Réseau profond de matrices symétriques définies positives pour la reconnaissance en temps réel des gestes de la main en 3D *

Mostefa Ben Naceur, Luc Brun, Olivier Lézoray
Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France
{mostefa.bennaceur,luc.brun,olivier.lezoray}@unicaen.fr

Résumé – Cet article propose un nouveau réseau neuronal de matrices symétriques définies positives pour la reconnaissance des gestes de la main. Les données du squelette d'entrée sont représentées par des matrices SDP qui permettent d'encoder efficacement des statistiques d'ordre élevé telles que les covariances ou les corrélations entre les caractéristiques des articulations. La reconnaissance en ligne s'effectue à l'aide de deux réseaux explorant une séquence de gestes et qui permettent de simultanément détecter et reconnaître un nouveau geste.

Abstract – This paper proposes a new neural network of symmetric positive definite matrices for hand gesture recognition. The input skeleton data is represented by SDP matrices that allow to encode efficiently high-order statistics such as the covariances or the correlations between the joints' features. Online recognition is performed using two networks that explore a sequence of gestures and simultaneously detect and recognize a new gesture.

1 Introduction

La reconnaissance des gestes de la main est un des éléments fondamentaux de nombreuses applications en interaction homme machine. Avec le développement de caméras de détection à faible coût, il est possible d'obtenir une représentation de la main sous la forme d'un squelette (Fig. 2). Récemment, Son *et al*. [1] ont proposé une méthode par apprentissage profond de matrices symétriques définies positives (SDP) pour la reconnaissance de gestes. Ces matrices ont la capacité d'encoder des relations non linéaires entre les caractéristiques d'entrée et sont adaptées pour le traitement de données non Euclidiennes telles que des squelettes. Si la méthode [1] permet d'encoder explicitement les relations d'ordre supérieur entre les caractéristiques d'entrée, la complexité de l'approche la rend difficilement envisageable pour des applications en temps réel.

Dans cet article, nous proposons une architecture profonde pour la reconnaissance des gestes de la main en temps réel à partir de séquences de squelettes 3D de la main [2]. Nous proposons tout d'abord un réseau neuronal profond efficace basé sur les matrices SDP afin d'obtenir une description riche de chaque sous-séquence d'un geste. Ces matrices SDP extraient des informations temporelles et locales sur le mouvement et les trajectoires des articulations de la main sur un segment de temps autour des poses 3D. Ce réseau est ensuite combiné à une nouvelle architecture afin d'effectuer une reconnaissance des gestes en ligne.

2 Réseau SDP profond

Nous résumons notre réseau (nommé Deep SDPNet) dans la Figure 1. Manipulant des matrices SDP, il exploite des ingrédients issus de [1, 3]. Les données représentant une main sont encodées sous la forme d'un graphe contenant 22 articulations (Fig. 2) auxquelles sont associées des coordonnées $(x, y, z)^T$. Nous représentons cela sous la forme d'une grille 2D de taille 4×5 à partir des articulations 1 à 20 (4 articulations pour chaque doigt). Nous disposons de séquences vidéos de gestes et donc à chaque frame correspondra une grille.

2.1 Couche de convolution :

La première couche permet de débuter par des convolutions 2D comme dans un CNN classique. La couche de convolution reçoit en entrée une grille 2D représentant la main. Chaque élément i de la grille a au plus 9 voisins dont lui-même, désignés par $\mathcal{N}_i$. Partant de $p_j^t \in \mathbb{R}^3$, les coordonnées 3D de l'articulation j à la frame t, des convolutions de coefficients W sont appliquées afin d'obtenir $X_i^t \in \mathbb{R}^{d_{out}}$ avec d_{out} le nombre de convolutions et $l(i,j)$ l'index du voisin j de i :

$$X_i^t = \sum_{j \in \mathcal{N}_i} W_{l(i,j)}^i p_j^t \tag{1}$$

2.2 Traitement de chaque doigt :

Chaque séquence est ensuite décomposée en 6 sous-séquences s pour chaque doigt f. La première sous-séquence est la séquence originale. Les autres correspondent à un découpage de la première en deux et trois sous-séquences de mêmes longueurs. Cela permet de disposer d'une pyramide de séquences à différentes résolutions afin de capturer les variations à différentes échelles. Le rectangle gris de la Figure 1 présente les traitement effectués sur une sous séquence. Nous les détaillons dans la suite.

Couche de mapping Gaussien (Gmap) : Pour caractériser les variations temporelles de chaque articulation j parmi les articulations $\mathcal{J}_f$ du doigt f dans une frame d'une sous-séquence, nous divisons ces dernières en N sous séquences de même longueur. Soient $t_{b,sb}^s$ et $t_{e,sb}^s$ les frames de début et de fin

*Ce travail a bénéficié d'un financement de l'union Européenne (FEDER) et de la région Normandie à travers le projet IGIL (Intuitive Gesture Lab).

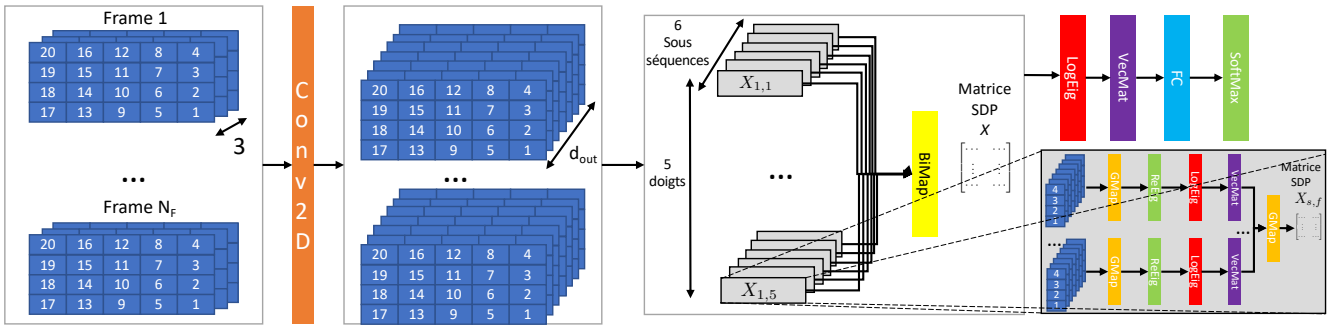

FIGURE 1 – Le réseau Deep SDPNet.

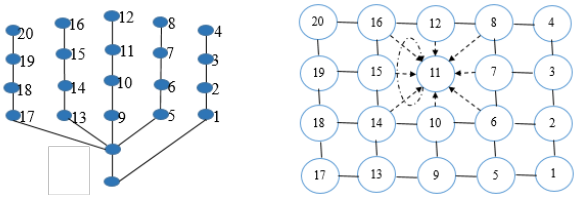

FIGURE 2 – (gauche) graphe des articulations de la main, (droite) représentation sous forme d'une grille 2D.

d'une sous-séquence $sb \in \{1, ..., N\}$. Les variations temporelles d'une articulation j à la frame sb dans une séquence s sont définies par un mapping Gaussien qui produit une matrice SDP définie par

$$X_{s,j}^{sb} = \begin{bmatrix} \Sigma_{s,j}^{sb} + \mu_{s,j}^{sb}(\mu_{s,j}^{sb})^T & \mu_{s,j}^{sb} \\ (\mu_{s,j}^{sb})^T & 1 \end{bmatrix} \quad (2)$$

avec les moyennes et covariance de la distribution Gaussienne obtenues comme suit : $\mu_{s,j}^{sb} = \frac{1}{t_{e,sb}^s - t_{b,sb}^s + 1} \sum_{t=t_{b,sb}^s}^{t_{e,sb}^s} X_{s,j}^t$, et $\Sigma_{s,j}^{sb} = \frac{1}{t_{e,sb}^s - t_{b,sb}^s + 1} \sum_{t=t_{b,sb}^s}^{t_{e,sb}^s} (X_{s,j}^t - \mu_{s,j}^{sb})(X_{s,j}^t - \mu_{s,j}^{sb})^T$ avec $X_{s,j}^t$ les coordonnées de X_j^t pour l'articulation $j \in \mathcal{J}_f$ à la frame t dans la sous séquence s

Couche rectifiant les valeurs propres (ReEig) : L'étape suivante utilise une couche ReEig pour appliquer des transformations non linéaires aux matrices SDP données par

$$X_{s,j,k}^{1,sb} = f_r(X_{s,j}^{sb}) = U_{k-1} M U_{k-1}^T \quad (3)$$

avec f_r la fonction ReEig et $X_{s,j}^{1,sb}$ la matrice SDP de sortie. M est une matrice diagonale

$$M(i,i) = \begin{cases} V_{k-1}(i,i), & V_{k-1}(i,i) > \epsilon, \\ \epsilon, & V_{k-1}(i,i) \leq \epsilon. \end{cases} \quad (4)$$

avec ϵ un seuil de rectification, $X_{s,j}^{sb} = U_{k-1} V_{k-1} U_{k-1}^T$ la décomposition en vecteurs propres de $X_{s,j}^{sb}$. Reig empêche les matrices SDP d'avoir des valeurs négatives ou nulles en plus d'ajuster leurs petites valeurs propres positives.

Couche logarithmique de valeurs propres (LogEig) : La couche suivante projète les matrices SDP rectifiées dans un espace Euclidien et est définie par

$$X_{s,j}^{2,sb} = f_l(X_{s,j}^{1,sb}) = log(X_{s,j}^{1,sb}) = U_{k-1} log(V_{k-1}) U_{k-1}^T \quad (5)$$

avec f_l la fonction LogEig, $X_{s,j}^{1,sb} = U_{k-1} V_{k-1} U_{k-1}^T$ une décomposition en vecteur propres et $log(V_{k-1})$ une matrice diagonale.

Couche de vectorisation (VecMat) : La couche suivante vectorise les matrices SDP en appliquant une transformation linéaire pour les convertir en vecteurs comme suit :

$$\begin{aligned} x_{s,j}^{sb} = f_v(X_{s,j}^{2,sb}) = [X_{s,j}^{2,sb}(1,1), \sqrt{2} X_{s,j}^{2,sb}(1,2) \\ , ..., \sqrt{2} X_{s,j}^{2,sb}(1,d_{out}), X_{s,j}^{2,sb}(2,2), \sqrt{2} X_{s,j}^{2,sb}(2,3) \\ , ..., X_{s,j}^{2,sb}(d_{out}, d_{out})]^T \end{aligned} \quad (6)$$

avec f_v la fonction VecMat, $X_{s,j}^{2,sb}(i,i)$ et $X_{s,j}^{2,sb}(i,j), i < j$ les entrées diagonales et la partie supérieure de $X_{s,j}^{2,sb}$.

Couche de Mapping Gaussien : Après avoir appliqué la couche VecMat, nous obtenons un ensemble de vecteurs $\{x_{s,j}^{sb}\}_{j \in \mathcal{J}_f}^{sb=1,...N}$ caractérisant la variation temporelle de chaque articulation sur chaque frame de s. Afin d'obtenir une caractérisation globale du doigt f le long de la sous-séquence s, nous agrégeons ces vecteurs en matrices SDP en utilisant l'opérateur Gmap comme suit :

$$X_{s,f} = \begin{bmatrix} \Sigma_{s,f} + \mu_{s,f}(\mu_{s,f})^T & \mu_{s,f} \\ (\mu_{s,f})^T & 1 \end{bmatrix} \quad (7)$$

avec les moyennes $\mu_{s,f} = \frac{1}{N|\mathcal{J}_f|} \sum_{j \in \mathcal{J}_f} \sum_{sb=1}^{N} x_{s,j}^{sb}$ et covariance $\Sigma_{s,f} = \frac{1}{N|\mathcal{J}_f|} \sum_{j \in \mathcal{J}_f} \sum_{sb=1}^{N} (x_{s,j}^{sb} - \mu_{s,f})(x_{s,j}^{sb} - \mu_{s,f})^T$.

2.3 Couche de mapping bilinéaire (BiMap) :

Pour chaque doigt f nous avons obtenu un ensemble de matrices SDP qui encodent les variations dans chaque sous séquence s. Nous les agrégeons en une seule matrice SDP à l'aide de poids afin de privilégier les matrices les plus pertinentes pour la reconnaissance du geste. Afin de simplifier les notations, nous désignons $(X_{s,f})_{(s,f) \in \{1,...,6\} \times \{1,...,5\}}$ par $(X_i)_{i \in \{1,...,N\}}$ avec $N = 30$. La couche BiMap transforme les matrices SDP en une seule matrice SDP par $X = \sum_{i=1}^{N} W_i X_i W_i^T$. La couche BiMap peut être vue comme un processus d'attention pour des matrices SDP qui garantit que la matrice résultante est SDP. Celle-ci est ensuite transformée en vecteur par deux couches LogEig et VecMat. Le vecteur obtenu est la représentation du geste et celle-ci peut être classée par une couche FC et un SoftMax.

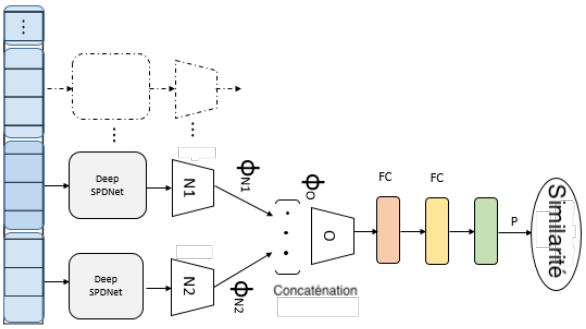

FIGURE 3 – Une représentation de clips de 10 frames est obtenue par deux Deep SDPNet et leur similarité est mesurée par un PMC nommé RDT.

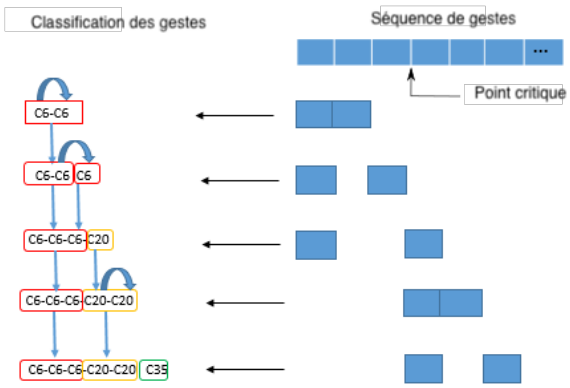

FIGURE 4 – Reconnaissance en ligne de gestes : deux réseaux Deep SDPNet se déplacent temporellement le long d'une séquence de geste et les points de rupture de gestes sont localisés. CX signifie classe X.

3 Reconnaissance des gestes en ligne

Le reconnaissance de gestes en ligne est difficile car i) le flux d'entrée ne contient aucune indication sur le début et la fin de chaque geste, ii) l'architecture doit être efficace, iii) certains gestes peuvent ne pas faire partie de la liste des gestes à reconnaître. Pour résoudre ces problèmes, nous combinons deux réseaux Deep SDPNet (présenté dans la section précédente), comme illustré dans la Figure 3. La combinaison des deux représentations vectorielles produites par les deux réseaux (via leur couche BiMap) est effectuée par un perceptron multi-couche que nous nommons RDT pour Réseau de Détection Temporelle. Chaque Deep SDPNet reçoit en entrée des séquences de 10 frames que nous nommons des clips. Puisqu'un même geste se retrouve dans plusieurs clips consécutifs, le réseau RDT a pour objectif d'estimer la similarité entre les deux clips. Afin de détecter quand un geste se termine, nous cherchons à localiser un point de rupture dans les clips. Pour cela, le premier Deep SDPNet a une position temporelle fixe alors que le second se déplace temporellement sur le flux de gestes. La similarité des deux représentations vectorielles des deux Deep SDPNet est estimée par le réseau des RDT et cela permet d'estimer si les deux clips correspondent au même geste. Dès que le réseau RDT détecte un nouveau geste, un point de rupture est détecté, ce qui signifie que le geste actuel s'est terminé pour laisser la place au geste suivant. Le premier réseau Deep SDPNet est alors déplacé à la position du clip de ce point de rupture et le second au clip suivant. Le processus est répété et cela permet de détecter en ligne les gestes de la main. La figure 4 illustre ce processus. Le premier réseau est placé au premier clip et le second au deuxième clip. Le second réseau se déplace sur les clips suivants jusqu'à ce qu'une rupture de geste soit détectée au quatrième clip. Le premier réseau se déplace alors au quatrième clip et le second au cinquième et le processus continue.

Pour une reconnaissance hors ligne, notre réseau Deep SDPNet peut être entrainé hors ligne sur des séquences décrivant des gestes complets avec des séquences de 500 frames (comme dans [1]). Cependant pour une reconnaissance en ligne des gestes, le temps d'inférence est un élément clé et nous procédons différemment. Tout d'abord notre réseau Deep SDPNet est entrainé sur des clips de 10 frames. Cela permet d'obtenir une représentation vectorielle pour ces clips. Le réseau RDT est ensuite entrainé avec différentes combinaisons de paires de clips positifs ou négatifs. Une paire positive désigne deux clips contenant le même geste et une paire négative deux clips ne contenant pas le même geste. Ces paires de clips sont choisies aléatoirement dans la base de gestes afin d'obtenir une base d'apprentissage balancée et de taille raisonnable. Dans la suite nous mesurerons les performances de notre architecture pour une reconnaissance en ligne et hors ligne des gestes non seulement sur leur qualité de prédiction mais également sur leur temps d'inférence qui est essentiel pour une classification des gestes en quasi temps réel.

4 Résultats

Nous menons nos expérience avec la base First-Person Hand Action (FPHA) [4]. Elle contient 1175 vidéos de 45 différents gestes de la main réalisés par 6 personnes. Elle est divisé en 600 séquences pour l'entraînement et 575 pour le test. Pour chaque vidéo les coordonnées 3D $(x, y, z)^T$ de 21 articulations de la main sont fournies. Nous prenons les même hyper-paramètres que [3] : taux d'apprentissage de 10^{-2}, taille de batch de 30, les poids de la couche BiMap sont de taille 200×56 et fixés par initialisation aléatoire semi-orthogonale, le seuil de rectification ϵ est fixé à 10^{-4}. Le nombre de convolutions est fixé à 9. La Table 1 présente les performances de notre réseau Deep SDPNet par rapport à l'état de l'art. Notre approche obtient de meilleurs résultats que la majorité de l'état de l'art et est moins performante que [1]. Cependant notre réseau Deep SDPNet est spécifiquement conçu pour la reconnaissance en ligne des gestes, ce que ne peut pas effectuer [1].

Pour évaluer les performances de notre approche pour la reconnaissance en ligne de gestes, nous avons créé 5 bases de longueurs différentes, chacune comporte des séquences de 10

M Learning [3]	84.35
Gram Matrix [5]	85.39
Two-stream NN [6]	90.26
Deep SDPNet	90.96
ST-TS-HGR-NET [1]	**93.22**

TABLE 1 – Comparaison des performances de notre approche avec l'état de l'art.

à 160 clips successifs. Le réseau RDT effectue une prédiction pour détecter si les caractéristiques de deux clips correspondent ou non à un même geste. S'ils appartiennent à un même geste, alors RDT exploite le premier Deep SDPNet pour classer le premier clip, et puisque les deux clips sont considérés comme similaires, nous supposons qu'ils correspondent à un même geste. Afin de valider cette hypothèse, nous évaluons dans la table 2 les performances de la classification binaire réalisée par le réseau RDT sur des paires de clips. L'ensemble des paires de clips est divisé en 3 sous-ensembles : un ensemble d'apprentissage avec 64% des données, un ensemble de validation avec 20% des données, un ensemble de test avec 16% des données. La précision obtenue sur l'ensemble de test est de 98%, ce qui confirme notre hypothèse. Cette mise en correspondance des deux clips sur un même geste permet d'éviter le temps d'inférence nécessaire à la reconnaissance de la classe du second clip, ce qui permet de fortement réduire le temps d'exécution. De plus, le réseau RDT mesure la similarité de deux clips en seulement 0,03 secondes. Dans la table 3, nous mesurons la

Base	Précision (%)
Entrainement	98.57
Validation	98.22
Test	98

TABLE 2 – Performances du réseau RDT pour prédire si deux clips sont similaires.

performance du processus proposé pour détecter et reconnaître des séquences de clips issues du jeu de données FPHA. Tout d'abord, nous mesurons l'effet du nombre de clips par séquence générée. Comme le montre la table 3, l'ajout de clips supplémentaires ne nuit pas fortement aux performances. Pour 10 C/S (nombre de clips par séquence), les performances du processus sont de 91,28%, et lorsque le nombre de gestes est multiplié par 16 (c'est-à-dire 160 C/S), les performances atteignent 88,35%, soit une baisse de seulement 2,93%, ce qui est acceptable pour l'application considérée. Deuxièmement, nous avons mesuré les effets du nombre de séquences sur le processus. Comme le montre la Table 3, doubler le nombre de séquences augmente les performances de 1% à 2%. Afin de modéliser les différentes positions et mouvements de la main pendant les phases qui se situent avant et après le geste, nous ajoutons aléatoirement un pourcentage de bruit aux séquences générées. Pour ce faire, nous insérons aléatoirement un pourcentage donné de gestes aléatoires dans toutes les séquences de test. Les coordonnées des articulations de ces gestes insérés

# de séquences	10 C/S	20 C/S	40 C/S	80 C/S	160 C/S
1000	91.28(±0.77)	89.7(±1.06)	88.11(±0.83)	87.58(±1.43)	88.35(±0.96)
2000	91.71(±0.65)	90.44(±1.4)	89.78(±1.5)	89.34(±1.56)	89.09(±1.51)

TABLE 3 – Précision moyenne et écart type du processus proposé pour détecter et reconnaître les séquences de clips. Les résultats sont obtenus sur 10 répétitions avec 600 clips.

sont générées dans l'échelle des valeurs des séquences originales. La table 4 montre les performances de notre processus proposé en présence de bruit. Nous avons fixé le nombre de clips à 1175 (c'est-à-dire toutes les données de l'ensemble de données FPHA) et le nombre de séquences à 2000. Le pourcentage de gestes aléatoires est progressivement augmenté de 10 à 20%. Les résultats, résumés dans la table 4, montrent que pour 10 C/S, l'ajout de 10% de bruit au nombre fixe de séquences diminue les performances de 17% par rapport aux meilleurs résultats (c'est-à-dire 91,71%). Avec 20% de bruit les performances sont impactées.

Bruit (%)	10 C/S	20 C/S	40 C/S	80 C/S	160 C/S
10	74.18(±2.39)	69.84(±5.23)	63.68(±9.89)	55.8(±16.28)	48.76(±20.65)
20	64.57(±1.84)	57.59(±7.45)	49.63(±12.93)	41.39(±18.21)	34.37(±21.53)

TABLE 4 – Précision moyenne et écart-type du processus proposé en présence de bruit. Les résultats sont obtenus sur 10 répétitions avec 1175 clips et 2000 séquences.

5 Conclusion

Nous avons proposé un réseau profond de matrices symétriques définies positives pour la reconnaissance en temps réel des gestes de la main en 3D. Un réseau SDP permet tout d'abord d'extraire des caractéristiques des gestes qui sont fournies à un réseau temporel de détection des gestes. Le processus proposé prend en charge la reconnaissance des gestes en temps réel ainsi que leur détection précoce.

Références

[1] X. S. Nguyen, L. Brun, O. Lezoray, and S. Bougleux, "A neural network based on SPD manifold learning for skeleton-based hand gesture recognition," in *IEEE CVPR*, 2019, pp. 12 036–12 045.

[2] M. B. Naceur, L. Brun, and O. Lézoray, "Lightweight deep spd manifold network for real-time 3d hand gesture recognition," in *International Conference on Automatic Face and Gesture Recognition (FG - IEEE)*, vol. to appear, 2021.

[3] Z. Huang and L. V. Gool, "A riemannian network for SPD matrix learning," in *AAAI*, 2017, pp. 2036–2042.

[4] G. Garcia-Hernando, S. Yuan, S. Baek, and T. Kim, "First-person hand action benchmark with RGB-D videos and 3d hand pose annotations," in *IEEE CVPR*, 2018, pp. 409–419.

[5] X. Zhang, Y. Wang, M. Gou, M. Sznaier, and O. I. Camps, "Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold," in *IEEE CVPR*, 2016, pp. 4498–4507.

[6] C. Li, S. Li, Y. Gao, X. Zhang, and W. Li, "A two-stream neural network for pose-based hand gesture recognition," *CoRR*, vol. abs/2101.08926, 2021.