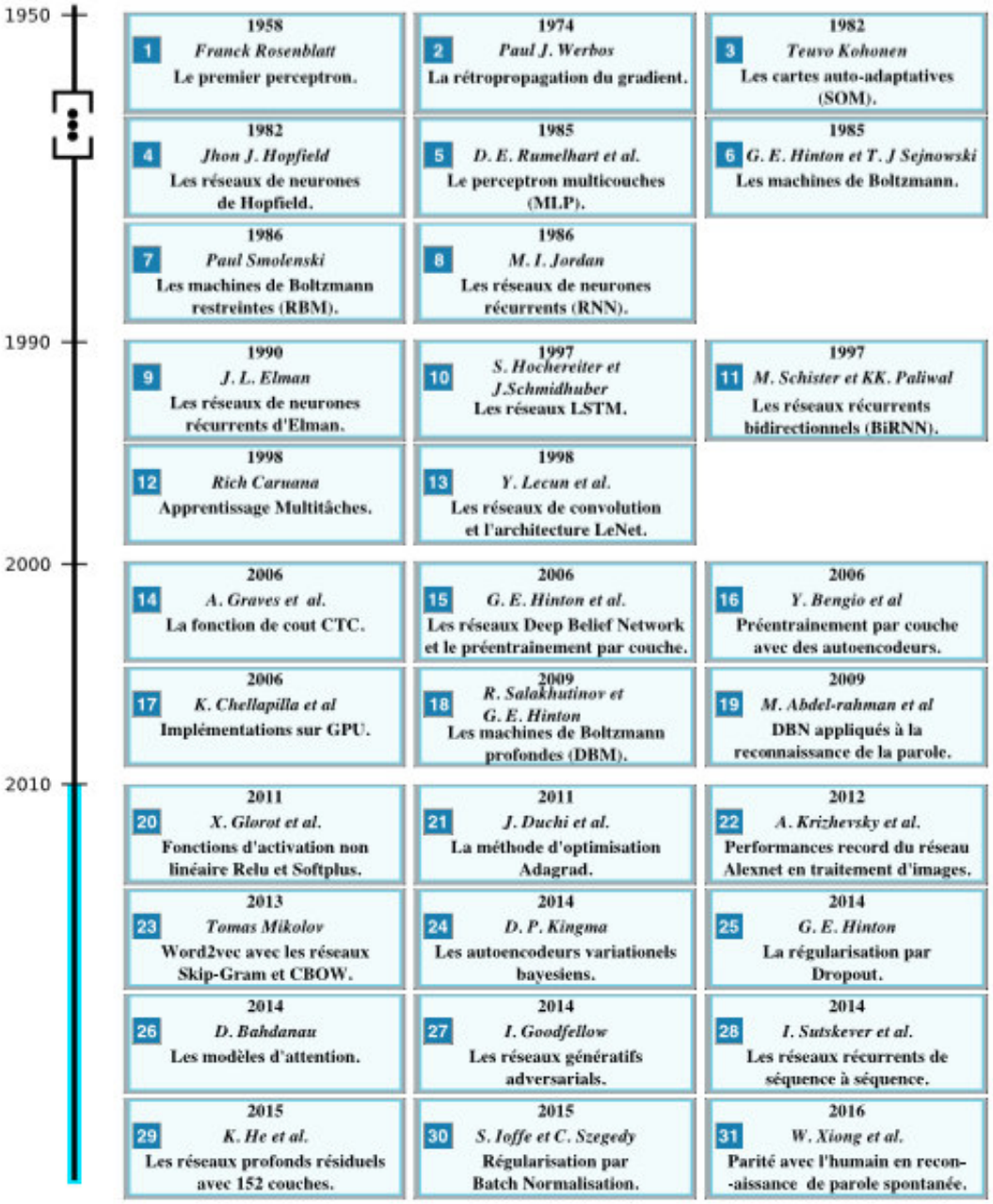


Figure 1. 2 Chronologie de l'évolution de l'apprentissage profond [27]

1.5 Réseaux de neurones

Les réseaux de neurones (Neural Networks) est l'un des algorithmes d'apprentissage automatique les plus populaires à l'heure actuelle. Au fil du temps, il a été prouvé de manière décisive que les réseaux de neurones surpassent les autres algorithmes en termes de précision et de rapidité. Avec diverses variantes telles que CNN (Réseaux de Neurones Convolutionnels (abrégé en CNN)), RNN (Réseaux de Neurones Récurrents), Auto-Encodeurs, etc…, les réseaux de neurones deviennent peu à peu pour les scientifiques ou les praticiens de l'apprentissage automatique, ce que la régression linéaire était pour les statisticiens.

Les réseaux de neurones profonds (CNN) ont connu un succès considérable dans la reconnaissance et la localisation d'objets dans des images. L'approche fondamentale qui a conduit aux CNN consiste à construire des systèmes artificiels basés sur le cerveau et la vision humaine. Pourtant, sous de nombreux aspects importants, les capacités des CNN sont inférieures à celles de la vision humaine. Un axe de recherche prometteur consiste à étudier les similitudes et les différences en comblant les lacunes, pour améliorer les CNN. Il existe plusieurs algorithmes de Deep Learning.

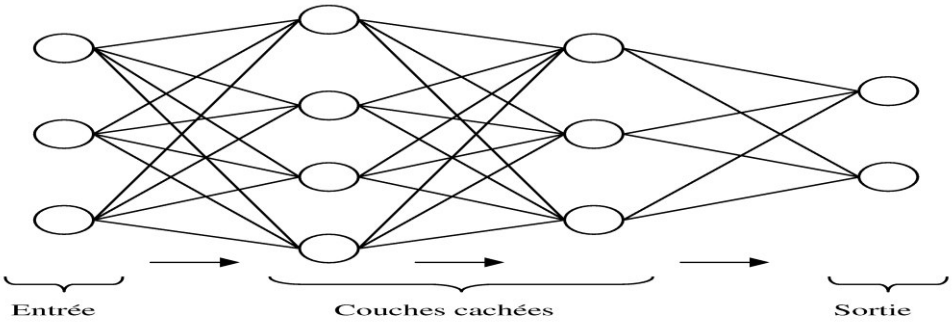


Figure 1. 3 Schéma d'un réseau de neurones artificiel

Dans notre travail, nous nous intéressons aux méthodes de réduction pour l'apprentissage en profondeur, il est essentiel de présenter le CNN qui est fondamental pour comprendre la reconnaissance de visage basée sur le Deep Learning.

Dans ce qui suit, nous présentons un aperçu sur le CNN.

1.5.1 Réseaux de neurones convolutifs (CNN)

La vision par ordinateur évolue rapidement de jour en jour. Une des raisons est le développement de l'apprentissage en profondeur. Lorsque nous parlons de vision par ordinateur, un terme réseau de neurones convolutionnel nous vient à l'esprit parce que CNN est fortement utilisé ici. Les exemples de CNN en vision par ordinateur sont la reconnaissance de visage, la classification d'images, etc. Il est similaire au réseau de neurones de base. CNN possède également des paramètres pouvant être appris, tels que le réseau de neurones, à savoir les pondérations, les biais…Un exemple de schéma de principe est illustré par la **figure 1.4**.

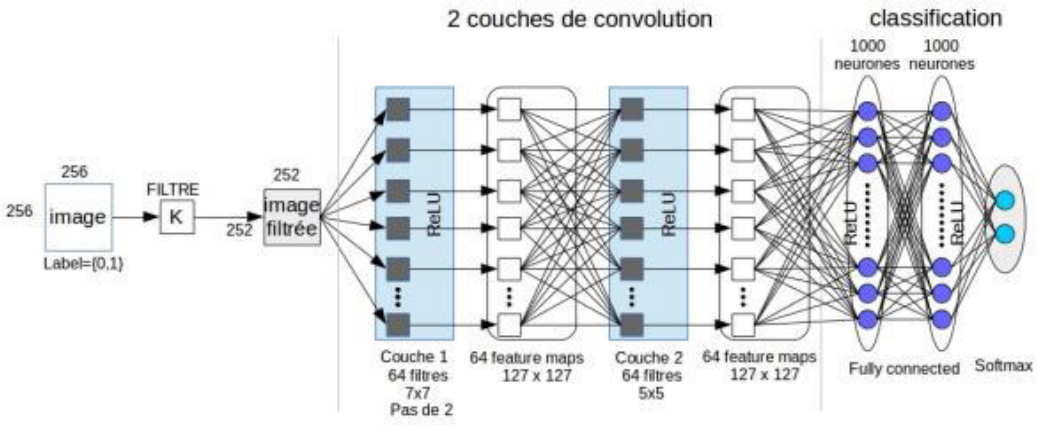


Figure 1. 4 Exemple de schéma de principe du CNN [28]

1.5.2 Les couches de réseaux de neurones convolutionnels

Il y a plusieurs couches différentes dans CNN comme le montre la **figure 1.5** :

- Couche d'entrée (Input layer).
- Couche de convolution (Convo layer : Convolution + ReLU).
- Couche de Pooling.
- Couche entièrement connectée (Couche Fully connected).
- Couche Softmax/logistique.
- Couche de sortie (Output layer).

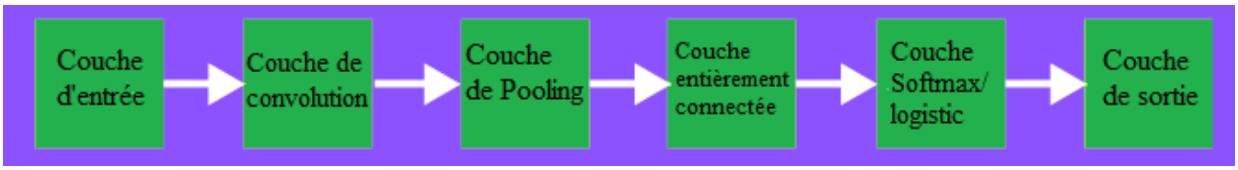


Figure 1. 5 Les couches de CNN [29]

Un exemple de l'architecture du CNN est montré par la **Figure 1.6**

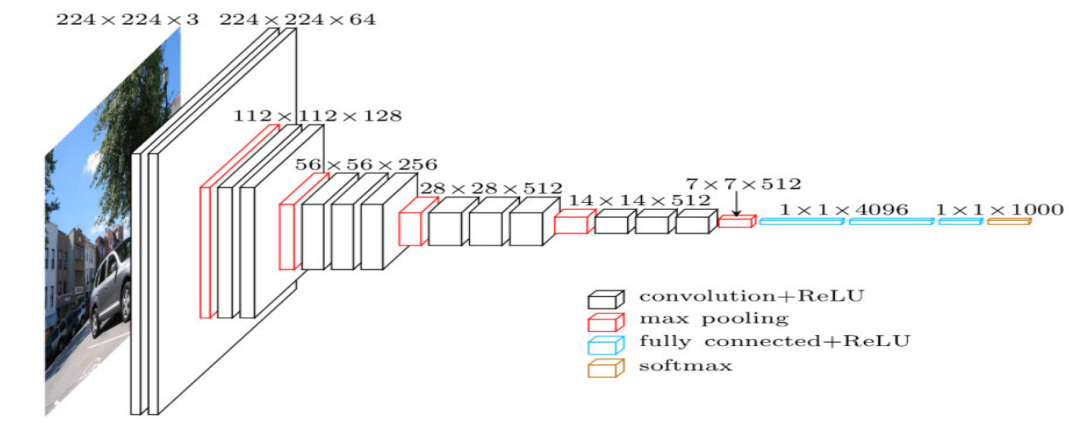


Figure 1. 6 Exemple d'architecture CNN [29]

1.5.2.1 Couche d'entrée du CNN

La couche d'entrée dans CNN doit contenir des données décrivant l'image. Les données d'image sont représentées par une matrice tridimensionnelle qui en général doit être remodelée en une seule colonne (représentation vectorielle).

1.5.2.2 Couche de convolution

La couche de convolution est parfois appelée couche d'extraction de caractéristiques, car les caractéristiques de l'image sont extraites dans cette couche. Tout d'abord, une partie de l'image est connectée à la couche Convo pour effectuer une opération de convolution et calculer le produit scalaire entre le champ récepteur (c'est une région locale de l'image d'entrée ayant la même taille que celle du filtre) et le filtre comme le montre la **figure 1.6**. Le résultat de l'opération est un entier unique du volume de sortie. Ensuite, nous faisons glisser le filtre sur le champ récepteur suivant de la même image d'entrée par une foulée et refaisons la même opération. Cette opération est répétée par le même processus encore et encore jusqu'à ce que toute l'image soit parcourue.

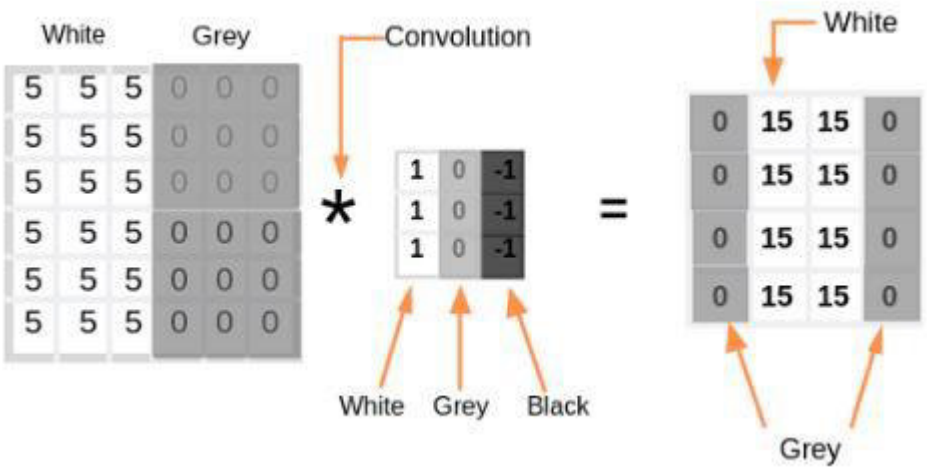


Figure 1. 7 Exemple de principe du filtre convolutionnel [29]

La couche Convo contient également l'activation ReLU voir **figure 1.7** pour que toutes les valeurs négatives soient mises à zéro.

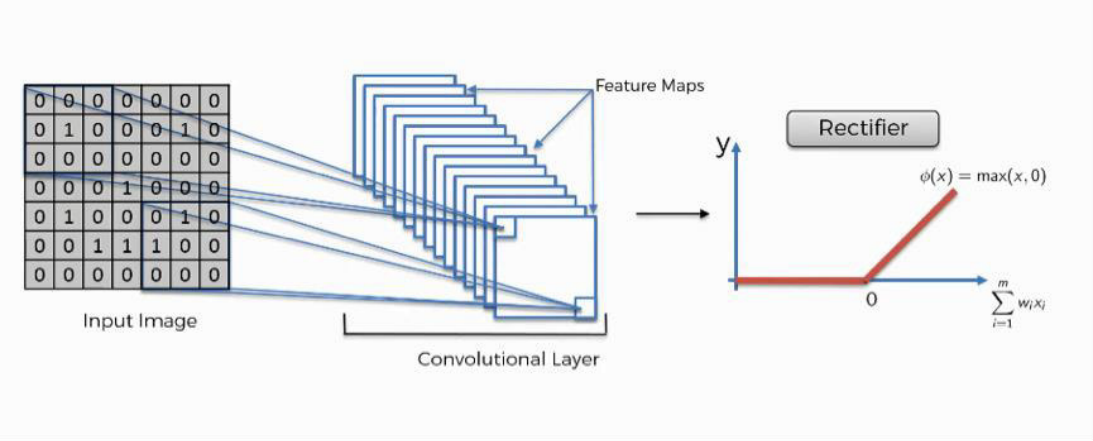


Figure 1. 8 Principe de la fonction ReLu [30]

1.5.2.3 Couche de mise en commun (Pooling)

La couche de Pooling est utilisée pour réduire le volume spatial de l'image d'entrée après la convolution. Elle est utilisée entre deux couches de convolution. Si nous appliquons ***fc*** (Fully Connected) après la couche Convo sans appliquer le pooling ou le pooling maximum, le calcul sera coûteux. Ainsi, la mise en commun maximale est le seul moyen de réduire le volume spatial de l'image d'entrée en codant l'information.

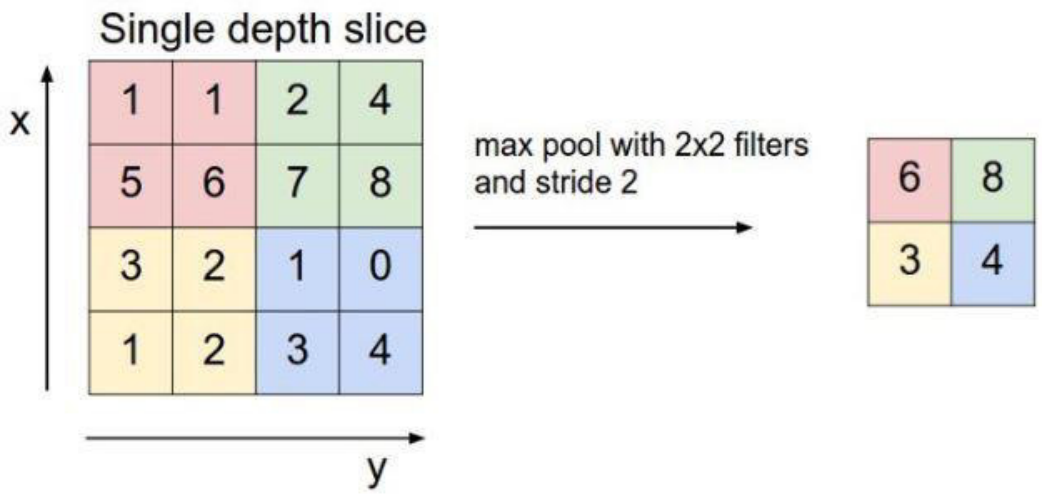


Figure 1. 9 Exemple de principe du Pooling [29]

1.5.2.4 Couche entièrement connecté (Fully Connected)

Une couche entièrement connectée implique des poids, des biais et des neurones. Il connecte les neurones d'une couche aux neurones d'une autre couche. Il est utilisé pour classer les images entre différentes catégories par formation.

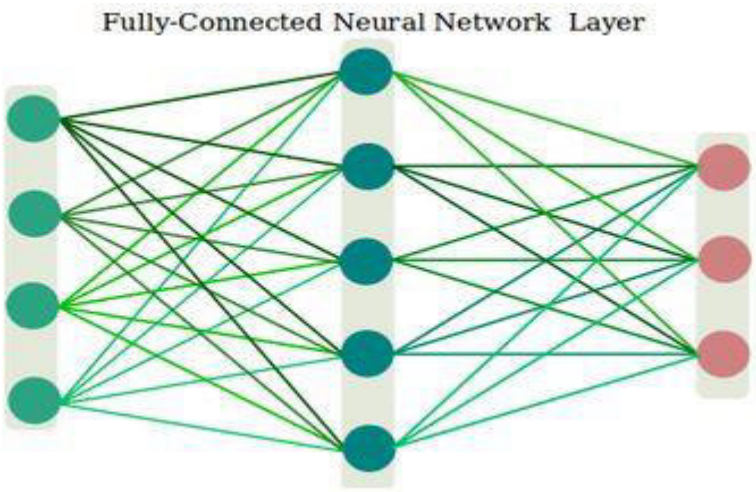


Figure 1. 10 Principe de la couche entièrement connectée (fc)

1.5.2.4 Couche Logistique ou Softmax

Softmax ou couche logistique est la dernière couche de CNN. Elle réside à la fin de la couche FC. La logistique est utilisée pour la classification binaire et Softmax est pour la multi-classification.

1.5.2.5 Couche de sortie (output layer)

La couche de sortie contient l'étiquette qui est sous forme codée comme le montre la **figure 1.11**.

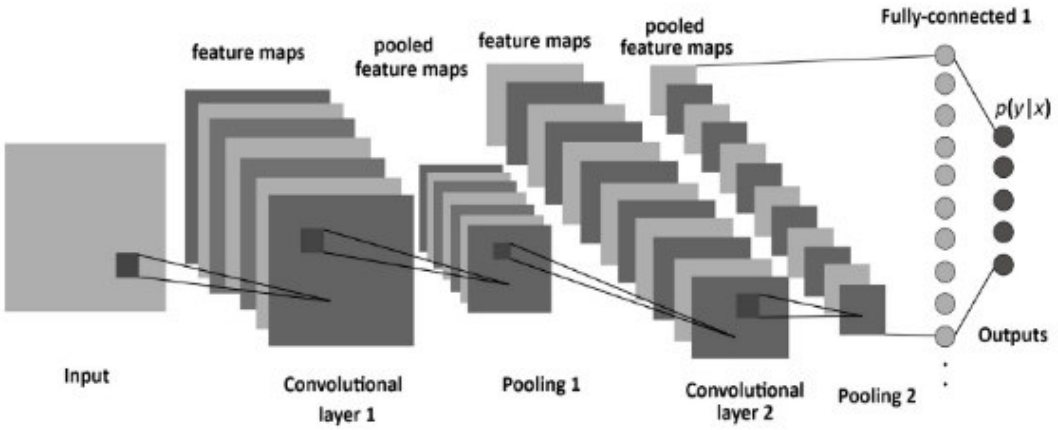


Figure 1. 11 Exemple montrant l'étiquette codée de la couche de sortie CNN

1.6 Deep Learning et Système de Reconnaissance de Visage

1.6.1 Deep Learning pour la Reconnaissance d'objets

Les approches actuelles de la reconnaissance d'objet font un usage essentiel des méthodes d'apprentissage automatique. Pour améliorer leurs performances, on peut collecter des ensembles de données plus volumineux, apprendre des modèles plus puissants et utiliser de meilleures techniques pour éviter les sur-ajustements. Jusqu'à récemment, les ensembles de données d'images étiquetées étaient relativement petits - de l'ordre de dizaines de milliers d'images (par exemple, NORB [31], Caltech-101/256 [32,33] et CIFAR-10/100 [34]). Les tâches de reconnaissance simples peuvent très bien être résolues avec des jeux de données de cette taille, en particulier s'ils sont complétés par des transformations préservant les étiquettes. Par exemple, le taux d'erreur actuel courant sur la tâche de reconnaissance de chiffres MNIST (<0,3%) est proche de celui de la performance humaine [35]. Mais les objets dans des environnements réalistes présentent une variabilité considérable, aussi, pour apprendre à les reconnaître, il est nécessaire d'utiliser des ensembles d'entraînement beaucoup plus volumineux. Et en effet, les lacunes des petits ensembles de données d'image ont été largement reconnues (par exemple, Pintoetal. [36]), mais il est récemment apparu qu'il est parfois possible de collecter des données étiquetées avec des millions d'images. Les nouveaux jeux de données plus volumineux incluent LabelMe [37], qui contient des centaines de milliers d'images entièrement segmentées, et ImageNet [38], qui contient plus de 15 millions d'images haute résolution étiquetées réparties dans plus de 22 000 catégories. Pour en savoir plus sur des milliers d'objets à partir de millions d'images, nous avons besoin d'un modèle doté d'une grande capacité d'apprentissage. Cependant, l'immense complexité de la tâche de reconnaissance d'objets signifie que ce problème ne peut pas être spécifié même par un jeu de données aussi grand que ImageNet. Notre modèle doit donc également disposer de nombreuses connaissances préalables pour compenser toutes les données que nous n'avons pas. Les réseaux de neurones convolutifs (CNN) constituent l'une de ces classes de modèles [31, 39, 40, 41, 42, 43, 44]. Leur capacité peut être contrôlée en fonction de leur largeur minimale et de leurs hypothèses approximatives sur la nature des images (à savoir la stationnarité des statistiques et la localisation des dépendances de pixels). Ainsi, par rapport aux réseaux neuronaux à réaction standard avec des couches de taille similaire, les CNN ont beaucoup de connexions et de paramètres en plus, même si leur meilleure performance théorique n'est que légèrement pire.

1.6.2 Méthodes de réduction associées au Deep Learning

- **PCANet :** une très simple réseau d'apprentissage en profondeur pour la classification d'images ne comprenant que les composants de base du traitement de données: analyse en cascade des composants principaux (PCA), hachage binaire et histogrammes par blocs [45].

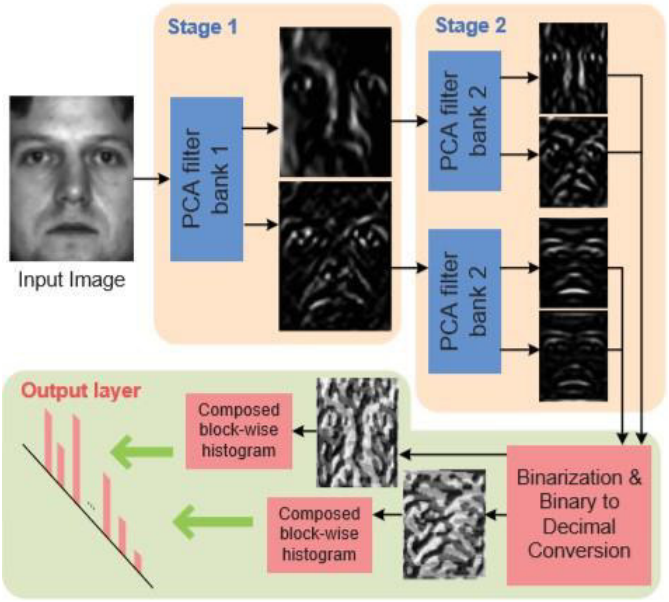


Figure 1. 12 Illustration de la manière dont le PCANet proposé extrait les caractéristiques d'une image par le biais de trois composants de traitement les plus simples: PCA filtres, binary hashing, and histogram [45]

- **Analyse de Discrimination linéaire profonde (DeepLDA)**
  Apprend des représentations latentes séparables linéairement de manière intégrale. Classique LDA extrait des fonctionnalités qui préservent la séparabilité des classes et sont utilisées pour la réduction de la dimensionnalité dans de nombreux problèmes de classification. L'idée centrale est de mettre LDA sur un réseau de neurones profond. Cela peut être vu comme une extension non linéaire du LDA classique

1.7 Performances du système de Reconnaissance de Visage

Pour évaluer les performances d'un système RV, trois critères principaux doivent déjà être clairement définis:

- **Taux de faux rejet ("False Reject Rate" ou FRR(TFR))**
  Ce taux représente le pourcentage d'individus qui devraient être reconnus, mais ils sont néanmoins rejetés par le système [2].

- **Taux de fausse acceptation ("False Accept Rate" ou FAR (TFA))**
  Ce taux représente le pourcentage d'individus qui ne devraient pas être reconnus, mais ils sont néanmoins acceptés par le système [2].

- **Taux d'égale erreur ("Equal Error Rate" ou EER (TEE))**
  Ce taux représente la mesure de performance optimale et est calculé en fonction des deux premiers critères. Il atteint lorsque TFR = TFA, c'est-a`-dire le meilleur compromis entre faux rejet et faux acceptations [2].

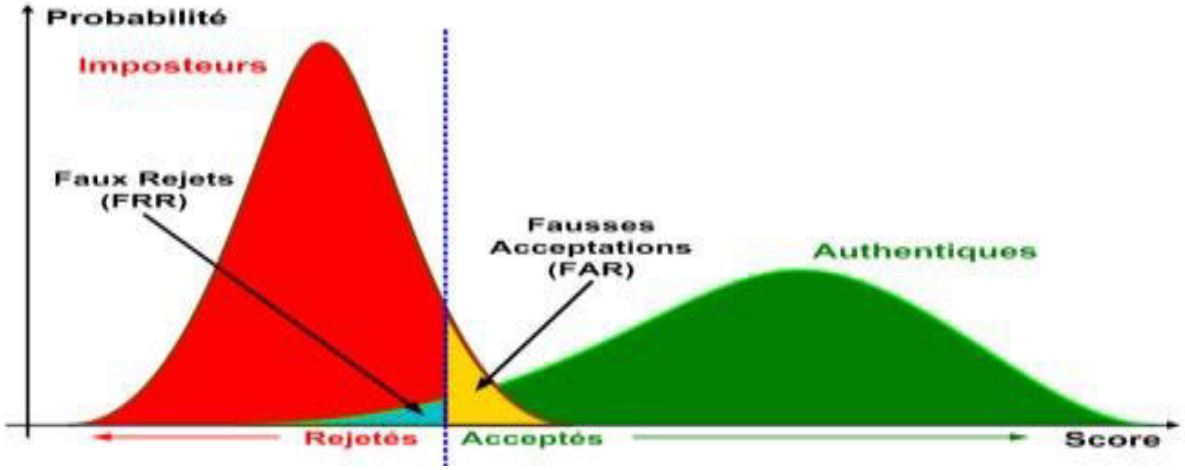


Figure 1. 13 Illustration du FRR et du FAR [2].

Lorsque le système opère en mode authentification, on utilise ce que l'on appelle une courbe ROC ("Receiver Operating Characteristic" en anglais). La courbe ROC (Figure 1.14) trace le taux de faux rejet en fonction du taux de fausse acceptation. Plus cette courbe tend à épouser la forme du repère, plus le système est performant, c'est-a`-dire possédant un taux de reconnaissance global élevé. La performance globale d'un système de vérification d'identité est mieux caractérisée par sa courbe caractéristique de fonctionnement (ROC), qui représente le TFA en fonction du TFR.

Figure 1. 14 Courbe ROC [2].

Conclusion

Les réseaux de neurones offrent un cadre de travail souple et robuste dont les intérêts ont été largement démontrés expérimentalement. Ils ont permis de franchir des étapes importantes pour le développement de l'apprentissage automatique, notamment, et en traitement de l'image (reconnaissance de visage) [47] où des performances de systèmes automatiques sont comparables à celles obtenues par les humains dans certaines conditions [48] [49]. Les très grands corpus rendus disponibles récemment par exemple le Yahoo news feeds dataset9, le Google audio set10 et le Google video set11 couplés aux puissantes capacités de modélisation des réseaux profonds qui évoluent rapidement, laissent présager l'exploration de nouveaux domaines de recherche ainsi que de nouvelles applications intéressantes.

Chapitre 2

État de l'Art sur Deep Learning et Méthodes de Réduction

Introduction

Depuis quelques années, on observe un besoin croissant pour des systèmes automatiques d'identification de personnes. Rien n'est plus naturel que d'utiliser le visage pour identifier une personne. Les images faciales sont probablement la caractéristique biométrique la plus communément employée par l'homme pour effectuer une identification personnelle.

La reconnaissance et l'identification de visages joue un rôle fondamental dans nos interactions sociales, elle est basée sur notre capacité de reconnaître les personnes, elle ne présente pas de difficultés énormes pour un être humain, mais elle constitue pour tout système informatique une situation extrêmement délicate. Dans notre travail, nous utilisons le Deep Learning pour automatiser et améliorer la vitesse du système de reconnaissance de visage, son pourcentage de réussite ainsi que sa robustesse.

Lorsque nous traitons de vrais problèmes de données réelles, le traitement se fait souvent avec des données de grandes dimensions pouvant aller jusqu'à des millions d'images. Pour ce cas de structure d'origine à haute dimension, nous avons besoin de réduire leur dimensionnalité. La nécessité de réduire la dimensionnalité est donc réelle et a de nombreuses applications.

Une longue liste de méthodes bien connues est utilisée pour réduire la dimension des données et les visualiser dans un espace réduit. Seulement, la plus part des méthodes d'entre elles restent supervisées et nécessitent des phases nécessaires d'extraction de caractéristiques et de classification.

Avec le développement du matériel, beaucoup d'algorithme d'apprentissage en profondeur ont vu le jour, cette dernière décennie. Dans ce chapitre, nous présentons l'essentiel de l'état de l'art des méthodes de réduction pour un apprentissage en profondeur pour la reconnaissance et la classification des visages.

2.1 Méthode de réduction PCANet (Principal Component Analysis Network)

Dans ce travail [18], un réseau très simple d'apprentissage en profondeur est proposé pour la classification des images, qui ne comprend que les composants de base du traitement de données: l'analyse en cascade des composants principaux (PCA), le hachage binaire et les histogrammes par blocs. Dans l'architecture proposée, PCA est utilisé pour apprendre des banques de filtres à plusieurs étages. Il est suivi d'un simple hachage binaire et d'histogrammes de bloc pour l'indexation et la mise en commun (pool). Cette architecture est donc appelée réseau PCA (PCANet) et peut être conçue et apprise de manière extrêmement simple et efficace. À des fins de comparaison et de compréhension, deux variantes simples de PCANet sont étudiées et présentées, à savoir RandNet et LDANet. Ils partagent la même topologie de PCANet mais leurs filtres en cascade sont soit choisis au hasard, soit appris de LDA. Ces réseaux de base sont largement testés sur de nombreux jeux de données visuels de référence pour différentes tâches, telles que LFW pour la vérification des visages, MultiPIE,

Extended Yale B, AR, FERET pour la reconnaissance des visages, ainsi que MNIST pour la reconnaissance de chiffres manuscrits.

La classification des images basée sur le contenu visuel est une tâche très difficile, en grande partie à cause de la grande variabilité intra-classe, résultant d'éclairages différents, d'un désalignement, de déformations non rigides, d'occlusion et de bruit. De nombreux efforts ont été déployés pour contrer la variabilité intra-classe en concevant manuellement des fonctionnalités de bas niveau pour les tâches de classification en cours. Des exemples représentatifs sont les caractéristiques de Gabor et les motifs binaires locaux (LBP) pour la classification des textures et des visages, ainsi que les caractéristiques SIFT et HOG pour la reconnaissance des objets. Bien que les fonctionnalités de bas niveau puissent être conçues avec le plus grand succès pour certaines données et tâches spécifiques, la conception de fonctionnalités efficaces pour les nouvelles données et tâches requiert généralement de nouvelles connaissances de domaine, certaines fonctionnalités spécialement conçues à la main ne pouvant être simplement adoptées à de nouvelles conditions [50], [51].

Un exemple de telles méthodes est l'apprentissage par l'intermédiaire de réseaux de neurones profonds DNN (Deep Neural Network), qui a récemment fait l'objet d'une attention considérable [50]. L'apprentissage en profondeur vise à découvrir plusieurs niveaux de représentation, en espérant que les entités de niveau supérieur représentent une sémantique plus abstraite des données. De telles représentations abstraites apprises à partir d'un réseau profond devraient fournir plus d'invariance à la variabilité intra-classe. Un ingrédient clé du succès de l'apprentissage en profondeur dans la classification des images et l'utilisation des architectures convolutionnels [52] - [53].

2.1.1 Architecture de la PCANet

Une architecture de réseau de neurones profonds convolutifs (ConvNet) [52] - [39], [47], [54] se compose de plusieurs étages pouvant être entraînés, empilés les uns sur les autres, suivis d'un classifieur supervisé. Chaque étape comprend généralement des «trois couches» - une couche de banque de filtres de convolution, une couche de traitement non linéaire et une couche de pool de fonctionnalités.

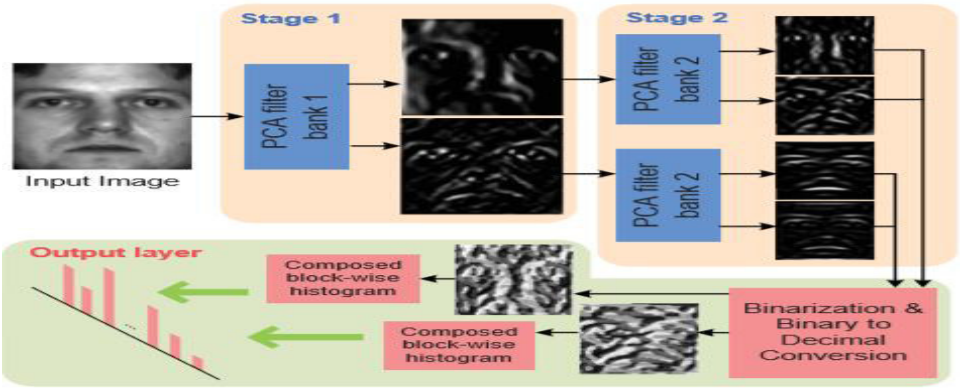


Figure 2. 1 Illustration de l'architecture de PCANet pour la RV [18]

La PCANet extrait les caractéristiques d'une image par le biais de trois composants de traitement les plus simples : filtres PCA, hashing binaire et les histogrammes.

1.2.2 Reconnaissance de visage faciale sur plusieurs bases de données

Une partie du jeu de données MultiPIE est utilisée pour apprendre les filtres PCA dans PCANet, puis ce PCANet formé à l'extraction des caractéristiques est appliqué à de nouveaux sujets dans les jeux de données MultiPIE, Extended Yale B, AR et FERET pour la reconnaissance de visage. Chaque base de données possède des caractéristiques spécifiques. PCANet est également évaluée sur la base de données LFW (under unsupervised setting) pour la vérification de visage sans contrainte. LFW contient 13 233 images de visages de 5 749 personnes différentes, recueillies à partir du Web, avec de grandes variations de pose, d'expression, d'éclairage, de vêtements, de coiffures, etc.

Tableau 2. 2 Comparison de performances de vérification (%) de PCANet sur LFW [18]

Methods	Accuracy
POEM	82.70±0.59
High-dim. LBP [3]	84.08
High-dim. LE [3]	84.58
SFRD [4]	84.81
I-LQP	86.20±0.46
OCLBP [5]	**86.66±0.30**
PCANet-1	81.18 ± 1.99
PCANet-1 (sqrt)	82.55 ± 1.48
PCANet-2	85.20 ± 1.46
PCANet-2 (sqrt)	86.28 ± 1.14

Tableau 2. 1 Comparaison des performances des méthodes de réduction conventionnelles [50]

Method	$\mu \pm S_E$
Eigenfaces, original [6]	0.6002±0.0079
Nowak, original	0.7245±0.0040
Nowak, funneled	0.7393±0.0049
Hybrid descriptor-based, funneled	0.7847±0.0051
3x3 Multi-Region Histograms	0.7295±0.0055
Pixels/MKL, funneled	0.6822±0.0041
V1-like/MKL, funneled	0.7935±0.0055
APEM (fusion), funneled	0.8408±0.0120
MRF-MLBP	0.7908±0.0014
Fisher vector faces [7]	0.8747±0.0149
Eigen-PEP	0.8897±0.0132
Weighted SILD after 8 feature combination	0.8768±0.0159
MRF-MBSIF-CSKDA [8]	0.9363±0.0127
MRF-Fusion-CSKDA	**0.9589±0.0194**
MSBSIF-SIEDA	**0.9463±0.0095**

Étonnamment, pour toutes les tâches, un tel modèle apparemment naïf de PCANet est à la hauteur des fonctionnalités de pointe, soit prédéfinies, hautement fabriquées manuellement ou soigneusement apprises (par les DNN). Encore plus surprenant, il établit de nouveaux enregistrements pour de nombreuses tâches de classification dans les jeux de données Extended Yale B, AR, FERET et MNIST. Des expériences supplémentaires sur d'autres ensembles de données publics démontrent également le potentiel du PCANet en tant que base simple mais très compétitive pour la classification des textures et la reconnaissance d'objets.

2.2 Analyse Discriminante Linéaire en profondeur (DeepLDA)

Ici [55], l'Analyse Discriminante Linéaire en profondeur (DeepLDA) qui permet d'apprendre des représentations latentes séparables linéairement de manière complète est introduite. La LDA classique extrait des fonctionnalités qui préservent la séparabilité des classes et sont utilisées pour la

réduction de la dimensionnalité dans de nombreux problèmes de classification. L'idée centrale de ce travail est de placer le LDA au sommet d'un réseau de neurones profonds. Cela peut être vu comme une extension non linéaire du LDA classique. Au lieu de maximiser la probabilité des étiquettes cibles pour des échantillons individuels, on propose une fonction objective qui pousse le réseau à produire des distributions de caractéristiques qui : (a) ont une faible variance dans la même classe et (b) une forte variance entre différentes classes. L'objectif est dérivé du problème général des valeurs propres de la LDA et permet toujours de s'entraîner avec une descente de gradient stochastique et une propagation en retour.

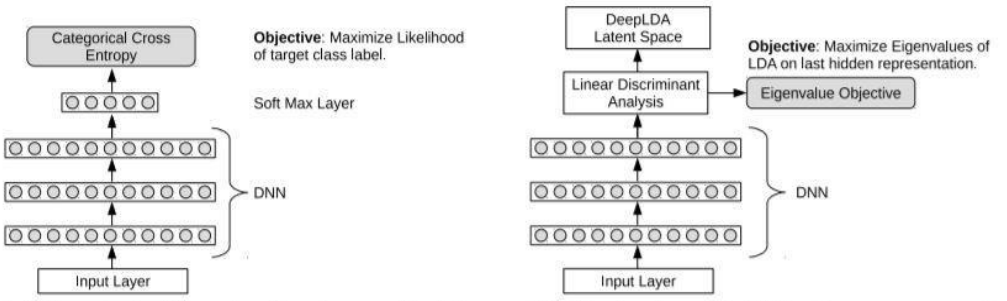
Rappelons que L'analyse discriminante linéaire (ADL) est une méthode de statistique multivariée qui cherche à trouver une projection linéaire d'observations de grande dimension dans un espace de plus petite dimension [57]. Lorsque ses conditions sont remplies, LDA permet de définir des limites de décision linéaires optimales dans l'espace latent résultant. Le but de cet article est d'exploiter les propriétés bénéfiques de la LDA classique (faible variabilité intra-classe, variabilité élevée entre classes, limites de décision optimales) en reformulant son objectif d'apprendre des représentations séparables linéairement basées sur un réseau neuronal profond (DNN). Récemment, les méthodes liées à la LDA ont obtenu un grand succès en combinaison avec des réseaux de neurones profonds. Andrew et al. a publié une version complète de Deep Canonical Correlation Analysis (DCCA) [58]. Dans leurs évaluations, DCCA est utilisé pour produire des représentations corrélées de données d'entrée multimodales de données de parole acoustiques et articulatoires enregistrées simultanément. Clevert et al. proposent des réseaux de facteurs rectifiés (RFN) qui sont une interprétation de réseau neuronal de l'analyse factorielle classique (Clevert et al., 2015)[56]. Les RFN sont utilisés pour la pré-formation non supervisée et aident à améliorer les performances de classifi cation sur quatre jeux de données de référence différents. Une méthode similaire appelée PCANet - ainsi qu'une variante basée sur la LDA - a été proposée par Chan et al. [18]. PCANet peut être considéré comme une simple approche d'apprentissage en profondeur par convolution non supervisée. Le procédé procède à une analyse en cascade de la composante principale (PCA) en cascade, à un hachage binaire et à des calculs d'histogramme en blocs. Cependant, l'un des obstacles majeurs à leur approche est leur limitation aux architectures très peu profondes (deux étapes) [18].

Pour l'évaluation, l'approche DeepLDA est validée sur trois jeux de données de référence différents (MNIST, CIFAR-10 et STL-10). DeepLDA produit des résultats compétitifs sur MNIST et CIFAR-10 et surpasse un réseau formé à l'entropie croisée catégorique (ayant la même architecture) sur un cadre supervisé de STL-10.

2.2.1 Idée Principale

L'apprentissage en profondeur est devenu le réseau le plus ultra en matière d'apprentissage automatique de fonctionnalités et remplace les approches existantes basées sur des fonctionnalités

conçues manuellement dans de nombreux domaines, tels que la reconnaissance d'objets [47]. DeepLDA est motivé par le fait que, lorsque les conditions préalables de LDA sont remplies, il est capable de trouver des combinaisons linéaires des caractéristiques en entrée qui permettent des limites de décision linéaires optimales. En général, LDA prend les fonctionnalités en entrée. L'intuition de cette méthode est d'utiliser LDA comme objectif au-dessus d'un puissant algorithme d'apprentissage de fonctionnalités. Au lieu de maximiser la probabilité des étiquettes cibles pour des échantillons individuels, une fonction objective est proposée, celle-ci est basée sur les valeurs propres LDA qui pousse le réseau à produire des distributions de caractéristiques discriminantes. Les paramètres sont optimisés en propageant en retour l'erreur d'un objectif basé sur LDA sur l'ensemble du réseau. Le problème de l'apprentissage des fonctionnalités est abordé en se concentrant sur les directions dans l'espace latent avec le plus petit pouvoir discriminant. Ceci remplace le schéma de pondération de (Stuhlsatz et al., 2012)[59] et permet d'opérer sur la formulation d'origine de la LDA. On s'attend à ce que DeepLDA produise des représentations masquées séparables linéairement avec un pouvoir discriminant similaire dans toutes les directions de l'espace latent. De telles représentations devraient également être associées à un potentiel de classification élevé des réseaux respectifs. Les résultats de la classification expérimentale présentés ci-dessous confirment cet effet positif sur la précision de la classification, et deux expériences supplémentaires) donnent une première confirmation qualitative selon laquelle les représentations acquises montrent les propriétés attendues.



(a) The output of the network gets normalized by a soft max layer to form valid probabilities. The CCE objective maximizes the likelihood of the target class under the model.

(b) On the topmost hidden layer we compute an LDA which produces corresponding eigenvalues. The optimization target is to maximize those eigenvalues.

Figure 2. 2 Schéma d'un DNN et de DeepLDA [55].

Pour les deux architectures, les données d'entrée sont d'abord propagées à travers les couches des deux réseaux. Cependant, la couche finale et la cible d'optimisation sont différentes.

2.2.2 Configuration de modèle DeepLDA

La figure 2.2.b montre un schéma de DeepLDA. Au lieu d'une optimisation par échantillon de la perte de CCE sur les probabilités de classe prédites, une couche LDA est placée au-dessus du DNN. Cela signifie notamment que l'on ne pénalise pas le classement erroné d'échantillons individuels. Au lieu de cela, on essaie de produire des fonctionnalités qui montrent une faible

variabilité intra-classe et élevée entre classes. Le problème de maximisation est abordé par une version modifiée du problème général des valeurs propres de la. Contrairement à CCE (Categorical-Cross-Entropy) l'optimisation DeepLDA agit sur les propriétés des paramètres de distribution de la représentation cachée produite par le réseau neuronal. L'optimisation des valeurs propres étant liée aux vecteurs propres correspondants (une matrice de projection linéaire), DeepLDA peut également être considéré comme un cas particulier d'une couche dense.

Les résultats expérimentaux montrent que les représentations apprises avec DeepLDA sont discriminantes et ont un effet positif sur la précision de la classification. Nos modèles DeepLDA obtiennent des résultats compétitifs sur les modèles MNIST et CIFAR-10 et surpassent CCE dans une configuration entièrement supervisée du STL-10 avec une précision de plus de 9%. Les résultats et les investigations ultérieures suggèrent que DeepLDA fonctionne mieux lorsqu'il est appliqué à des images de taille raisonnable (dans le cas présent, 96 × 96 pixels).

2.3 Réseau de Transformations en Cosinus Discrètes (DCTNet)

PCANet a été proposé comme un réseau d'apprentissage en profondeur léger qui utilise principalement l'analyse en composantes principales (PCA) pour apprendre des banques de filtres à plusieurs étages, suivies de la binarisation et de l'histogramme par blocs. On a montré que PCANet fonctionnait étonnamment bien dans diverses tâches de classification d'images. PCANet dépend toutefois des données et est donc inflexible. Dans cet article [60], les auteurs proposent un réseau indépendant des données, baptisé DCTNet pour la reconnaissance faciale, dans lequel ils adoptent la transformée en cosinus discrète (DCT) en tant que banques de filtres à la place de la PCA. Ceci est motivé par le fait que la base de la DCT 2D est une bonne approximation pour les vecteurs propres de rang supérieur de la PCA. La DCT 2D et la PCA ressemblent toutes deux à une sorte de motifs d'onde sinusoïdale modulés, qui peuvent être perçus comme une banque de filtres passe-bande. DCTNet est libre d'apprentissage car les bases 2D DCT peuvent être calculées à l'avance. En outre, une méthode efficace est également proposée pour réguler le vecteur de caractéristiques d'histogramme par bloc de DCTNet afin de le rendre plus robuste. Il a été démontré que son efficacité augmentait de manière surprenante lorsque l'image test était considérablement différente de celle de la base d'apprentissage. Les performances de DCTNet sont évaluées de manière approfondie sur un certain nombre de bases de données de visage de référence et elle est capable d'atteindre des performances équivalentes, voire supérieures, à celles de PCANet.

2.3.1 Architecture du réseau Transformations Cosinus Discretes (DCTNET)

DCTNet adopte une structure similaire à PCANet sauf qu'il y'a une couche supplémentaire à la sortie de l'histogramme pour la normalisation de l'histogramme comme représenté sur la **figure 2. 3.**

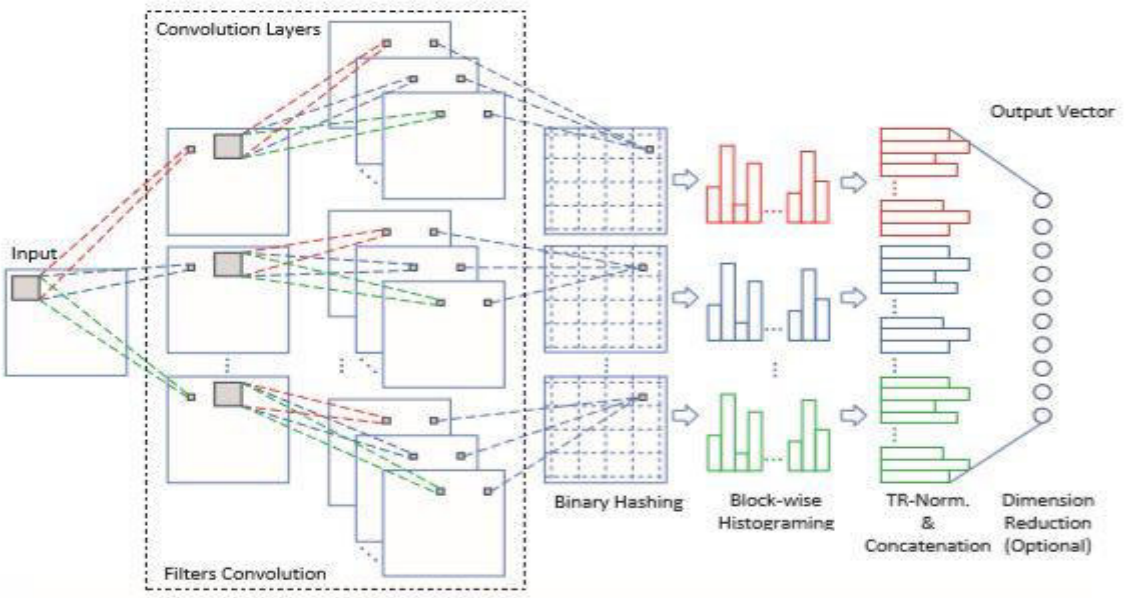


Figure 2. 3 Architecture de DCTNet [60]

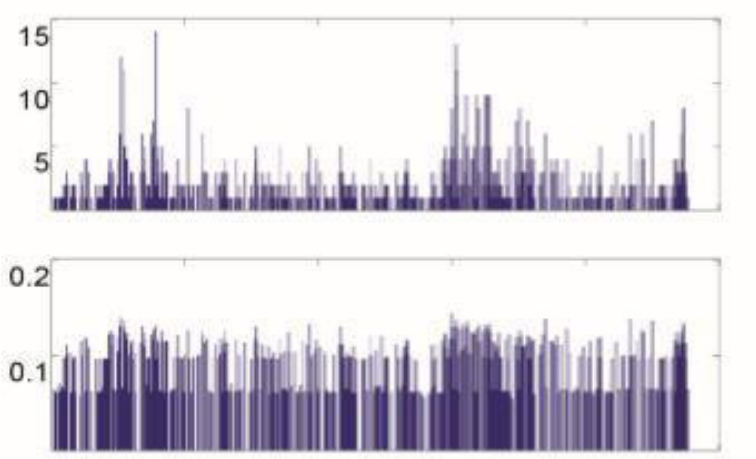


Figure 2. 4 Histogrammes du vecteur cactéristiques par bloc [60]

En haut de la **figure 2.4** une partie du vecteur de caractéristiques de l'histogramme d'origine par bloc ; en bas, le vecteur de caractéristiques d'histogramme par blocs normalisé TR résultant. Notez que la différence d'échelle entre l'entrée et la sortie est due au processus de normalisation.

2.3.2 Application et résultats du réseau Transformations Cosinus Discretes (DCTNET)

DCTNet est un apprentissage en profondeur qui donne une perspective différente des filtres appris par PCANet. La nature de la caractéristique de corrélation locale de l'image qui peut être modélisée avec un processus de Markov stationnaire du premier ordre en supposant que les pixels voisins sont fortement corrélés nous conduit à un réseau convolutionnel beaucoup plus simple et sans apprentissage. Les tableaux ci-dessous recensent l'essentiel du réseau DCTNet et sa comparaison avec d'autres méthodes.

Tableau 2. 3 Taux de reconnaissance (%) sur FERET [60]

TR Norm.	Method	Bc	Bd	Be	Bf	Bg	Bh	Avg
No	PCANet-A	51.5	91.0	99.0	99.5	93.0	51.5	80.92
	PCANet-B	62.0	92.5	**100**	**100**	95.5	55.5	84.25
	DCTNet	**70.5**	**97.0**	**99.5**	**100**	**96.0**	**73.0**	**89.33**
Yes	PCANet-A	82.0	97.0	**100**	**100**	98.5	76.0	92.25
	PCANet-B	**88.5**	**99.5**	**100**	**100**	**99.5**	**86.0**	**95.58**
	DCTNet	85.5	98.5	**100**	**100**	**99.5**	85.0	94.75

Tableau 2. 4 Taux de reconnaissance (%) sur FERET avec d'autres méthodes [60]

Method	Fb	Fc	Dup-I	Dup-II	Avg
LBP [12]	93.00	51.00	61.00	50.00	63.75
DMMA [20]	98.10	98.50	81.60	83.20	90.35
G-LBP [21]	98.00	98.00	90.00	85.00	92.75
WPCA-POEM [22]	99.60	99.50	88.80	85.00	93.23
G-LQP [23]	**99.90**	**100**	93.20	91.00	96.03
LGBP-LGXP [24]	99.00	99.00	94.00	93.00	96.25
sPOEM+POD [25]	99.70	**100**	94.90	94.00	97.15
GOM [26]	**99.90**	**100**	**95.70**	93.10	97.18
PCANet-2 [9]	99.58	**100**	95.43	**94.02**	97.26
PCANet-A	99.25	**100**	94.46	93.16	96.72
DCTNet	99.67	**100**	95.57	**94.02**	**97.32**

La relation entre la fréquence et la variance de la PCA et de la DCT 2D nous amène à classer l'importance de la base de la DCT 2D à partir de la fréquence la plus basse en tant que sélection de filtre et il est démontré que différents jeux de données de faces fonctionnent très bien. En revanche, DCTNet peut ne pas fonctionner correctement si la nature de l'image d'entrée ne suit pas l'hypothèse de corrélation locale élevée, telle qu'une image contenant une activité spectrale élevée et des détails fins, tels que des images de texture. De telles données d'image peuvent nécessiter différents schémas de sélection de base DCT.

2.4 Réseau en profondeur 2DPCA pour la reconnaissance de visage (2DPCANet)

Cet article propose un réseau bidimensionnel d'analyse en composantes principales (2DPCANet), qui est un nouveau réseau d'apprentissage en profondeur pour la reconnaissance des visages. Dans l'architecture, 2DPCA est utilisé pour apprendre les filtres de couches à plusieurs étages, puis le hachage binaire est exploité et les histogrammes par bloc pour générer les caractéristiques locales. Les machines à vecteurs de support (SVM) et extrême learning machine (ELM) sont adoptés en tant que classificateur [4].

2.4.1 Architecture de 2DPCANet

Dans cette section, nous présentons un réseau d'apprentissage en profondeur simple, 2DPCANet. Pour faire face au défaut de PCANet, la structure de PCANet est adoptée, mais on utilise le 2DPCA plutôt que la PCA en tant que banques de filtres. La différence entre 2DPCANet et PCANet est indiquée dans **figure 2.** La principale différence entre 2DPCANet et PCANet réside dans le fait que PCANet utilise PCA comme les banques de filtres, qui doivent transformer la matrice 2D en vecteur1D avant de calculer les vecteurs propres, tandis que 2DPCANet utilise le 2DPCA en tant que banques de filtres, qui calcule directement les vecteurs propres de la soi-disant matrice de covariance d'image sans conversion matrice-vecteur [4].

- ***La première couche 2DPCANet***
- ***La seconde couche de 2DPCANet*** : Hashing and histogram

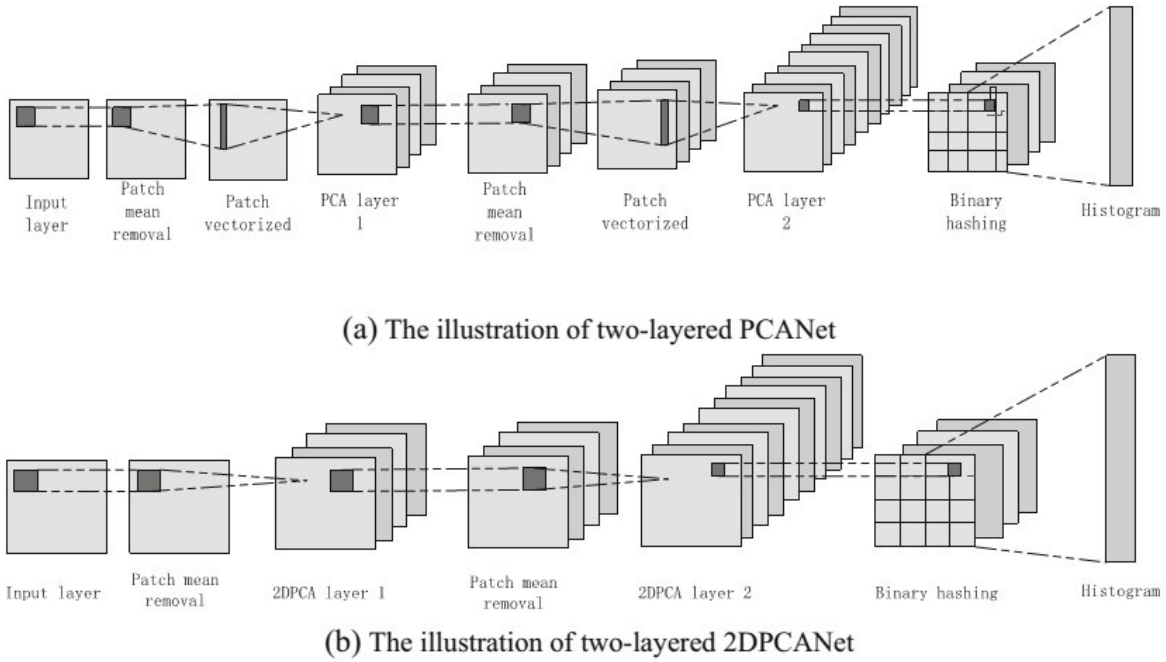


Figure 2. 5 la différence entre 2DPCANet et PCANet [4]

2.4.2 Application et résultats 2DPCANet

2DPCANet est également comparée avec LDANet [45], RandNet [45], DLANet [62] et CNN. Caffe est utilisé dans le cadre et l'architecture de CNN c'est similaire à AlexNet proposé par Krizhevsky et al [47]. Le CNN est formé sur les images de la galerie pour 2000 époques.

Tableau 2. 6 Recognition rate on YALE database [4]

	2	3	4	5	6	7
PCANet	0.9133 ±0.0217	0.9458 ±0.0168	0.9705 ±0.0116	0.9895 ±0.0089	0.9853 ±0.0111	0.9867 ±0.0100
2DPCANet	0.9593 ±0.0145	0.9817 ±0.0090	0.9895 ±0.0089	0.9933 ±0.0073	0.9947 ±0.0040	0.9983 ±0.0050
PCANet(ELM)	0.9289 ±0.0217	0.9675 ±0.0180	0.9714 ±0.0217	0.9844 ±0.0158	0.9867 ±0.0157	0.9917 ±0.0145
2DPCANet(ELM)	0.9770 ±0.0130	0.9842 ±0.0114	0.9905 ±0.0085	0.9962 ±0.0111	0.9987 ±0.0065	0.9983 ±0.0050

Tableau 2. 5 Recognition rate on XM2VTS database

	2	3	4	5	6	7
PCANet	0.8787 ±0.0049	0.9332 ±0.0052	0.9586 ±0.0043	0.9714 ±0.0052	0.9785 ±0.0043	0.9814 ±0.0061
2DPCANet	0.9246 ±0.0044	0.9555 ±0.0032	0.9700 ±0.0030	0.9763 ±0.0014	0.9847 ±0.0021	0.9892 ±0.0054
PCANet(ELM)	0.8855 ±0.0068	0.9368 ±0.0034	0.9610 ±0.0055	0.9718 ±0.0072	0.9797 ±0.0054	0.9834 ±0.0044
2DPCANet(ELM)	0.9491 ±0.0052	0.9712 ±0.0020	0.9725 ±0.0025	0.9856 ±0.0031	0.9919 ±0.0023	0.9932 ±0.0032

Tableau 2. 8 Performance on AR database [4]

	Sunglasses		Scarf		Sunglasses and scarf	
	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)
PCANet	302.13	0.21	296.16	0.21	350.84	0.22
2DPCANet	**193.86**	0.23	**181.34**	0.21	**231.62**	0.23
PCANet(ELM)	384.33	**0.12**	374.91	**0.13**	429.47	**0.14**
2DPCANet(ELM)	**262.52**	**0.13**	**270.18**	**0.13**	**341.82**	**0.14**

Tableau 2. 7 Recognition rate on LFW-a database [4]

	2	3	4	5	6	7
PCANet	0.2722 ±0.0112	0.3483 ±0.0141	0.4101 ±0.0099	0.4503 ±0.0177	0.4780 ±0.0176	0.5314 ±0.0271
2DPCANet	0.2926 ±0.0093	0.3638 ±0.0076	0.4331 ±0.0071	0.4863 ±0.0136	0.5127 ±0.0126	0.5586 ±0.0176
PCANet(ELM)	0.2763 ±0.0075	0.3507 ±0.0125	0.4071 ±0.0125	0.4602 ±0.0085	0.4965 ±0.0132	0.5312 ±0.0178
2DPCANet(ELM)	0.2979 ±0.0063	0.3738 ±0.0127	0.4363 ±0.0125	0.4848 ±0.0068	0.5218 ±0.0095	0.5675 ±0.0130

Tableau 2. 9 Recognition rate on FERET database [4]

	Fb	Fc	Dup-1	Dup-2	Avg.
LDANet	0.9502	0.9948	0.9312	0.9205	0.9492
RandNet	0.9113	0.9323	0.8374	0.8598	0.8852
DLANet	0.9540	1	0.9448	**0.9372**	0.9590
CNN-2	0.8285	0.8368	0.7874	0.8000	0.8132
PCANet	0.9481	0.9812	0.9122	0.9037	0.9363
2DPCANet	**0.9565**	1	**0.9452**	0.9368	**0.9596**

Tableau 2. 10 Recognition rate on Extended Yale B database [4]

	Subset1&2	Subset3	Subset4	Avg.
LDANet	0.9941	0.9780	0.9658	0.9793
RandNet	0.9400	0.8793	0.8421	0.8871
CNN-2	0.8603	0.3202	0.1134	0.4322
DLANet	**0.9956**	0.9868	0.9737	0.9854
PCANet	0.9927	0.9868	0.9658	0.9817
2DPCANet	**0.9956**	**0.9890**	**0.9816**	**0.9887**

Et 2DPCANet est beaucoup plus rapide que PCANet en formation. De plus, 2DPCANet est insensible aux variations d'éclairage et robuste à l'occlusion. Par conséquent, 2DPCANet est un moyen efficace et réseau robuste pour la reconnaissance faciale. Cependant, la comparaison des performances de SVM et ELM suggère que, bien que ELM fournit un meilleur taux de reconnaissance précis, cet avantage se détériorera lorsque l'ampleur du problème devient grand. De plus, le temps d'entraînement d'ELM est apparemment supérieur à celui de SVM en particulier pour les grandes bases de données [4].

Il est à noter que 2DPCANet a été proposé pour résoudre le problème de classification des images.

Les résultats expérimentaux obtenus sur la base de données faciale YALE, XM2VTS, AR, LFW-a, FERET et Extended Yale B montrent que la performance de reconnaissance de 2DPCANet est supérieure à celles des méthodes rapportées [4].

Conclusion

Dans ce chapitre, nous avons présenté l'état de l'art de méthodes de réduction dimensionnel basé sur le Deep Learning pour améliorer l'extraction des caractéristiques des images.

On a conclu qu'il existe plusieurs méthodes de réduction à la base de DL et nous allons détailler quelques méthodes dans le chapitre 3 pour bien utilisé dans notre travail.

Introduction

Les techniques de réduction de dimensionnalité dans les tâches d'apprentissage supervisées ou non supervisées ont attiré l'attention de la vision par ordinateur et de la reconnaissance des formes. Parmi eux, les algorithmes linéaires d'analyse en composantes principales (PCA) et l'analyse discriminante linéaire (LDA) ont été les deux populaire en raison de leur simplicité et de leur efficacité et Side-Information based Exponential Discriminant Analysis (SLDA). Lorsque nous parlons de vision par ordinateur, un terme réseau de neurones convolutionnel (CNN) vient à l'esprit. L'utilisation de CNN simplifie la tâche et automatise la reconnaissance faciale.

Après avoir étudié dans le deuxième chapitre l'état de l'art les différentes méthodes utilisées pour la réduction de dimension associé au **Deep Learning** pour la reconnaissance de visage. Nous allons présenter dans ce chapitre les méthodes étudiées à savoir :

1. **PCA (Principales Components Analysis)**
2. **PCANet (Principales Components Analysis Network)**
3. **LDANet**
4. **SVDNet**

Nous présentons le principe du CNN au début du chapitre pour aider à mieux comprendre les méthodes que nous projetons de concevoir.

L'approche proposée utilise le classifieur SVM, nous consacrons lui dernière partie de ce chapitre.

3.1 Les réseaux de neurones convolutionnels (CNN)

Les réseaux de neurones convolutionnels (CNN pour Convolutional Neural Networks) proposés initialement par Le Cun [63]. Ce choix a été motivé principalement par ce qu'il intègre implicitement une phase d'extraction de caractéristiques et il a été utilisé avec succès dans de nombreuses applications [64]. Ils sont réputés pour leur robustesse aux faibles variations d'entrée et le faible taux de prétraitement nécessaire à leur fonctionnement.

Le CNN est un réseau de neurone multicouche qui est spécialisé dans des tâches de reconnaissance de forme. [65] Ces réseaux ont été inspirés par les travaux de Hubel et Wiesel sur le cortex visuel chez les mammifères [66] qui combine **trois idées principales** : i) ***les champs récepteurs locaux***, ii) ***les poids partagés*** et iii) ***le sous-échantillonnage***.

L'architecture de CNN repose sur plusieurs réseaux de neurones profonds consistant en une succession de couches de convolution et d'agrégation (pooling) est dédié à l'extraction

automatique de caractéristiques, tandis que la seconde partie, composée de couches de neurones complètement connectées, est dédiée à la classification [64].

Chaque cellule des couches de convolution est connectée à un ensemble de cellules regroupées dans un voisinage rectangulaire sur la couche précédente. Les champs récepteurs locaux permettent d'extraire des caractéristiques basiques. Les couches sont dites « à convolution » car les poids sont partagés et chaque cellule de la couche réalise la même combinaison linéaire (avant d'appliquer la fonction sigmoïde) qui peut être vue comme une simple convolution. Ces caractéristiques sont alors combinées à la couche suivante afin de détecter des caractéristiques de plus haut niveau.

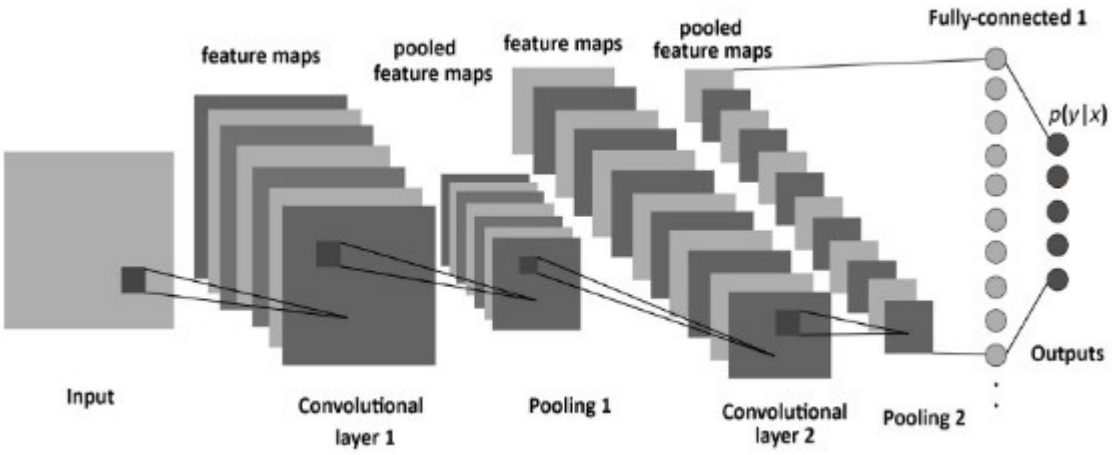


Figure 3. 1 L'architecture d'un réseau de neurone convolutionnel [67]

Entre deux phases d'extraction de caractéristiques, le réseau réduit la résolution de la carte des caractéristiques par un moyen de sous-échantillonnage. Cette réduction se justifie à deux titres : diminuer la taille de la couche et apporter de la robustesse par rapport aux faibles distorsions.

- ***Couche de convolution*** : La convolution est une opération mathématique comme l'addition et la multiplication, il est très utile de simplifier des équations plus complexe, cette opération est largement utilisée dans le traitement du signal numérique. Lorsque l'on applique la convolution au traitement d'image, on réalise la convolution (combiner) l'image d'entrée avec une sous-région de cette image (filtre). Le filtre est aussi connu sous le nom du noyau de convolution, il consiste en des poids de cette sous-région. La sortie de cette couche est l'image entrée avec des modifications qui est souvent appelée une carte de caractéristique (Feature Map).

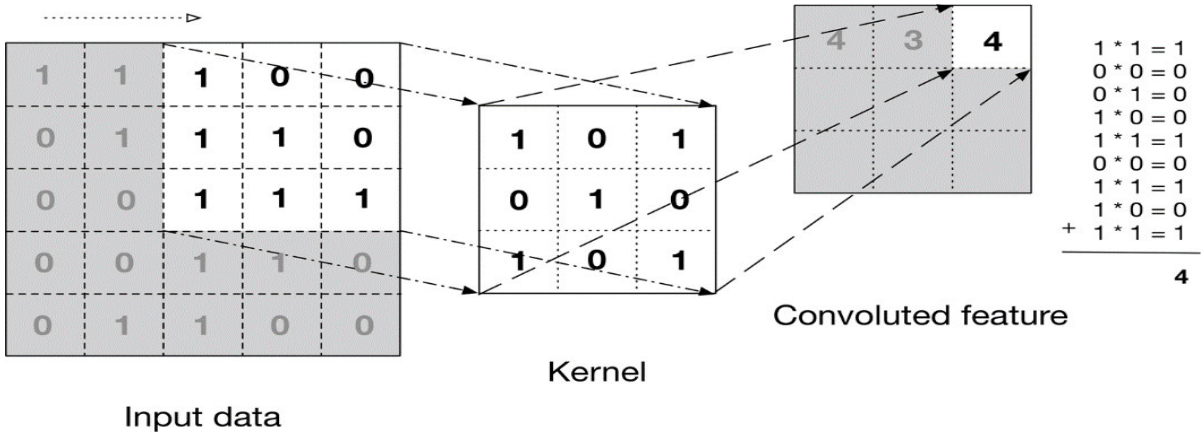


Figure 3. 2 L'opération de convolution [68]

En terme mathématique, une couche de convolution C_i (couche i du réseau) est paramétrée par son nombre **N** de cartes de convolution $\boldsymbol{M_j^i(j \in \{1 \ldots . N\})}$, la taille des noyaux de convolution K_x K_y(souvent carrée), et le schéma de connexion à la couche précédente $\boldsymbol{L^i}$. Chaque carte de convolution est le résultat d'une somme de convolution des cartes de la couche précédente par son noyau de convolution respectif. Un biais est ensuite ajouté et le résultat passe à une fonction de transfert non-linéaire. Dans le cas d'une carte complètement connectée aux cartes de la couche précédente, le résultat est alors calculé par :

$$\boldsymbol{M_j^i = \varphi(b_j^i + \sum_{n=1}^{n} M_j^{i-1} \quad K_j^i)}$$

- ***Couche de sous-échantillonnage (Pooling) :*** Dans les architectures classiques de réseaux de neurones convolutionnels, les couches de convolution sont suivies par des couches de sous échantillonnage (couche d'agrégation). Cette dernière réduit la taille des cartes de caractéristique pour but de diminuer la taille de paramètre, et renvoie les valeurs maximales des régions rectangulaires de son entrée [65].

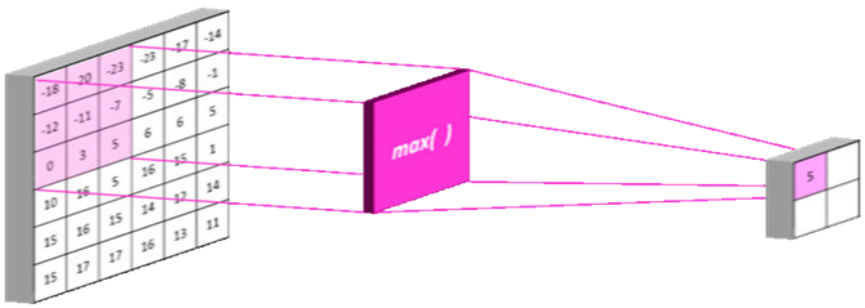


Figure 3. 3 L'opération de sous-échantillonnage (Pooling Layer) [30]

- ***Couche entièrement connectée :*** Les paramètres des couches de convolution et de max agrégation sont choisis de sorte que les cartes d'activation de la dernière couche soient de taille 1, ce qui résulte en un vecteur 1D d'attributs. Des couches classiques complètement connectées composées de neurones sont alors ajoutées au réseau pour réaliser la classification [65]. La dernière couche, dans le cas d'un apprentissage supervisé, contient autant de neurones que de classes désirées. Cette dernière couche contient N neurones (nombre des classes dans la base), et une fonction d'activation de type sigmoïde est utilisée afin d'obtenir des probabilités d'appartenance à chaque classe.

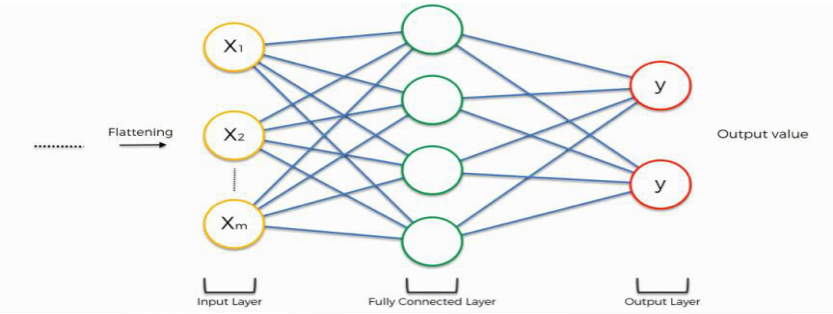


Figure 3. 4 Couche entièrement connectée (Fully connected)

3.2 Analyse en Composantes Principales (ACP)

La méthode d'analyse en composantes principales (ACP ou PCA) est une technique linéaire de réduction de dimension, qui signifie qu'il réduit la dimensionnalité en incorporant les données dans un sous-espace linéaire de dimensionnalité inférieure.

C'est une procédure mathématique qui transforme un nombre de variables corrélées en un nombre (plus petit) de variables non corrélées appelées composantes principales. L'objectif est de réduire l'espace d'attribut d'un plus grand nombre de variables à un plus petit nombre de facteurs.

3.3 Réseau d'Analyse en Composantes Principales (PCANet)

Le réseau PCANet représente une phase principale dans notre projet. L'architecture de cette PCANet est similaire à celle du réseau convolutionnel présenté dans la section 3.2 de ce chapitre. Elle combine les forces de l'analyse en composantes principales PCA et l'apprentissage en profondeur [45]. En comparaison avec le CNN qui tente de trouver les filtres optimaux pour le mappage des fonctionnalités, PCANet est sous-optimal dans la mesure où il apprend les banques de filtres en appliquant PCA sur les données d'entrée. D'autre part, ses avantages

résident dans le fait qu'il n'exige pas de grandes quantités de données ni un temps d'apprentissage long, tout en utilisant le concept de base de l'architecture de réseau à convolution profonde.

3.3.1 Définition du réseau PCA (PCANet)

Le PCANet s'initialise en appliquant l'analyse en composantes principales aux patches qui se chevauchent de toutes les images. Les composants principaux sélectionnés forment les filtres de la première couche et les projections des patchs sur les composants principaux forment la réponse des unités de la première couche. Cette méthodologie se répète pour former une linéaire map en cascade dans les couches suivantes de l'architecture de réseau à convolution profonde. Ensuite, le procédé utilise une quantification binaire et un hachage pour les ensembles d'images filtrées à plusieurs étapes afin de les concaténer sous forme décimale. Enfin, des histogrammes locaux sont extraits des blocs des images quantifiées et une méthode de regroupement de pyramides spatiales est appliquée à ces histogrammes afin d'extraire des caractéristiques. **Dans notre travail, nous nous intéressons à la PCANet à deux étages (PCANet2).** Dans ce qui suit, nous décrivons en détail celle méthode.

3.3.2 Structure du réseau PCA à deux étages (PCANet2)

Les données d'apprentissage contiennent $\boldsymbol{i} = 1, 2, ..., \mathbf{N}$ images $\boldsymbol{I_i}$ de taille $\mathbf{m} \times \mathbf{n}$. Supposons que nous recevons doc ces N images en entrée $\{\boldsymbol{I_i}\}_{i=1}^{N}$ de taille $\mathbf{m} \times \mathbf{n}$, et supposons que la taille du patch (ou la taille du filtre **2D**) est $\boldsymbol{k_1} \times \boldsymbol{k_2}$ à tous les stades. Le modèle PCANet2 proposé est illustré à la **figure 3.5**

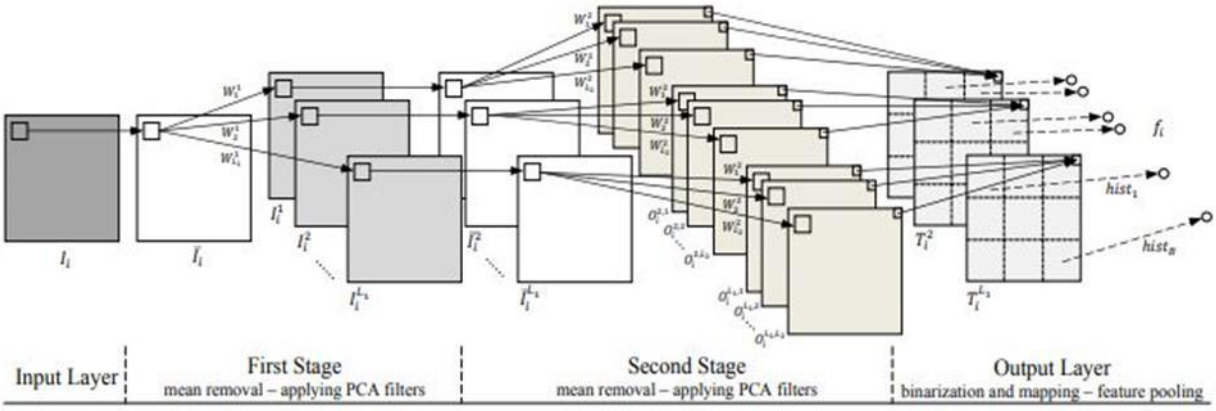


Figure 3. 5 Schéma fonctionnel d'un PCANet à deux étages (PCANet2) [69]

L'algorithme est expliqué en détail comme suit :

Dans la première étape, des patches de taille $\boldsymbol{k_1 \times k_2}$ pixels sont extraits autour de chaque pixel de l'image $\boldsymbol{I_i}$ et seuls les filtres **PCA** doivent être appris à partir des images d'entrée $\{\boldsymbol{I_i}\}_{i=1}^{N}$.

Dans la deuxième étape, nous répétons presque le même processus que le premier étage de PCANet2.

- ***Première étape :*** ***Couche premier étage PCANet2***

Autour de chaque pixel, on prend un patch $\boldsymbol{k_1 \times k_2}$, ces patches se chevauchant sont collectés, vectorisés et soustraits de la moyenne pour obtenir $\bar{\boldsymbol{X}}_{\boldsymbol{i}}$. En construisant la même matrice pour toutes les images d'entrée et en les assemblant, nous obtenons $\boldsymbol{X}$ [45].

$$\boldsymbol{X} = [\bar{\boldsymbol{X}}_1, \bar{\boldsymbol{X}}_2, \ldots, \bar{\boldsymbol{X}}_N] \in \mathbb{R}^{k_1 k_2 \times Nmn}$$

En supposant que le nombre de filtres dans la couche $\boldsymbol{i}$ soit $\boldsymbol{L_i}$, **PCA** minimise l'erreur de reconstruction dans une famille de filtres orthonormés [45].

$$\min_{\boldsymbol{V} \in \mathbb{R}^{k_1 k_2 \times L_1}} \|\boldsymbol{X} - \boldsymbol{V}\boldsymbol{V}^T\boldsymbol{X}\|_F^2, \text{ s.t. } \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_{L_1}$$

où $\boldsymbol{I_{L_1}}$ est la matrice d'identité de taille $\boldsymbol{L_1 \times L_1}$. La solution est connue sous le nom de vecteurs propres **(Eigenvectors)** principaux $\boldsymbol{L_1}$ de $\boldsymbol{XX^T}$. Les filtres PCA sont donc exprimés selon l'équation suivante :

$$\boldsymbol{W}_l^1 \doteq \text{mat}_{k_1,k_2}(\boldsymbol{q}_l(\boldsymbol{X}\boldsymbol{X}^T)) \in \mathbb{R}^{k_1 \times k_2}, \ l = 1, 2, \ldots, L_1$$

où $\boldsymbol{mat}_{\boldsymbol{k_1,k_2}}(\boldsymbol{v})$ est une fonction qui mappe $\boldsymbol{v} \in \mathbb{R}^{\boldsymbol{k_1}\,\boldsymbol{k_2}}$ à une matrice $\mathbf{W} \in \mathbb{R}^{\boldsymbol{k_1}\,\boldsymbol{k_2}}$, et $\boldsymbol{q_l}$ $(\boldsymbol{XX^T})$ désigne le $(\boldsymbol{l})$ième vecteur propre de $\boldsymbol{XX^T}$. Les principaux vecteurs propres capturent la principale variation de tous les correctifs d'apprentissage supprimés. Nous pouvons empiler plusieurs étapes de filtres **PCA** afin d'extraire des caractéristiques de niveau supérieur [45].

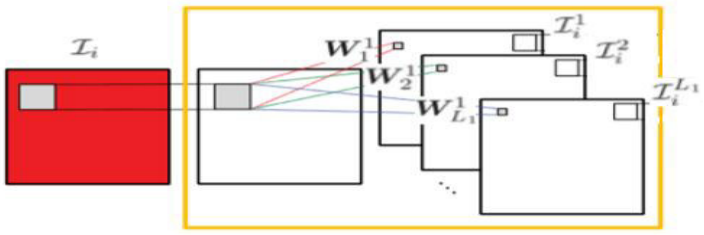


Figure 3. 6 Schéma du premier étage de l'architecture PCANet2 [45]

Dans cette première couche, W_l^1 représente le i^{ème} filtre de la première couche. La première couche contient $L1$ filtres.

- ***Deuxième étape :*** ***Couche deuxième étage PCANet2***

Répétant presque le même processus que la première étape. Laissons la sortie du $l^{ème}$ filtre du premier étage être :

$$\mathcal{I}_i^l \doteq \mathcal{I}_i * \boldsymbol{W}_l^1, \quad i = 1, 2, \ldots, N,$$

où $*$ désigne une convolution 2D et la limite de $\mathbf{I_1}$ est complétée de zéro avant de convoluer avec $\boldsymbol{W_l^1}$ de manière que $\mathbf{I_i^l}$ ait la même taille que $\mathbf{I_i}$. Comme pour la première étape, nous pouvons collecter tous les patchs superposés de $\mathbf{I_i^l}$, soustraire la moyenne des patchs à chaque patch, et ainsi former $\overline{\boldsymbol{Y}}_{\boldsymbol{i}}^{\boldsymbol{l}} = [\overline{\boldsymbol{y}}_{\boldsymbol{i,l,1}}, \overline{\boldsymbol{y}}_{\boldsymbol{i,l,2}}, \ldots, \overline{\boldsymbol{y}}_{\boldsymbol{i,l,mn}}] \in \mathbb{R}^{\boldsymbol{k_1\, k_2\, x\, mn}}$ où $\overline{\boldsymbol{y}}_{\boldsymbol{i,l,1}}$ est le j^{ème} patch à moyenne suppression de $\mathbf{I_i^l}$. Nous définissons en outre $\boldsymbol{Y^l} = [\overline{\boldsymbol{Y}}_{\boldsymbol{1}}^{\boldsymbol{l}}, \overline{\boldsymbol{Y}}_{\boldsymbol{2}}^{\boldsymbol{l}}, \ldots, \overline{\boldsymbol{Y}}_{\boldsymbol{N}}^{\boldsymbol{l}}] \in \mathbb{R}^{\boldsymbol{k_1\, k_2\, x\, Nmn}}$ pour la matrice de collecte tous les correctifs de moyennes-retiré de la sortie du filtre à $l^{ième}$, et concaténer $\boldsymbol{Y^l}$ pour toutes les sorties de filtre [45].

Les filtres PCA de la deuxième étape sont ensuite obtenus comme

$$\boldsymbol{Y} = [\boldsymbol{Y}^1, \boldsymbol{Y}^2, \ldots, \boldsymbol{Y}^{L_1}] \in \mathbb{R}^{k_1 k_2 \times L_1 N m n}$$

$$\boldsymbol{W}_\ell^2 \doteq \mathrm{mat}_{k_1,k_2}(\boldsymbol{q}_\ell(\boldsymbol{Y}\boldsymbol{Y}^T)) \in \mathbb{R}^{k_1 \times k_2}, \quad \ell = 1, 2, \ldots, L_2$$

Pour chaque entrée $\mathbf{I_i^l}$ du deuxième étage, nous aurons des sorties $\boldsymbol{L_2}$, chacune convoluer $\mathbf{I_i^l}$

$$\mathcal{O}_i^l \doteq \{\mathcal{I}_i^l * \boldsymbol{W}_\ell^2\}_{\ell=1}^{L_2}$$

avec $\boldsymbol{W_l^2}$ pour $\boldsymbol{l = 1, 2, \ldots, L_2}$. Le nombre de sorties du deuxième étage est $\boldsymbol{L_1 L_2}$. On peut simplement répéter le processus ci-dessus pour construire plus d'étapes **PCA** si une architecture plus profonde s'avère bénéfique [45].

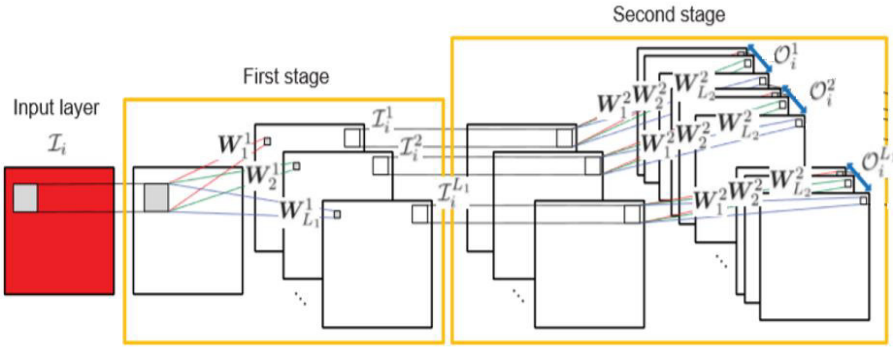


Figure 3. 7 Schéma des deux étages de l'architecture PCANet2[45]

- ***<u>Etape de sortie :</u> Couche hachage et histogramme***

Pour chacune des images d'entrée $\boldsymbol{L_1}$ de $\mathbf{I_i^l}$ pour le deuxième étage, il possède $\boldsymbol{L_2}$ sorties à valeurs réelles $\{\boldsymbol{I_i^l} * \boldsymbol{W_l^2}\}_{\boldsymbol{l=1}}^{\boldsymbol{L_2}}$ du deuxième étage. Nous binarisons ces sorties et

$$\mathcal{T}_i^l \doteq \sum_{\ell=1}^{L_2} 2^{\ell-1} H(\mathcal{I}_i^l * \boldsymbol{W}_\ell^2)$$

obtenons $\{H(I_i^l * W_l^2)\}_{l=1}^{L_2}$, où **H** (·) est une fonction de type Heaviside dont la valeur est un pour les entrées positives et zéro pour les autres.

Dont chaque pixel est un entier dans la plage $[0, 2^{L_2} - 1]$. L'ordre et les poids pour les sorties

$$f_i \doteq [\mathrm{Bhist}(\mathcal{T}_i^1), \ldots, \mathrm{Bhist}(\mathcal{T}_i^{L_1})]^T \in \mathbb{R}^{(2^{L_2})L_1 B}$$

L_2 sont sans importance, car nous traitons ici chaque entier comme un «mot» distinct. Pour chacune des images L_1 T_i^l, l = 1, ...,L_1, nous le partitionnons en blocs **B**. Nous calculons l'histogramme (avec 2^{L_2} cases) des valeurs décimales de chaque bloc et concaténons tous les histogrammes **B** en un seul vecteur et nous désignons $Bhist(T_i^l)$. Après ce processus de codage, le "caractéristique" de l'image d'entrée I_i est alors défini comme étant l'ensemble des histogrammes par bloc [45].

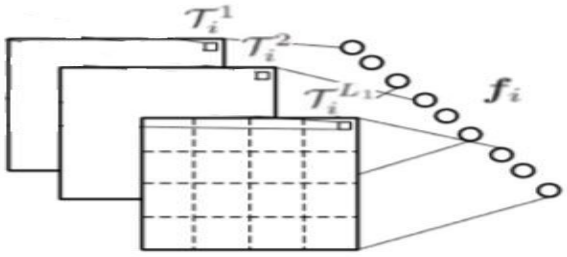


Figure 3. 8 Hachage et concaténation des histogrammes de l'architecture PCANet2[45]

3.4 Décomposition en Valeurs Singulières SVD

La théorie de la décomposition en valeurs singulières a été établie pour les matrices réelles carrées dans les armées 1870 par Beltrami et Jordan et pour les matrices complexes par Autonne en 1902. Récemment, la décomposition en valeurs singulières a été utilisée dans différentes applications du traitement d'image telle que la compression, la dissimulation de l'information et la réduction du bruit [70].

En mathématiques, le procédé d'algèbre linéaire de Décomposition en Valeurs Singulières (ou **SVD**, de l'anglais *singular value decomposition*) d'une matrice est un outil important de factorisation des matrices rectangulaires réelles ou complexes. Ses applications s'étendent du traitement du signal aux statistiques, en passant par la météorologie.

Le théorème spectral énonce qu'une matrice normale peut être diagonalisée par une base orthonormée de vecteurs propres. On peut voir la décomposition en valeurs singulières comme une généralisation du théorème spectral à des matrices arbitraires, qui ne sont pas nécessairement carrées [70].

3.4.1 Principe mathématique

Soit A une matrice quelconque de taille $m \times n$ dont les coefficients appartiennent au corps K, où $K = \mathbb{R}$ ou $K = \mathbb{C}$, et de rang r (le rang de la matrice A est le nombre de valeurs singulières non nulles). Alors il existe une matrice orthogonale U d'ordre $m \times n$, Une matrice orthogonale V d'ordre $n \times n$ et une matrice "pseudo-diagonale" (tous les éléments hors de la diagonale principale sont nuls, mais la matrice n'est pas carrée) S de dimension $m \times n$ (et donc de même dimension que A), il existe donc une factorisation de la forme [70]:

$$A = U * S * V^T$$

$$\begin{cases} U * U^T = I(m) \\ V * V^T = I(n) \\ S(m,n) = \begin{pmatrix} S1 & 0 & 0 \\ 0 & S2 & 0 \\ 0 & 0 & Sn \end{pmatrix} \end{cases}$$

Ici, * c'est l'opérateur produit.

I est la matrice identité et s_1, s_2 ….s_n sont les valeurs singulières de A. Ce se sont des nombres réels et non négatifs et qui respectent la condition : $s_1 > s_2 > s_3 > \ldots > s_n$

L'intérêt de la SVD pour le traitement d'images

Le principal intérêt de cette méthode vient du fait que :

1. Les valeurs singulières représentent l'énergie de l'image, c'est-à-dire que la SVD range le maximum d'énergie de l'image dans un minimum de valeurs singulières.

$$S(m,n) = \begin{pmatrix} S1 & 0 & 0 \\ 0 & S2 & 0 \\ 0 & 0 & Sn \end{pmatrix}$$

2. Les valeurs singulières d'une image ont une très bonne stabilité, c'est-a-dire que quand une petite perturbation (par exemple une marque) est ajoutée à une image, les valeurs singulières ne changent pas significativement.
3. En plus, la factorisation en SVD est unique.

3.4.2 Application de la SVD à l'image

Nous supposons M la matrice image $m \times n$. La décomposition en valeurs singulières dans le cas d'une matrice réelle à 2 dimensions M est de la forme : $M = U\Sigma V^*$

avec U une matrice unitaire $m \times m$ sur K, Σ une matrice $m \times n$ dont les coefficients diagonaux sont des réels positifs ou nuls et tous les autres sont nuls, et V^* est la matrice

adjointe à *V*, matrice unitaire $n \times n$ sur *K*. On appelle cette factorisation la *décomposition en valeurs singulières* de *M* [70].

Cette transformation déforme, par exemple, un cercle unitaire bleu ci-dessous à gauche en une ellipse dans le coin supérieur droit de l'image. La transformation M peut alors être décomposée en une rotation V* suivie d'une compression ou étirement Σ le long des axes de coordonnées suivie en fin d'une nouvelle rotation U. Les valeurs singulières $\sigma 1$ et $\sigma 2$ correspondent aux longueurs des grand et petit axes de l'ellipse [70].

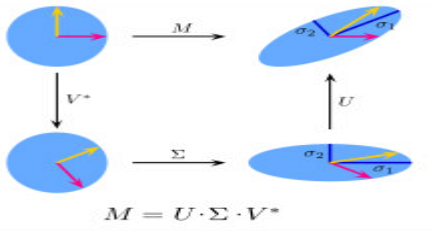


Figure 3. 9 Illustration de la projection dans l'espace réduit par SVD [70]

- La matrice V contient un ensemble de vecteurs de base orthonormés de K^n, dits « d'entrée » ou « d'analyse » ;
- La matrice U contient un ensemble de vecteurs de base orthonormés de K^m, dits « de sortie » ;
- La matrice Σ contient dans ses coefficients diagonaux les valeurs singulières de la matrice M.

Une convention courante est de ranger les valeurs $\Sigma_{i,i}$ par ordre décroissant. Alors, la matrice Σ est déterminée de façon unique par M (mais U et V ne le sont pas).

3.4.3 La décomposition en valeurs singulières SVD

La matrice image est un tableau de nombres dont il est parfois difficile d'extraire les caractéristiques intéressantes pour résoudre un problème donné. Une stratégie efficace pour mettre en évidence les propriétés d'une matrice est de la décomposer (ou factoriser) en un produit de matrices plus simples et dont les caractéristiques sont clairement identifiables et interprétables. La factorisation la plus générale, et peut-être la plus utile, est la Décomposition en Valeurs Singulières [70].

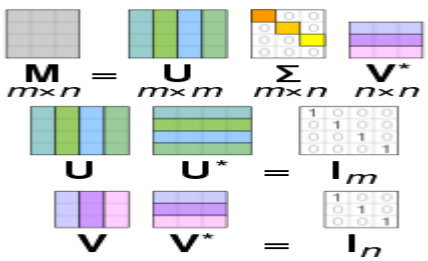


Figure 3. 10 Principe de décomposition en valeurs singulières [70]

3.4.5 Valeurs singulières et vecteurs singuliers

On appelle **valeur singulière** de M toute racine carrée d'une valeur propre de M^*M, autrement dit tout réel positif σ tel qu'il existe un vecteur unitaire u dans K^m et un vecteur unitaire v dans K^n vérifiant :

$$M^*u = \sigma v \quad \text{et} \quad Mv = \sigma u$$

Les vecteurs u et v sont appelés **vecteur singulier à gauche** et **vecteur singulier à droite** pour σ, respectivement. Dans toute décomposition en valeurs singulières,

$$M = U\Sigma V^*$$

Les coefficients diagonaux de Σ sont égaux aux valeurs singulières de M. Les colonnes de U et de V sont, respectivement, vecteur singulier à gauche et à droite pour les valeurs singulières correspondantes [70].

Par conséquent :

- Une matrice M de type $m \times n$ possède au moins une et au plus $p = \min(m,n)$ valeurs singulières distinctes ;
- Il est toujours possible de trouver une base unitaire pour K^m constituée des vecteurs singuliers à gauche de M ;
- Il est toujours possible de trouver une base unitaire pour K^n constituée des vecteurs singuliers à droite de M ;

Une valeur singulière pour laquelle on peut trouver deux vecteurs singuliers à gauche (respectivement, à droite) qui sont linéairement indépendants est dite *dégénérée* [70].

Exemple :

Soit la matrice :
$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \end{pmatrix}$$

La décomposition en valeurs singulières de M est alors :

$$U = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2{,}236 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad V^* = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0{,}447 & 0 & 0 & 0 & 0{,}894 \\ 0 & 0 & 0 & 1 & 0 \\ -0{,}894 & 0 & 0 & 0 & 0{,}447 \end{pmatrix}$$

(Les valeurs non entières sont en fait des approximations à 10^{-3})

Ainsi, on a :

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2{,}236 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0{,}447 & 0 & 0 & 0 & 0{,}894 \\ 0 & 0 & 0 & 1 & 0 \\ -0{,}894 & 0 & 0 & 0 & 0{,}447 \end{pmatrix}$$

On vérifie que Σ ne possède des valeurs non nulles que sur sa diagonale. De plus, comme montré ci-dessous, en multipliant les matrices U et V^* par leurs transposées, on obtient la matrice identité :

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Et de même :

$$\begin{pmatrix} 0 & 0 & 0{,}447 & 0 & -0{,}894 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0{,}894 & 0 & 0{,}447 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0{,}447 & 0 & 0 & 0 & 0{,}894 \\ 0 & 0 & 0 & 1 & 0 \\ -0{,}894 & 0 & 0 & 0 & 0{,}447 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

3.5 Le réseau PCA-SVDNet

Il est primordial de réduire la redondance dans le descripteur fc (full connected) afin de rendre cela fonctionnel sous la distance euclidienne.

Pour résoudre le problème de corrélation, nous proposons SVDNet, qui est présenté par une couche fc contenant des vecteurs de pondération décorrélés. Nous présentons également un nouveau schéma d'apprentissage de trois étapes.

Pourquoi la SVDNet est-elle utilisée?

Notre idée clé est de trouver un ensemble de directions de projection orthogonales basées sur ce que PCANet2 a déjà appris de l'entraînement. Fondamentalement, pour une couche linéaire, un ensemble de base dans l'espace de distance de W (c'est-à-dire un sous-espace linéaire recouvert par les vecteurs de colonne de W) est une solution potentielle. En fait, il existe de nombreux ensembles de bases orthogonales. Dans notre travail, entre autres nous nous intéressons à la réduction du vecteur caractéristique extrait du réseau de PCA par la SVD. Pour cela, nous utilisons les vecteurs singuliers de W comme nouvelles directions de projection et pondèrons les résultats de projection avec les valeurs singulières correspondantes. Autrement dit, nous remplaçons W = USV^T avec US. Ce fait, maintient la capacité discriminante de fonctionnalité sur la totalité de l'espace échantillon de représentation. Nous faisons une preuve mathématique comme suit:

Étant donné deux images x_i et x_j, on note $\vec{h_i}$ et $\vec{h_j}$ comme les caractéristiques correspondantes avant Eigenlayer, respectivement. $\vec{f_i}$ et $\vec{f_j}$ sont les fonctionnalités de sortie de Eigenlayer. La distance euclidienne D_{ij} entre les caractéristiques de x_i et x_j est

$$D_{ij} = \|\vec{f_i} - \vec{f_j}\|_2 = \sqrt{(\vec{f_i} - \vec{f_j})^{\mathrm{T}}(\vec{f_i} - \vec{f_j})}$$
$$= \sqrt{(\vec{h_i} - \vec{h_j})^{\mathrm{T}} W W^{\mathrm{T}} (\vec{h_i} - \vec{h_j})}$$
$$= \sqrt{(\vec{h_i} - \vec{h_j})^{\mathrm{T}} U S V^{\mathrm{T}} V S^{\mathrm{T}} U^{\mathrm{T}} (\vec{h_i} - \vec{h_j})},$$

$$D_{ij} = \sqrt{(\vec{h_i} - \vec{h_j})^{\mathrm{T}} U S S^{\mathrm{T}} U^{\mathrm{T}} (\vec{h_i} - \vec{h_j})}$$

3.6 Réseau PCA-LDANet

Rappelons brièvement le principe de l'Analyse Discriminante Linéaire (ADL ou LDA).

3.6.1 Analyse Discriminante Linéaire LDA

L'analyse discriminante linéaire est définie comme une transformation linéaire orthogonale qui sépare au mieux deux ou plus classes d'objets. Il trouve l'ensemble de la projection des vecteurs qui maximisent le rapport entre les classes dispersion contre dispersion dans la classe. La combinaison résultante peut être utilisée pour la classification [45]. **LDA** trouve un sous-espace de dimension inférieure dans lequel le rapport entre la variance inter-classe et la variance intra-classe est maximisé. C'est-à-dire qu'une transformation linéaire

$$\mathbf{W}_{opt} = \arg\max_{\mathbf{W}} \frac{trace(\mathbf{W}^T \mathbf{S}_b \mathbf{W})}{trace(\mathbf{W}^T \mathbf{S}_w \mathbf{W})},$$

discriminante $\boldsymbol{W_{opt}}$ est le maximum du critère suivant :

Dont $\boldsymbol{S_b}$ et $\boldsymbol{S_w}$ sont inter-classe et intra-classe matrices de covariance, respectivement, définies comme suit :

$$\mathbf{S}_w = \frac{1}{N}\sum_{k=1}^{L}\sum_{\mathbf{x}_i \in C_k}(\mathbf{x}_i - \mathbf{u}_k)(\mathbf{x}_i - \mathbf{u}_k)^T, \quad \mathbf{u}_k = \frac{1}{N_k}\sum_{\mathbf{x}_i \in C_k}\mathbf{x}_i,$$

$$\mathbf{S}_b = \frac{1}{N}\sum_{k=1}^{L} N_k(\mathbf{u}_k - \mathbf{u})(\mathbf{u}_k - \mathbf{u})^T.$$

$$\mathbf{\Gamma}_c = \sum_{i \in S_c} \bar{\boldsymbol{X}}_i / |S_c|,$$
$$\mathbf{\Sigma}_c = \sum_{i \in S_c} (\bar{\boldsymbol{X}}_i - \mathbf{\Gamma}_c)(\bar{\boldsymbol{X}}_i - \mathbf{\Gamma}_c)^T / |S_c|.$$

3.6.2 Linear Discriminant Analysis Network (LDANet)

La construction du **LDANet**. Supposons que $\boldsymbol{N}$ les images d'apprentissage sont classées dans **C** classes $\{\boldsymbol{Ii}\}_{\boldsymbol{i \in Sc}}\ \boldsymbol{c} = \mathbf{1}, \mathbf{2}, \ldots, \boldsymbol{C}$, où $\boldsymbol{S_c}$ est l'ensemble des indices des images en classe **c**, et des zones de moyenne suppression associées à chaque image de classes distinctes $\bar{\boldsymbol{X}}_{\boldsymbol{i}} \in \mathbb{R}^{K_1 K_2 \times mn}$, $\mathbf{i \in Sc}$ sont donnés. On peut d'abord calculer la moyenne de classe $\boldsymbol{\Gamma_C}$ et la variabilité intra-classe $\boldsymbol{\Sigma_C}$ pour tous les patchs comme suit, [45]

Chaque colonne de $\boldsymbol{\Gamma_C}$ indique la moyenne des patchs autour de chaque pixel de la classe **C**, et $\boldsymbol{\Sigma_C}$ est la somme de toutes les covariances de l'échantillon de patch de la classe **c**. De même, la variabilité inter-classe des patchs est définie comme suit:

$$\boldsymbol{\Phi} = \sum_{c=1}^{C} (\boldsymbol{\Gamma}_c - \boldsymbol{\Gamma})(\boldsymbol{\Gamma}_c - \boldsymbol{\Gamma})^T / C,$$

où Γ est la moyenne des classes. L'idée de la **LDA** est de maximiser le rapport entre la variabilité inter-classe et la somme de la variabilité intra-classe au sein d'une famille de filtres orthonormés [45] où $Tr(\cdot)$ est l'opérateur de trace. La solution est connue sous le nom de vecteurs propres

$$\max_{\boldsymbol{V} \in \mathbb{R}^{k_1 k_2 \times L_1}} \frac{\mathrm{Tr}(\boldsymbol{V}^T \boldsymbol{\Phi} \boldsymbol{V})}{\mathrm{Tr}(\boldsymbol{V}^T (\sum_{c=1}^{C} \boldsymbol{\Sigma}_c) \boldsymbol{V})}, \text{ s.t. } \boldsymbol{V}^T \boldsymbol{V} = \boldsymbol{I}_{L_1},$$

principaux L1 de $\overline{\Phi} = (\sum_{c=1}^{C} \boldsymbol{\Sigma_C})\dagger\, \Phi$, où l'exposant † désigne le pseudo-inverse. Le pseudo inverse consiste à traiter le cas où $\sum_{c=1}^{C} \boldsymbol{\Sigma_C}$ n'est pas de rang complet, bien qu'il puisse exister un autre moyen de gérer cela avec une meilleure stabilité numérique. Les filtres **LDA** sont donc exprimés par $W_l^1 = mat_{K_1,K_2}\,(q_l\,(l)) \in \mathbb{R}^{K_1 K_2}, l = 1,2,\ldots,L_1$. Un réseau plus profond peut être construit en répétant le même processus [45].

Comme nous l'avons indiqué PCANet est basé sur une structure convolutionnelle, il apprend les filtres de plusieurs couches en appliquant PCA et exploite les histogrammes par bloc des codes binaires des cartes de caractéristiques pour générer les descripteurs locaux. Une couche DLA est rajoutée pour maximiser la marge entre les patchs inter-classes et minimiser la distance des patchs intra-classe dans la région locale. En particulier, nous construisons LDA suite à PCANet deux couches en empilant deux couches PCA, une couche LDA et une couche de fonctions. Elle est suivie par un cadre de classification très répandu une machine de vecteur de support linéaire (SVM).

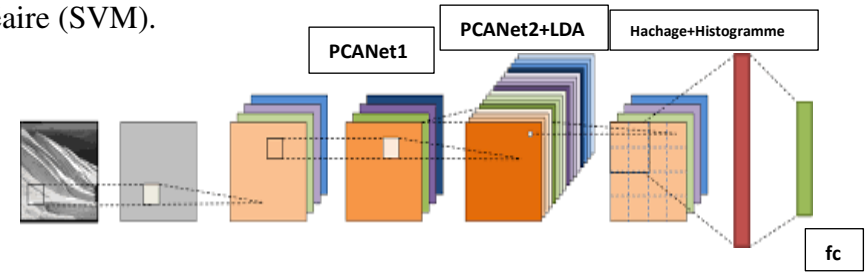


Figure 3. 11 L'approche proposée PCA-LDANet

3.7 Classification Support Vector Machine (SVM)

L'apprentissage automatique implique la prédiction et la classification des données. Pour ce faire, nous utilisons divers algorithmes d'apprentissage automatique en fonction de l'ensemble de données.

SVM ou (Support Vector Machine) La machine à vecteurs de support est une généralisation d'un classificateur appelé classificateur de marge maximale. Le classificateur de marge maximale est simple, mais il ne peut pas être appliqué à la majorité des bases de données [71]. Il peut résoudre des problèmes linéaires et non linéaires et fonctionne bien pour de nombreux problèmes pratiques. L'idée de SVM est simple: l'algorithme crée une ligne ou un hyperplan qui sépare les données en classes. Il peut être utilisé pour la classification binaire ou multi class. SVM est une extension du classifieur de vecteurs de support résultant de l'élargissement de l'espace des fonctions à l'aide de noyaux (Kernels). L'approche par noyau est simplement une

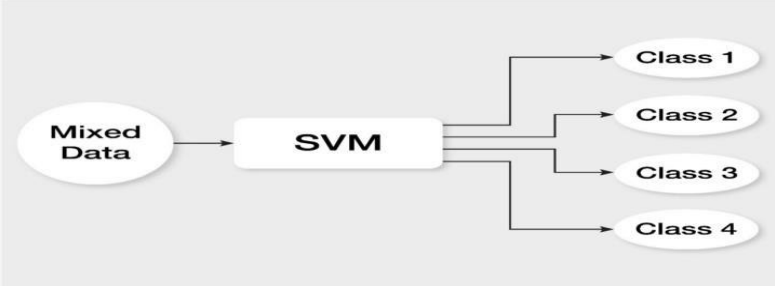


Figure 3. 12 Schéma de principe d'un SVM multiclasses[71]

approche informatique efficace pour tenir compte d'une limite non linéaire entre les classes [71]. Le noyau(Kernel) est une fonction qui quantifie la similarité de deux observations. Le noyau peut être de n'importe quel degré. L'utilisation d'un noyau avec un degré supérieur à un conduit à une limite de décision plus flexible [71].

La conception de l'architecture de classification SVM est très simple et nécessite principalement le choix du noyau (Kernel). SVM trouve l'hyperplan laissant la plus grande fraction possible de points de la même classe du même côté, tout en maximisant la distance entre l'une ou l'autre classe et l'hyperplan. Cet hyperplan minimise le risque d'exemples de classification erronés de l'ensemble de tests [71].

Séparation optimale des hyperplans : Base d'exemples d'apprentissage $(\boldsymbol{x_i}, \boldsymbol{y_i})_{\mathbf{1 \le i \le N}}$, chaque exemple $\boldsymbol{x_i} \in \mathbb{R}^{\boldsymbol{d}}$, **d** est le dimension l'espace d'entrée, appartient à une classe marquée par $\boldsymbol{y_i} \in \{-\mathbf{1}, \mathbf{1}\}$. L'objectif est de définir un hyperplan qui divise l'ensemble des exemples de manière à ce que tous les points ayant la même étiquette se trouvent du même côté de l'hyperplan. Cela revient à trouver $\boldsymbol{w}$ et $\boldsymbol{b}$ pour que [71]

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 0,\ i = 1, \ldots, N \qquad (1)$$

S'il existe un hyperplan satisfaisant l'**Eq (1)**, on dit que l'ensemble est séparable linéairement. Dans ce cas, il est toujours possible de redimensionner $\boldsymbol{w}$ et $\boldsymbol{b}$ pour que

c'est-à-dire que le point le plus proche de l'hyperplan est à une distance de $\mathbf{1} / ||\boldsymbol{w}||$. Ensuite, **Eq. (1)** devient

$$\min_{1\le i\le N} y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\ge 1,\ i=1,\ldots,N$$

$$y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\ge 1 \qquad (2)$$

La marge peut être vue comme une mesure de la capacité de généralisation: plus la marge est grande, plus la généralisation devrait être bonne.

Puisque $||\ w^2\ ||$ est convexe, il est possible de le minimiser sous des contraintes linéaires (2) avec des multiplicateurs de Lagrange. Si on note $\alpha = (\alpha_1,\ldots,\alpha_N)$ les multiplicateurs de Lagrange N non négatifs associés aux contraintes (2), notre problème d'optimisation revient à maximiser [71]

Une fois que le vecteur $\alpha 0 = (\alpha_1^0,\ldots,\alpha_N^0)$ du problème de maximisation (3) a été trouvé, **OSH** (w_0, b_0) présente l'extension suivante:

$$W(\boldsymbol{\alpha}) = \sum^{N}\alpha_i - \frac{1}{2}\sum^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j \qquad (3)$$

$$\mathbf{w}_0 = \sum_{i=1}^{N}\alpha_i^0 y_i\mathbf{x}_i \qquad (4)$$

Les vecteurs de support sont les points pour lesquels $\alpha_i^0 > 0$ vérifie $Eq.(2)$ avec égalité.

Considérant le développement (4) de w_0, la fonction de décision de l'hyperplan peut donc être écrite comme suit:

$$f(\mathbf{x}) = \quad f(\mathbf{x}) = sgn\left(\sum_{i=1}^{N}\alpha_i y_i K(\mathbf{x}_i,\mathbf{x}) + b\right)$$

Conclusion

Dans ce chapitre, nous avons présenté un rappel de plusieurs méthodes de réduction basiques. Ces méthodes sont fondamentales aux transformations étudiées. Nous avons présenté aussi les techniques que nous avons retenu pour notre conception selon différents critères : simplicité de l'algorithme, optimalité d'apport en information, pouvoir discriminant et aussi pouvant être rapide en temps de calcul. Selon le dernier critère les techniques de réduction peuvent être regroupées en deux catégories travaillant dans le domaine fréquentiel et spatial. Dans cette dernière catégorie plusieurs transformées peuvent être utilisées telles que ACP, SVD et LDA.

Par la suite nous avons présenté une étude détaillées des trois méthodes PCANet2, PCASVDNet, PCALDANet et leurs conceptions afin de pouvoir les implémenter dans le chapitre 4. Nous avons vu aussi les caractéristiques et intérêts de chacune d'elles. Nous nous sommes intéressés aux terminologies et notions liées aux techniques des méthodes de

réduction en profondeur. Ces terminologies sont nécessaires pour les chapitres suivants tels que les définitions mathématiques et les structures des réseaux Deep Learning pour l'implémentation et a validation du modèle de système de reconnaissance de visage.

Chapitre 4
Implémentation et Résultats

Introduction

Dans ce chapitre, nous présentons la conception et la mise en œuvre de notre application. En commençant tout d'abord par une présentation des outils de développement ; langage de programmation choisi (Matlab) et le matériel (PC). Nous proposons les implémentations de deux méthodes proposées : PCANet2-SVD et PCANet2-LDA en mettant en œuvre la PCANet2 sur la base de données FERET. Ensuite nous présenterons quelques résultats obtenus par ces différentes méthodes. Nous terminerons le chapitre par une étude comparative avec d'autres méthodes de l'état de l'art récent.

4.1 Présentation des outils de développement

Le matériel utilisé est un PC personnel Dell (Inspiron 15 3000 Series) avec un processeur Intel® core™ i5-4210U CPU @ 1.70GHz et une capacité mémoire de 8 Gb avec Windows 10 professionnel, 64 bit type système.

Pour le développement de nos algorithmes nous utilisons le Matlab (Matrix Laboratory) version 2018 qui est un logiciel interactif basé sur le calcul matriciel. Il est utilisé dans les calculs scientifiques et les problèmes d'ingénierie parce qu'il permet de résoudre des problèmes numériques complexes en moins de temps requis par les langages de programmation courant, et ce grâce à une multitude de fonctions intégrées et à plusieurs programmes outils testés et regroupés selon usage dans des dossiers appelés boites à outils ou "toolbox". Son objectif, par rapport aux autres langages, est de simplifier au maximum la transcription en langage informatique d'un problème mathématique, en utilisant une écriture la plus proche possible du langage naturel scientifique.

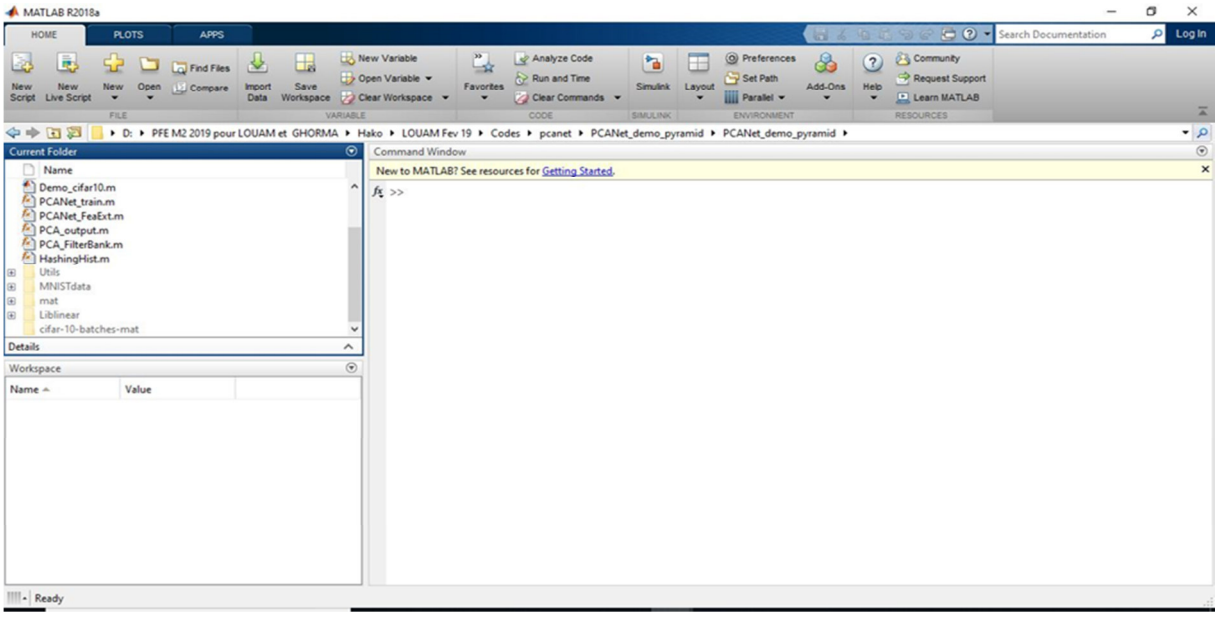


Figure 4. 1 Environnement Matlab

4.2 Base de données utilisée

La base de données FERET (Face Recognition Technology) est une vaste base de données d'images faciales, divisée en parties de développement et parties isolées. La partie développement est mise à la disposition des chercheurs et la partie isolée est réservée au test des algorithmes de reconnaissance faciale [72]. La procédure d'évaluation de FERET est un test indépendant d'algorithmes de reconnaissance faciale. Le test a été conçu pour :

- Permettre une comparaison directe entre différents algorithmes.
- Identifier les approches les plus prometteuses.
- Evaluer l'état de la technique en matière de reconnaissance faciale.
- Identifier les orientations futures de la recherche.
- Faire progresser l'état de l'art de reconnaissance des visages.

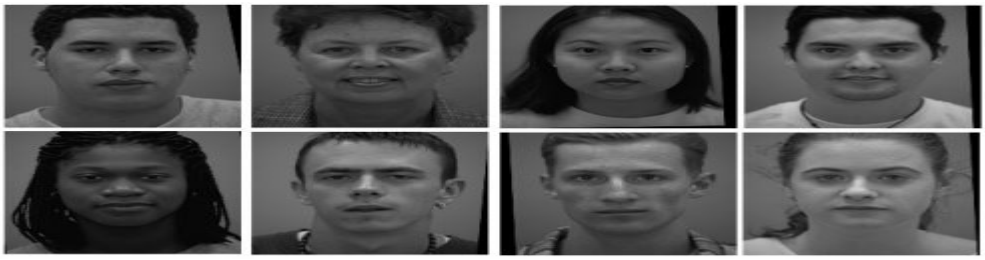


Figure 4. 2 Exemples d'images de visage de base de données FERET [72].

Tableau 4.1 Informations sur les données d'apprentissage et de test pour FERET [72].

Catégories de Probe	Taille de galerie	Taille de Probe
FB	1196	1195
Duplicate I	1196	722
FC	1196	194
Duplicate II	1196	234

4.3 Implémentation des méthodes de réduction en profondeur pour la RV

Nous utilisons le principe du Deep Learning appliqué aux méthodes de réduction telles :

PCANet2, PCANet2-SVD, PCANet2-LDA. Toutes ces trois méthodes sont implémentées et validées sur la base de données Feret. Elles sont détaillées dans ce qui suit. Le SVM est utilisé pour la classification. L'objectif de ce travail se présente comme suit :

- Etudier les méthodes de réduction de dimension en profondeur séparément pour améliorer les performances du SRV et aussi réduire les temps de calcul.

- Puis, faire une comparaison entre les trois méthodes conçues et implémentées du côté de la précision, le taux d'erreur, le temps de calcul et nombre de caractéristiques utilisé.

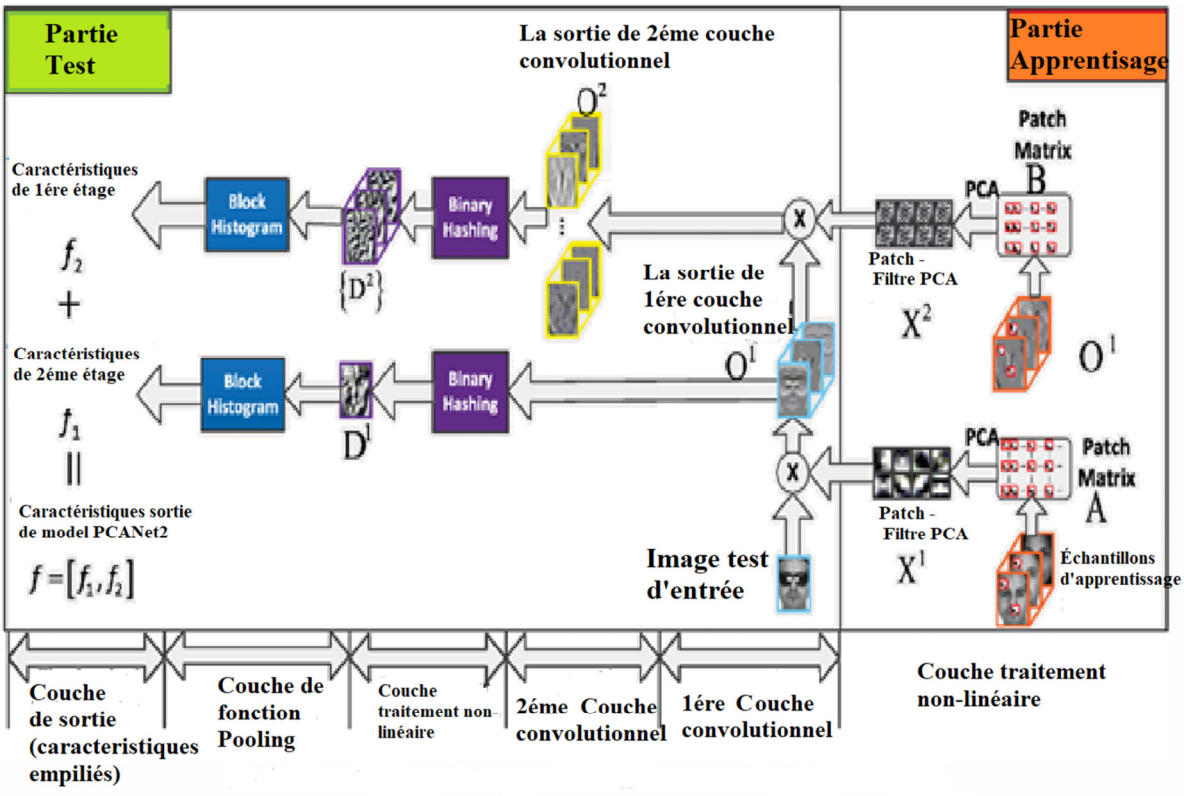
Dans ce qui suit, nous présentons les résultats obtenus grâce à des expériences réalisées par l'application des méthodes de réduction de dimension (PCANet2, PCANet2-SVD et PCANet2-LDA) sur la base de données FERET.

4.4 Implémentation de la PCANet pour la RV

4.4.1 Méthode de PCANet2

PCANet est un réseau d'apprentissage en profondeur pour la classification d'images permettant d'extraire des caractéristiques dans une image. Il se compose principalement de filtres PCA en cascade, de hachage binaire et d'histogrammes de blocs. Il est possible d'appliquer plusieurs ensembles de données et les paramètres de structure et de paramètres sont simples.

4.4.2 Structure de PCANet2



La structure PCANet est comme le montre la figure suivante:

Figure 4. 3 Structure de la PCANet2

La **figure 4.3** illustre l'extraction de caractéristiques par les filtres PCANet dans les phases d'apprentissage et test.

Le tableau d'entraînement de taille N = m × n, la taille du patch dans toutes les couches est $k_1 \times k_2$, seuls les filtres PCA doivent apprendre l'image d'entrée.

4.4.3 Principe de PCANet2

1ère couche : Chaque image d'entrée est divisée en $(m-k_1 + 1)(n-k_2 + 1)$ ème patch, une valeur moyenne est calculée sur la matrice d'entrée comme suit :

$$\boldsymbol{X} = [\bar{\boldsymbol{X}}_1, \bar{\boldsymbol{X}}_2, ..., \bar{\boldsymbol{X}}_N] \in \mathbb{R}^{k_1 k_2 \times Nmn}.$$

Afin de minimiser le recouvrement d'erreur :

$$\min_{\boldsymbol{V} \in \mathbb{R}^{k_1 k_2 \times L_1}} \|\boldsymbol{X} - \boldsymbol{V}\boldsymbol{V}^T\boldsymbol{X}\|_F^2, \text{ s.t. } \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_{L_1}$$

(Équivalente à la procédure ci - dessus utilisé dans le calcul de la covariance de l'image d'entrée). XX^T est la matrice des valeurs caractéristiques dans l'ordre décroissant, on prendra L_1 vecteurs propres caractéristiques d'image extraite de la première couche :

$$\boldsymbol{W}_l^1 \doteq \text{mat}_{k_1,k_2}(\boldsymbol{q}_l(\boldsymbol{X}\boldsymbol{X}^T)) \in \mathbb{R}^{k_1 \times k_2}, \ l = 1, 2, ..., L_1$$

.

2ème couche : La caractéristique extraite de la couche précédente est convertie en convolution avec la matrice d'entrée après le remplissage à zéro

$$\mathcal{I}_i^l \doteq \mathcal{I}_i * \boldsymbol{W}_l^1, \ i = 1, 2, ..., N, \quad l = 1,2,3 .. L_1$$

Elle est centrée et chaque bloc est vectorisé pour obtenir la matrice d'entrée de la deuxième couche (la taille de la matrice d'entrée est L_1 fois la couche précédente).

Le vecteur de caractéristiques de la deuxième couche est obtenu de manière similaire:

$$\boldsymbol{Y} = [\boldsymbol{Y}^1, \boldsymbol{Y}^2, ..., \boldsymbol{Y}^{L_1}] \in \mathbb{R}^{k_1 k_2 \times L_1 Nmn}$$

Nous obtenons de la même manière le vecteur de caractéristiques de la deuxième couche:

$$\boldsymbol{W}_\ell^2 \doteq \text{mat}_{k_1,k_2}(\boldsymbol{q}_\ell(\boldsymbol{Y}\boldsymbol{Y}^T)) \in \mathbb{R}^{k_1 \times k_2}, \ \ell = 1, 2, ..., L_2.$$

$$\mathcal{O}_i^l \doteq \{\mathcal{I}_i^l * \boldsymbol{W}_\ell^2\}_{\ell=1}^{L_2}.$$

Donc, les caractéristiques de sortie de la deuxième couche sont L_1 x L_2

Nous répétons les étapes ci-dessus pour créer plus de niveaux de structure.

Couche de sortie : les entités en sortie de la couche précédente sont binarisées :

$\{H(\mathcal{I}_i^l * \boldsymbol{W}_\ell^2)\}_{\ell=1}^{L_2}$ (H (.) Signifie> 0 = 1, les autres prennent 0)

On suppose ici que le nombre de filtres de la couche précédente est égal à L_2, la sortie de chaque filtre est quantifiée et la combinaison de poids est ajoutée :

$$\mathcal{T}_i^l = \sum_{\ell=1}^{L_2} 2^{\ell-1} H(\mathcal{I}_i^l * \boldsymbol{W}_\ell^2)$$

Par conséquent, chaque pixel a une plage de valeurs de [0, 2 ^ L2-1].

Un histogramme est généré pour chaque valeur statistique de bloc et transformé en un vecteur : $\boldsymbol{f}_i = [\mathrm{Bhist}(\mathcal{T}_i^1), \ldots, \mathrm{Bhist}(\mathcal{T}_i^{L_1})]^T \in \mathbb{R}^{(2^{L_2})L_1 B}$.

Ensuite, le résultat final est $\mathrm{Bhist}(\mathcal{T}_i^l)$ bloc pouvant se chevaucher en fonction de la situation réelle. En général, la reconnaissance de visage ne se chevauche pas par contre la discrimination de texture se chevauchent.

Complexité informatique, le choix des paramètres

$$\mathcal{O}(mnk_1k_2(L_1 + L_2) + mn(k_1k_2)^2)$$

Les paramètres de modèle de PCANet2 incluent :

- l'ordre du filtre PCA et la taille k_1, k_2,
- le nombre de filtres dans chaque ordre L_1L_2
- la taille de bloc de l'histogramme local de la couche en sortie.

Le groupe de filtres PCA requiert $k_1k_2 > L_1, L_2$.

Dans l'expérience, huit directions peuvent être définies, comme le filtre de Gabor, et $L_1 = L_2 = 8$ est fixe, mais l'ajustement de la valeur L_1L_2 peut améliorer légèrement les performances.

En règle générale, le PCANet en deux étapes est suffisant pour la performance et la définition d'un plus grand nombre de commandes n'améliore pas significativement la performance. Les histogrammes locaux de blocs de tailles plus grandes utilisent les fonctions de distorsion extraites.

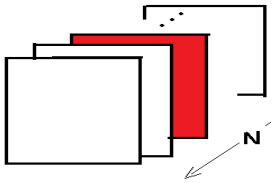
4.4.4 Algorithme PCANet

L'algorithme principal contient la mise en cascade de deux convolutions de banque de filtres avec une étape de normalisation moyenne intermédiaire, suivie d'une étape de hachage de bits et d'une étape finale de concaténation d'histogramme. PCANet est un modèle qui intègre **PCA** dans un modèle d'apprentissage en profondeur (CNN)

L'apprentissage en profondeur nous aide à découvrir plusieurs niveaux de représentation avec l'espoir que les caractéristiques de niveau supérieur peuvent représenter une sémantique plus abstraites.

L'algorithme principal comprend la mise en cascade de deux convolutions de banc de filtres avec une étape de normalisation moyenne intermédiaire, suivie d'une étape de hachage binaire et d'une étape de formation d'histogramme finale.

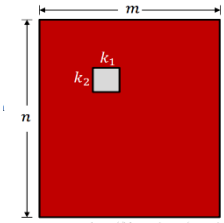
- **Algorithme couche 1 de PCANet2**



Entrées : **N** images d'apprentissage de taille $\boldsymbol{m} \times \boldsymbol{n}$.

Considérons '**i**' image.

1. Taille du patch k_1 x k_2 dans l'image,
2. Convertir chaque patch en vecteur,
3. Trouver le patch moyen et le soustraire de tous les patchs,



$$\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,\tilde{m}\tilde{n}} \in \mathbb{R}^{k_1 k_2}$$

4. Calcul de la matrice **X** qui est une matrice concaténée de toutes les images normalisées moyennes.
5. Effectuer une **ACP** sur ces images de moyenne nulle et conserver les huit principaux composants,
6. **PCA** minimise l'erreur de reconstruction avec la famille de filtres orthonormaux et la solution est (cela peut être réglé en fonction de l'erreur de validation) les vecteurs propres principaux de $\boldsymbol{XX^T}$
7. Nous obtenons les filtres:

$$\boldsymbol{W_l^1} = \boldsymbol{mat_{k_1 \times k_2}}\{\boldsymbol{q_l}(\boldsymbol{XX^T})\} \in \mathbb{R}^{k1 \times k2}, \ \textit{where } l = 1,2, \ldots L_1$$

Sortie : matrice W

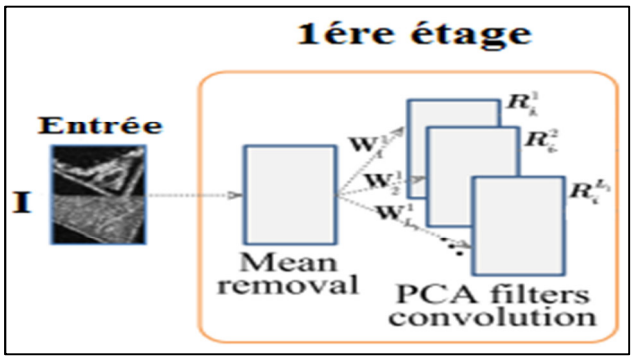


Figure 4.4 Illustration du 1ère étage de PCANet

- **Couche 2 de PCANet2**

La 2ème couche est construite en itérant l'algorithme de la 1ère couche sur chacune des images de sortie obtenues par convolution avec des filtres de 1ère étape.

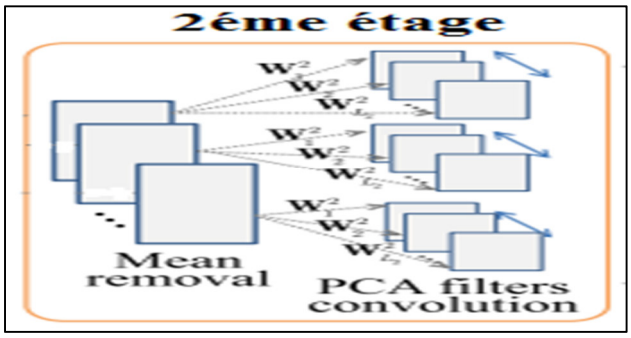
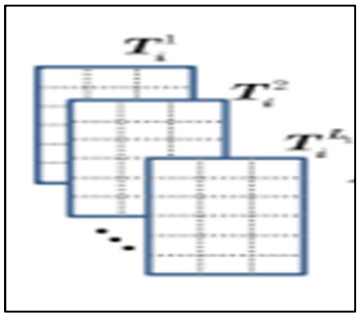


Figure 4.5 Illustration du 2ème étage de PCANet

- **Couche du hachage binaire**

Ces vecteurs de caractéristiques sont convertis en décimales à l'aide d'une étape de

$$\mathcal{T}_i^l \doteq \sum_{\ell=1}^{L_2} 2^{\ell-1} H(\mathcal{I}_i^l * \boldsymbol{W}_\ell^2).$$



Heaviside semblable à la fonction $\boldsymbol{H}$ **:**

Figure 4.6 Résultat de Hachage

- **Couche de sortie – Histogramme**

Chacune des $\boldsymbol{L_1}$ images (décimales) de la sortie est divisée en blocs **B**.

- Calculer l'histogramme (avec $\mathbf{2^{L_2}}$ **cases**) de valeurs décimales dans chaque

$$\boldsymbol{f}_i \doteq [\mathrm{Bhist}(\mathcal{T}_i^1), \ldots, \mathrm{Bhist}(\mathcal{T}_i^{L_1})]^T \in \mathbb{R}^{(2^{L_2})L_1 B}.$$

bloc et les concaténer en un seul vecteur pour former f_i.

Ainsi, la caractéristique d'une image de l'ensemble de données est obtenue. Nous obtenons les vecteurs de caractéristiques pour toutes les images puis nous appliquons le classificateur SVM.

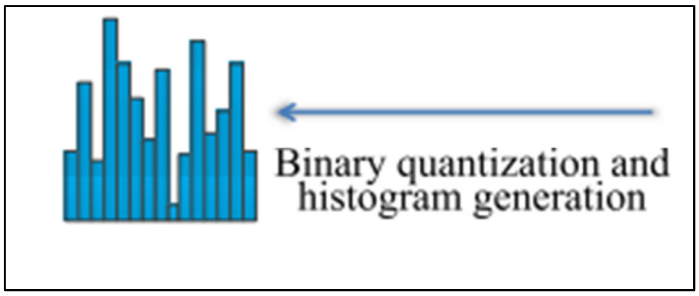


Figure 4.7 Quantification binaire et génération d'histogramme

4.4.5 Tests et résultats

Nous utilisons PCANet2 pour effectuer des expériences sur la reconnaissance faciale dans cette partie du travail. Les fonctionnalités extraites sont formées et prédites par libSVM. Nous évaluons maintenant la performance de la proposition PCANet2 par différents paramètres sur la BDD FERET (**Dup-II)**.

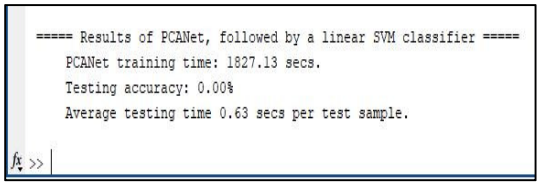
1. **Le test sur FERET**

Nous appliquons ensuite le PCANet2 avec les filtres PCA l'apprentissage est réalisée sur la BDD MultiPIE. L'ensemble test de la base de données FERET est subdivisé en quatre catégories :

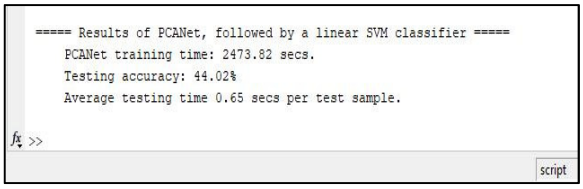
Fb avec différents changements d'expression; **Fc** avec différentes conditions d'éclairage; **Dup-I** acquis dans le délai de trois à quatre mois; **Dup-II** (utilisé pour nos expériences) acquis au moins sur un an et demi.

2. **Le choix des paramètres de PCANet2**
    a. **NumStages** le nombre d'étages dans PCANet est fixé à 2
    b. **PatchSize** la taille du patch (taille du filtre) pour les patchs carrés est fixé à [5 5] signifie que la taille du patch est égale à 5 et 3 dans le premier et deuxième étage respectivement.
    c. **NumFilters** le nombre de filtres dans chaque étape est égal à [8 8] ce qui signifie 8 dans le premier étage et 8 filtres dans le deuxième étage, respectivement.
    d. **HistBlockSize** la taille de chaque bloc pour l'histogramme local est [15 15]
    e. **BlkOverLapRatio** pourcentage de régions de blocs superposés; 0 signifie pas de chevauchement entre blocs et 0,3 signifie que 30% de la taille de bloc est superposé
3. **Résultats**

Nous avons testé le PCANet2 avec plusieurs nombres de paramètres caractéristiques dans l'intervalle [0, 1196] et nous avons obtenu les résultats suivants :

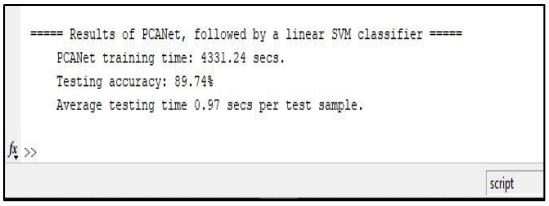


Résultats de 10 paramètres

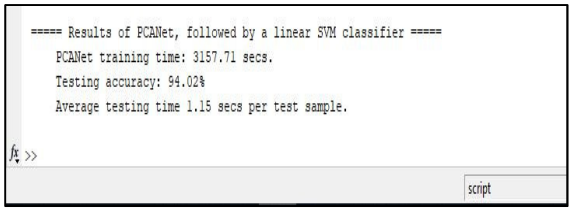


Résultats de 100 paramètres

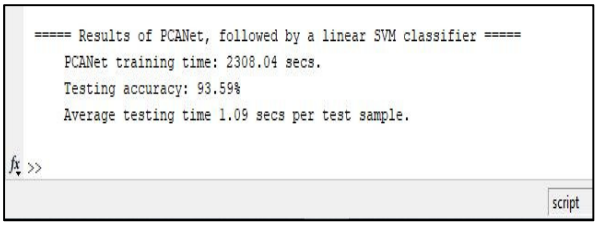
Résultats de 500 paramètres



Résultats de 1000 paramètres



Résultats de 1196 paramètres



Tous les résultats de PCANet2 sont recensés sur le tableau 4. Ci-dessous :

Table 4. Performances de PCANet2 en fonction du nombre de paramètres (FERET-Dup II)

Nombre de paramètres	**Accuracy (%)**	**TEE (%)**	**Temps (training /test (s))**
10	0	1	1827.13/0.63
50	18.38	0.8162	2176.19/0.66
100	44.02	0.5598	2473.82/0.65
500	89.74	0.1026	4331.24/0.97
1000	**94.02**	**0.0598**	**3157.71/1.15**
1196	93.59	0.0641	2308.04/1.09

La figure 4.4 nous révèle les courbes de la précision (Accuracy) et de l'erreur (TEE (%)) par rapport au nombre de paramètres.

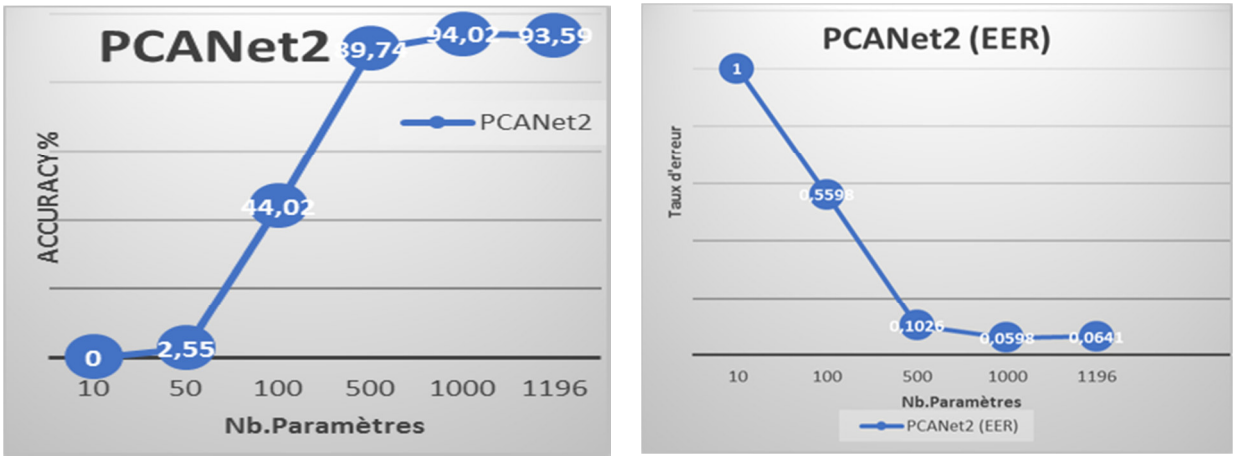


Figure 4. 4 Précision (Accuracy) et TEE pour PCANet2 sur FERET (Dup-II)

D'après la figure 4.4 nous remarquons que le meilleur résultat est obtenu pour un nombre de paramètres égale à 1000 avec une **accuracy = 94.02**%, un taux d'erreur **TEE = 0.0598**%, un **temps d'apprentissage = 3157.71 s** et un **temps de test = 1.15 s.**

Dans les deux expériences suivantes, nous essayons d'améliorer les performances de notre système par la réduction du vecteur de paramètres caractéristiques pour la préparation à une bonne classification et aussi pour réduire le temps de classification.

4.5 La méthode de PCANet2-SVD

La structure de notre approche PCANet2-SVD se présente sous la forme du schéma représenté par la figure 4.5 ci-dessous :

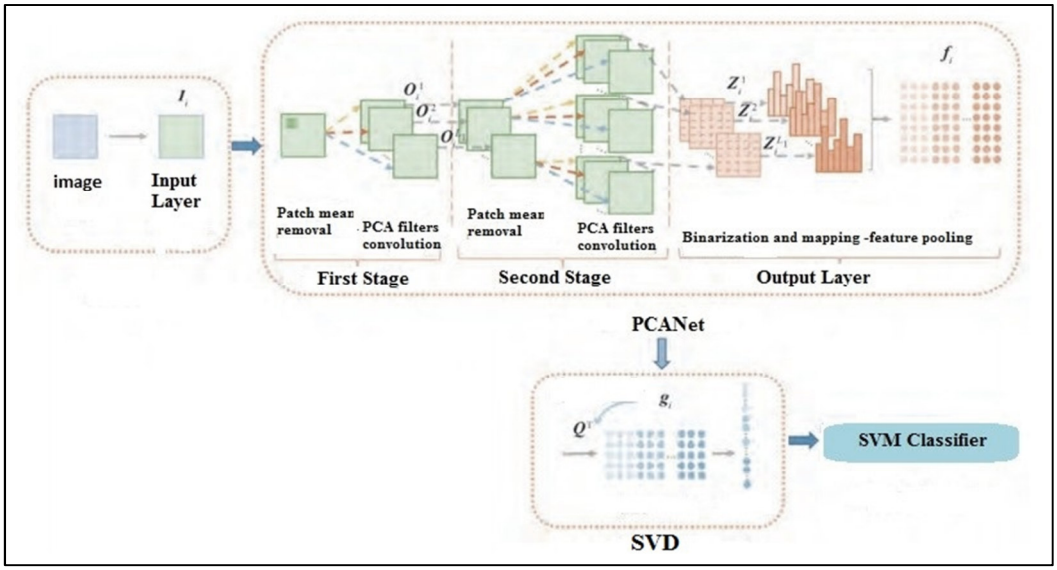


Figure 4. 5 Schéma bloc de PCANet2-SVD

Nous avons utilisé le **SVDNet** pour résoudre le problème de corrélation. Il est présenté par une couche fc contenant des vecteurs de pondération décorrélés. L'algorithme de SVNet est présenté dans ce qui suit.

4.5.1 Algorithme SVDNet

Entrée: les données (sorties de ACPNet2).

0. Ajoutez le Eigenlayer et ajustez le réseau.

pour **t** ← **1** à **T** faire

1. Décorrélation: décomposer **W** avec **SVD** puis mettre à jour: **W** ← **US**

2. Restraint: ajuster le réseau avec le Eigenlayer fixé

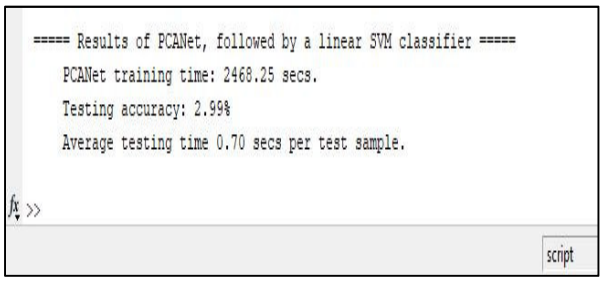
3. Relaxation: ajuster le réseau avec le Eigenlayer non fixé

Fin.

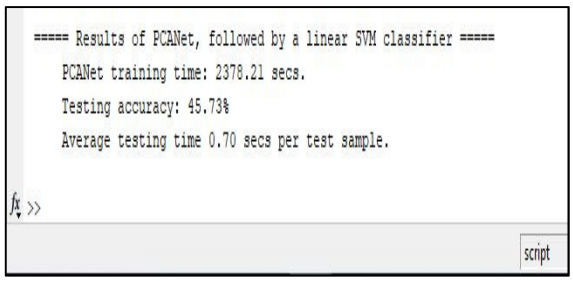
Sortie: SVDNet

4.5.2 Tests et résultats

Nous évaluons maintenant la performance de l'architecture basée sur le **PCANet2** et **SVD** en variant le nombre de paramètres caractéristiques sur la BDD FERET (Dup II).

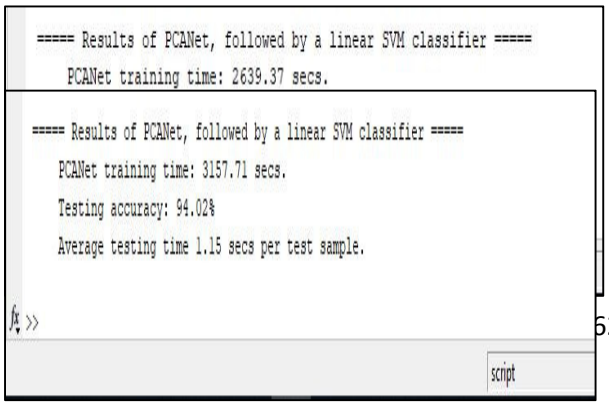


Résultats pour 10 paramètres



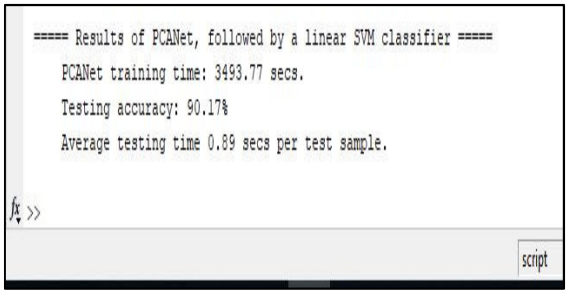
Résultats pour 100 paramètres

Résultats pour 300 paramètres



Résultats pour 600 paramètres

Résultats pour 1000 paramètres



Résultat pour 1196 paramètres

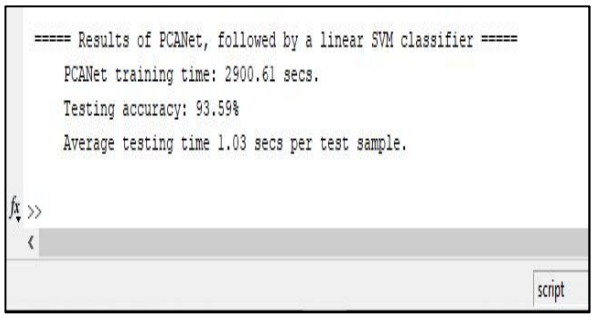


Tableau 5. Performances du SRVPCANet2-SVD pour différents paramètres (FERET– Dup-II)

Nombre de paramètres	Accuracy (%)	TEE (%)	T (training/test time s)
10	2.99	0.9701	2468.25
100	45.73	0.5427	2378.21
300	81.62	0.1838	2639.37
600	90.17	0.0983	3493.77/0.89
1000	**94.02**	**0.0598**	**3157.71/1.15**
1196	93.59	0.0641	2900.61/1.03

La figure suivante nous montre les courbes de précision (Accuracy(%)) et le taux d'erreur (TEE (%)) par rapport au nombre de paramètres.

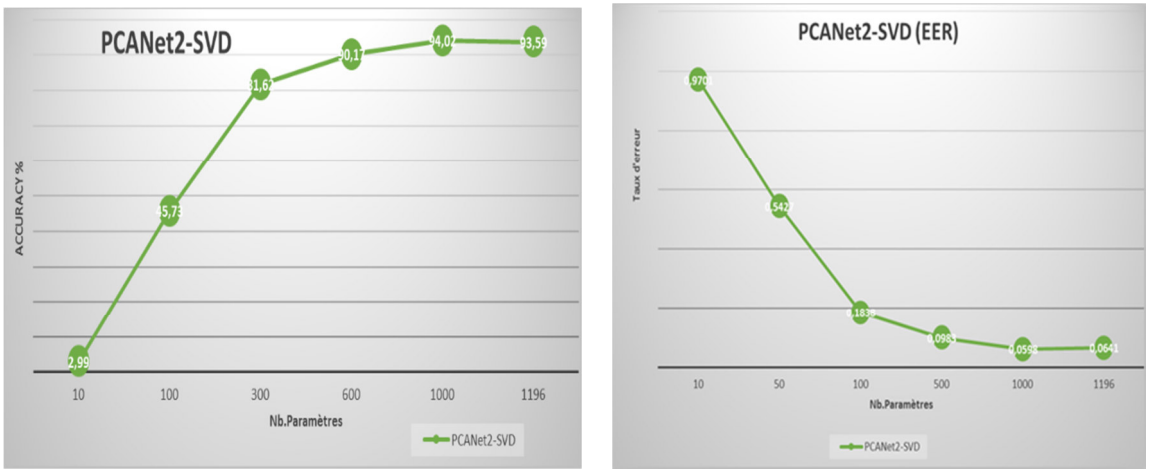


Figure 4. 6 Courbes de performances du SRVPCANet2-SVD sur FERET (Dup II)

Les résultats obtenus montrent que les résultats obtenus par PCANet2 et PCANet2SVD restent identiques. Ceci est prévisible car la SVD est là pour la décorrélation davantage des paramètres caractéristiques. C'est aussi peut être dû à la taille de la base de données, car les

méthodes Deep Learning montrent leur efficacité sur les bases de grandes tailles. Ceci dit, la méthode PCANet2-SVD est retenue, car on voit que ses résultats sont meilleurs que ceux de la PCANet2 pour un nombre réduit de paramètres caractéristiques.

4.6 Implémentation de la méthode de PCANet2-LDA pour RV

Après la conception et de l'implémentation des deux méthodes PCANet2 et PCANet2-SVD, nous nous intéressons à la PCANet2-LDA qui va nous aider sûrement à améliorer la classification de notre système, car les deux expériences précédentes montrent bien que les méthodes utilisées (PCANet2 ou PCANet2-SVD) assurent l'extraction des caractéristiques, mais la classification reste à améliorer. La figure 4. ci-dessous montre le schéma de PCANet2-LDA.

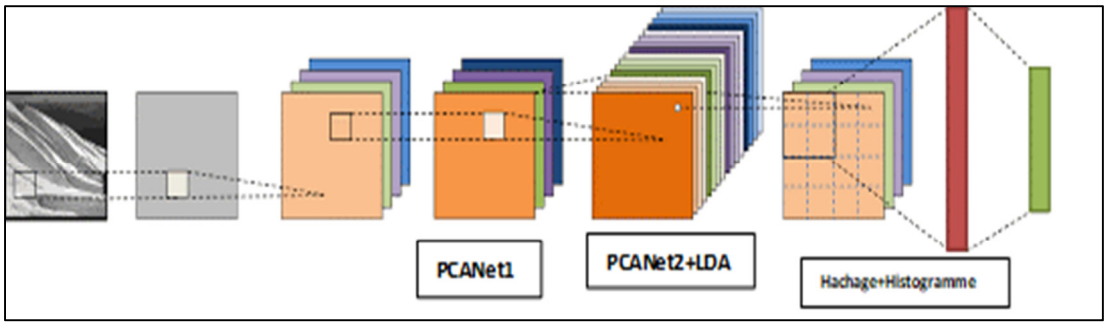


Figure 4. 7 Schéma bloc PCANet2-LDA

4.6.1 Principe de PCANet2-LDA

Pour la classification: Analyse Discriminante Linéaire Multicouche.

Construction du modèle : diviser N images d'apprentissage en catégories C, Sc est l'étiquette de classe de chaque image, les informations sur la classe et les correctifs de réduction de la moyenne sont combinés dans une matrice, et les changements dans la classe et dans la classe intra-classe sont calculés selon la formule suivante:

$$\boldsymbol{\Gamma}_c = \sum_{i \in S_c} \bar{\boldsymbol{X}}_i / |S_c|,$$

$$\boldsymbol{\Sigma}_c = \sum_{i \in S_c} (\bar{\boldsymbol{X}}_i - \boldsymbol{\Gamma}_c)(\bar{\boldsymbol{X}}_i - \boldsymbol{\Gamma}_c)^T / |S_c|$$

Où: $\tilde{\boldsymbol{X}}_i \in \mathbb{R}^{k_1 k_2 \times mn},\ i \in S_c$

le changement de patch entre classes définit :

$$\boldsymbol{\Phi} = \sum_{c=1}^{C} (\boldsymbol{\Gamma}_c - \boldsymbol{\Gamma})(\boldsymbol{\Gamma}_c - \boldsymbol{\Gamma})^T / C$$

L'idée de LDA par:

$$\max_{\boldsymbol{V} \in \mathbb{R}^{k_1 k_2 \times L_1}} \frac{\mathrm{Tr}(\boldsymbol{V}^T \boldsymbol{\Phi} \boldsymbol{V})}{\mathrm{Tr}(\boldsymbol{V}^T (\sum_{c=1}^{C} \boldsymbol{\Sigma}_c) \boldsymbol{V})},\ \text{s.t. } \boldsymbol{V}^T \boldsymbol{V} = \boldsymbol{I}_{L_1}$$

4.6.2 L'algorithme de PCANet2-LDA

Algorithme 1 Modèle approximatif de LDA pour obtenir des filtres

Entrées : X (sortie de PCANet), nombre de filtres L.

1. Initialise U ; initialiser V ;
2. $V = (U^T U)^{-1} U^T X$.
3. Mettre à jour $\bar{V}$
4. Mettre à jour U
5. Mise à jour V

Fin While

6. $\mathbf{W_i = mat^{k1, k2}(U_{:,i})}$, **i = 1, 2, · · ·, L.**

Sortie: Les filtres W = [W_1, W_2, · · ·, W_L].

Algorithme 2 Le processus d'apprentissage de PCANet2-LDA pour la classification

Entrées :

Base de données **P** ; nombre d'étapes LDA N_l; coefficients de pénalité α_i et β_i

au (i) ème étage LDA, **i =1, 2, · · ·, N_l;** nombre de filtres $\mathbf{L_i}$ au (i)ème étage LDA, **i = 1, 2, ·, N_l**

1. Prendre X_1 = P;

Pour chaque **i = 1: N_l** faire

2. W_i = LDA (X_i, a_i, $ß_i$, L_i);
3. $X_i + 1 = X_i W_i$;

fin pour

4. étape de hachage et d'histogrammes: prenez $X_{N_l + 1}$ en entrée;
5. Train linéaire SVM: prendre les codes d'histogrammes en entrée;

Sortie : Résultats de la classification.

4.5.3 Tests et résultats :

Nous évaluons maintenant la performance de l'architecture basée sur le **PCANet2** et **LDA** en variant le nombre de paramètres caractéristiques sur la BDD FERET (Dup II).

Nombre de paramètres	Accuracy %	ERR	T (training time s)/ time test
10	**0**	**1**	**2284.22**
50	**0**	**1**	
100	**100**	**0**	

500	**100**	**0**	
1000	**100**	**0**	
1196	**100**	**0**	

Table. Taux de reconnaissance, taux d'erreur et temps d'apprentissage de PCANet2-LDA en différents paramètres (FERET – DUP-II)

Conclusion

PCANet n'a pas besoin d'ajuster les paramètres et de résoudre les problèmes d'optimisation numérique. L'établissement du réseau nécessite uniquement une carte en cascade et un étage de sortie non linéaire.

Il est plus pratique pour PCANet d'extraire les informations de classification.

PCANet fournit une base de référence précieuse pour des structures de réseau d'apprentissage en profondeur plus complexes. Il est pratique d'ajouter ou de modifier les parties correspondantes pour créer un réseau d'apprentissage plus efficace. Comparativement PCANet2, PCANet2SVD et à LDANet, les performances sont meilleures pour PCANet2-LDA.