Classification de gestes avec le radar Soli

Ali TOUALBI
alitoualbi@gmail.com
Encadré par : olivier schwander et Sylvain Lamprier

1 Introduction

La reconnaissance des gestes est un thème largement traité , l'intérêt pour ce sujet vient de son importance dans les interactions humaines. Le radar Soli est un radar miniature conçu pour la réalisation d'interfaces homme-machine, il est capable d'extraire des informations de vitesse à très haute fréquence (60 GHz) et donc d'obtenir des données avec une résolution suffisante pour analyser des gestes. L'objectif de ce projet est la reconnaissance de gestes à l'aide de la classification supervisée en exploitant plusieurs architectures CNN, RNN ainsi qu' architecture hybride sur un ensemble de données fourni par les auteurs de [1]

2 Contexte et état de l'art

Les recherches effectuées dans le domaine de l'Interaction Homme Machine visent à améliorer les performances des machines par l'évolution des techniques de détection ainsi que les algorithmes de reconnaissance.

2.1 La détection du geste

Les modalités de détection pour la reconnaissance de geste a connu une évolution très impressionnante, certaines approches de l'état de l'art utilisent de la 2D [2] d'autres de la 3D [3], néanmoins ces pratiques ne sont pas très précises du à la difficultés d'imiter les capacités humaines à comprendre un geste effectué par une personne qu'on a jamais connu auparavant. Les auteurs de [4] proposent une structuration du geste en quatres phases afin de mieux comprendre son codage, mais réussir à déterminer les instants exacts où démarrent chacune des phases restent assez délicat. [1] pallient à ces problèmes en exploitant les ondes de fréquences retournées par le radar Soli. La différence majeure de leur approche et celle des recherches se basant sur la détection du signal, est la nature et la fréquence de ce dernier. En effet, les approches existantes reposent sur des signaux à bandes passantes basses (ce qui limite la résolution spatiale) tandis que [1] utilisent de hautes fréquences afin de mieux reconnaître les gestes.

2.2 La reconnaissance des gestes

L'apprentissage des représentations en utilisant des réseaux de neurones profonds est la solution la plus adaptée pour la reconnaissance des gestes, en effet, elle a fait ses preuves dans la reconnaissance vocale [5] ou même en Music Information Retrieval [6]. L'une des premières applications des CNNs est la reconnaissance de l'écriture manuscrite [7], mais ils ont connu rapidement un champ d'applications plus élargis. [8] ont montré que les CNNs sont puissants pour la reconnaissance du langage des signes, tandis que [9] mettent en évidence leur importance dans l'analyse vidéo.
Une autre architecture largement utilisée est celle des Réseaux de neurones récurrents, leur but est de prédire ou de classifier des séquences temporelles. Parmi leurs nombreuses applications, nous pouvons citer la modélisation du langage[10], la traducation automatique [11] et l'analyse vidéo [12].
Dans une approche similaire que [1] nous proposons une architecture hypride combinant un CNN et un RNN, après plusieurs expériences que nous allons détailler par la suite.

3 Approches

Nous proposons de comparer les résultats obtenus d'un réseau neuronal convolutif, un réseau de neurones récurrent (plus précisément un réseau GRU) et une architecture hybride combinant les deux.

3.1 Apprentissage de la représentation

La première étape de tout modèle d'apprentissage est l'extraction des caractéristiques, en vision par ordinateur, cette étape est primordiale car elle permet de mettre en évidence plusieurs propriétés visuelles de l'image. Les CNNs répondent le mieux à cette problématique et sont largement utilisées dans les approches de l'état de l'art.

3.2 Reconnaissance des gestes

Plusieurs approches peuvent être utilisées pour modéliser les processus dynamiques, nous avons choisis d'exploiter les réseaux neuronaux récurrents, plus précisément les Gated Reccurent Unit qui est une variante des LSTM introduite en 2014. Le GRU possède deux " portes ", qui sont des paires d'un vecteur et d'une matrice : la porte de mise à jour et la porte de réinitialisation.

4 Travail expérimental

4.1 Dataset

Nous avons utilisé le dataset proposé par [1] où 10 sujets ont effectué 11 gestes qui ont été capturés 25 fois. Nous avons donc au total 11 x 25 x 10 = 2750 séquences en tout. Pour la répartition du train et le test set, nous avons effectué un split 50-50. s

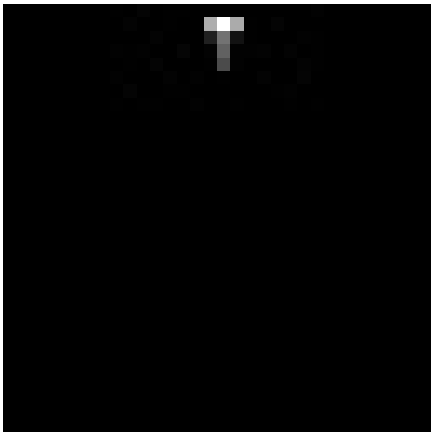

FIGURE 1 – Exemple d'une image

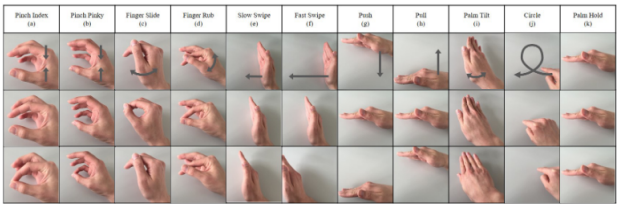

FIGURE 2 – Les gestes utilisés pour les expériences

4.2 Conditions d'expérimentation

CNN. Dans un premier lieu, nous avons pris en compte un CNN seul comme base de comparaison pour mettre en évidence l'importance de la représentation des caractéristiques.
Premièrement, comme mesure d'évaluation nous avons classifié chaque channel de chaque séquence de chaque geste individuellement pour avoir :

$$nbr\_prediction = 4{*}nbr\_exemple{*}len(sequence) \quad (1)$$

. Cette approche naïve a été considérée comme une tentative d'augmentation de données, afin d'avoir un nombre considérable de données d'entrainement.
Puis, nous avons regroupé tous les channels de chaque séquence pour chaque exemple pour avoir un total de nombre de prédictions égale à :

$$nbr\_prediction = nbr\_exemple * len(sequence) \quad (2)$$

Pour finir, nous avons établi un système de vote sur la deuxième mesure d'évaluation pour avoir une seule prédiction pour chaque exemple, et donc :

$$nbr\_prediction = nbr\_exemple \quad (3)$$

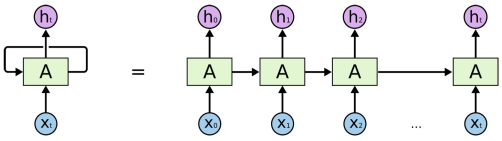

FIGURE 3 – Architecture d'un RNN

RNN. Pour évaluer l'impact de la cohérence temporelle, nous avons pris en considération un RNN seul pour l'extraction et l'évaluation des caractéristiques. L'objectif d'un RNN est de prédire l'élément suivant d'une séquence, qui dépend des éléments précédents et d'un état mémoire latent. Le réseau prédit pour chaque passage le dernier état latent. On aura donc Nombre d'exemples de prédictions.

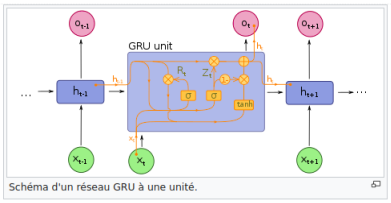

FIGURE 4 – Architecture d'un GRU

CNN + RNN. Enfin, nous avons utilisé une pipeline où le CNN et le RNN ont été entrainé conjointement, le RNN prend en entrée les caractéristiques extraites grâce au CNN.

4.3 Perte

La perte d'entropie croisée, ou perte logarithmique, mesure les performances d'un modèle de classification dont la sortie est une valeur comprise entre 0 et 1. La perte d'entropie croisée augmente à mesure que la probabilité prédite diverge de l'étiquette réelle. Ainsi, prédire une probabilité de 0,012 alors que l'étiquette de l'observation réelle est 1 serait mauvais et entraînerait une perte élevée. Un modèle parfait aurait une perte logarithmique de 0.[13]

4.4 Détails d'implémentation

Dataset. Comme pré-traitement, nous avons organisé tout le dataset sous une forme hiérarchique afin de faciliter son utilisation.

/geste1
../exemple1
../ch1-seq1
../ch2-seq1
../chan4-seqn
../exemple25
../ch1-seq1
../ch2-seq1
../chan4-seqn

/geste11
../exemple1
../ch1-seq1
../ch2-seq1
../chan4-seqn
../exemple25
../ch1-seq1
../ch2-seq1
../chan4-seqn

Nous avons implémenté un sampler adapté à notre dataset afin d'accéder à nos données à travers un dataloader.

Architectures.

Ls	CNN	RNN	CNN+RNN
1	conv1 3×3×32-pool1 2×2	fc1 512	conv1 3x3x32
2	conv2 3×3×64-pool2 2×2	gru2 512	conv2 3x3x64
3	conv3 3×3×128-pool3 2×2	fc3 - softmax 11	conv3 3x3x128
4	fc4 512	-	fc4 512
5	fc5 512	-	fc5 512
6	fc6 - softmax 11	-	gru6 512
7	-	-	fc7 - softmax 11

TABLE 1 – Architectures

4.5 Résultats et discussion

CNN. Les figures 5, 6 et 7 montrent nos résultats d'évaluation pour les expériences menées sur un réseau neuronal convolutif.
Nous remarquons que le CNN naïf arrive à une accuracy de 57.5%, le CNN mesure 1 a une accuracy de 72.5% tandis que le CNN mesure 2 arrive à 88% comme accuracy. Nous pouvons donc conclure que le système de vote établi sur la deuxième mesure d'évaluation est la solution la plus adaptée à cette problématique car un geste est constitué de plusieurs séquences, les traiter séparément va biaiser la véracité du geste, de plus chaque channel capté apporte une information importante pour discerner le geste, on ne peut donc pas traiter channel par channel.

RNN. Comme le résume la figure 8, le RNN seul a 96.18% comme score (après 5 epochs), cette différence de score avec le CNN est dû à la définition même d'un geste, un geste est enchainement de séquences la temporalité et l'ordre de ces dernières est ce qui rend un geste unique.

CNN + RNN. Comme affiché dans la figure 9 l'architecture hybride arrive à un score de 96.14%. On remarque que le CNN+RNN apprend lentement mais d'une manière stable ce qui garantit l'obtention de résultats concrets après plusieurs epochs.

Mesure	CNN	RNN	CNN+RNN
Accuracy	84.78%	96.18%	96.14%

TABLE 2 – Résultats de comparaison

Références

[1] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. Interacting with soli : Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, page 851–860, New York, NY, USA, 2016. Association for Computing Machinery.

[2] Ali Erol, George Bebis, Mircea Nicolescu, Richard Boyle, and Xander Twombly. Vision-based hand pose estimation : A review. *Computer Vision and Image Understanding*, 108 :52–73, 10 2007.

[3] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. Digits : Freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, page 167–176, New York, NY, USA, 2012. Association for Computing Machinery.

[4] R. Chellali, I. Renna, E. Bernier, and Cyrille Achard. Détection et Reconnaissance des Gestes Emblématiques. In *RFIA 2012 (Reconnaissance des Formes et Intelligence Artificielle)*, pages 978–2–9539515–2–3, Lyon, France, January 2012. Session "Atelier IHMA".

[5] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition : The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6) :82–97, 2012.

[6] Keunwoo Choi, György Fazekas, Kyunghyun Cho, and Mark B. Sandler. A tutorial on deep learning for music information retrieval. *CoRR*, abs/1709.04396, 2017.

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998.

[8] Natalia Neverova, Christian Wolf, Graham W. Taylor, and Florian Nebout. Multi-scale deep learning for gesture detection and localization. In Lourdes Agapito, Michael M. Bronstein, and Carsten Rother, editors, *Computer Vision - ECCV 2014 Workshops*, pages 474–490, Cham, 2015. Springer International Publishing.

[9] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[10] Tomas Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, 2010.

[11] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[12] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, abs/1411.4389, 2014.

[13] Cross entropy.

5 Annexe

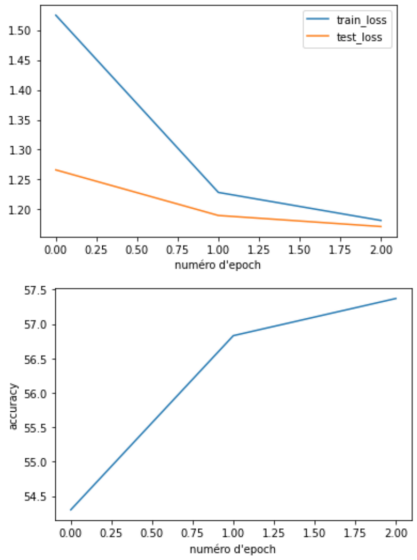

FIGURE 5 – CNN naïf

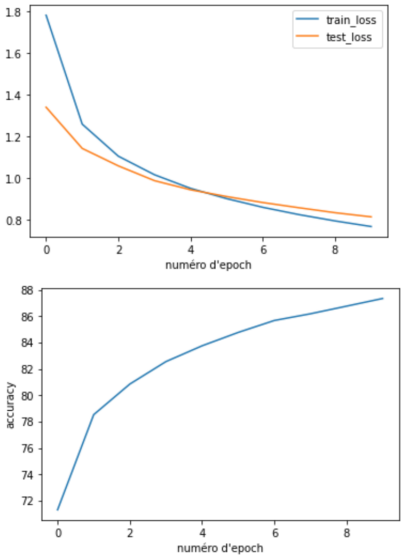

FIGURE 7 – CNN mesure 2

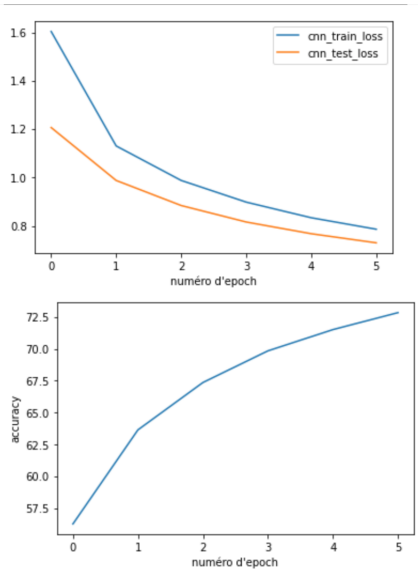

FIGURE 6 – CNN mesure 1

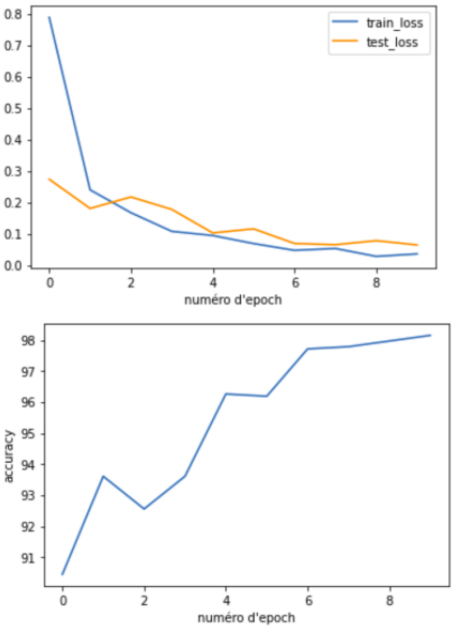

FIGURE 8 – RNN

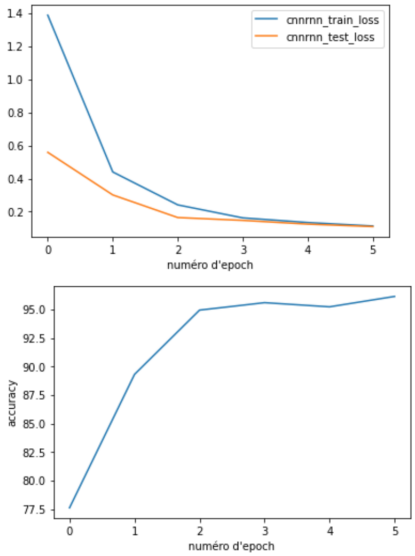

FIGURE 9 – CNN+RNN