THÈSE

présentée pour obtenir

le titre de docteur délivré par Sorbonne Université

École doctorale: Science Mécanique, Acoustique, Électronique et Robotique de Paris

par

Mégane MILLAN

L'Apprentissage Profond pour l'Évaluation et le Retour d'Information lors de l'Apprentissage de Gestes

---

Mathias QUOY	Professeur des Universités à l'Université de Cergy Pontoise	Président du Jury
Fabienne POREE	Maîtresse de Conférence à l'Université Rennes 1 - HDR	Rapporteur
Jean-Philippe VANDEBORRE	Professeur à l'IMT Lille-Douai	Rapporteur
Bertrand LUVISON	Ingénieur de Recherche au CEA-LIST Saclay	Examinateur
Hichem SAHBI	Chargé de Recherche à Sorbonne Université	Examinateur
Catherine ACHARD	Professeure des Universités à Sorbonne Université	Directrice de thèse

Institut des Systèmes Intelligents et de Robotique (ISIR)
Pyramide Tour 55, 4 place Jussieu
UMR CNRS 7222, Paris, France

Résumé

Apprendre un nouveau sport, ou un métier manuel est complexe. En effet, de nombreux gestes doivent être assimilés afin d'atteindre un bon niveau de compétences. Cependant, l'apprentissage de ces gestes ne peut se faire seul. En effet, il est nécessaire de voir la réalisation du geste d'un œil expert afin d'indiquer les corrections pour s'améliorer. Or les experts, que ce soit en sport ou dans les métiers manuels, sont peu disponibles pour analyser et évaluer les gestes d'un novice.

Afin d'aider les experts dans cette tâche d'analyse, il est possible de développer des coachs virtuels. Selon les domaines, le coach va posséder plus ou moins de compétences, mais une évaluation selon des critères précis est toujours à privilégier. Fournir un retour sur les erreurs commises est également essentiel pour l'apprentissage d'un novice.

Dans cette thèse, différentes solutions pour développer des coachs virtuels les plus efficaces possibles sont proposées.
Dans un premier temps, et comme évoqué précédemment, il est nécessaire d'évaluer les gestes. Dans cette optique, un premier travail a consisté à comprendre les enjeux de l'analyse de gestes automatique, afin de développer un algorithme d'évaluation automatique qui soit le plus performant possible. Par la suite, deux algorithmes d'évaluation automatique de la qualité de gestes sont proposés. Ces deux algorithmes fondés sur l'apprentissage profond, ont par la suite été testés sur deux bases de données de gestes différentes afin d'évaluer leur généricité.

Une fois l'évaluation réalisée, il est nécessaire de fournir un retour d'information pertinent à l'apprenant sur ses erreurs. Afin de garder une continuité dans les travaux réalisés, ce retour est également fondés sur les réseaux de neurones et l'apprentissage profond. En s'inspirant des méthodes d'explicabilité de réseaux de neurones, une méthode a été développée. Elle permet de remonter aux instants du gestes où des erreurs ont été commises selon le modèle d'évaluation. Enfin coupler cette méthode à de la segmentation sémantique, permet d'indiquer aux apprenants quel partie du geste a été mal réalisée, mais également de lui fournir des statistiques et une courbe d'apprentissage.

Mots-clés : Analyse de Gestes, Apprentissage Profond, Évaluation Automatique de la Qualité d'un Geste, Retour d'Information, Segmentation Sémantique

Abstract

Learning a new sport or manual work is complex. Indeed, many gestures have to be assimilated in order to reach a good level of skill. However, learning these gestures cannot be done alone. Indeed, it is necessary to see the gesture execution with an expert eye in order to indicate corrections for improvement. However, experts, whether in sports or in manual works, are not always available to analyze and evaluate a novice's gesture.

In order to help experts in this task of analysis, it is possible to develop virtual coaches. Depending on the field, the virtual coach will have more or less skills, but an evaluation according to precise criteria is always mandatory. Providing feedback on mistakes is also essential for the learning of a novice.

In this thesis, different solutions for developing the most effective virtual coaches are proposed.
First of all, and as mentioned above, it is necessary to evaluate the gestures. From this point of view, a first part consisted in understanding the stakes of automatic gesture analysis, in order to develop an automatic evaluation algorithm that is as efficient as possible. Subsequently, two algorithms for automatic quality evaluation are proposed. These two algorithms, based on deep learning, were then tested on two different gestures databases in order to evaluate their genericity.

Once the evaluation has been carried out, it is necessary to provide relevant feedback to the learner on his errors. In order to maintain continuity in the work carried out, this feedback is also based on neural networks and deep learning. A method has been developed based on neural network explanability methods. It allows to go back to the moments of the gestures when errors were made according to the evaluation model. Finally, coupled with semantic segmentation, this method makes it possible to indicate to learners which part of the gesture was badly performed, and to provide them with statistics and a learning curve.

Keywords : Gesture Analysis, Deep Learning, Automatic Quality Assessment, Feedback, Semantic Segmentation

Table des matières

Liste des Figures

Liste des Tables

Chapitre 1

Introduction Générale

1.1 Contexte

L'analyse de gestes existe depuis toujours. Nécessaire pour les interactions sociales, afin de pouvoir décoder les micro-gestes de communication non-verbales, c'est une tâche que nous réalisons inconsciemment. De plus, après plusieurs années à pratiquer un sport ou une activité manuelle, il est également facile à un œil avisé de reconnaître les gestes d'un novice et de prodiguer des conseils afin que cette personne puisse s'améliorer.

Cependant, analyser ses propres gestes reste complexe. En effet, dans le cadre d'apprentissage d'un nouveau geste, il est nécessaire de voir la réalisation d'un œil extérieur afin de pouvoir déceler les erreurs. Or, il est impossible de se voir soi-même de manière globale, il est donc difficile de faire une auto-évaluation. Même en regardant un film de sa propre réalisation, déceler des erreurs est complexe et fournir une correction l'est encore plus. Enfin comme il a été dit précédemment, il est nécessaire d'avoir des années d'expérience avant de pouvoir noter les erreurs commises. En effet elles sont souvent complexes à percevoir, de par leur aspect spatial et temporel (savoir quelle partie du corps est mal placée, ou mal synchronisée avec le reste du corps...). Par exemple, le service au tennis, est un geste qui peut paraître simple lorsque des experts le réalisent, mais qui demande en réalité une synchronie parfaite entre le haut et le bas du corps afin de pouvoir réaliser un service puissant et précis. Ainsi, même si l'auto-évaluation à partir d'une vidéo est difficile, l'analyse de cette vidéo par un expert est possible. Cependant, avoir le retour d'un expert dans le domaine est parfois impossible :ils sont souvent très

demandés et ont donc très peu de temps à accorder à l'évaluation et à l'apprentissage des novices, que ce soit dans le domaine médical, sportif ou encore artistique (poterie, sculpture).

Pour soulager les experts de cette tâche d'évaluation, il est intéressant de développer des coachs virtuels. Les besoins diffèrent selon les domaines, mais il faut au minimum être capable de fournir un score selon des critères et une échelle prédéfinies. Ensuite, être capable de fournir un retour d'information, que ce soit *a posteriori* ou en temps réel, permettrait de créer un coach virtuel qui aurait le même impact qu'un expert dans l'apprentissage.
Pour créer ce coach virtuel, il est donc nécessaire de développer des méthodes d'évaluation automatique de la qualité d'un geste et également des méthodes fournissent un retour d'information.

1.2 Problématique

Depuis quelques années, l'analyse de geste tend à s'automatiser. En effet, l'évolution et le développement de nouveaux systèmes de capture de mouvement, tels que la Kinect® (Microsoft Corporation, Washington, États-Unis) ou des systèmes moins portables comme l'Optitrack® (NaturalPoint, Inc., Corvallis, États-Unis), ont permis de récolter beaucoup de données de gestes de bonne qualité. En parallèle de l'évolution des systèmes de capture de mouvement, l'apprentissage statistique a lui aussi bénéficié d'avancées matérielles, permettant de développer des modèles qui donnent, d'année en année, de meilleurs résultats.

L'analyse de gestes automatique s'est ainsi démocratisée et commence à être présente dans certaines applications, comme les systèmes de télésurveillance ou les jeux vidéo. Elle reste cependant un challenge à relever dans de nombreux autres contextes, de par la complexité des signaux acquis. En effet, le même geste peut être réalisé à différentes vitesses, amenant à des signaux de longueurs variables qu'il va falloir comparer. ainsi une grande variabilité temporelle existe. De la même façon, un même geste peut-être réalisé avec différentes amplitudes, que ce soit à cause de l'expressivité de la personne le réalisant ou à cause de sa morphologie. Ceci amène donc également à une grande variabilité spatiale.

Á tout cela s'ajoutent d'autres difficultés inhérentes à l'évaluation de gestes. En effet, certains gestes sont liés à une tâche mais ne demandent pas une trajectoire précise. Par exemple, lors d'une suture, le chirurgien a une grande liberté dans la façon de tenir l'aiguille ou d'approcher de la plaie, sans que cela ait de conséquences sur la qualité du geste. Comment à partir de la trajectoire de l'aiguille déterminer automatiquement si le geste est correct ?

Une autre difficulté lorsque l'on s'intéresse aux gestes réside dans la petite taille des bases de données. En effet, leur acquisition est longue car elle demande de capter complètement le même geste, réalisé plusieurs fois, par plusieurs personnes. Il

est également parfois nécessaire d'équiper les personnes avec des marqueurs, actifs ou non (Optitrack® par exemple), ce qui complexifie l'acquisition. Enfin, de tels systèmes d'acquisition sont lourds à mettre en place et difficilement déployables.

L'annotation des bases de données est également un problème difficile. Évaluer un geste demande d'observer toute sa réalisation. Quelle note donner lorsque le geste débute bien mais finit mal ? Faut-il annoter le geste indépendamment ou non de la performance de la tâche à réaliser lorsqu'il y en a une ? Toutes ces considérations font qu'il existe une grande variabilité dans les annotations, que ce soit inter- ou intra-annotateur. Cette variabilité, qui peut être considérée comme un bruit important, rend la tâche d'évaluation automatique encore plus complexe.

1.3 Contributions

Au cours des travaux de cette thèse, nous avons abordé la problématique globale de l'évaluation automatique de la qualité d'un geste. Ces travaux de recherche ont pu bénéficier de l'expertise de l'équipe PIRoS (Perception, Interaction et Robotique Sociale) de l'Institut des Systèmes Intelligents et de Robotique (ISIR) qui s'intéresse depuis de nombreuses années à l'analyse de gestes, notamment dans le cadre de signaux sociaux [Fang et al., 2016] et plus récemment pour l'évaluation de la qualité des gestes sportifs comme le tsuki au judo ou le service au tennis [Morel et al., 2017a].

L'objectif final de ces travaux de recherche est de fournir une approche de bout en bout, capable de fournir à la fois une note et un retour sur chaque réalisation de geste. L'idée finale est de mettre en place des coachs virtuels aptes à faciliter l'apprentissage des gestes, que ce soit dans le cadre professionnel (chirurgiens, sportifs de haut niveau) ou dans le cadre du loisir (yoga, stretching). Pour parvenir à mettre en place cette solution, plusieurs sous-problèmes ont été adressés, ce qui a amené à plusieurs contributions.

Dans un premier temps, nous nous sommes intéressés à l'évaluation de la qualité d'un geste, *i.e.*, fournir une note, ou un score, à chaque réalisation. Nous nous sommes pour cela tournés vers les réseaux siamois qui, plutôt que de donner une note à chaque réalisation, comparent deux gestes afin de déterminer le meilleur. Intuitivement, il parait en effet plus facile de se focaliser sur les différences entre deux réalisations pour justifier de l'écart de notes. Un inconvénient de ces réseaux est cependant qu'ils n'amènent pas à des notes absolues mais seulement à un classement des gestes, allant du pire au meilleur. Nous avons ainsi proposé deux nouvelles architectures, fondées sur ces réseaux, permettant de remédier à ce problème. Elles permettent, non seulement d'attribuer une note à chaque geste, mais également d'améliorer le classement des gestes selon leur qualité. Ces deux architectures ont été validées sur deux bases de données de la littérature correspondant à des applications très différentes (chirurgie et sport) mais également à

des données très différentes (données cinématiques ou vidéos). Ceci a permis de valider les méthode proposées dont les résultats dépassent ceux de l'état de l'art sur ces deux bases.

Dans un second temps, nous avons travaillé sur le retour d'information à fournir à l'utilisateur. Il s'agit ici de pouvoir expliquer la note donnée et surtout, déterminer les erreurs à corriger. Comme la méthode mise en place pour évaluer le geste est fondée sur les réseaux de neurones, nous nous sommes naturellement tournés vers les méthodes relatives à l'explicabilité des réseaux et plus particulièrement à celles utilisant le gradient estimé lors de la rétropropagation. Ainsi, en utilisant le réseau appris, et en calculant le gradient nécessaire pour obtenir la note maximale, nous pouvons faire évoluer l'entrée de manière à ce que la note attribuée évolue progressivement jusqu'à la note parfaite. Les différences entre l'entrée modifiée et l'entrée initiale permettent alors de remonter aux instants du geste à corriger. Cette approche a été validée sur des gestes de synthèse où la vérité-terrain est connue. Elle est assez générique pour s'appliquer à tout type de réseau convolutionnel et pourra donc être exploitée sur de nombreuses architectures.

Dans une dernière contribution, nous avons proposé de réaliser une segmentation sémantique des gestes. Elle réalise la double tâche de segmenter temporellement le geste et de reconnaître la classe de chaque segment. Dans le cadre des applications liées à la chirurgie, ceci permet de réaliser des statistiques sur l'apprentissage et de voir l'évolution fine des compétences sur chaque partie du geste.

En répondant à ces trois problématiques, de manière générique, il est possible de créer des coachs virtuels permettant de s'entraîner à la réalisation d'un geste, dans divers milieux applicatifs.

1.4 Plan du Document

Ce mémoire de thèse se décompose en 5 chapitres :

- **Chapitre 2** - Ce chapitre présente une vue générale des différents domaines de l'analyse de gestes. Ceci permet de situer l'estimation de la qualité d'un gestes dans ce vaste domaine. Par la suite, une étude des méthodes courantes d'apprentissage statistique est proposée, avec leur force et leur faiblesse.
- **Chapitre 3** - Ce chapitre se concentre sur l'évaluation automatique de la qualité d'un geste. Après une étude bibliographique des approches existantes, nous avons proposé deux architectures, fondées sur les réseaux siamois, permettant de répondre à la problématique. Elles ont été testées et comparées à l'état de l'art, sur deux bases de données de la littérature.
- **Chapitre 4** - Ce chapitre propose une solution pour fournir un retour d'information sur les défauts d'un geste. Elle utilise les principes de l'explicabilité

des réseaux de neurones afin de modifier l'entrée du système de manière à optimiser la note. Afin de tester la méthode proposée, une base de données de synthèse a été créée où les vérité-terrains, non seulement sur les notes, mais également sur les défauts des entrées sont connues. Ceci a permis de valider la méthode proposée quant à son aptitude à trouver les défauts d'une série temporelle.

- **Chapitre 5** - Ce chapitre propose une nouvelle méthode de segmentation temporelle, permettant à la fois de segmenter et d'identifier chaque segment dans une application de chirurgie par laparoscopie où la difficulté principale réside dans le faible nombre d'exemples de la base. Une fois la méthode de segmentation sémantique validée, nous proposons de l'utiliser afin de fournir des statistiques sur l'apprentissage d'un chirurgien et ses compétences dans les différentes tâches.
- **Chapitre 6** - Ce dernier chapitre conclut le manuscrit en rappelant les principales contributions et en présentant des perspectives pour poursuivre ces travaux.

1.5 Publications

Durant cette thèse, plusieurs réponses à des problématiques concernant l'évaluation de geste ont été apportées et ont donné lieu à plusieurs publications et communications :

— M. Millan and C. Achard. *Segmenting Surgical Tasks using Temporal Convolutional Neural Network* SURGETICA - 2019

— M. Millan and C. Achard. *Fine-tuning Siamese Networks to Assess Sport Gestures Quality*. 15^{th} International Conference on Computer Vision Theory and Applications (VISAPP) - 2020.

De plus un autres article est en cours de révisions :

— M. Millan and C. Achard. *Multiple Loss for Gesture Quality Evaluation using Siamese Networks*. Machine, Vision and Application (MVA)

Enfin durant cette thèse, différents colloques internes (Journées des Doctorants 2017/2018/2019 de l'ISIR, Journées de l'Ecole Doctorale SMAER 2017) ont permis de présenter ces travaux sous formes de communications orales. Ces travaux ont également été présentés lors d'une journée du GDR ISIS organisé le 14 Novembre 2019, nommée : "Journée Action, Visage, Geste, Action et Comportement".

Chapitre 2

L'Analyse de Gestes et l'Apprentissage Statistique

Un geste est défini par le Larousse (https://www.larousse.fr/dictionnaires/francais/geste/36848) comme "*un mouvement du corps, principalement de la main, des bras, de la tête, porteur ou non de signification*". On pourra par exemple considérer le geste sportif du boxeur, le coup de pied du footballeur, le geste du boulanger qui pétrit son pain ou encore, les gestes réalisés lors de conversations. De manière inconsciente ou non, nous analysons ces gestes en permanence car ils sont continuellement présents dans notre vie de tous les jours.

Dans ce chapitre, nous allons explorer les différentes branches de l'analyse de gestes et dresser un état de l'art pour chacune d'elles. Nous allons également aborder les différentes méthodes d'apprentissage statistique, avec leurs avantages et leurs inconvénients.

2.1 Analyse de Gestes

Au travers du terme générique "*d'analyse de gestes*" se cachent différents domaines bien distincts avec des applications et des problématiques différentes, *i.e* la synthèse, la segmentation, la reconnaissance ou encore l'évaluation automatique. Nous reprenons ces différents aspects dans les sections suivantes.

2.1.1 La Synthèse de Gestes

L'objectif de la synthèse de gestes est de recréer des gestes à partir de mouvements existants ou de connaissances fournies par un expert. La synthèse est très utilisée dans les jeux vidéo, la médecine, ou encore les films.
Selon les études, la synthèse de gestes va consister à fusionner deux mouvements en les synchronisant [Ménardais et al., 2004], ou bien à retrouver des descripteurs robustes dans le domaine fréquentiel afin de recréer des mouvements réalistes [Unuma et al., 1995]. Cependant, il est aussi important de prendre en compte l'environnement dans lequel les mouvements vont être réalisés, notamment si le but est synthétiser des mouvements pour un robot [Kulpa et al., 2005].

Avec le développement massif, des modèles d'apprentissage statistique, de nouvelles méthodes de synthèse se sont développées [Holden et al., 2016; Zhou et al., 2018]. Par exemple, Zhou *et al.* [Zhou et al., 2019] proposent de fusionner deux mouvements grâce à des réseaux convolutionnels, tout en ajoutant des contraintes sur la taille des membres et sur les articulations afin d'avoir le rendu le plus réaliste possible.

2.1.2 La Reconnaissance de Gestes

Dans le domaine de la sécurité, la reconnaissance d'actions dites suspectes, a été étudiée par Bouma *et al.* [Bouma et al., 2018], afin de protéger des bâtiments ou des zones sensibles. Dans le domaine industriel, Coupeté [Coupeté, 2016], améliore l'interaction homme-robot dans les usines afin de permettre à des ouvriers de travailler sur les lignes de production en toute sécurité. Dans le domaine médical, plus particulièrement la chirurgie, la reconnaissance d'actions est de plus en plus présente [Lea et al., 2016a; Sharghi et al., 2020]. Padoy *et al.*, par exemple, proposent d'analyser les vidéos d'opération afin de reconnaître les actions et d'être capable de créer le bloc opératoire du futur [Padoy et al., 2008]. On peut également citer des applications dans la reconnaissance du langage des signes [Nasreddine and Benzinou, 2016] ou la domotique [de Carvalho Correia et al., 2013].

La reconnaissance d'actions impliquant plusieurs individus est encore peu explorée dans la littérature [Sozykin et al., 2018]. Dans le domaine sportif, Bialkowski *et al.* [Bialkowski et al., 2013], propose une nouvelle représentation des joueurs pour le hockey sur gazon, afin de faciliter l'analyse de vidéos par la suite.

Bien souvent, ces méthodes de reconnaissance nécessitent une segmentation préalable du geste afin d'en connaître le début et la fin [Lei et al., 2016; Mavroudi et al., 2018].

2.1.3 La Segmentation de Flux de Données

La segmentation d'actions peut être définie comme une tentative de prédire, pour une vidéo ou une série temporelle, l'action qui se produit à chaque instant. Elle peut aussi consister à découper une action en plusieurs gestes élémentaires [Nakamura et al., 2017; Caramiaux et al., 2012; Mousas et al., 2014]. Souvent utilisée comme un pré-traitement à la reconnaissance, ou à l'évaluation de gestes, cette tâche reste complexe, notamment lorsque l'on utilise des données "*in the wild*", qui ne présentent pas de pauses entre les gestes ou de retour à un état de repos, mais qui ont l'avantage de bien représenter la réalité. Il est parfois possible de réaliser la segmentation et la reconnaissance en même temps [DiPietro et al., 2016]. On parle alors de segmentation sémantique.

2.1.4 L'Évaluation de la Qualité d'un Geste

L'évaluation de la qualité d'un geste est un autre domaine d'étude qui vise, non pas à reconnaître un geste (il est déjà connu), mais à déterminer s'il a été correctement réalisé ou non. A ce jour, elle a déjà été étudiée dans le domaine sportif [Burns et al., 2011] et dans le domaine médical [Zia et al., 2016; Poddar et al., 2014].
Selon les annotations fournies avec les bases de données, l'évaluation du geste pourra amener à catégoriser le niveau de l'exécutant en « novice », « intermédiaire » ou « expert » (tâche de classification) ou à évaluer finement la qualité du geste au travers de l'attribution d'un score (tâche de régression).

Il est même possible d'aller plus loin et de fournir un retour d'information sur les erreurs commises lors de la réalisation du geste afin d'en aider l'apprentissage. Deux difficultés majeures apparaissant alors : comment déterminer les défauts du geste [Morel et al., 2017a] et comment fournir un retour d'information compréhensible [Ninon and Szewczyk, 2017].

Dans le cadre de cette thèse, nous proposons de nous intéresser à l'évaluation automatique du geste de par l'attribution d'un score à chaque réalisation (Chapitre 3), ainsi qu'à la recherche des défauts dans les gestes (Chapitre 4). Un état de l'art plus conséquent sur ces deux tâches sera donné dans ces chapitres.

2.2 Méthodes d'Apprentissage Statistique Supervisées

Afin de pouvoir analyser un geste, il est fréquent d'utiliser des outils statistiques, et/ou des méthodes utilisant l'apprentissage machine qui consistent à utiliser des approches mathématiques pour permettre aux ordinateurs d'"*apprendre*"

à résoudre des tâches, sans que la résolution soit explicitement programmée.
L'analyse de gestes et plus généralement de signaux temporels, peut-être considérée selon deux approches. Dans le premier cas, les signaux sont caractérisés par des vecteurs, ou matrices, représentant l'information dans sa globalité. Dans le deuxième cas, chaque signal est caractérisé par des descripteurs reliés entre eux temporellement.
Selon la représentation choisie, les méthodes d'apprentissage diffèrent. On pourra par exemple utiliser des SVM ou des arbres de décision dans le premier cas tandis que l'analyse de signaux temporels requiert des méthodes spécifiques telles que les HMM, le DTW ou encore les réseaux de neurones récurrents.

Dans les sections suivantes, nous revenons sur les principales méthodes de la littérature utilisées pour ces deux types de représentation des signaux.

2.2.1 Apprentissage sur des Données Non-Temporelles

Bien souvent, l'apprentissage est réalisé en deux temps : une première étape de codage suivie de l'apprentissage en lui-même. Dans cette section, on suppose que les gestes sont codés de manière globale, menant à un vecteur de caractéristiques de taille fixe représentant le geste dans sa globalité . Ce type de codage, ou l'information temporelle est perdue, peut-être traité avec des méthodes d'apprentissage classiques que nous détaillons dans la suite de la Section.

2.2.1.1 Algorithme des Plus Proches Voisins

La méthode de recherche du plus proche voisin est une méthode facile à mettre en place et très populaire dans tous les domaines [Moldagulova and Sulaiman, 2017; Laptev et al., 2007]. Dans l'espace de représentation des données, le but est de trouver, grâce à une mesure de distance ou de similarité, l'exemple de la base d'apprentissage qui ressemble le plus à un exemple inconnu. La classe de cet exemple inconnu est déterminée par celle de l'exemple le plus proche. Une illustration est proposée Figure 2.1. Dans le cas d'un problème de régression, la valeur de l'exemple le plus proche est attribuée à l'exemple inconnu.

Une variante consiste à utiliser les K voisins les plus proches (*K-Nearest Neighbors - KNN*) pour prendre une décision. Dans le cas de la classification, la classe de l'exemple inconnu sera la classe majoritaire parmi les K plus proches voisins. En régression, la moyenne ou la médiane des valeurs des K voisins est attribuée à l'exemple inconnu.

La méthode des KNN est souvent utilisée pour la reconnaissance de geste. Par exemple, Liu *et al.* reconnaissent des gestes de la main dans le but d'améliorer les interactions homme-machine [Liu et al., 2016]. Mokhber *et al.* [Mokhber et al.,

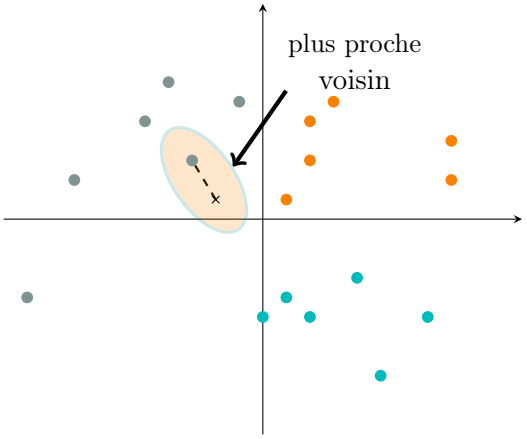


FIGURE 2.1: Illustration de l'algorithme du plus proche voisin.

2008] classifient des comportements humains à partir des moments géométriques, en utilisant la distance de Mahalanobis.

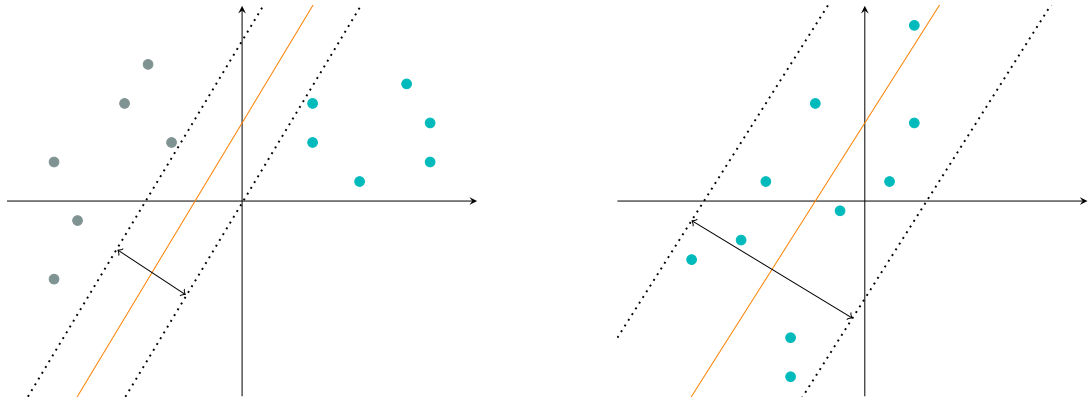
2.2.1.2 Machine à Vecteurs Supports

Les machines à vecteurs supports (*Support Vector Machine - SVM*) ont été proposées par Vapnik au milieu des années 90 [Vapnik, 1995]. L'hypothèse principale sur laquelle repose ces classifieurs binaires est la séparabilité des données de manière linéaire. L'apprentissage consiste à trouver l'hyperplan séparateur "*optimal*" qui va maximiser la distance minimale entre l'hyperplan séparateur et les exemples de la base d'apprentissage.

Malheureusement, les données sont rarement linéairement séparables. Pour répondre à cette problématique, il est possible de projeter les données d'apprentissage dans un espace de plus grande dimension dans lequel il sera possible de trouver un hyperplan séparateur entre les classes. Pour éviter une projection explicite des données dans ce nouvel espace, il est possible d'utiliser l'astuce du noyau (*Kernel Trick*), car seul le produit scalaire entre les descripteurs est requis pour le calcul de l'hyperplan.

Au-delà de la classification, il est également possible d'utiliser des machines à vecteurs supports en régression. On parle alors de SVR (*Support Vector Regression*) [Drucker et al., 1996]. Les deux approches, classification et régression, sont présentées Figure 2.2.

De la même manière que la méthode des plus proches voisins, les SVM sont souvent utilisés pour reconnaître des gestes, comme par exemple par Saha *et al.* ainsi que Nagarajan *et al.* dans une application liée au langage des signes [Saha et al., 2018; Nagarajan and Subashini, 2013]. Les SVR ont été utilisés pour évaluer la qualité d'un geste dans un contexte chirurgical [Zia and Essa, 2017], le but étant de prédire un score pour chaque réalisation.



(A) Illustration en 2D des SVM pour un problème à 2 classes. On retrouve en orange l'hyperplan séparateur.

(B) Illustration en 2D des SVR. On retrouve en orange l'hyperplan qui définit au mieux la tendance des points.

FIGURE 2.2: Illustration en 2D des SVM et des SVR.

2.2.1.3 Arbres de Décision

Les arbres de décision sont une classe d'algorithme d'apprentissage souvent utilisés en classification. Un arbre est un graphe non orienté et sans cycle, composé d'un nœud racine, de nœuds internes et enfin de feuilles. Chaque nœud représente un test réalisé sur les données et les feuilles amènent à la décision finale, *i.e.* la classe. Une illustration d'arbre de décision est présentée Figure 2.3.

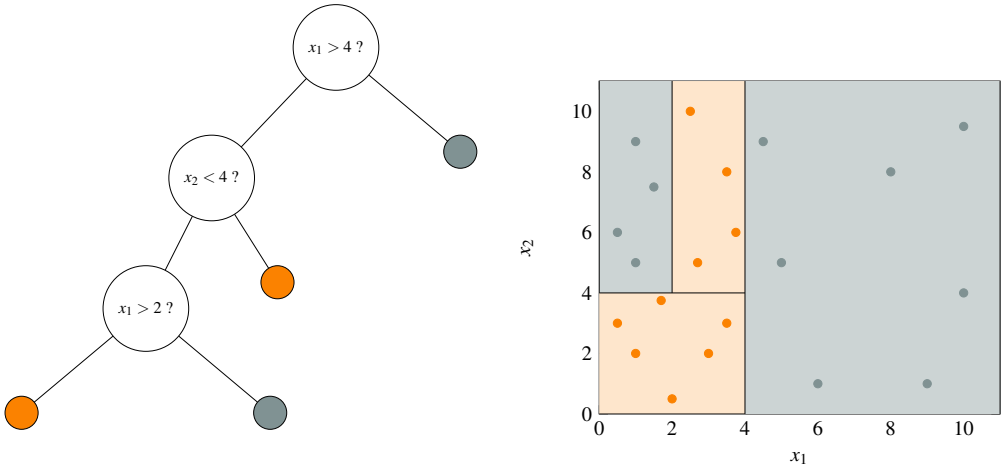


FIGURE 2.3: Illustration d'un arbre de décision sur un problème de classification binaire.

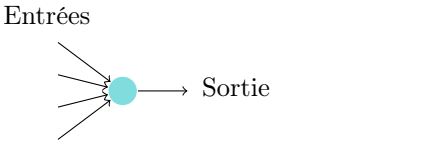
Lorsque les problèmes sont complexes, utiliser un seul arbre peut amener à une mauvaise généralisation. Pour cette raison, les forêts d'arbres aléatoires ont été introduites par Breiman *et al.* en 2001 [Breiman, 2001]. Plusieurs arbres sont entraînés à partir d'un tirage aléatoire des exemples et amènent à une décision individuelle. La décision finale de classification est la classe majoritaire des décisions de chaque arbre.

Ces arbres de décision ont été utilisés pour reconnaître ou segmenter des gestes. Par exemple, Joishi *et al.* [Joshi et al., 2015] classifient des fenêtres de temps glissantes, ce qui permet à la fois de segmenter et de reconnaitre les gestes.
Dans le domaine des interfaces Homme-Machine, Zhao *et al.* [Zhao et al., 2012] s'intéressent aux gestes de la main et créent une interface fluide en utilisant les forêts aléatoires.

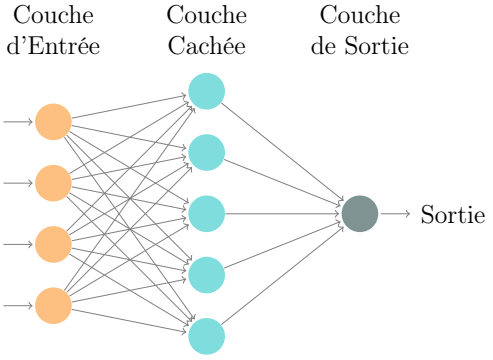
2.2.1.4 Réseaux de Neurones

Dès le début des années 40, les chercheurs se sont intéressés au fonctionnement du neurone biologique et à sa modélisation [McCullogh and Pitts, 1943]. Ceci a conduit au modèle du neurone formel (Figure 2.4A) encore utilisé aujourd'hui.
En 1949, Hebb [Hebb, 1949] propose d'associer plusieurs neurones pour réaliser l'apprentissage. Ceci a amené quelques années plus tard au perceptron [Rosenblatt, 1958] composé d'une couche de neurones d'entrée et d'une couche de neurones de sortie. Ce réseau très simple s'est avéré incapable d'apprendre des fonctions non linéaires mais a servi de base pour construire le perceptron multicouches [Hopfield, 1982] illustré Figure 2.4B.

Avec les progrès matériels, le développement de cartes graphiques de plus en plus performantes et l'arrivée des couches convolutionnelles, les réseaux de neurones se sont développés et accumulent de plus en plus de couches, on parle alors de *Deep Learning*.



(A) Illustration d'un neurone artificiel.



(B) Illustration d'un réseau de neurones avec une couche cachée.

FIGURE 2.4: Illustration des réseaux de neurones artificiels.

Ainsi, un réseau de neurones est composé de plusieurs couches de neurones et chaque neurone contient des poids pour le relier au neurone précédant. L'apprentissage du réseau consiste à trouver l'ensemble des poids qui répond au mieux à un problème posé.

À partir d'une base de données d'apprentissage D composée d'exemples d'entrée X et de sorties désirées Y, il va être possible d'apprendre les poids du réseau

de neurones, grâce à une descente de gradient et un processus de rétropropagation de l'erreur. Si $\mathcal{L}$ est la fonction de coût (erreur) entre la sortie désirée et la sortie du réseau pour les exemples d'apprentissage, la mise à jour des poids w_i^j du neurone i de la couche j est réalisée avec :

$$w_i^j = w_i^j - \lambda * \frac{\partial \mathcal{L}}{\partial w_i^j} \tag{2.1}$$

où λ est le pas d'apprentissage. Plusieurs itérations sont nécessaires pour obtenir une convergence et un bon apprentissage.

En plus des réseaux avec des neurones artificiels "*classiques*", d'autres "*types*" de neurones se sont développés en parallèle, *i.e.* les réseaux de neurones convolutifs (*Convolutional Neural Networks - CNN*).
Contrairement au perceptron multicouches, les neurones ne sont plus connectés à tous les neurones de la couche précédente. Ceci permet d'augmenter la taille des signaux d'entrée sans voir exploser le nombre de paramètres à apprendre. Pour la même raison, ceci permet d'augmenter le nombre de couches et la puissance d'apprentissage des réseaux.

Ces réseaux de neurones convolutifs ont été développés dès la fin des années 90 par LeCun *et al.* [LeCun et al., 1999]. Particulièrement efficace pour la reconnaissance de forme dans des images, les CNN reposent sur trois principes :

— *La connectivité locale* : un neurone n'est plus lié à tous les neurones de la couche précédente mais seulement aux neurones qui lui sont voisins, ce qui limite son nombre de poids.

— *Les poids partagés* : les neurones d'une couche partagent tous les mêmes poids.

— *L'invariance à la translation* : tous les neurones d'une même couche ayant les mêmes poids, cela induit qu'ils produiront la même sortie pour une entrée identique à deux position différentes.

La Figure 2.5 illustre une couche de convolution composée de D_1 filtres qui transforment l'entrée de taille $W \times H \times D$ en un nouveau tenseur de taille $W_1 \times H1 \times D$. W_1 et H_1 dépendent de la taille du filtre, de la gestion des bords et du déplacement lors du calcul de la convolution. En plus des couches convolutionnelles, les architectures modernes ont vu l'apparition de couches de "*pooling*" aidant à l'invariance en translation. Ces apports sont illustrés dans le réseau VGG16 qui a popularisé les CNN [Simonyan et al., 2013].

Plusieurs architectures spécifiques ont également vu le jour pour aider à la généralisation et donc améliorer les résultats de classification. On citera par exemple le module Inception [Szegedy et al., 2015] ou l'architecture ResNet [He et al., 2016].

Citons pour finir les réseaux siamois [Bromley et al., 1993] conçus pour comparer des exemples en entrées. Ce type de réseaux est très populaire pour des tâches

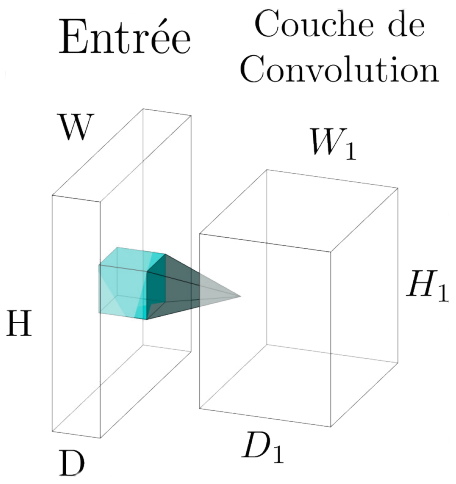


FIGURE 2.5: Illustration d'une couche de convolution dans un CNN.

de vérification d'identité [Taigman et al., 2014], d'empreintes digitales [Zhong et al., 2018], de signatures [Bromley et al., 1993] ou de la réidentification [Chung et al., 2017]. Par exemple, Chung *et al.* [Chung et al., 2017] comparent le flux vidéo de deux caméra grâce à un réseau siamois et arrivent ainsi à relier les images des deux caméras représentant d'une personne.

Les réseaux siamois possèdent deux branches identiques, qui partagent leur poids (une pour chaque exemple) et dont la sortie est utilisée pour produire la sortie finale du réseau.

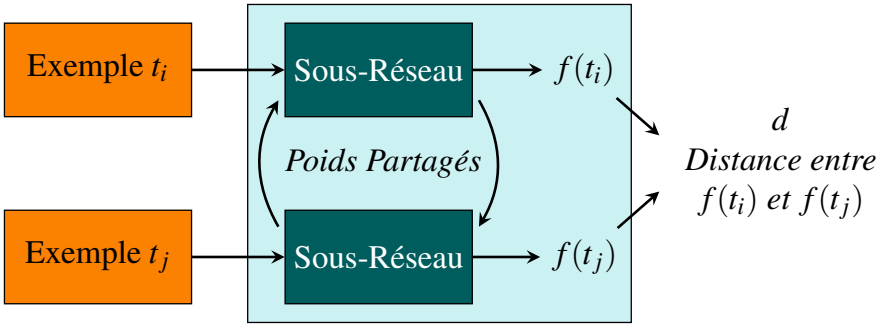


FIGURE 2.6: Illustration d'un réseau siamois.

Comme illustré Figure 2.6, les sorties du sous-réseau $f(t_i)$ et $f(t_j)$ sont utilisés pour comparer les exemples t_i et t_j.
La sortie désirée du réseau siamois est telle que :

$$y(t_i, t_j) = \begin{cases} 1 & \text{si } t_i \text{ et } t_j \text{ sont dans la même classe} \\ 0 & \text{si } t_i \text{ et } t_j \text{ ne sont pas dans la même classe} \end{cases} \tag{2.2}$$

La fonction de coût est donc [Hadsell et al., 2006] :

$$\mathcal{L}(W, t_i, t_j) = \frac{y}{2} d^2 + \frac{1-y}{2} max(0, m-d)^2 \tag{2.3}$$

avec d, la distance entre $f(t_i)$ et $f(t_j)$ et m le paramètre de marge.
Dans le cas où les exemples t_i et t_j n'appartiennent pas à la même classe, $y = 0$, la fonction de coût devient $\mathcal{L}(W, t_i, t_j) = \frac{max(0, m-d)^2}{2}$. Minimiser cette fonction de coût revient donc à rendre la distance d supérieure à la marge m. Enfin dans le cas où (t_i, t_j) sont de la même classe, $y = 1$, la fonction de coût devient $\mathcal{L}(W, t_i, t_j) = \frac{D^2}{2}$. Minimiser ce coût revient à minimiser d, *i.e.* $f(t_i)$ et $f(t_j)$ se rapprochent dans l'espace de représentation, comme représenté Figure 2.7.

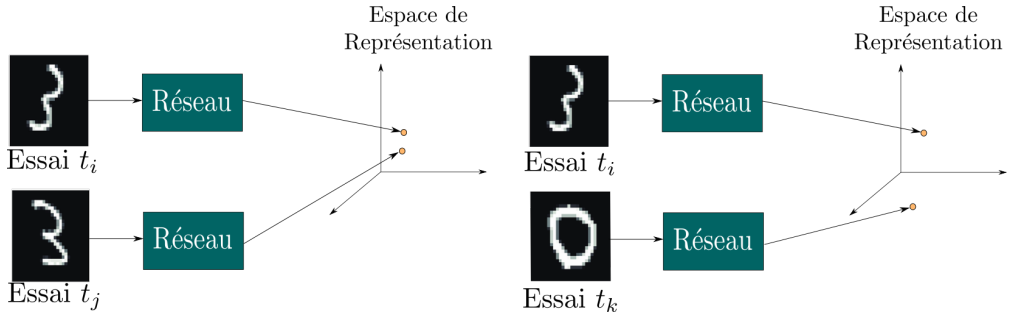


FIGURE 2.7: Illustration de la fonction de coût contrastive.

Afin d'avoir, un sous réseau plus robuste, une autre fonction de coût est souvent utilisée : la "*triplet loss*" [Chechik et al., 2009]. Avec cette fonction de coût une ancre A est définie comme étant la référence. Cette ancre va ensuite être comparée à deux autres exemples : un positif P et un négatif N, l'exemple positif est de la même classe que l'ancre et le négatif d'une autre classe. Le but, comme précédemment, est de minimiser la distance entre les représentations de l'ancre et l'exemple positif et de maximiser la distance entre l'ancre et le négatif. En utilisant la distance euclidienne comme métrique, on obtient la fonction de coût suivante :

$$\mathcal{L}(A, P, N) = max(||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + m, 0) \tag{2.4}$$

avec m, le paramètre de marge.

2.2.2 Apprentissage sur des Données Temporelles

Pour analyser les gestes, le codage peut consister à coder chaque instant du geste. On arrive ainsi à une série temporelle. Les méthodes présentées précédemment ne s'appliquent pas au traitement de ces données plus complexes qui peuvent avoir des longueurs variables. D'autres méthodes, présentées par la suite, sont plus adaptées.

2.2.2.1 Déformation Temporelle Dynamique

Pour reconnaître des signaux temporels, il est possible d'utiliser la méthode des KNN. Néanmoins, la mesure de similarité nécessite quelques adaptations. En effet, même s'il est possible d'utiliser la distance euclidienne, elle est inutilisable dans la plupart des cas, car elle nécessite que les signaux aient le même nombre d'échantillons et qu'aucune déformation temporelle ne se soit produite. Pour comparer deux signaux ayant des déformations temporelles et un nombre d'échantillons différent, il faut préférer des méthodes non linéaires, comme la Déformation Temporelle Dynamique (*Dynamic Time Warping - DTW*). Cette méthode permet de recaler les signaux entre eux par déformation temporelle élastique [Sakoe and Chiba, 1978]. Une comparaison de la distance euclidienne et du DTW est proposée Figure 2.8.

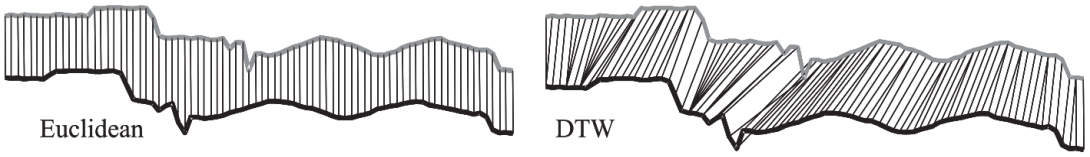


FIGURE 2.8: Figure extraite de [Keogh and Ratanamahatana, 2004]. Comparaison d'une métrique linéaire et non linéaire pour les signaux temporels. À gauche la distance euclidienne mesure la distance instant par instant et ne prend donc pas en compte la différence temporelle entre les deux signaux. À droite, le DTW prend en compte le décalage temporel, en appariant les échantillons similaires entre les deux signaux. Il permet donc d'aligner les signaux et d'arriver à une mesure de similarité entre les deux signaux.

Le DTW a été fréquemment utilisé afin de reconnaître des gestes. Ainsi, Pham *et al.* comparent des mouvements d'outils chirurgicaux grâce à l'algorithme du DTW, en utilisant la courbure des mouvements [Pham et al., 2010]. D'autres travaux se concentrent sur l'évaluation de la qualité de gestes sportifs. Morel *et al.*, utilisent l'algorithme du DTW afin de créer un modèle de geste et par la suite comparent un geste inconnu à ce modèle [Morel et al., 2017a].

2.2.2.2 Modèles de Markov Cachés

Les chaînes de Markov cachées (*Hidden Markov Model - HMM*) sont souvent utilisées pour modéliser des séries temporelles (Figure 2.9). La modélisation fait intervenir des états (décrits grâce à une matrice d'observation), des transitions entre ces états (décrites grâce à une matrice de transition) et une probabilité d'état initial [Rabiner, 1989]. Ces trois éléments décrivent entièrement une chaîne de Markov. Leur utilisation, avec des algorithmes dédiés, permet de reconnaître des séquences ou de les segmenter.

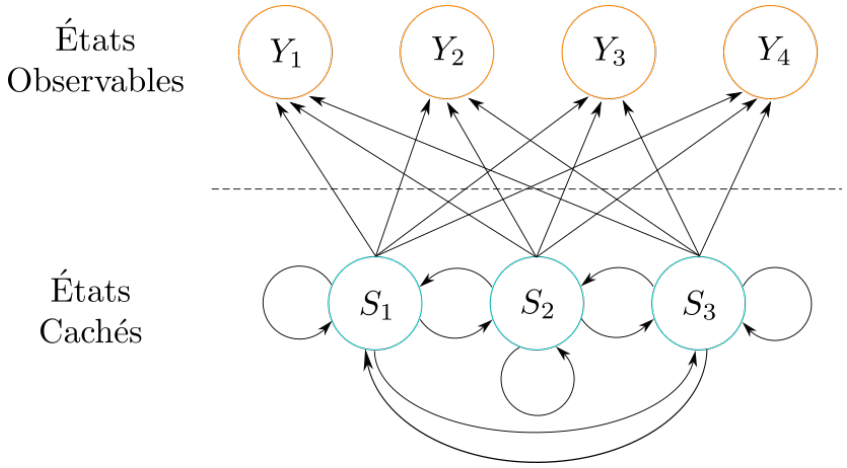


FIGURE 2.9: Illustration d'un modèle de Markov caché.

En effet, une fois les paramètres des HMM appris, reconnaître un geste consiste à trouver le modèle de Markov qui a la plus grande probabilité d'avoir généré la séquence d'observations [Park et al., 2005; Parcheta and Martínez-Hinarejos, 2017; Saha et al., 2017; Chen et al., 2003]. D'autre part, en associant un état à un geste, les HMM peuvent être utilisés pour segmenter des séquences [Nakamura et al., 2017; Tao et al., 2012; Lei et al., 2016]. En effet, l'algorithme de Viterbi [Forney, 1973] permet de retrouver la séquence d'états à partir d'une séquence d'observations.

2.2.2.3 Réseaux de Neurones

Certaines architectures ou utilisations des réseaux de neurones peuvent également traiter des séries temporelles, comme illustrée Figure 2.10.

Une première solution consiste à "*piocher*" des clips de taille fixe dans les séries temporelles et à les classer indépendamment les uns des autres [Karpathy et al., 2014; Tran et al., 2015]. La classe de la série entière est ensuite déterminée à partir de la classification des clips. Le problème est donc ramené à la classification de clips temporels de taille fixe. Pour simplifier, considérons des clips vidéos, de taille $L \times H \times 3 \times T$ où L et H représentent les dimensions de l'image et T, la longueur temporelle du clip. Certains auteurs appliquent des convolutions 2D [Karpathy et al., 2014] selon les dimensions spatiales. Un filtre amène ainsi à une image de taille $L \times H$. D'autres auteurs [Tran et al., 2015] préfèrent appliquer des convolutions 3D, spatio-temporelles. Un filtre amène alors à un volume de taille $L \times H \times T$. Dans les deux cas, les architectures des réseaux ressemblent à celles utilisées pour les images avec des couches convolutionnelles, des couches de *pooling* et des couches entièrement connectées. En utilisant les neurones de la dernière couche convolutionnelle, un descripteur de vidéo peut être obtenu, comme le descripteur C3D [Tran et al., 2015] appris sur une base de sport mais utilisable dans d'autres contextes. Enfin, une approche "*two streams*" s'est avérée

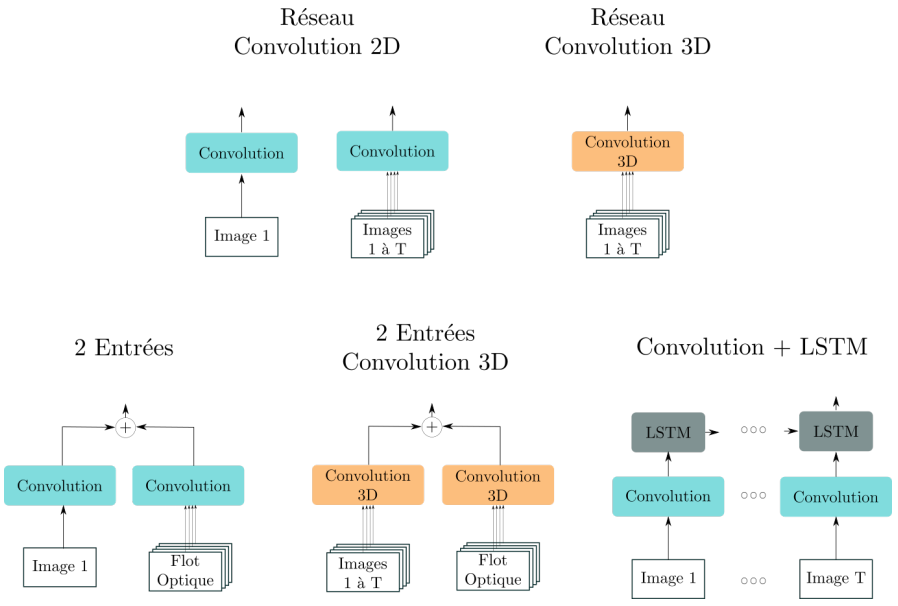


FIGURE 2.10: Illustration des différentes architectures de réseaux de neurones adaptés à l'analyse de geste, tirée de [Carreira and Zisserman, 2017].

très performante pour reconnaître des actions à partir de vidéos [Simonyan and Zisserman, 2014]. Elle consiste à traiter séparément le flux spatial et le flot optique puis de fusionner les résultats. Ces approches convolutionnelles ont été utilisées pour segmenter des tâches chirurgicales [Lea et al., 2017b] ou pour reconnaître des gestes [Jie Huang et al., 2015]. Elles peuvent aussi être utilisées directement sur les signaux, sans extraire de clips. Pour gérer la variabilité dans la taille des signaux, un *pooling* temporel global devra alors être intégré avant les couches entièrement connectées.

Des architectures spécifiques de réseaux de neurones sont dédiées aux séries temporelles, il s'agit des réseaux récurrents (*Recurrent Neural Network - RNN*). Ils mettent en place des états cachés h_t qui transportent l'information utile au cours du temps (Figure 2.11).

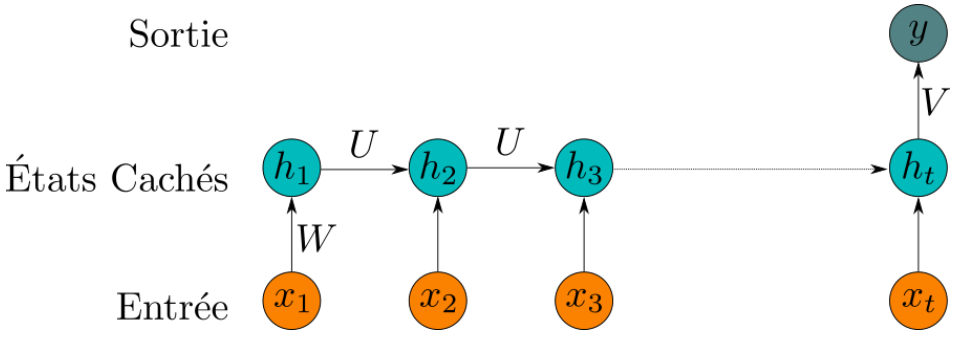


FIGURE 2.11: Illustration des RNN où une séquence d'entrée x_t produit une seule sortie y. L'information est transportée grâce à un état caché h_t. Les poids U, V et W du réseau sont partagés par tous les temps.

Selon l'application visée, une séquence peut fournir une seule sortie (reconnaissance d'actions), une séquence de même taille (segmentation) ou une séquence de taille différente (traduction linguistique). Contrairement aux réseaux de neurones traditionnels qui utilisent différents paramètres pour chaque couche, les RNN partagent les paramètres (U, V et W Figure 2.11) à travers le temps, ce qui permet de gérer des signaux de longueur variable. L'apprentissage des RNN fait appel à une rétropropagation particulière, à travers le temps [Werbos, 1990] qui s'avère sensible au problème de disparition du gradient. Même si l'application reine des RNN est la traduction automatique, ils ont été utilisés pour reconnaître [Du et al., 2015; Veeriah et al., 2015], détecter [Singh et al., 2016], ou segmenter [DiPietro et al., 2016] des actions. En théorie, l'information utile peut être transportée tout au long de la séquence, mais en pratique, elle a du mal à traverser quelques échantillons. Ceci a conduit à la mise en place d'une autre cellule récurrente : la Long Short Term Memory (*Long Short Term Memory - LSTM*) [Hochreiter and Schmidhuber, 1997].

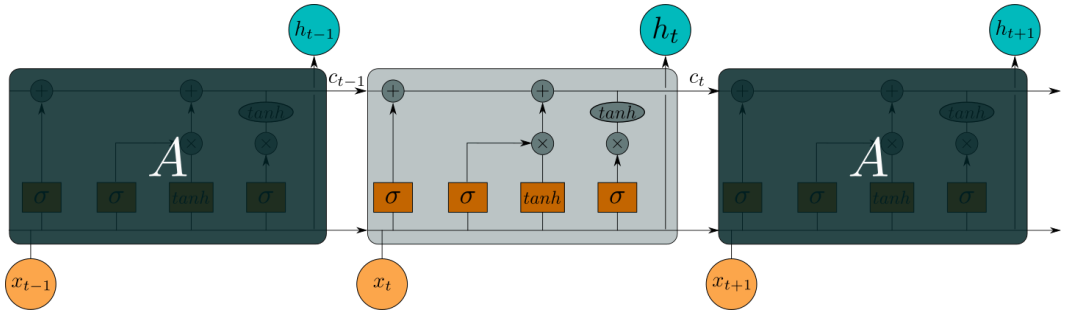


FIGURE 2.12: Illustration des LSTM

En plus d'un état caché h_t, les LSTM propagent une cellule mémoire c_t (Figure 2.12) à travers le temps, apte à transporter l'information sur de longues périodes. Des fonctions portes spécifiques sont mises en place pour gérer cette mémoire et y ajouter ou supprimer de l'information. Les LSTM ont été utilisées pour reconnaître [Zhu et al., 2017] ou segmenter [Wang et al., 2017] des gestes. Ils sont maintenant souvent combinés avec des processus d'attention afin d'améliorer les performances [Zhang et al., 2018].

2.3 Bilan

Dans ce chapitre, nous avons tout d'abord dressé un état des lieux des différents domaines de l'analyse de gestes afin de mieux positionner nos travaux. Ainsi, nous souhaitons estimer la qualité d'un geste, en attribuant un score à chaque réalisation d'une part et en fournissant un retour sur les erreurs commises d'autre part.
Dans un second temps, nous avons présenté les méthodes d'apprentissage statistiques utilisées lors de l'analyse des gestes. Comme il s'agit de signaux spécifiques,

dont la longueur varie d'une réalisation à l'autre, nous avons fait ressortir deux grandes catégories de méthodes. La première vise à coder le geste de manière globale par un vecteur de caractéristiques. Ceci permet de retrouver des exemples de taille fixe qui peuvent être traités par des méthodes d'apprentissage classiques comme les KNN, les SVM/SVR ou les forêts d'arbres aléatoires. Malheureusement, ce codage global rend impossible le retour à la dimension temporelle et donc l'accès aux erreurs commises lors de la réalisation du geste. La seconde catégorie de méthodes est explicitement dédiée aux signaux temporels. Parmi celles-ci, le DTW est très utilisé et une méthode a déjà été proposée pour estimer la qualité d'un geste et fournir un retour à l'utilisateur sur les défauts réalisés. Cette méthode n'est cependant pas généralisable à tout type de gestes puisqu'elle requiert une trajectoire précise afin de construire un modèle de geste sous forme de "*trajectoire moyenne*". Elle ne peut donc pas être utilisée pour des tâches chirurgicales par exemple. Les HMM sont également très employés pour l'analyse de séries temporelles. Il s'agit d'une méthode générative qui vise à construire un modèle pour modéliser un geste. S'ils sont très utilisés en classification, leur emploi pour estimer la qualité d'un geste semble difficile. La dernière méthode présentée et celle envisagée dans cette thèse, est donc d'utiliser des réseaux de neurones. De par leur polyvalence, et les résultats performants obtenus ces dernières années dans différents domaines, cette solution semble être la plus adaptée à l'évaluation de la qualité d'un geste. Cependant la question du retour d'information demeure. En effet, un réseau de neurone permet, comme vu précédemment, de classifier des gestes dans différentes classes ou de régresser un score de qualité, mais fournir un retour d'information à partir de ce genre de modèle est encore peu exploré.

Dans les prochains chapitres, nous présentons dans un premier temps une stratégie pour évaluer la qualité d'un geste en utilisant des réseaux de neurones. Par la suite, nous verrons comment utiliser ces réseaux pour fournir un retour d'information précis à l'utilisateur.

Chapitre 3

Évaluation de la Qualité d'un Geste

3.1 Introduction

Tout au long de la vie, nous passons notre temps à apprendre, que ce soit pour des loisirs, comme apprendre à jouer au tennis, ou bien pour des besoins professionnels, en apprenant à opérer quelqu'un dans le cas d'un chirurgien. Mais apprendre, ne consiste pas juste à apprendre de nouveaux gestes, il faut également être capable d'absorber de nouvelles connaissances. Cet apprentissage commence dès le plus jeune âge, à l'école, où l'on demande aux élèves d'engranger beaucoup de nouvelles connaissances tous les jours, qu'elles soient intellectuelles ou bien gestuelles pendant les cours de sport. Cependant, apprendre seul est très difficile, surtout dans le cas de gestes car les seules informations disponibles résident dans

les conséquences de la réalisation du geste si elles existent. De la même manière qu'un apprenti boulanger a un maître boulanger comme référent pour lui montrer les bons gestes, les bonnes recettes et également pour le corriger en cas d'erreur, tout type d'apprentissage nécessite un maître dans la discipline, afin de fournir un retour et de guider l'enseignement dans la bonne direction. Ainsi, les examens scolaires ou universitaires sont, bien sûr, nécessaires pour valider une compétence, mais permettent également aux étudiants d'avoir un retour sur les connaissances acquises. Cependant, il est plus aisé d'évaluer des connaissances, que des compétences gestuelles. En effet, au-delà, d'un certain niveau d'expertise dans une tâche ou un sport, chaque personne va développer une technique propre et il devient donc très complexe de comparer deux personnes de niveau égal, mais ayant une technique différente. Néanmoins, repérer un débutant reste facile, ainsi qu'être capable de lui fournir un retour et des conseils sur comment se corriger et s'améliorer. Par exemple, les salles de sport engagent toujours un coach qui vérifie que toutes les personnes présentes réalisent les gestes de manière adéquates. En effet, lors de la réalisation de squats, par exemple, le risque de blessure au dos pour un débutant est très haut, notamment dû à sa mauvaise posture [Diggin et al., 2011]. S'il est repéré par le coach, le débutant obtiendra des conseils sur comment bien se corriger afin de ne pas se blesser. Toutefois, en cas de forte affluence, il est possible que le coach ne le voit pas et la blessure est alors presque inévitable. Mettre en place des coachs virtuels est alors une solution pour permettre à des débutants d'apprendre les bons gestes. De plus, cela permettrait de faire du sport chez soi, sans risques de blessures. Ce genre de solution nécessite la mise en place de base de données et d'algorithme d'apprentissage afin de pouvoir repérer les erreurs et les corriger.
Un autre domaine qui peut bénéficier de la mise en place de coachs virtuels est la chirurgie et plus particulièrement la chirurgie mini-invasive (*Minimally Invasive Surgery - MIS*). La MIS consiste à faire de petites incisions sur le ventre du patient afin d'insérer des outils et un endoscope afin d'opérer via ces incisions. Ce type de chirurgie a permis de réformer les opérations abdominales, en réduisant considérablement le temps de convalescence du patient. Cependant, pour le chirurgien, la tâche devient plus difficile en raison du manque de perception de la profondeur, de la vision indirecte due à l'endoscope et des degrés de liberté limités induits par les trocarts et l'environnement clos. Apprendre les bons gestes de MIS est donc long et difficile et nécessite un guidage constant de la part d'un expert. Or les chirurgiens experts sont peu disponibles et rares. Alors, comment s'assurer que les apprenants ont atteint un certain niveau de compétence ? Comment les aider dans l'apprentissage de manière à l'accélérer ? La mise en place d'outils d'apprentissage en réalité virtuelle est désormais un sujet majeur de recherche et a mené à la commercialisation de plusieurs simulateurs, par exemple le Laparo Analytic de la société Laparo® (illustré Figure 3.1) qui enregistre les gestes pendant une tâche et donne un retour sur la performance.

On peut encore citer d'autres domaines où des gestes techniques sont requis :

FIGURE 3.1: Photographie du simulateur Laparo Analytic de la société Laparo^{TM}(https://laparo.pl/en/).

par exemple le milieu artistique comme la peinture, la sculpture ou la poterie. Le métier de potier nécessite de nombreuses connaissances ainsi que la maîtrise de certains gestes. En effet, savoir manipuler l'argile est un art qui demande en plus de nombreuses heures de pratique, d'avoir des démonstrations de la part d'experts dans le domaine.

La mise en place de coachs virtuels, que ce soit dans le sport, la chirurgie ou le milieu artistique, implique un travail conséquent en amont. L'avantage est que contrairement à une évaluation manuelle qui peut être longue et rare, l'évaluation fournie par un coach virtuel sera rapide, objective et aussi fréquente qu'on le souhaite, comme illustré Figure 3.2.

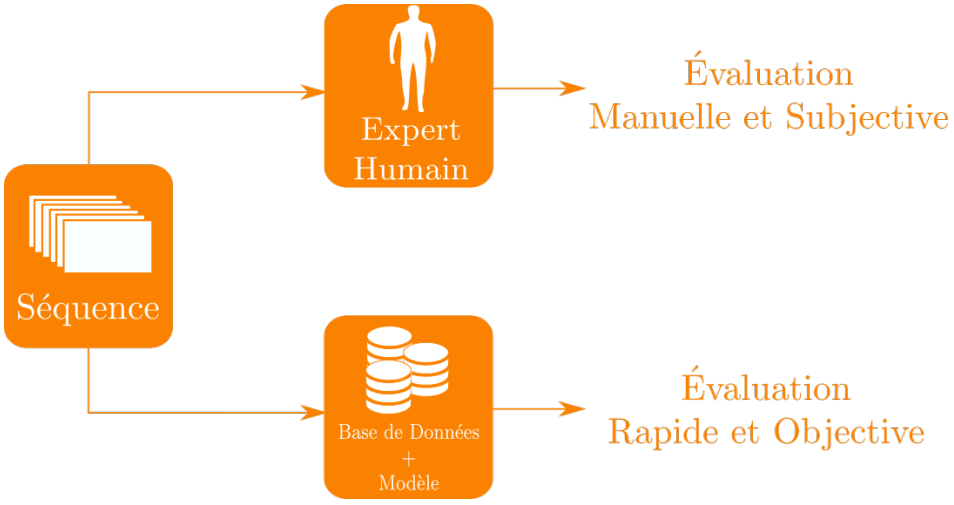


FIGURE 3.2: Processus pour l'évaluation de la qualité d'un geste.

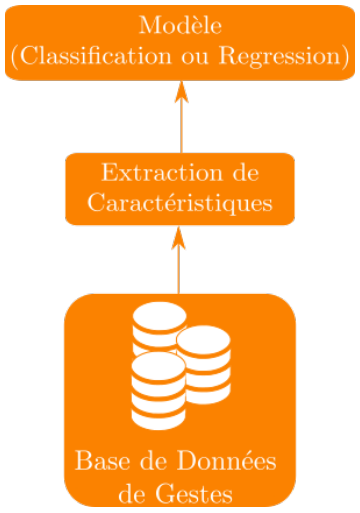
L'inconvénient est que le développement d'un coach virtuel nécessite d'enregistrer des bases de données de gestes, en grand nombre, annotées par un expert qui en déterminera la qualité, au travers d'une note par exemple. La constitution

et l'annotation de la base de données sont importantes puisque cela conditionne le type de méthode d'apprentissage à utiliser. De la même manière, selon que le coach virtuel fournisse juste une note ou un retour d'information sur ce qui a été mal réalisé, le type de méthode diffère.

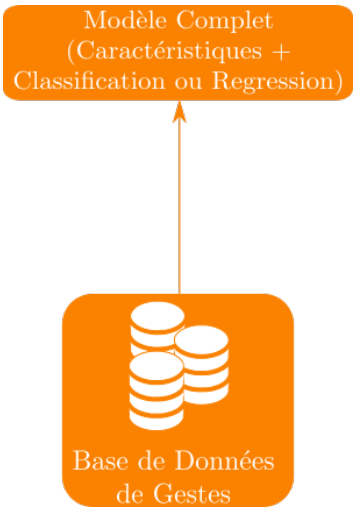
Dans ce chapitre, nous proposons tout d'abord de faire un état de l'art des méthodes existantes pour l'évaluation de la qualité d'un geste, en nous concentrant particulièrement sur les gestes sportifs et chirurgicaux. Après un bilan, des méthodes existantes par rapport à nos attentes, nous présentons deux nouvelles méthodes d'évaluation automatique de la qualité d'un geste. Ces deux approches sont testées sur des gestes provenant de domaines bien différents : la chirurgie et le sport.

3.2 État de l'Art

Dans cette section nous introduisons différentes méthodes permettant l'évaluation de la qualité d'un geste, en nous concentrant plus particulièrement sur les méthodes évaluant les gestes chirurgicaux et les gestes sportifs. Deux grandes familles de méthodes ont émergé de cette étude bibliographique : les méthodes utilisant des caractéristiques choisies et prédéfinies manuellement et les méthodes "*end-to-end*", partant de l'enregistrement du geste et évaluant directement sa qualité, comme illustré sur la Figure 3.3



(A) Illustration des méthodes avec des caractéristiques choisies manuellement.



(B) Illustration des méthodes "*end-to-end*".

FIGURE 3.3: Illustration des deux grandes familles de méthodes pour l'évaluation de la qualité d'un geste.

3.2.1 Évaluation avec des Caractéristiques Manuellement Définies

Afin d'évaluer automatiquement la qualité d'un geste, il est nécessaire de trouver des caractéristiques capturant des informations sur les dimensions spatiales et temporelles. A cette fin, il est possible de définir des caractéristiques qui seront extraites de chaque geste et ensuite utilisées pour l'évaluation par un algorithme d'apprentissage. Comme évoqué précédemment, nous nous intéressons plus particulièrement aux travaux réalisés sur les gestes chirurgicaux et les gestes sportifs, qui sont les plus étudiés aujourd'hui.

3.2.1.1 L'évaluation de Gestes Chirurgicaux

Pour un chirurgien novice, apprendre à opérer est long et complexe. Cela est partiellement dû aux gestes, très spécifiques et très précis, mais également au manque de retour lors de séances d'entraînement en solitaire. Ces deux facteurs rendent l'apprentissage de la laparoscopie long. Pour le faciliter, certains travaux se concentrent sur le développement de méthodes d'aide à l'apprentissage, en particulier en donnant un retour d'information aux novices pendant le processus d'apprentissage.
Ainsi Judkins *et al.* proposent d'étudier des mesures et de les évaluer afin de savoir si elles sont assez discriminatives entre des experts et des novices [Judkins et al., 2009]. Cette évaluation est fondée sur trois tâches de chirurgie mini-invasive réalisées à l'aide d'un robot télémanipulé(*Robotic Minimally Invasive Surgery - RMIS*) :

- Déplacer des objets avec les deux pinces en simultanée ;
- Passer une aiguille dans un parcours ;
- Faire des nœuds de suture.

L'utilisation des robots d'aide à la chirurgie aide grandement l'acquisition des données puisque l'on connaît la position des outils à chaque instant. Les caractéristiques proposées sont les suivantes : le temps pour réaliser la tâche, la distance totale parcourue par l'outil, la vitesse, la courbure des mouvements ainsi que l'angle entre les deux outils. Selon leur étude, ces caractéristiques sont effectivement discriminantes et permettent d'entraîner un classifieur afin de différencier les novices des experts en chirurgie.
Pour aller plus loin, Fard *et al.* proposent de rajouter des caractéristiques en plus des cinq proposées précédemment [Fard et al., 2016]. Les deux caractéristiques rajoutées sont la distance parcourue par l'outil le long de son axe et la fluidité du mouvement, ce qui porte le nombre de caractéristiques pour décrire le mouvement à quatorze (sept pour chaque outil). En utilisant ces caractéristiques, deux classifieurs binaires sont entraînés (il n'y a que deux classes : novice et expert) : une régression logistique et un SVM. Avec ces deux classifieurs, les résultats de

classifications sont correctes ($\approx 70\%$) et permettent donc de valider le choix des caractéristiques pour décrire le mouvement chirurgical, même si des progrès restent à faire.
Au-delà des caractéristiques, il est intéressant de savoir à quel niveau de décomposition les calculer, afin d'avoir une évaluation qui soit la plus juste possible. C'est le but d'une étude proposée par Vedula *et al.* [Vedula et al., 2016], qui évalue la qualité de sous-gestes afin ensuite de déterminer la qualité de la tâche globale. En utilisant des tâches de RMIS, ils proposent de tester trois des caractéristiques précédemment évoquées : le temps, la longueur du chemin et le nombre de mouvements (un mouvement étant défini entre deux pics de vitesse). Les résultats de l'étude montrent que le classifieur donne les mêmes résultats que l'on utilise les tâches globales ou bien des sous-gestes pour extraire les caractéristiques.
Les caractéristiques extraites des trajectoires des outils donnent des résultats intéressants et permettent ainsi d'établir une base de résultat, mais il est possible de les améliorer.
L'évaluation de gestes chirurgicaux ne s'arrête pas à la MIS. En effet, Poddar *et al.* proposent d'évaluer le niveau de chirurgiens réalisant une septoplastie nasale [Poddar et al., 2014]. Pour cela, ils proposent d'enregistrer le chemin parcouru par les outils et par la suite de le segmenter en traits (*strokes*), chaque trait représentant un mouvement, comme illustré Figure 3.4.

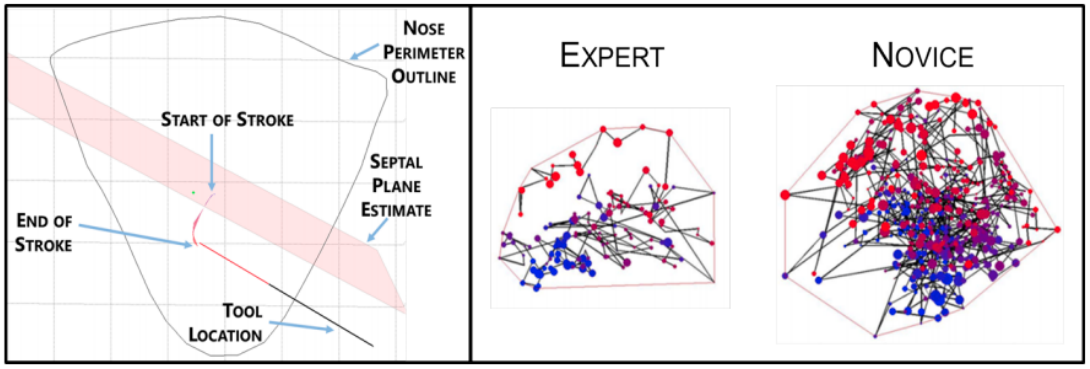


Figure 3.4: Illustration des traits utilisés pour l'évaluation des compétences, extraite de [Poddar et al., 2014].

À partir de ces traits, les auteurs sont capables d'extraire des caractéristiques du même type que précédemment, *i.e.* la courbure, la durée, et la couverture du septum nasal d'un trait. Un HMM entraîné avec ces caractéristiques permet ensuite de prédire le niveau d'expertise de la personne réalisant la septoplastie.
Zia *et al.* proposent quant à eux d'utiliser des caractéristiques fréquentielles (Transformée de Fourier Discrète (TFD), Transformée en cosinus discrète (TCD)...) afin de soit prédire un score, soit prédire le niveau du chirurgien (Novice, Intermédiaire ou Expert) [Zia and Essa, 2017]. Chaque caractéristique fréquentielle (TFD, TCD,...) est traitée indépendamment, suivie d'une analyse en composante principale (ACP) puis un classifieur (ou régresseur) Une agglomération des résultats des

différents classifieurs est par la suite réalisée. Ceci permet d'entraîner des classifieurs plus simples tout en gardant un bon niveau de performance.

Plutôt que d'utiliser les données cinématiques des outils, certains travaux extraient des caractéristiques directement en utilisant l'enregistrement vidéo des tâches. Utiliser les vidéos rajoute une difficulté supplémentaire à cause du grand nombre de dimensions. En effet trouver des caractéristiques, à la fois temporelles et spatiales, en utilisant un flux vidéo est complexe.
Dans leurs travaux, Sharma *et al.* utilisent les vidéos de personnes réalisant des sutures [Sharma et al., 2014]. Afin de pouvoir exploiter ces vidéos, il est nécessaire d'en extraire des caractéristiques. Leur choix s'est porté sur des caractéristiques de texture, telles que la matrice de co-occurrence des niveaux de gris [Haralick et al., 1973] ou encore le Motif Binaire Local (*Local Binary Pattern - LBP*) [Ojala et al., 1996]. Comme les caractéristiques choisies restent dans un espace de grande dimension, une Analyse Linéaire Discriminante (*Linear Discriminant Ananlysis - LDA*) est d'abord réalisée afin de projeter les données dans un espace de plus petite dimension. Par la suite, une régression linéaire est effectuée afin d'évaluer la qualité des gestes. Toujours en utilisant des caractéristiques de texture, Zia *et al.* proposent d'ajouter les données de capteurs posés sur les mains afin d'obtenir des données cinématiques [Zia and Sharma, 2017]. Le travail étant principalement focalisé sur le choix des caractéristiques, la classification utilisée est une simple recherche du plus proche voisin dans l'espace des caractéristiques. Par la suite une étude comparative de plusieurs caractéristiques extraites des vidéos a été réalisée. Les méthodes envisagées sont : les caractéristiques symboliques (*bag of words*, HMM), les caractéristiques de texture (matrice de co-occurence) et les caractéristiques fréquentielles (TFD, TCD) [Zia et al., 2016]. Les résultats de cette étude montrent qu'utiliser des caractéristiques fréquentielles extraites de vidéos donne de meilleurs résultats de classification, comparées aux autres caractéristiques, pour un classifieur donné.

3.2.1.2 Évaluation de Gestes Sportifs

Apprendre un nouveau sport de la même manière, qu'apprendre à opérer en MIS, est difficile, surtout en l'absence de conseil, et donc d'évaluation des compétences. Dans le but de pouvoir faire cette évaluation de manière automatique des études se sont intéressées à la prédiction de score pour les gestes sportifs. Cependant, l'évaluation d'un geste sportif est plus complexe que celle d'un geste chirurgical. En effet, l'évaluation automatique des compétences dans le sport nécessite de connaître la partie du corps sur laquelle on doit se concentrer et le moment où il faut le faire, contrairement à la MIS où l'on peut se concentrer uniquement sur les outils. Par exemple, lors du coup droit d'un tennisman, l'étude du bras portant la raquette est bien plus importante que l'étude de la jambe ou de l'autre bras. A l'opposé, le piétinement lors de l'attente de la balle demande de se focaliser plus sur les jambes. Comme pour la chirurgie, l'étude du geste peut être

faite à partir des données cinématiques des articulations, ou directement à partir des flux vidéos.

Créer un système de coach virtuel demande tout d'abord d'être capable d'évaluer les performances et les réalisations de certains gestes. Dans ce but, Burns *et al.* s'intéressent à un geste de karaté et à l'utilité d'un coach virtuel pour l'apprentissage de ce geste [Burns et al., 2011]. Afin d'évaluer le niveau des participants, un enregistrement des gestes grâce à un système de capture de mouvement est d'abord réalisé. L'évaluation dans ces travaux reste manuelle. Cependant l'efficacité d'un coach virtuel a elle été démontrée, car les personnes s'étant entraînées avec ce coach ont eu la même courbe de progression que celles ayant eu un coach humain.
Toujours dans la même idée de coach virtuel et pour la boxe, Komura *et al.* proposent d'aller plus loin, et d'effectuer une évaluation automatique de la performance [Komura et al., 2006]. En utilisant des mouvements enregistrés à l'aide d'un système de motion capture et des connaissances *a priori* sur les gestes réalisés, ils extraient des caractéristiques qui permettent de différencier les novices des experts pour un évaluateur humain. Ces caractéristiques sont les suivantes : le nombre de mouvements total de défense, le temps de réaction pour répondre à une attaque, le nombre de feintes et de combinaisons.
Pour la danse, Maes *et al.* se sont intéressés à la synchronie entre les mouvements d'un expert et ceux d'un novice [Maes et al., 2013]. En effet, en faisant l'hypothèse que la musique choisie impose la vitesse d'exécution, il va être possible de comparer des fenêtres de temps des mouvements du novice et de l'expert, ce qui permet d'évaluer la performance du novice.

Utiliser un flux vidéo reste aussi complexe pour le sport que pour la chirurgie. Cependant des travaux se basant sur les images pour faire l'évaluation de la qualité d'un geste ont été réalisés.
Pour les routines d'aérobic, John *et al.* proposent d'évaluer les postures [John et al., 2019]. Dans un premier temps, une extraction de l'arrière-plan est réalisée afin d'obtenir uniquement la personne réalisant la posture. L'évaluation est ensuite fondée sur un calcul d'intersection entre les images du novice et celles de l'expert. Plus les postures extraites vont s'intersecter, plus la posture sera considérée comme bien réalisée. Ce mode d'évaluation est bien adapté aux postures fixes, mais a du mal à s'adapter aux gestes complexes et longs dans le temps.

3.2.2 Évaluation sans Caractéristiques Prédéfinies

Des méthodes d'apprentissage profond sont également apparues pour prédire le niveau de compétences dans différentes applications. L'avantage de ce genre de méthodes réside dans leur capacité à généraliser. En effet que ce soit en classification ou en régression, les résultats obtenus sont très souvent bien supérieurs à

ceux des méthodes utilisant des caractéristiques prédéfinies associées à un autre algorithme d'apprentissage. Cependant, afin de généraliser au mieux, il est nécessaire d'avoir des bases de données de taille importante.
De la même manière que pour les méthodes d'évaluation avec des caractéristiques prédéfinies, nous présentons les travaux sur les gestes chirurgicaux dans un premier temps, puis ceux sur les gestes sportifs.

3.2.2.1 Évaluation de Gestes Chirurgicaux

La base de données de gestes chirurgicaux de RMIS présentée Section 3.3.1 a permis le développement de nombreuses méthodes d'apprentissage profond. En effet, il est possible de faire deux types d'évaluation sur cette base de données : une évaluation du niveau de l'utilisateur (Classification Expert/Novice/Intermédiaire) ou bien une évaluation au travers d'un score de la qualité du geste (Régression sur les scores OSATS présentés Section 3.3.1).

La base de données JIGSAWS (Section 3.3.1) possède assez peu d'exemples par tâche, ce qui rend complexe l'utilisation de méthodes fondées sur l'apprentissage profond. Pour augmenter la taille de la base de données, Wang *et al.* divisent chaque essai en petits segments à l'aide d'une fenêtre glissante de taille fixe [Wang and Fey, 2018]. La classe donnée à chaque segment est identique à celle de l'exemple de base. Par la suite, un réseau convolutionnel est entraîné afin de classifier les segments. L'utilisation de cette stratégie permet d'augmenter considérablement le nombre d'exemples, néanmoins ils ne sont plus non corrélés. Le modèle défini par Wang *et al.* est présenté Figure 3.5. Il est assez classique dans le sens où il est composé de couches convolutionnelles et de *max-pooling* dans une première partie, puis de couches entièrement connectées. Les signaux d'entrées étant de taille fixe (extraction à partir de la fenêtre glissante), il n'est pas nécessaire de gérer le problème de la longueur des signaux temporels.

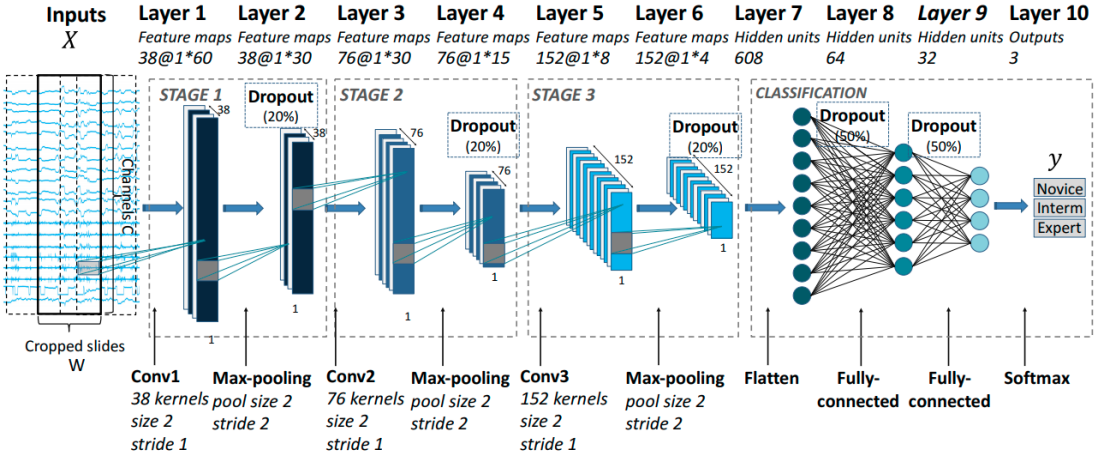


FIGURE 3.5: Modèle utilisé pour la classification des gestes chirurgicaux en fonction du niveau du participant [Wang and Fey, 2018].

Toujours dans le but de prédire le niveau d'expertise des personnes réalisant la tâche, Fawaz *et al.* utilisent également les données cinématiques de la base mais de manière différente [Fawaz et al., 2018]. En regroupant ces données selon leur type (position, accélération, rotation), ils proposent d'extraire des caractéristiques inhérentes au type de données à l'aide d'un CNN. A chaque couche, les caractéristiques sont regroupées, jusqu'à arriver à un vecteur latent représentant la séquence d'entrée, utilisé pour prédire le niveau d'expertise. Grâce à l'architecture du réseau convolutionnel, il est également possible de retrouver les moments qui ont motivé la décision du réseau.
Enfin Funke *et al.* proposent d'utiliser les enregistrements vidéos de chaque tâche pour entraîner un réseau convolutionnel 3D (images 2D + temps) à reconnaître le niveau d'expertise [Funke et al., 2019]. Le nombre de paramètres à apprendre pour ce type de réseau est énorme, il est donc nécessaire d'avoir un grand nombre de données. Cependant pour répondre à cette problématique, les auteurs utilisent un réseau pré-entrainé sur une autre base de données. Cette stratégie permet d'avoir un réseau 3D capable d'extraire des caractéristiques spatio-temporelles qui sont utilisée par un autre réseau sur des bases de données de taille réduite.

Pour avoir une évaluation plus fine des compétences, il est intéressant d'entraîner des modèles à régresser les scores plutôt que de prédire les trois classes d'expertise.
Ainsi, Doughty *et al.* utilisent une architecture siamoise pour estimer le score des exemples d'entrée [Doughty et al., 2018]. L'utilisation de réseaux siamois permet d'augmenter la taille de la base d'apprentissage, car l'entrée est constituée d'une paire d'exemples et non plus d'un exemple seul. Dans leur principe, les réseaux siamois permettent de comparer les deux exemples d'entrée pour prédire lequel est le meilleur. Ceci permet donc de trier les exemples selon leur niveau de réalisation. Par la suite et afin de pouvoir fournir un retour sur les erreurs commises pendant la réalisation d'une tâche, deux modules d'attention ont été ajoutés [Doughty et al., 2019]. Ces modules permettent au réseau de se focaliser, pour l'un sur les instants avec un haut niveau de compétence et pour l'autre sur les instants correspondant à un faible niveau de compétences. Par la suite, il est assez aisé de montrer ces instants aux participants afin qu'ils puissent apprendre à corriger leurs gestes. Une illustration du tri des exemples selon leur niveau de réalisation, extraite de [Doughty et al., 2019], est présentée Figure 3.6.

Les deux méthodes présentées précédemment ont toutes utilisé une architecture siamoise pour entraîner leur réseau, avec la même fonction de coût : une fonction de coût par paire. Ce type de fonction de coût permet de n'avoir que des annotations relatives des exemples, *i.e.* quel exemple est le meilleur dans une paire donnée. Cependant son utilisation limite l'application puisqu'il est impossible de prédire les véritables scores. En effet les méthodes de classement (tri) par paires ne conduisent qu'à une valeur qui respecte l'ordre des scores, mais qui peut être très éloignée de la vérité.

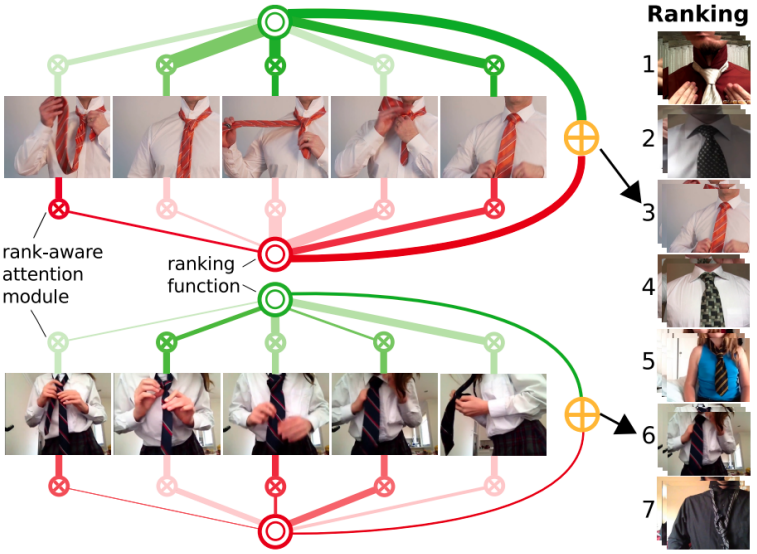


FIGURE 3.6: Illustration du processus de tri des exemples selon leur niveau de réalisation avec un module d'attention [Doughty et al., 2019].

3.2.2.2 Évaluation de Gestes Sportifs

De même que pour les gestes chirurgicaux, le développement de modèles de réseaux de neurones pour l'évaluation de gestes sportifs a été encouragé par la mise à disposition de bases de données publiques annotées regroupant des gestes ou des tâches sportives évaluées.

Pour le basket-ball, une base de données a ainsi été développée [Bertasius et al., 2016]. Composée de vidéos filmées du point de vue du joueur (une caméra a été fixée sur le torse des joueurs) et ensuite annotée par un expert, cette base de données a permis de développer des réseaux capables d'évaluer les performances d'un joueur à partir des vidéos, grâce à un réseau LSTM. De la même manière que pour la chirurgie, le réseau est entraîné dans une architecture siamoise afin de comparer deux joueurs et de trouver le meilleur. De plus, il est également possible de localiser les instants où des erreurs ont été commises. En effet, en regardant la sortie du réseau LSTM à chaque temps, cela donne une évolution du score prédit à l'instant t. Si ce score est positif, alors les gestes ont été bien réalisés et sinon des fautes ont été commises.
Une autre base de données de sport, présentée Section 3.3.2, a fait l'objet de nombreux travaux. Tout d'abord, Parmar *et al.* se sont concentrés uniquement sur des vidéos tirées des jeux olympiques de deux sports : le plongeon à 10m et le patinage artistique [Parmar and Morris, 2017]. Comme évoqué précédemment, traiter des vidéos avec un réseau convolutionnel 3D nécessite un grand nombre de vidéos annotées. Pour palier le manque de vidéos dans la base de données, un modèle pré-entraîné [Tran et al., 2015] est d'abord utilisé afin d'extraire des caractéristiques spatio-temporelles pertinentes. Cela permet de réduire le nombre de dimensions décrivant la vidéo.

Pour réaliser l'évaluation, deux méthodes sont proposées :

— La première consiste à prédire le score une fois que toute la vidéo est passée au travers du LSTM. Cette méthode amène donc uniquement à un score global.

— La deuxième méthode consiste à associer à chaque segment de la vidéo, une partie du score et à l'incrémenter à chaque nouveau segment. Ainsi le calcul de l'erreur pour l'apprentissage ne se fait plus uniquement sur le score global, mais à chaque segment de vidéo et donc à chaque fragment de score.

Le processus de calcul du score incrémental est présenté Figure 3.7.

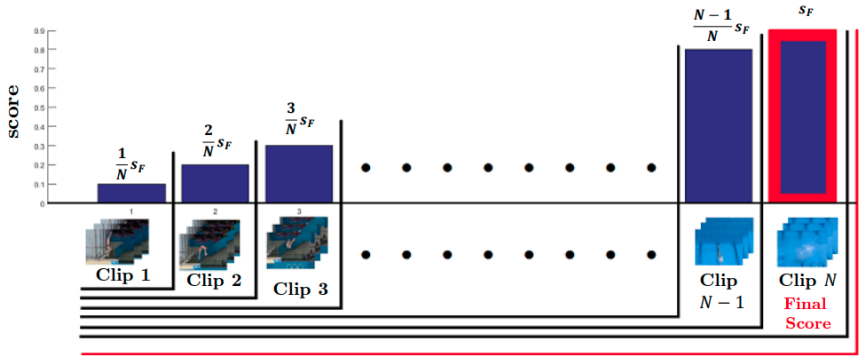


FIGURE 3.7: Score Incrémental proposé par Parmar *et al.* [Parmar and Morris, 2017].

Dans la suite de leurs travaux [Parmar and Morris, 2018], Parmar *et al.* étudient plus de sports, comme le ski et le snowboard big air. En remarquant que des figures, telles que les saltos ou les vrilles, sont présentes dans tous les sports de la base de données, les auteurs ont fait l'hypothèse qu'entraîner un seul modèle pour tous donnerait de meilleurs résultats de régression que d'entraîner un modèle par sport. Cette mise en commun est effectivement bénéfique pour la régression des scores. Le modèle utilisé pour la prédiction des scores est présenté Figure 3.8.

Enfin, une dernière approche développée par Parmar *et al.* a été introduite [Parmar and Morris, 2019]. Cette approche multi-tâche se concentre sur un seul sport, le plongeon. Créer un seul réseau, qui soit capable à la fois de prédire le score, de reconnaître les figures et de générer une légende pour un saut, permet au modèle d'apprendre à extraire des caractéristiques plus générales et donc d'améliorer les résultats pour les trois tâches.

3.2.3 Bilan

Dans cette partie, nous avons présenté les méthodes existantes pour évaluer automatiquement la qualité d'un geste. Au vu de l'application finale qui est de fournir un retour d'information sur la réalisation, utiliser des caractéristiques définies manuellement, telles que les profils de vitesse et d'accélération, ne semble pas

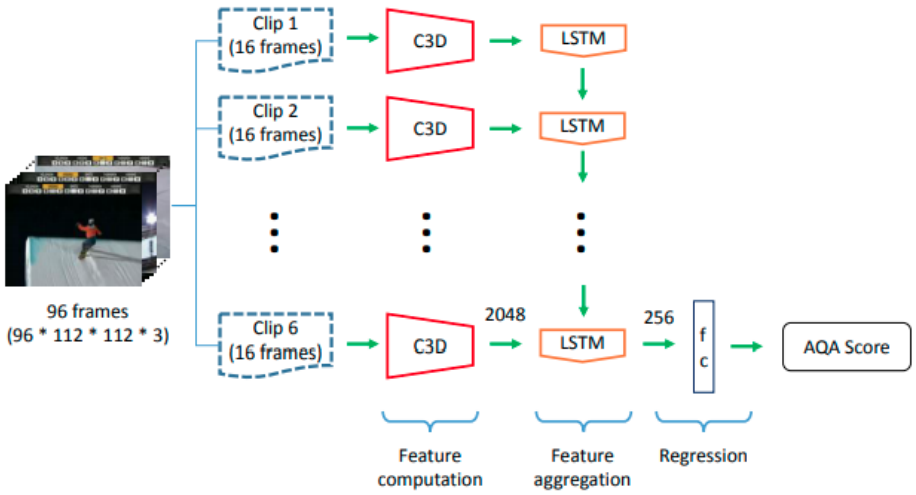


FIGURE 3.8: Modèle utilisé par Parmar *et al.* pour la prédiction de score [Parmar and Morris, 2018].

être une bonne solution. En effet, ces caractéristiques, définies manuellement par un expert pour un geste donné ne sont pas généralisables à tout type de gestes. De plus, ces méthodes dénommées "*hand-crafted features*" dans la littérature, sont bien souvent bien moins performantes que les méthodes récentes utilisant des réseaux de neurones profonds.

Se tourner vers des méthodes utilisant l'apprentissage profond semble donc être la piste à envisager : d'une part, elles permettent une meilleure généralisation à tous types de gestes, dès lors qu'une base de données a été acquise et d'autre part, elles ont démontré leur supériorité dans l'état de l'art, aussi bien sur des gestes sportifs que sur les gestes chirurgicaux.

Parmi les méthodes existantes pour régresser des scores, des architectures classiques ou utilisant des réseaux siamois ont été présentées. Les réseaux siamois permettent de comparer des exemples. Ceci permet d'une part d'augmenter la taille de la base de données (n exemples pour un réseau simple et $\frac{n^2}{2}$ pour un réseau siamois) et d'autre part, intuitivement, d'extraire des caractéristiques plus pertinentes. En effet, étudier deux réalisations de gestes avec des notes différentes en même temps permet de se focaliser sur les différences pour établir la note.

Nous avons donc décidé d'utiliser des réseaux siamois pour estimer la qualité d'un geste.

Dans la suite du chapitre, nous présentons les deux méthodes d'évaluation proposées et les résultats de ces méthodes sur deux bases de données différentes : une de gestes sportifs et une autre de gestes chirurgicaux. Ces deux bases sont très différentes dans les gestes réalisés mais aussi, dans le processus de captation des données : tandis que pour les gestes chirurgicaux, nous utilisons les données cinématiques des outils de laparoscopie, l'étude des gestes sportifs est directement

réalisée à partir de données vidéos. Montrer les performances de la méthode sur deux bases aussi différentes permettra donc de conclure quant à sa généricité.

3.3 Base de Données d'Évaluation de Gestes

Que ce soit pour l'évaluation automatique ou la reconnaissance de gestes, il est nécessaire de créer un modèle et de l'entraîner à réaliser cette tâche. L'entraînement nécessite d'avoir recours à des bases de données, annotées ou non, qui vont représenter au mieux les gestes que l'on cherche à reconnaître par exemple.

Dans cette section, nous présentons les deux bases de données utilisées dans la suite du manuscrit pour réaliser l'évaluation automatique de la qualité d'un geste.

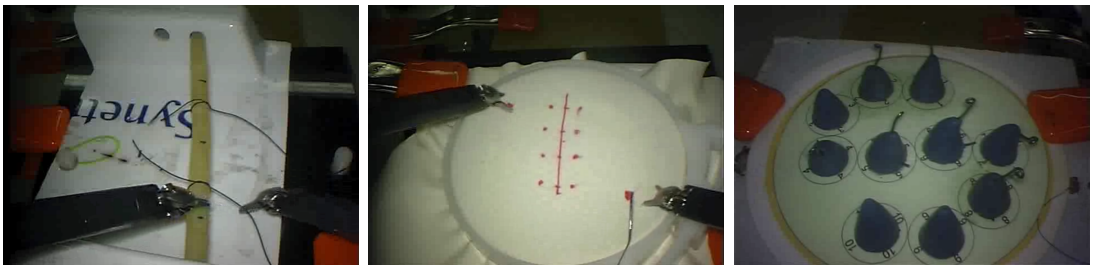
3.3.1 Gestes Chirurgicaux

La base de données JIGSAWS [Gao et al., 2014] comprend trois tâches standards figurant dans la plupart des programmes de formation à la formation en RMIS, comme illustré Figure 3.9 :

— Faire deux nœuds ;

— Faire quatre points de suture ;

— Passer une aiguille à travers un parcours.



(A) Faire deux nœuds (B) Suture (C) Parcours d'aiguille

FIGURE 3.9: Illustration des 3 tâches présentes dans la base de données JIGSAWS [Gao et al., 2014].

Toutes ces tâches sont exécutées à l'aide du système chirurgical daVinci® (Intuitive Surgical Inc., Sunnyvale, États-Unis). Ce robot télémanipulé, qui comporte une partie maître (la partie que le chirurgien manipule) et une partie esclave (la partie du robot avec les outils qui opère), a révolutionné la MIS. En effet, contrairement aux outils classiques de MIS, les pinces du robot ont deux rotations suivant les axes orthogonaux à la tige de l'outil. Cela donne donc des degrés de liberté en plus et permet de se déplacer plus librement dans le corps du patient. Cependant, le robot ne fournit aucun retour de force, le chirurgien ne ressent pas la force qu'il met lorsqu'il prend un organe, ou bien lorsqu'un de ses mouvements touche un

organe hors de son champs de vision.

FIGURE 3.10: Le robot daVinci® utilisé dans la base de données JIGSAWS

L'enregistrement des tâches est effectué grâce à l'interface du robot pour les données cinématiques et grâce aux caméras endoscopiques. Les caméras enregistrent à une fréquence de 30 Hz avec une résolution de 640x480 pixels. Les données cinématiques sont elles aussi enregistrées à une fréquence de 30HZ et sont synchronisées avec la vidéo. Chaque tâche de la base est effectuée cinq fois par huit chirurgiens de différents niveaux, ce qui donne 39 essais pour la tâche de suture, 36 pour la tâche de nœuds et 28 pour la dernière tâche du parcours d'aiguille, car des essais étaient inutilisables. La durée moyenne, la même pour les trois tâches, est de deux minutes.

Les mesures de la base de données comprennent les vidéos de la caméra endoscopique, ainsi que des données cinématiques du robot. Ces données cinématiques comprennent la position $(x,\ y,\ z)$, la vitesse $(v_x,\ v_y,\ v_z)$, la matrice de rotation (R de dimension 9), la vitesse de rotation (α, β, γ), et l'angle de préhension (θ), pour chaque manipulateur (maître et esclave), ce qui amène à un vecteur temporel de dimension 76 (19×4). Pour nos expériences, nous n'utilisons que les données cinématiques.

Chaque essai a été annoté manuellement par un chirurgien expert selon une approche OSATS (*Objective Structured Assessment of Technical Skills*) [Martin et al., 1997]. Les critères OSATS sont présentés dans le tableau 3.1. Cette approche contient six éléments différents notés sur une échelle de Likert, de 1 à 5. À partir de ces scores, un score global de classement (GRS) est calculé en additionnant les six scores précédents, *i.e.* le GRS va de 5 à 30. Nous proposons ici d'estimer automatiquement ce score à partir des données cinématiques.

Dans les annotations, les tâches sont également découpées en sous-gestes. Ainsi, chaque essai est étiqueté avec l'index du sous-geste effectué G_1 à G_{15}. Tous les sous-gestes ne sont pas présents dans toutes les tâches.
Les sous-gestes et leurs descriptions sont présentés Table 3.2.

Élement à Évaluer	Echelle de Notation
Respect des Tissus	1 - Trop de force lors de la manipulation des tissus 3 - Manipulation soigneuse des tissus mais cause parfois des dommages 5 - Manipulation adéquate des tissus
Prise en Main de l'Aiguille	1 - Gestes incertains et enchevêtrements répétés du fil 3 - La majorité des noeuds sont bien placés avec une tension adéquate 5 - Excellent contrôle de l'aiguille et du fil
Temps et Mouvement	1 - Trop de mouvements inutiles 3 - Mouvements efficaces malgré quelques mouvements inutiles 5 - Efficacité maximale et aucun mouvement inutile
Fluidité de l'Opération	1 - Nombreuses pauses pour penser aux prochains mouvements 3 - Réussit à planifier quelques mouvements à l'avance 5 - Planification parfaite de l'opération avec des transitions efficaces entre les mouvements
Performance Globale	1 - Novice 3 - Compétent 5 - Très compétent et experimenté
Qualité Finale de la Réalisation	1 - Novice 3 - Compétent 5 - Très compétent et experimenté

TABLE 3.1: Tableau des critères pour l'attribution du score OSATS

3.3.2 Gestes Sportifs

La base de données de gestes sportifs est la base AQA-7 [Parmar and Morris, 2018]. Dans cette base de données, acquise lors des Jeux Olympiques d'hiver ou d'été, 7 sports sont présents, allant du plongeon au ski, pour un total de 1106 vidéos. Les détails de la base sont présentés Table 3.3.

Dans chaque vidéo, une seule figure est exécutée, sauf pour le trampoline où une vidéo est une séquence de figures. Comme cela a été fait dans [Parmar and Morris, 2018], le trampoline est exclu de nos tests. En effet, les annotations pour ce sport n'étant pas présentes dans la base de données, il est impossible de les utiliser. Les différents sports sont présentés Figure 3.11.

Index	S	PA	N	Description
G_1	X	X	X	Prendre l'aiguille avec la pince droite
G_2	X	X		Positionner l'aiguille
G_3	X	X		Passer l'aiguille à travers un tissu
G_4	X	X		Transférer l'aiguille de la pince gauche à la droite
G_5	X	X		Aller au centre avec l'aiguille
G_6	X	X		Tirer la suture avec la pince gauche
G_7	X			Tirer la suture avec la pince droite
G_8	X	X		Orienter l'aiguille
G_9	X			Utiliser la pince droite pour resserrer la suture
G_{10}	X			Tirer le fil pour le libérer
G_{11}	X	X	X	Laisser tomber le fil et se déplacer vers les points finaux
G_{12}			X	Prendre l'aiguille avec la pince gauche
G_{13}			X	Faire une boucle autour de la main droite
G_{14}			X	Atteindre la suture avec la pince droite
G_{15}			X	Tirer la suture avec les deux pinces

TABLE 3.2: Description des sous-gestes inclus dans la Suture (S), le Passage de l'Aiguille (PA) et les Nœuds (N) comme proposé par [Gao et al., 2014].

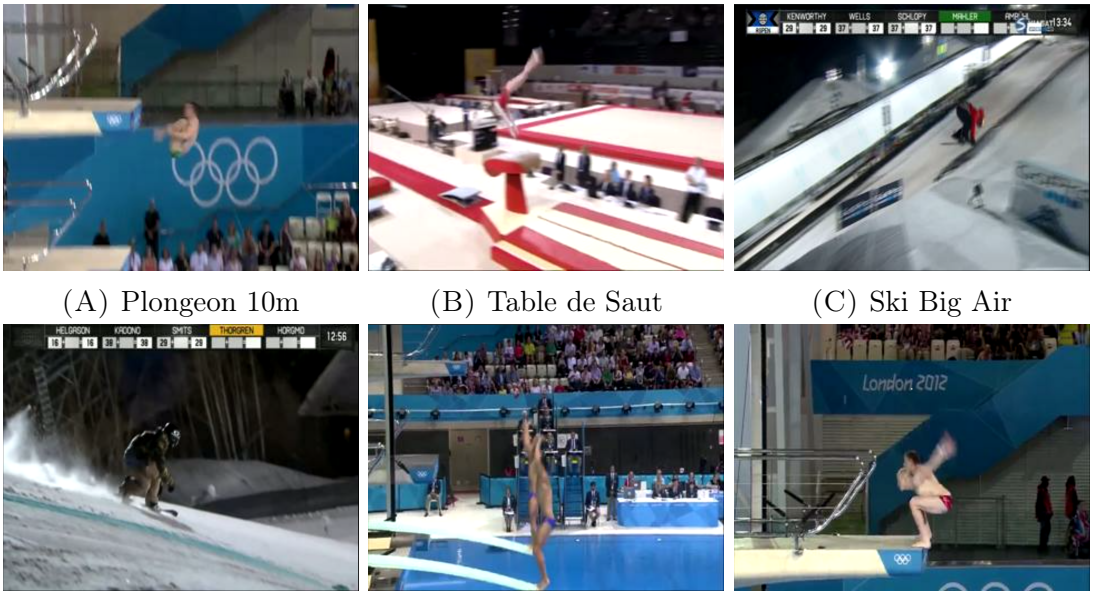
Sport	Nombre d'exemples	Longueur Moyenne (Échantillons)	Variation des Scores
Plongeon 10m	370	97	21.60 - 102.60
Table de Saut	176	87	12.30 - 16.87
Ski Big Air	175	132	8 - 50
Snowboard Big Air	206	122	8 - 50
Plongeon Sync. 3m	88	156	46.20 - 104.88
Plongeon Sync. 10m	91	105	49.80 - 99.36
Trampoline	83	634	6.72 - 62.99

TABLE 3.3: Description des différents sports de la base de données AQA-7 [Parmar and Morris, 2018].

Afin de palier les différences de durée entre les différents sports et essais, toutes les vidéos de la base de données ont été rééchantillonnées par les auteurs à une longueur fixe de 103 images. Les scores dépendent à la fois de la réalisation et de l'atterrissage. Chaque sport a sa propre échelle de score. Par exemple, les scores pour le plongeon vont de 45 à 100 tandis que les scores en gymnastique vont de 10 à 20.

3.4 Les Réseaux Siamois pour l'Évaluation de la Qualité d'un Gestes

L'évaluation automatique de la qualité d'un geste passe par deux grandes étapes : extraire des caractéristiques spatio-temporelles et entraîner un modèle statistique à prédire le niveau de qualité du geste. En utilisant des réseaux de



(A) Plongeon 10m (B) Table de Saut (C) Ski Big Air

(D) Snowboard Big Air (E) Plongeon Synchronisé 3m (F) Plongeon Synchronisé 10m

FIGURE 3.11: Illustration des sports de la base de données AQA-7.

neurones, il est possible de rassembler ces deux étapes en une. Les réseaux siamois ayant montré de bons résultats à la fois sur des gestes chirurgicaux mais également sur des gestes sportifs [Doughty et al., 2019, 2018; Li et al., 2019], nous nous sommes orientés vers cette architecture. Une représentation de l'architecture d'un réseau siamois est présentée Figure 3.12.

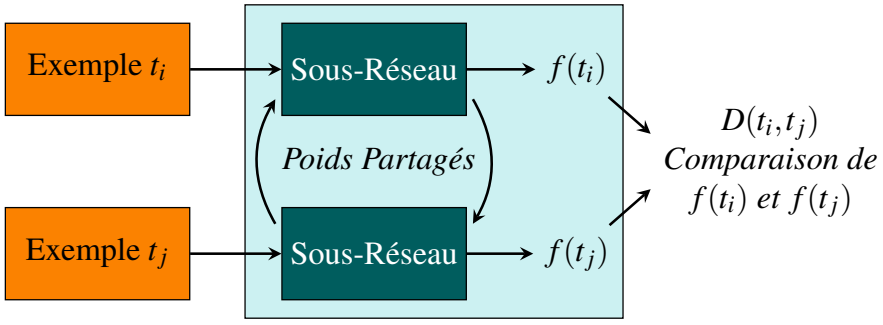


FIGURE 3.12: Illustration d'un réseau siamois.

Deux exemples t_i et t_j sont présentés en entrée de l'architecture. Ils passent chacun dans le même sous-réseau qui fournit les sorties $f(t_i)$ et $f(t_j)$ respectivement. Les sorties de ces deux branches sont ensuite comparées pour arriver à la sortie finale $D(t_i, t_j)$ du réseau. Le but du réseau siamois étant de comparer les exemples, seule $D(t_i, t_j)$ est supervisée pour déterminer lequel des exemples d'entrée est le meilleur. Ainsi, les sorties intermédiaires du sous-réseau $f(t_i)$ et $f(t_j)$ sont valides dès lors qu'elles respectent l'ordre de qualité des exemples d'entrée.

3.4.1 Fonction de Coût pour Réseaux Siamois

Ainsi, et comme présenté Section 2.2.1.4, les réseaux siamois permettent de comparer deux entrées. En triant les sorties intermédiaires du réseau $f(t_i)$, ceci permet d'aboutir à un classement des exemples par ordre de qualité. Du coté des annotations de la base de données, ce type d'architecture ne demande pas d'attribuer un score à chaque exemple, mais de déterminer, pour chaque paire d'exemples, lequel est le meilleur. Ces annotations relatives sont bien plus faciles à obtenir et demandent moins d'expertise.
Ainsi, la sortie idéale $D(t_i, t_j)$ du réseau siamois est, pour l'évaluation :

$$D(t_i, t_j) = \begin{cases} 1 & \text{si } t_i \text{ est meilleur que } t_j \\ -1 & \text{si } t_j \text{ est meilleur que } t_i \\ 0 & \text{s'il y a égalité.} \end{cases}$$

Compte tenu de cette sortie, les réseaux siamois sont entraînés en utilisant une fonction de coût par paire [Wang et al., 2014; Yao et al., 2016; Doughty et al., 2019; Li et al., 2019]. En considérant uniquement les paires d'entrées (t_i, t_j) telles que $D(t_i, t_j) = 1$, la fonction de coût par paire $L_{Siam_{Pair}}$ est définie par :

$$L_{Siam_{Pair}} = \sum_{(t_i,t_j)/D(t_i,t_j)=1} max\big(0, m - f(t_i) + f(t_j)\big) \tag{3.1}$$

où m, la marge, est généralement égale à 1. Elle permet d'imposer une grande différence entre les valeurs de $f(t_i)$ et $f(t_j)$ dès lors que l'exemple t_i et meilleur que t_j. Ceci force le réseau à se positionner lors de la comparaison, plutôt que de donner à chaque fois des valeurs très similaires. En effet, la paire (t_i, t_j) arrête de contribuer à la fonction de coût dès lors que $f(t_i) - f(t_j) > 1$ et continue d'y contribuer sinon.
Si la contrainte sur les paires $D(t_i, t_j) = 1$ est libérée, toutes les paires d'entrée sont considérées et la fonction de perte $L_{Siam_{Pair}}$ peut être réécrite :

$$L_{Siam_{Pair}} = \sum_{(t_i,t_j)} max\big(0, m - sgn(D(t_i, t_j))(f(t_i) - f(t_j))\big) \tag{3.2}$$

où $sgn()$ est la fonction de signe.
Avec cette fonction perte, le modèle produit des scores $f(t_i)$, qui peuvent être utiles pour classer les exemples. Cependant, cette fonction de perte n'estime qu'un ordre de classement, de sorte que la mesure de qualité $f(t_i)$ peut être éloignée de la valeur réelle du score, s_i. Comme illustré Figure 3.13, il y a une multitude de possibilités dans la prédiction des scores. En effet, si la seule contrainte est de respecter l'ordre de classement des exemples, alors il est possible de prédire des valeurs très proches les unes des autres (fonction f_1 Figure 3.13), ou bien de respecter la différence de

scores, mais de prédire des valeurs très éloignées de la vérité terrain (fonction f_2 Figure 3.13).

t_i	s_i	$f_1(t_i)$	$f_2(t_i)$
t_1	4	0	6
t_2	9	1	11
t_3	10	2	12
t_4	20	3	22

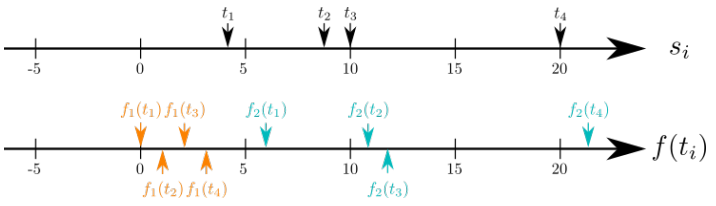


FIGURE 3.13: Illustration des problèmes induits par la fonction de coût $L_{Siam_{Pair}}$ et des différences autorisées entre les scores estimés $f(t_i)$ et les scores réels dans deux cas f_1 et f_2.

Ainsi, dès que $sgn(D(t_i, t_j))(f(t_i) - f(t_j)) > m$, la paire $(t_i; t_j)$ respecte les contraintes d'ordre et cesse donc de contribuer à la perte, même si les scores sont très éloignés des scores réels.

Comme notre objectif est d'estimer les scores s_i, nous proposons de changer la sortie du réseau siamois et d'estimer l'écart de score entre t_i et t_j, $\Delta_{ij} = s_i - s_j$. Pour y parvenir, nous proposons d'utiliser comme fonction de perte, l'erreur quadratique moyenne (*Mean Square Error - MSE*) :

$$L_{Siam_{MSE}} = \sum_{(t_i,t_j)} \left(f(t_i) - f(t_j) - \Delta_{i,j}\right)^2 \tag{3.3}$$

L'estimation $f(t_i)$ du modèle siamois respecte bien maintenant le classement mais aussi l'écart entre les scores. Cependant, les estimations peuvent être décalées dans l'espace de sortie. En effet rien ne contraint la sortie $f(t_i)$ à respecter la valeur des scores, ce qui peut entraîner une constante additive entre la sortie et la vérité-terrain, comme illustré par la fonction f_2 Figure 3.13. Nous proposons deux solutions pour remédier à ce problème. La première est un fine-tuning du réseau et la seconde consiste à apprendre un réseau siamois multi-tâches.

3.4.2 Réseau Siamois à Sorties Multiples

La première solution consiste à utiliser une fonction de coût multiple. En effet, en rajoutant une contrainte pendant l'apprentissage sur une des deux branches du réseau siamois, les sorties $f(t_i)$ et $f(t_j)$ vont correspondre aux scores réels et non plus seulement conserver le classement des exemples.

L'objectif de ce réseau siamois à deux sorties est à la fois de comparer les sorties des deux sous-réseaux, mais aussi d'utiliser la mesure estimée $f(t_i)$ pour prédire le véritable score s_i. La fonction de coût ajoutée L_{Single} est également fondée sur la MSE :

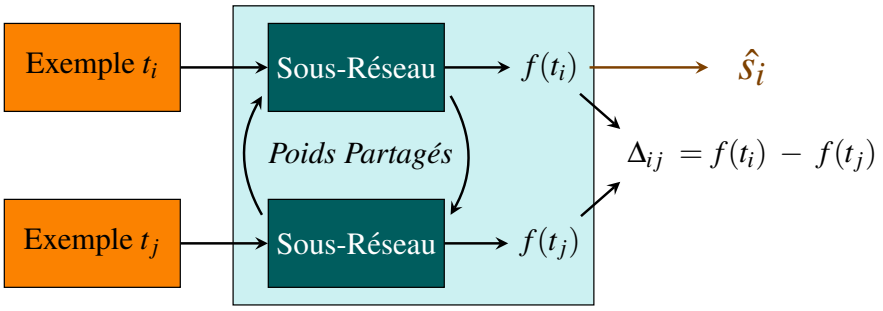


FIGURE 3.14: Illustration de la fonction du réseau siamois à sorties multiples.

$$L_{Single} = \sum_{t_i} \big(f(t_i) - s_i\big)^2 \tag{3.4}$$

Une seule sous-sortie est ajoutée à la sortie globale du réseau, puisque les poids et les paramètres sont partagés entre les deux sous-réseaux. L'ajout des deux sorties à la sortie globale pourrait conduire à un entraînement inefficace, en raison de conflits lors de la rétro-propagation du gradient. Pour former le réseau, la fonction de coût L_{Single} est ajoutée à l'une des deux fonctions de coût siamoises, soit $L_{Siam_{MSE}}$ ou $L_{Siam_{Pair}}$.

$$L = (1 - \lambda)L_{Siam} + \lambda L_{Single} \tag{3.5}$$

Un paramètre de pondération λ est introduit afin d'obtenir le meilleur compromis entre le classement et la prédiction du score. Avec cette fonction de coût globale, la mesure estimée $f(t_i)$ n'est plus décalée du score réel s_i. Une fois le réseau siamois entraîné, seul le sous-réseau est utilisé pour prédire le score $f(t_i)$ associé à l'exemple t_i.

3.4.3 *Finetuning* de Réseau Siamois

Pour enlever cette constante additive, une seconde solution consiste à réentraîner le réseau afin qu'il se centre sur la vérité-terrain. Un *finetuning* est souvent utilisé dans le cas dans de base de données ayant une taille réduite. Par exemple, Gong *et al.*, ne disposant que de peu de données réelles, entraînent un réseau sur des données de simulation dans un premier temps et dans un second temps sur les données réelles [Gong et al., 2019]. Ainsi le réseau apprend à extraire des caractéristiques pertinentes, à partir du premier jeu de données. Une fois cela réalisé, il suffit juste d'ajuster finement les poids des neurones afin que des résultats similaires soient obtenus sur les données réelles.
En s'inspirant de ce principe, nous avons décidé de réaliser un fine-tuning d'une des branches du réseau siamois. Dans un premier temps, le réseau siamois est entraîné

de manière classique avec une des deux fonctions de coût présentées Section 3.4.1. Dans un second temps, une branche est extraite de l'architecture siamoise et les poids de la dernière couche sont ajustés lors d'un second entraînement sur les entrées individuelles avec leurs scores. Les poids des autres couches sont gelées, lors de ce second apprentissage. La fonction de coût utilisée pour apprendre cette dernière couche est la MSE, comme représenté Figure 3.15.

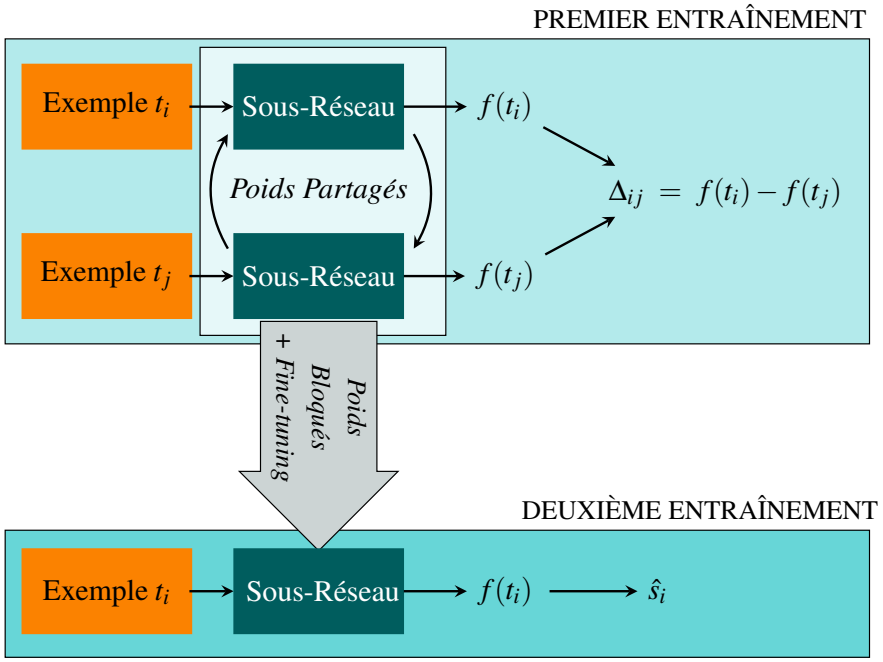


FIGURE 3.15: Illustration de la méthode de *finetuning* d'un réseau siamois.

Ce second apprentissage permet d'ajuster les poids de la dernière couche du réseau afin qu'il ne prédise pas simplement une mesure respectant le classement ou l'écart entre les scores, mais les scores en eux-même.

3.5 Résultats

Les approches présentées Section 3.4 ont été testées avec deux types de sous-réseaux : un réseau convolutif temporel et un réseau fondé sur les LSTM. Deux applications sont également étudiées : l'évaluation du sport sur la base de données AQA-7 [Parmar and Morris, 2018] et l'évaluation des compétences chirurgicales sur la base de données JIGSAWS [Gao et al., 2014]. Ces applications nous permettent de travailler sur différents signaux d'entrée tels que des données cinématiques ou des vidéos. Afin de mettre en évidence les avantages de la méthode proposée, nous la comparons avec d'autres méthodes :

— Une simple régression sur le score en utilisant la fonction de coût MSE ;

— Un réseau siamois avec la fonction de coût par paire (équation 3.2) ;

— Un réseau siamois avec la fonction de coût MSE (équation 3.3) ;

— Un réseau siamois avec la fonction de coût par paire et un *finetuning* (équation 3.2) ;

— Un réseau siamois avec la fonction de coût MSE et un *finetuning* (équation 3.3).

— Un réseau siamois avec la fonction de coût par paire et un apprentissage multi-tâches (équation 3.2) ;

— Un réseau siamois avec la fonction de coût MSE et un apprentissage multi-tâches (équation 3.3).

Pour évaluer à la fois la prédiction des scores et le classement des scores, deux mesures d'évaluation sont utilisées :

— La Racine carrée de l'erreur quadratique moyenne (*Root Mean Square Error - RMSE*) qui représente bien l'erreur de prédiction des scores est définie par :

$$RMSE = \sqrt{\sum_i \frac{(\hat{s}_i - s_i)^2}{n}} \tag{3.6}$$

Cette valeur doit être minimale, afin de montrer une prédiction des scores précises.

— La précision par paire, Acc_{pair}, définie comme le pourcentage de paires correctement classées dans l'ensemble de test :

$$Acc_{pair} = \frac{N_{paires\ correctes}}{N_{total}} \tag{3.7}$$

Cette mesure est représentative de la validité du classement des paires d'exemples, et doit être la plus grande possible afin de refléter un bon classement.

Chaque expérience a été réalisée dix fois, la moyenne et l'écart-type sont présentés dans les résultats.

3.5.1 Gestes Chirurgicaux

Pour évaluer des gestes chirurgicaux, il a été décidé d'utiliser la base de données publique JIGSAWS, présentée Section 3.3.1. Cela permet ainsi de comparer notre approche avec l'état de l'art.

3.5.1.1 Modèle utilisé dans le Réseau Siamois

Au vu du peu d'exemples présents dans la base JIGSAWS (environ 40 essais par tâche), entraîner un réseau à évaluer un geste à partir du flux vidéo paraît compliqué. C'est pourquoi nous avons décidé d'utiliser les données cinématiques disponibles. En entrée du réseau, nous retrouvons donc un signal temporel t_i de

dimension $1 \times 76 \times T$ où T est la longueur temporelle et 76 représente le nombre de caractéristiques extraites présentées Section 3.3.1. Chacune des 76 caractéristiques est normalisée en soustrayant sa valeur moyenne et en la divisant par son écart type.

Pour extraire les caractéristiques de ces signaux multi-dimensionnels, le sous-réseau comprend des couches de convolution 1D, des couches de *max-pooling* et une couche entièrement connectée.

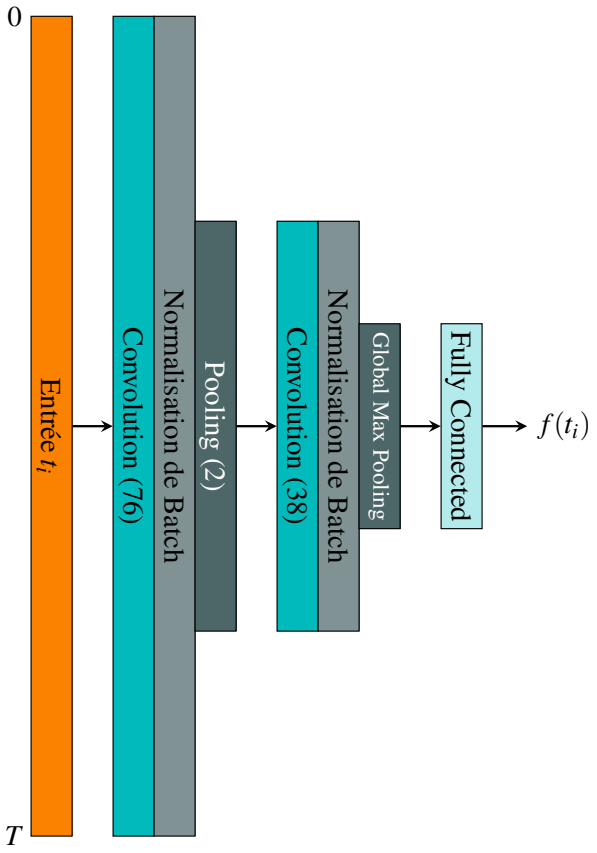


FIGURE 3.16: Représentation du réseau utilisé pour l'évaluation automatique de la qualité de gestes chirurgicaux.

Plus particulièrement, le sous-réseau est composé de deux couches à convolution 1D avec respectivement 76 et 38 filtres de taille 2 (Figure 3.16). Entre les deux couches, un *max-pooling* et une normalisation par batch sont appliqués, afin de ne conserver que la moitié de la dimension temporelle d'entrée et d'éviter d'avoir des entrées pour les couches de convolutions qui aient des valeurs trop importantes. Après la dernière couche de convolution, une couche de *pooling* global sur la dimension temporelle est effectuée sur chaque filtre pour obtenir une représentation indépendante de la longueur du signal d'entrée. Le vecteur latent possède donc 38 dimensions. Le réseau se termine par une couche entièrement connectée qui permet de prédire la mesure $f(t_i)$. Les poids de chaque couche sont initialisés avec un bruit gaussien de moyenne zéro et d'écart-type de 0.01. La régularisation L_2

est utilisée sur les poids de chaque couche avec un coefficient de 0.1, pour limiter le sur-apprentissage.

Ce sous-réseau est utilisé dans chaque branche du réseau siamois et le réseau global est entraîné pendant 100 itérations. L'algorithme Adam [Kingma and Ba, 2015] est utilisé pour la rétropropagation du gradient en utilisant un taux d'apprentissage initial de 0.001 et une taille de batch de 100 paires.

3.5.1.2 Résultats

Pour déterminer la meilleure pondération pour la fonction de coût globale, des tests ont été effectués en faisant varier λ de 0 à 1. Les résultats (moyenne de 10 apprentissages) de RMSE et de précision par paire sont présentés Figure 3.17.

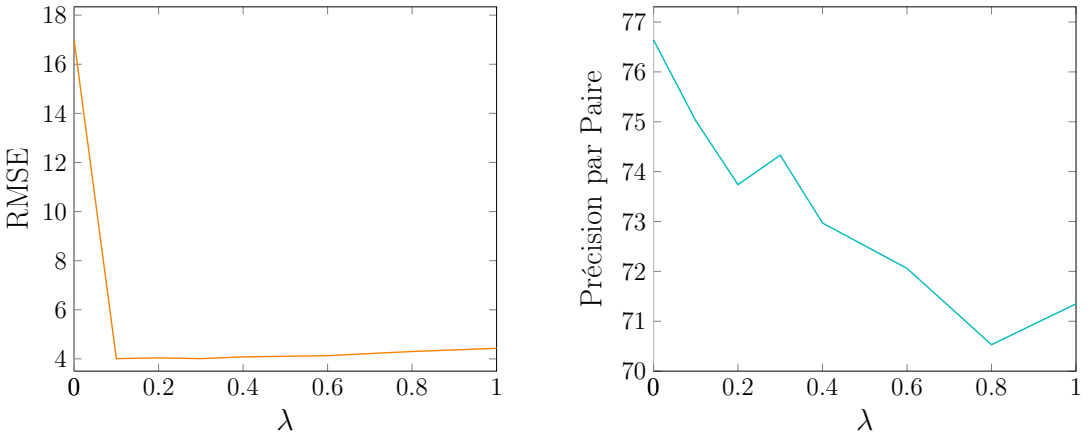


FIGURE 3.17: RMSE et précision par paire selon le paramètre de pondération λ de la fonction de coût pour les gestes chirurgicaux.

Lorsque le coefficient de pondération $\lambda = 0$, le fonction de coût ne comprend que la fonction de coût siamoise, il est donc normal d'obtenir de mauvais résultats en RMSE et de bons résultats en précision par paire. Dès que ce coefficient augmente, la fonction de coût prend en compte la prédiction des scores (Équation 3.5) et les résultats en RMSE s'améliorent. Il est important de noter, que même lorsque la fonction de coût n'inclut plus la fonction de coût siamoise ($\lambda = 1$), les résultats en RMSE restent très bons mais les résultats de précision par paire sont eux dégradés. Le meilleur compromis entre la RMSE et la précision est atteint lorsque $\lambda = 0.1$. Cette valeur est utilisée pour le reste des tests.

En raison du petit nombre d'occurrences dans cette base de données (environ 40 exemples pour chacune des 3 tâches), les travaux précédents étaient principalement axés sur la prédiction du niveau de compétence, *i.e.* une tâche de classification à 3 classes : Expert, Novice ou Intermédiaire. En effet, parallèlement à l'annotation présentée Section 3.3.1, chaque chirurgien s'est auto-proclamé Novice, Intermédiaire ou Expert. Parmi les quelques travaux sur la prédiction du Global Rating Score (GRS), la plupart d'entre eux effectuent un classement par paire en

utilisant le réseau siamois avec une fonction de coût par paire et évaluent leur méthode en utilisant la précision par paire. D'autre part, chacune des 3 tâches a été traitée indépendamment.

Le premier test réalisé a pour but à mesurer l'intérêt des réseaux siamois et la pertinence de la fonction coût MSE qui vise à faire respecter l'écart de score entre les exemples. Les résultats sont présentés Tableau 3.4.

		Regression	**Réseau Siamois $L_{Siam_{Pair}}$**	**Réseau Siamois $L_{Siam_{MSE}}$**
Suture	*RMSE*	**5.05 ± 0.29**	19.42 ± 0.31	18.97 ±0.29
	Acc_{pair}	68.77 ± 3.1	**76.49 ± 2.3**	75.73 ±2
Parcours de l'Aiguille	*RMSE*	**4.94 ± 0.23**	15.26 ± 0.54	15.14 ±0.39
	Acc_{pair}	63.93 ± 4.4	75.48 ± 3.6	**76.79 ±2.6**
Noeuds	*RMSE*	**4.32 ± 0.33**	15.50 ± 0.27	14.63 ±0.21
	Acc_{pair}	69.17 ± 3.6	76.39 ± 2	**79.86 ±3.3**
Moyenne	*RMSE*	**4.77 ± 0.28**	16.73 ± 0.37	16.24 ±0.23
	Acc_{pair}	67.29 ± 3.7	76.12 ± 2.7	**77.46 ±2.6**

TABLE 3.4: Comparaison des fonctions de coût siamoises avec une régression.

Concernant la RMSE, qui représente l'erreur de prédiction des scores, la méthode la plus efficace est une simple régression. On retrouve ainsi les explications de la Section 3.4.1 qui montrent que les deux réseaux siamois étudiés ici ne permettent pas de retrouver les scores réels.
Concernant maintenant la précision par paire, qui représente le pourcentage de paires bien classées, entraîner le même réseau dans une architecture siamoise permet d'améliorer considérablement les résultats. D'autre part, même si la fonction de coût MSE ne permet pas encore de retrouver les scores réels, elle permet d'améliorer la précision par paire et donc, le classement des exemples. Ainsi, la prédiction des écarts de scores aide le réseau pour la tâche de classement des exemples. Ceci s'explique intuitivement par l'apport d'annotations bien plus riches : savoir qu'un exemple a un score supérieur de 3 à un autre exemple est bien plus riche que de simplement savoir qu'il est supérieur.

Pour améliorer les résultats d'estimation des scores, les deux méthodes présentées dans la Section 3.4 sont testées sur cette base de données. Les résultats sont présentés Tableau 3.5.

Comme prévu, l'entraînement du modèle avec une architecture siamoise donne une meilleure capacité de classement (Acc_{pair}) qu'une simple régression. Toutefois, en ce qui concerne la prédiction des scores (RMSE), les deux réseaux formés dans une architecture siamoise classique obtiennent de loin les pires résultats. Cela est logique, car leur objectif est de prédire les écarts de scores ($L_{Siam_{MSE}}$) ou de classer

			Classique		Multi-tâches		*Finetuning*	
		Régression	Réseau Siamois $L_{Siam_{Pair}}$	Réseau Siamois $L_{Siam_{MSE}}$	Réseau Siamois $L_{Siam_{Pair}}$	Réseau Siamois $L_{Siam_{MSE}}$	Réseau Siamois $L_{Siam_{Pair}}$	Réseau Siamois $L_{Siam_{MSE}}$
Suture	*RMSE*	5.05 ± 0.29	19.42 ± 0.31	18.97 ±0.29	4.65 ± 0.17	**4.33 ± 0.2**	4.73 ± 0.23	4.93 ± 0.36
	Acc_{pair}	68.77 ± 3.1	**76.49 ± 2.3**	75.73 ±2	71.59 ± 2.02	74.32 ± 2.4	72.87 ± 3.5	69.12 ± 4.6
Parcours de l'Aiguille	*RMSE*	4.94 ± 0.23	15.49 ± 0.54	14.63 ± 0.39	3.75 ± 0.15	**3.38 ± 0.18**	4.34 ± 0.24	4.46 ± 0.34
	Acc_{pair}	63.93 ± 4.4	76.38 ± 3.6	76.79 ±2.6	74.42 ± 2	**77.9 ± 2.04**	70.69 ± 1	72.86 ± 2.7
Noeuds	*RMSE*	4.32 ± 0.33	15.25 ± 0.27	15.14 ±0.21	4.69 ± 0.13	**4.31 ± 0.18**	4.53 ± 0.28	4.05 ± 0.24
	Acc_{pair}	69.17 ± 3.6	75.47 ± 2	**79.86 ±3.3**	74.43 ± 2.9	72.84 ± 2.9	70.95 ± 1.2	71.53 ± 3.3
Moyenne	*RMSE*	4.77 ± 0.28	16.73 ± 0.37	16.24 ±0.23	4.33 ± 0.15	**4.13 ± 0.18**	4.53 ± 0.25	4.48 ± 0.31
	Acc_{pair}	67.29 ± 3.7	76.12 ± 2.7	**77.46 ±2.6**	72.27 ± 2.31	75.02 ± 2.45	71.5 ± 1.9	71.17 ± 3.5

TABLE 3.5: Comparaison des méthodes pour les tâches chirurgicales.

les essais ($L_{Siam_{Pair}}$) et non d'estimer les scores.
L'utilisation d'une fonction de coût multi-tâches et du *finetuning* se traduit par des RMSE plus faibles et donne de meilleurs résultats qu'une simple régression. Cependant, la capacité de classement est légèrement réduite, probablement en raison des fortes contraintes de prédiction des scores ajoutées par la fonction de perte L_{single} pour la méthode de fonction de coût multiple. En effet, avec la fonction de perte multiple, un compromis entre le classement et la prédiction des scores est trouvé.

Un autre résultat important sur le réseau siamois est que l'utilisation d'une fonction de coût MSE sur les écarts améliore les résultats par rapport à la fonction de coût standard de la littérature fondée sur le classement, même pour une application de classement (Acc_{pair}).

Nous comparons également nos résultats avec les travaux de la littérature Table 3.6.

	Précision par paire Acc_{pair}
Doughty *et al.*[Doughty et al., 2018]	70.2
Li *et al.* [Li et al., 2019]	73.1
Nous (*Finetune* + $L_{Siam_{MSE}}$)	71.17
Nous (Multi-tâches + $L_{Siam_{MSE}}$)	**75.02**

TABLE 3.6: Précision par paire pour les méthodes de l'état de l'art.

Comme on peut le constater, le réseau multi-tâches proposé dépasse l'état de l'art pour la tâche de classement des scores, même si le but recherché n'était pas celui-ci. Nous ne présentons pas une comparaison concernant la vraie estimation des scores car cette tâche n'a pas été traitée dans la littérature sur cette base de données, les réseaux siamois classiques n'étant pas adaptés.

3.5.2 Gestes Sportifs

Pour les gestes sportifs, notre choix s'est porté sur la base AQA-7 [Parmar and Morris, 2018], présentée Section 3.3.2. Des modèles ayant déjà été entraînés sur cette base, il est aisé de comparer nos résultats à ceux de l'état de l'art. Afin de pouvoir comparer facilement les résultats, les scores de tous les sports ont été normalisés, en enlevant la valeur moyenne et en la divisant par l'écart type, *i.e.*, les scores se situent pour la plupart entre -1 et 1.

3.5.2.1 Modèle utilisé dans le Réseau Siamois

L'entrée du modèle est composée uniquement des vidéos. Les vidéos comportant de nombreuses dimensions et la taille de la base de données étant assez faible, entraîner directement un réseau sur les vidéos est délicat. Aussi, nous extrayons des vecteurs de caractéristiques à partir des vidéos en utilisant le réseau C3D créé par Tran *et al.* [Tran et al., 2015], comme [Parmar and Morris, 2018]. Le réseau C3D a prouvé son efficacité dans la préservation des informations temporelles et spatiales dans les vidéos, puisqu'il surpasse des réseaux CNN 2D, lorsqu'il est utilisé pour des tâches de classification [Tran et al., 2015]. En outre, ce modèle a été appris sur la base de données Sports-1M [Karpathy et al., 2014], qui comprend de nombreux sports qui sont également présents dans la base de données AQA-7. Pour avoir un vecteur latent représentatif, les vecteurs sont récupérés de la sixième couche de neurones du réseau C3D, ce qui amène à un vecteur de dimension 4096. Ce vecteur est extrait à partir de segments de vidéo de taille 16, découpés grâce à une fenêtre glissante et un pas de la taille du segment, soit 16. Les images de la base étant plus grandes, que celles sur lesquelles le réseau C3D a été entraîné, elles sont donc redimensionnées. $((320 \times 240 pixels) \rightarrow (172 \times 128 pixels))$

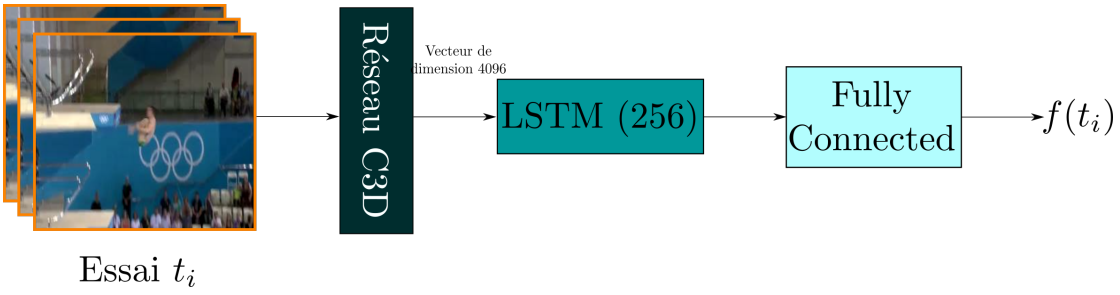


FIGURE 3.18: Illustration du modèle utilisé pour la régression des scores pour les gestes sportifs.

Un réseau LSTM [Hochreiter and Schmidhuber, 1997], comme dans [Parmar and Morris, 2018], est ensuite utilisé pour régresser les scores. Cette couche LSTM prend les caractéristiques C3D en entrée et produit un vecteur latent de dimension 256. Une couche de neurones entièrement connectés est ensuite ajoutée pour prédire la mesure $f(t_i)$. Le sous-réseau correspondant est illustré Figure 3.18.

La régularisation L_2 est utilisée sur les poids de chaque couche avec un coefficient de 0.1, pour limiter le sur-apprentissage. Ce sous-réseau est utilisé dans chaque branche du réseau siamois et le réseau global est entraîné pendant 100 itérations. L'algorithme Adam [Kingma and Ba, 2015] est utilisé pour la rétropropagation du gradient en utilisant un taux d'apprentissage initial de 0.001 et une taille de batch de 15 paires.

3.5.2.2 Résultats

Pour déterminer la meilleure pondération, des tests ont été effectués en faisant varier λ de 0 à 1. Les résultats moyens de la RMSE et de la précision par paire sont présentés Figure 3.19.

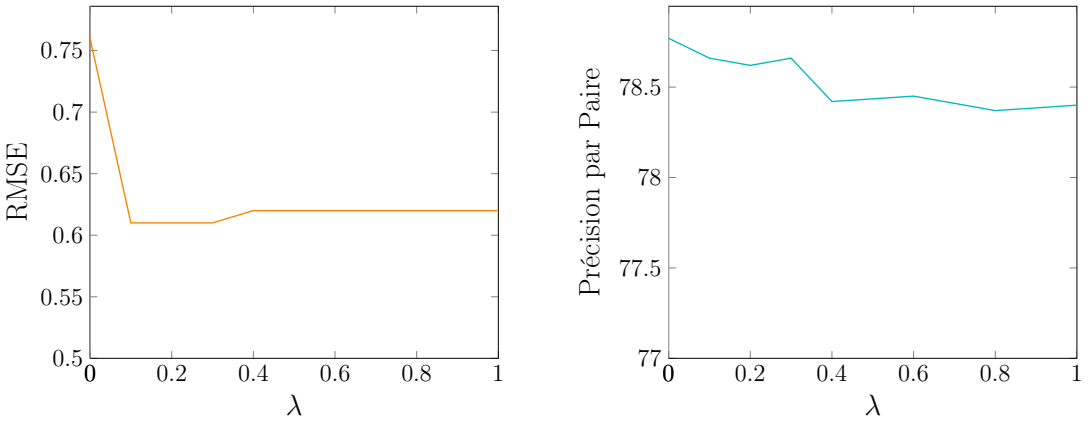


FIGURE 3.19: RMSE et précision par paire selon le paramètre de pondération λ de la fonction de coût pour les gestes sportifs.

La même tendance que pour les gestes chirurgicaux est observée sur les gestes sportifs. Lorsque le coefficient de pondération est nul ($\lambda = 0$), la RMSE est au plus haut de même que la précision par paire. Dès que ce coefficient est non nul, les résultats de RMSE baissent et sont donc meilleurs. Cependant les résultats de précision par paire sont eux moins bons. Enfin, lorsque $\lambda = 1$, les résultats de précision par paire sont au plus bas et ceux en RMSE ne sont pas les meilleurs, comme avec les gestes chirurgicaux. Le meilleur compromis est obtenu pour λ entre 0.1 et 0.3. Nous utilisons $\lambda = 0.1$ dans les résultats suivants.

Les méthodes ont été évaluées de la même manière que les méthodes d'évaluation des compétences chirurgicales (Section 3.5.1.2). Le premier test visant à montrer l'intérêt d'une fonction coût MSE sur les écarts de scores, sont présentés Table 3.7.

Les mêmes observations que pour la chirurgie peuvent être faites : entraîner un réseaux dans une architecture siamoise améliore la précision par paire mais augmente les erreurs de régression sur les scores, car ces réseaux ne sont pas conçus pour régresser des scores. Pour améliorer les résultats en RMSE, les deux méthodes

		Régression	Réseau Siamois $L_{Siam_{Pair}}$	Réseau Siamois $L_{Siam_{MSE}}$
Plongeon	*RMSE*	**0.71 ± 0.05**	1.1 ± 0.08	0.98 ± 0.08
	Acc_{pair}	**77.26 ± 1.01**	77.08 ± 0.96	76.41 ± 1.04
Table de Saut	*RMSE*	**0.67 ± 0.014**	1.13 ± 0.14	0.88 ± 0.07
	Acc_{pair}	75.12 ± 0.82	**77.59 ± 0.71**	77.01 ± 0.65
Ski	*RMSE*	**0.73 ± 0.012**	1.34 ± 0.15	0.87 ± 0.03
	Acc_{pair}	71.52 ± 1	**73.62 ± 0.65**	72.74 ± 0.79
Snowboard	*RMSE*	**0.72 ± 0.015**	1.04 ± 0.07	0.91 ± 0.05
	Acc_{pair}	69.30 ± 0.98	**71.26 ± 0.75**	69.81 ± 0.53
Plongeon Sync. 3m	*RMSE*	0.79 ± 0.09	**0.71 ± 0.13**	0.81 ±0.08
	Acc_{pair}	76.53 ± 3.6	87.42 ± 0.84	**87.81 ± 1.03**
Plongeon Sync. 10m	*RMSE*	**0.74 ± 0.11**	0.82 ± 0.09	**0.74 ± 0.02**
	Acc_{pair}	78.04 ± 5.03	82.60 ± 0.93	**82.92 ±1.6**
Moyenne	*RMSE*	**0.73 ± 0.05**	1.02 ± 0.11	0.87 ±0.06
	Acc_{pair}	74.63 ± 2.09	**78.26 ± 0.81**	77.78 ± 0.94

TABLE 3.7: Comparaison des fonctions de coût siamoises avec une régression.

présentées Section 3.4 sont testées sur cette base de données. Les résultats sont présentés Table 3.8.

			Classique		Multi-tâche		Fine-tuned	
		Régression	Réseau Siamois $L_{Siam_{Pair}}$	Réseau Siamois $L_{Siam_{MSE}}$	Réseau Siamois $L_{Siam_{Pair}}$	Réseau Siamois $L_{Siam_{MSE}}$	Réseau Siamois $L_{Siam_{Pair}}$	Réseau Siamois $L_{Siam_{MSE}}$
Plongeon	*RMSE*	0.71 ± 0.05	1.1 ± 0.08	0.98 ± 0.08	0.80 ± 0.01	**0.66 ± 0.03**	0.82 ± 0.24	0.67 ± 0.19
	Acc_{pair}	77.26 ± 1.01	77.08 ± 0.96	76.41 ± 1.04	**77.34 ± 0.71**	77.16 ± 1.09	76.23 ± 2.14	76.62 ± 2.19
Table de Saut	*RMSE*	0.67 ± 0.014	1.13 ± 0.14	0.88 ± 0.07	0.85 ± 0.03	**0.62 ± 0.016**	0.66 ± 0.19	0.63 ±0.19
	Acc_{pair}	75.12 ± 0.82	77.59 ± 0.71	77.01 ± 0.65	77.65 ± 0.89	**78.06 ± 0.76**	75.44 ± 2.14	77.73 ± 2.2
Ski	*RMSE*	0.73 ± 0.012	1.34 ± 0.15	0.87 ± 0.03	0.88 ± 0.02	**0.72 ± 0.014**	0.73 ± 0.21	**0.72 ±0.2**
	Acc_{pair}	71.52 ± 1	73.62 ± 0.65	72.74 ± 0.79	73.43 ± 0.63	73.91 ± 0.97	73.35 ± 2.12	**74.56 ± 2.16**
Snowboard	*RMSE*	0.72 ± 0.015	1.04 ± 0.07	0.91 ± 0.05	0.81 ± 0.004	0.73 ± 0.012	**0.70 ± 0.20**	0.73 ± 0.22
	Acc_{pair}	69.30 ± 0.98	**71.26 ± 0.75**	69.81 ± 0.53	71.21 ± 0.75	70.98 ± 0.43	70.79 ± 2.03	70.80 ± 2.05
Plongeon Sync. 3m	*RMSE*	0.79 ± 0.09	0.71 ± 0.13	0.81 ±0.08	0.48 ±0.02	0.41 ± 0.02	0.75 ± 0.19	**0.39 ± 0.11**
	Acc_{pair}	76.53 ± 3.6	87.42 ± 0.84	87.81 ± 1.03	86.49 ± 0.85	88.20 ± 0.66	82.45 ± 2.31	**88.27 ±2.54**
Plongeon Sync. 10m	*RMSE*	0.74 ± 0.11	0.82 ± 0.09	0.74 ± 0.02	0.55 ± 0.03	**0.54 ± 0.03**	0.60 ± 0.16	**0.54 ±0.17**
	Acc_{pair}	78.04 ± 5.03	82.60 ± 0.93	82.92 ±1.6	83.13 ± 1	**83.65 ± 1.19**	83.72 ± 2.46	**83.65±2.39**
Moyenne	*RMSE*	0.73 ± 0.05	1.02 ± 0.11	0.87 ±0.06	0.73 ± 0.02	**0.61 ± 0.02**	0.71 ± 0.2	**0.61±0.18**
	Acc_{pair}	74.63 ± 2.09	78.26 ± 0.81	77.78 ± 0.94	78.21 ± 0.82	**78.66 ± 0.85**	76.99 ± 2.2	78.61±2.26

TABLE 3.8: Comparaison des méthodes pour les tâches sportifs.

L'utilisation de la fonction de coût multiple, avec $L_{Siam_{MSE}}$ ou $L_{Siam_{Pair}}$, améliore les résultats de RMSE et de précision par paire pour le modèle siamois. Le même constat est fait pour les deux réseau *finetuned*, les résultats de RMSE sont améliorés par rapport à une régression simple et un réseau siamois avec un seul apprentissage.
La fonction de coût multiple avec la MSE surpasse également toutes les autres

configurations sur la tâche de classement. L'utilisation de l'optimisation des paramètres par sport permettrait probablement d'obtenir de meilleurs résultats par sport.

Les travaux antérieurs sur cette base de données sont limités et se concentrent sur une seule mesure : la corrélation de Spearman [Parmar and Morris, 2018]. Cette métrique autorise la non-linéarité entre le score réel et le score prédit puisqu'elle prend en compte uniquement le classement. Elle est définie par :

$$\rho = 1 - \frac{6 \sum_i {d_i}^2}{N(N^2 - 1)} \tag{3.8}$$

où $d_i = rank(s_i) - rank(\widehat{s}_i)$ est la différence de rang entre le score réel s_i et le score prédit $\widehat{s}_i$ et N est le nombre d'exemples.
Ainsi, elle ne fournit pas vraiment d'informations sur la prédiction du score réel. Les résultats des comparaisons utilisant cette métrique sont donnés Table 3.9.

	Plongeon	**Table de Saut**	**Ski**	**Snowboard**	**Plongeon Sync. 3m**	**Plongeon Sync 10m**	**Moyenne**
Single-action C3D-SVR [Parmar and Morris, 2017]	**0.79**	0.68	0.52	0.4	0.59	**0.91**	0.69
Single-action C3D-LSTM [Parmar and Morris, 2018]	0.6	0.56	0.46	0.5	0.79	0.69	0.62
Nous (*Finetuning* + $L_{Siam_{MSE}}$)	0.69	0.72	**0.65**	**0.55**	**0.91**	0.86	**0.73**
Nous (Multi-tâches + $L_{Siam_{MSE}}$)	0.71	**0.73**	0.64	**0.55**	**0.91**	0.86	**0.73**

TABLE 3.9: Corrélation de Spearman des méthodes de l'état de l'art.

Comme on peut le voir, les méthodes proposées surpassent celles de l'état de l'art. Notons que les hyper-paramètres, tels que la taille des batchs et le taux d'apprentissage, ont été ajustés globalement et non par sport.

3.6 Bilan

À partir de la littérature existante sur l'évaluation de gestes, il a été possible dans ce chapitre d'en voir les forces et les défauts et par la suite d'améliorer une approche existante de deux manières différentes.

Ainsi, nous avons décidé d'utiliser les réseaux siamois qui permettent une comparaison par paire d'exemples. D'une part, leurs performances sont aujourd'hui au niveau de l'état de l'art et d'autre part, intuitivement, il semble plus facile de comparer deux exemples pour justifier leur écart de score plutôt que d'attribuer

une note à chacun des deux exemples. Dans le cadre de ces réseaux siamois, nous avons apporté deux contributions.

La première contribution réside dans l'utilisation de la MSE comme fonction de coût pour entraîner le réseau siamois. En effet, il a été montré que cette fonction de coût, comparée aux fonctions existantes, donnait de meilleurs résultats au niveau de la précision par paire, et donc de meilleurs résultats de classement. Ainsi, lorsque les scores sont disponibles en annotation, il est plus judicieux d'utiliser la MSE entre les écarts de score plutôt que des fonctions perte de classement, même pour réaliser des tâches de classement. Cette affirmation reste à valider dans le cadre d'autre applications comme la réidentification par exemple.

Si cette fonction coût MSE donne de bons résultats pour le classement, elle ne résout pas le problème de l'estimation de scores : un décalage systématique peut exister entre les scores prédits et les scores réels. Deux solutions ont été proposées pour résoudre ce problème. La première solution vise à ajouter une seconde phase d'apprentissage au sous-réseau : comme *a priori*, seul un offset existe entre les scores prédits et les scores réels, le sous-réseau est *finetuned* en gelant tous ses poids sauf ceux de la dernière couche qui s'adaptent pour corriger ce décalage. La deuxième solution consiste à réaliser un apprentissage multi-tâches : tandis qu'une fonction coût vise toujours à estimer l'écart entre les exemples, la seconde aura pour but d'estimer le score réel et donc poussera à apprendre l'origine des scores.

Les deux solutions ont été testées sur deux bases de données de gestes différentes : une de gestes chirurgicaux et l'autre de gestes sportifs. Sur ces deux bases de données les résultats obtenus à la fois sur la RMSE entre la vérité terrain et les scores prédits, et sur la précision par paire, sont meilleurs que ceux de l'état de l'art.

Afin d'obtenir une application qui soit la plus complète possible, il est également nécessaire de fournir un retour sur les erreurs commises lors de la réalisation de geste. Cette problématique est abordée dans les chapitres suivants.

Chapitre 4

Retour d'Information

4.1 Introduction

Évaluer est nécessaire à un bon apprentissage, que ce soit de connaissances ou de gestes, comme vu Chapitre 3. Cependant cette évaluation seule ne permet pas d'avoir une courbe d'apprentissage fluide. En effet, une évaluation seule indique globalement le niveau de la réalisation, mais ne va pas rentrer dans les détails des erreurs commises. Dans le cas de contrôle de connaissances, cette note globale peut parfois suffire si l'évaluation se concentre sur des connaissances bien précises. En revanche dans le cas d'un apprentissage gestuel, cette évaluation globale n'est pas suffisante car pour évoluer le mieux possible et apprendre de manière efficace, une information sur les erreurs commises est essentielle. Ainsi, en plus de cette évaluation globale, il est nécessaire de trouver des stratégies pour fournir un retour temporel et spatial sur les erreurs gestuelles. En effet, fournir uniquement un retour temporel est certes intéressant, mais n'est pas suffisant. Cette information temporelle permet de savoir qu'une erreur a été commise, mais sans correction.

Cette correction est apportée par un retour spatial. En effet, ce retour spatial indique comment cette erreur a été commise, comment le geste s'est décalé d'une trajectoire dite idéale. Ce retour est souvent donné naturellement par un coach dans le sport ou par un professeur dans la chirurgie. Afin de créer un coach virtuel qui soit le plus intéressant possible, il est essentiel de développer une méthode qui fournisse ce retour temporel et spatial en complément de l'évaluation globale automatique.

Dans le chapitre précédent, nous avons évoqué deux grandes familles de méthodes d'évaluation automatique de la qualité d'un geste : les méthodes avec des caractéristiques manuellement définies et les méthodes *"end-to-end"*. Selon le type de méthode choisie, il va être plus ou moins aisé de fournir un retour sur les erreurs commises.
Les méthodes avec des caractéristiques prédéfinies, par exemple, ne vont pas forcément permettre de fournir ce retour d'information. En effet, dans le cas de caractéristiques globales, qui représentent bien les gestes, ce retour d'information va être impossible à produire, car il sera sur ces caractéristiques, ce qui n'a aucun intérêt pour l'apprenant.
Fournir un retour d'information automatique à partir d'une méthode *end-to-end* est plus simple qu'à partir de l'autre famille de méthodes. En effet, ces méthodes *end-to-end* partent des données brutes, extraient des caractéristiques et évaluent le geste, il paraît donc plus simple, par exemple, de trouver le modèle inverse, *i.e.* qui part de la note et qui remonte jusqu'aux données brutes.

Dans ce chapitre, nous proposons tout d'abord de faire un état de l'art des méthodes existantes fournissant un retour d'information et également sur les méthodes permettant d'analyser les sorties d'un réseau de neurones. Après un bilan sur ces méthodes, nous présentons une nouvelle approche permettant de fournir un retour d'information à la fois spatial et temporel sur la réalisation d'un geste. Les bases de données permettant de savoir si le retour fourni est pertinent n'existant pas, nous présentons également une base de données de signaux synthétiques possédant une vérité terrain sur l'instant et le type d'erreurs commises. La méthode proposée et des méthodes de l'état de l'art sont ainsi testées sur cette nouvelle base de données afin de les comparer et de savoir laquelle donne le meilleur retour d'information.

4.2 État de l'Art

Évaluer un geste n'est pas suffisant pour permettre un apprentissage qui soit le plus efficace possible. En effet, fournir un retour est essentiel et nécessaire au développement des compétences. Dans ce but et en complément d'une évaluation automatique de la qualité d'un geste, il est nécessaire de développer des techniques afin de fournir ce retour d'information.

Dans un premier temps, nous nous intéressons aux méthodes fournissant l'évaluation et le retour d'information, déjà présentes dans la littérature. Dans un second temps, ayant développé une méthode d'évaluation performante fondée sur des réseaux de neurones, nous nous intéressons aux différentes techniques permettant d'expliquer les décisions prises par un réseau de neurones.

4.2.1 Évaluation et Retour d'Information

Avant de donner un retour d'information pertinent, il convient d'évaluer la réalisation. Deux types d'évaluation sont possibles : une évaluation globale et une évaluation instant par instant. Le retour d'information découle naturellement de l'évaluation instant par instant. En effet, indiquer les instants bien et/ou mal réalisés peut être considéré comme un retour d'information temporel.
Dans leurs travaux, Feygin *et al.* s'intéressent à des gestes simples : un simple suivi de trajectoire [Feygin et al., 2002]. En connaissant la trajectoire idéale, il est ainsi possible de calculer l'erreur à chaque instant et donc d'avoir une mesure d'évaluation à tout instant. Enfin, ils proposent différents type de retours d'information pour améliorer la réalisation de la tâche : un retour d'information visuel, un retour d'information tactile et un retour d'information regroupant les deux. Selon le mode de retour d'information utilisé, l'apprentissage du geste va être plus ou moins rapide, mais toujours plus rapide qu'un apprentissage sans retour d'information.
Dans la même veine, Candalh étudie l'influence de ces différents modes de retour d'information pendant l'apprentissage d'une tâche de MIS [Candalh-Touta, 2018]. Il est ainsi demandé aux apprenants de réaliser une forme de huit à l'aide d'un outil de MIS. La distance entre le chemin parcouru par l'outil et le chemin idéal permet de déterminer à chaque instant la qualité du geste et de proposer un retour d'information sur les erreurs commises. Trois types de retour sont évoqués :

— Le premier est un retour visuel, qui apparaît à la demande pendant la réalisation de la tâche. Cela permet de voir si la trajectoire suivie par les outils est correcte et de corriger ainsi la position et la trajectoire du geste.

— Le deuxième retour évoqué est un retour tactile. Le dispositif consiste en un vibreur placé sur la main ne réalisant pas la tâche. L'intensité de la vibration dépend de la distance entre la trajectoire réalisée et la trajectoire parfaite, *i.e.* plus l'outil est loin, plus la vibration est intense.

— Enfin le troisième type de retour est un retour kinesthésique. Ce retour est apporté par un robot qui tient l'instrument en même temps que le participant. Pour ce set-up, le robot va ramener l'outil au plus proche de la trajectoire idéale si le participant s'en éloigne trop.

Grâce à la simplicité de la tâche à réaliser et à l'existence d'une trajectoire idéale, il est facile d'évaluer les gestes réalisés. Cependant, transposer ces méthodes d'évaluation sur des tâches plus complexes et sans réalisation idéale, semble impossible. Dans le sport également, des travaux s'intéressent au retour d'information pour

l'apprentissage. Pour la danse, Kyan *et al.* proposent de comparer les mouvements d'un novice avec ceux d'un expert afin de fournir à la fois un score, mais également un retour d'information sur les erreurs commises [Kyan et al., 2015]. Les gestes enregistrés à l'aide d'une Kinect, sont projetés dans un espace de descripteurs définis grâce à une carte auto-adaptative. Par la suite, les mouvements de l'expert et du novice sont réalignés, grâce à l'utilisation d'un métronome, et comparés afin de fournir un retour d'information pertinent et une courbe de score.
Morel *et al.* proposent un calcul générique de l'erreur spatiale et temporelle pendant les services de tennis et le tsuki de karaté [Morel et al., 2017b]. L'évaluation des compétences est effectuée en construisant un modèle du geste : les gestes réalisés par des experts sont réalignés grâce à l'algorithme de *Dynamic Time Warping* (DTW) [Morel et al., 2018] et ensuite moyennés afin de créer un template du geste. Par la suite, pour évaluer un nouveau geste, il suffit de l'aligner avec ce template grâce à l'algorithme du DTW et de calculer la distance de Mahalanobis pour chaque articulation entre le template et le nouveau geste.
Pour des séquences de mouvements complexes, Pirsiavash *et al.* proposent une autre méthodologie [Pirsiavash et al., 2014]. Ils conçoivent deux types de caractéristiques : des caractéristiques de bas niveau qui capturent les gradients et les vitesses des pixels et des caractéristiques de haut niveau fondées sur les trajectoires des poses humaines. Un SVR est ensuite entraînée pour prédire les scores sur des séquences de plongeon à 10m ainsi que sur du patinage artistique. L'évolution du score tout au long de la séquence de mouvements est présentée Figure 4.1.

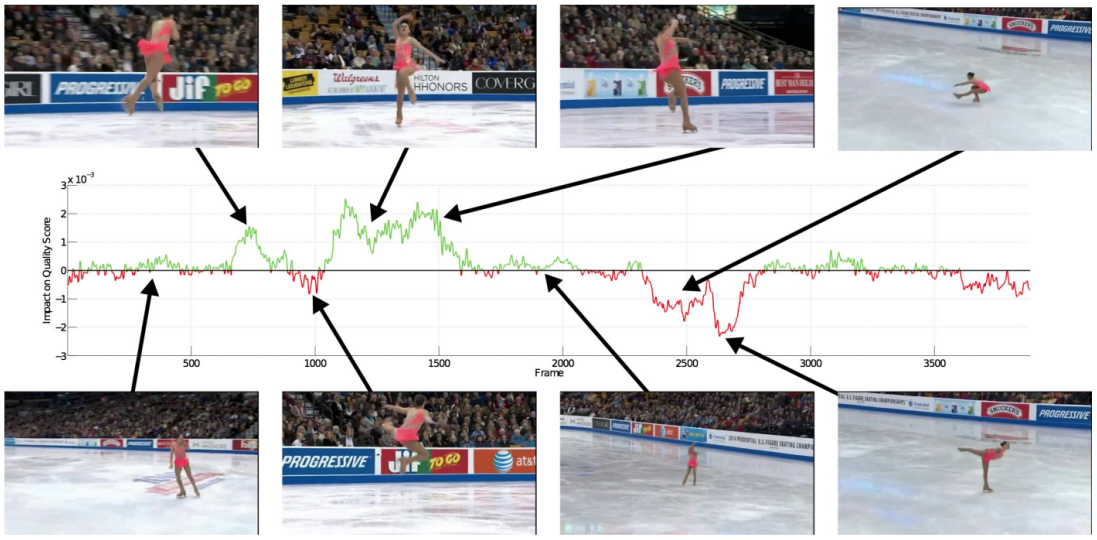


FIGURE 4.1: Illustration de l'évolution du score, extraite de [Pirsiavash et al., 2014].

4.2.2 Explicabilité des Réseaux de Neurones

Si l'on se base sur une évaluation automatique réalisée à l'aide d'un réseau de neurones, il est complexe d'expliquer les décisions prises par ces réseaux, mais des

méthodes existent pour tenter de proposer une explication.
Les méthodes expliquant les décisions prises par des réseaux de neurones ont pour but de trouver la contribution de chaque caractéristique d'entrée à la sortie et donc, de produire des cartes d'attribution. Quatre grands types d'approches sont possibles :

— les cartes d'attention, qui observent sur quelle partie de l'entrée le réseau se focalise tout en améliorant en général les résultats.
— les cartes d'activation, où le but va être de montrer, par exemple pour des images en entrée, les pixels qui ont le plus contribué à la décision finale.
— les approches basées sur des perturbations : l'entrée est modifiée afin de vérifier quels segments de cette entrée ont eu le plus d'influence sur la sortie.
— les approches fondées sur le gradient : en calculant le gradient sur l'entrée, il est possible d'obtenir une carte qui pointe les caractéristiques de l'entrée les plus importantes pour la décision finale.

Notons que toutes les méthodes proposées à ce jour travaillent sur des problèmes de classification, ce qui permet de mettre en compétition les différentes tâches lors de l'explicabilité du réseau.

4.2.2.1 Les Cartes d'Attention

Les cartes d'attention sont des mécanismes grâce auxquels un réseau de neurones pondère les caractéristiques en fonction de leur niveau d'importance. Ces cartes permettent au réseau de se concentrer uniquement sur les zones importantes pour la tâche à réaliser. Contrairement aux autres méthodes présentées dans la suite, la pondération est apprise en même temps que le reste du réseau, et permet d'améliorer les résultats. Ces cartes ont été introduites par Xu *et al.* pour sous-titrer des images automatiquement [Xu et al., 2015]. Comme illustré Figure 4.2, les cartes indiquent la partie de l'image sur laquelle le réseau a porté son attention pour prédire un mot du sous-titre.

Les cartes d'attention ont déjà été mises en place dans un contexte d'évaluation de gestes. En effet, comme évoqué Section 3.2.2.1 Doughty *et al.* on mis en place deux mécanismes d'attention : un se focalisant sur les instants erronés et un se focalisant sur les instants bien réalisés [Doughty et al., 2019]. Ces deux mécanismes d'attention sont ainsi temporels et non spatiaux, l'instant erroné est donc indiqué mais le type d'erreurs commises, n'est lui pas connu. D'autres part, aucune évaluation du retour d'information proposé n'est réalisée dans l'article.
De la même manière, Li *et al.* ont développé un réseau convolutionnel avec un module d'attention spatial [Li et al., 2019]. Ce module d'attention permet de déterminer la zone de l'image qui a mené à l'évaluation, comme illustré Figure 4.3. Cependant, cette zone ne correspond pas forcément au lieu des défauts. En effet, quelles sont les zones importantes pour prédire un score, celles qui correspondent à une bonne réalisation, celles qui correspondent aux défauts, ou les deux ? D'autre part, aucun retour temporel n'est fourni.

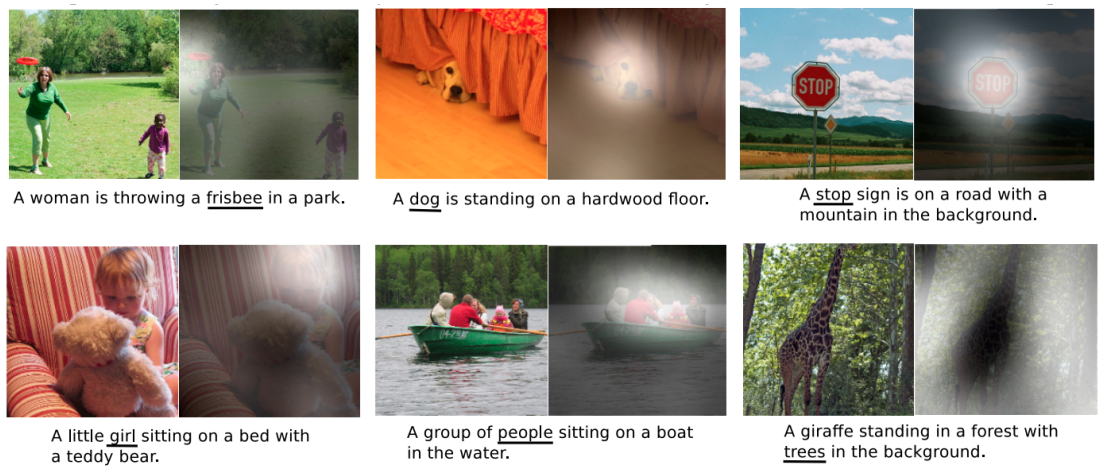


FIGURE 4.2: Illustration des cartes d'attention proposées par [Xu et al., 2015]. Le blanc indique la région sur laquelle le réseau se concentre et qui correspond au mot souligné.

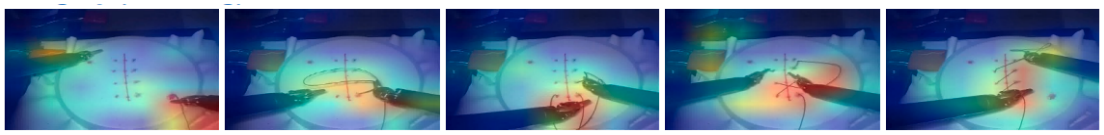


FIGURE 4.3: Illustration des cartes d'attention spatiales proposées par [Li et al., 2019] pour l'évaluation automatique de gestes.

4.2.2.2 Les Cartes d'Activation

L'objectif principal des cartes d'activation est de fournir des cartes qui mettront en lumière les régions qui ont été les plus importantes dans la décision prise par le réseau. Ces cartes sont souvent estimées dans un contexte de classification multi-classes.
Les premiers travaux portant sur ces cartes ont été réalisés par Zhou et al. [Zhou et al., 2016] qui proposent de remplacer la dernière couche de *max-pooling* du réseau GoogLe-Net [Szegedy et al., 2015], par une couche de *global-average-pooling* sur les cartes de caractéristiques puis d'utiliser ces caractéristiques dans une couche entièrement connectée qui produit la sortie désirée.
On peut alors identifier l'importance des régions en projetant les poids du réseau de sortie sur les cartes de caractéristiques. La somme pondérée des cartes de caractéristiques amène ensuite à la carte d'activation de la classe considérée (*Class Activation Mapping - CAM*), comme illustré Figure 4.4.

Cette méthode de cartes d'activation a, par la suite, été modifiée par Selvaraju *et al.* (*Grad-CAM*) [Selvaraju et al., 2017] . Les poids attribués à chaque carte de la dernière couche de convolution ne sont plus les poids de la dernière couche connectée, mais les gradients des cartes par rapport à la sortie du réseau. Le poids α_k^c, pour la classe c et la carte de caractéristiques A^k se calculent donc de la manière

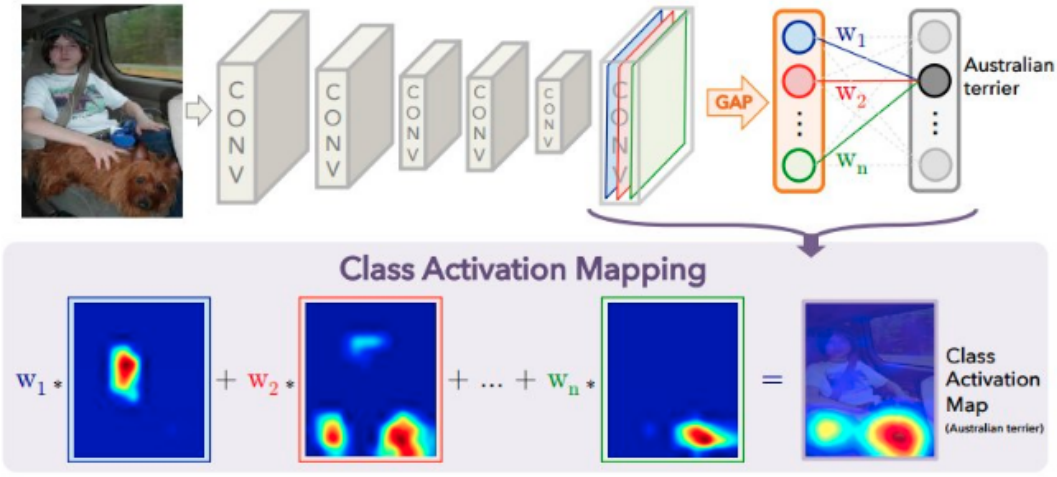


FIGURE 4.4: Illustration issue de [Zhou et al., 2016] qui montre la méthode CAM pour un modèle de classification.

suivante :

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y_c}{\partial A_{ij}^k} \tag{4.1}$$

Le gradient est ainsi calculé pour chaque pixel de la carte et est ensuite moyenné afin d'obtenir le poids α_k^c.
Une autre modification apportée sur les cartes d'activation a été proposée par Patro *et al.* [Patro et al., 2019], en rajoutant une sortie au réseau se concentrant uniquement sur les incertitudes de classification. Grâce à cette sortie, il est ensuite possible de calculer une carte d'attention en utilisant la méthode Grad-CAM, mais en intégrant au gradient cette nouvelle sortie d'incertitudes.

Les cartes d'activation ont déjà été utilisées dans un contexte d'évaluation de geste. En effet, Fawaz *et al.* [Fawaz et al., 2018] proposent de classifier des gestes chirurgicaux selon le niveau auto-proclamé des chirurgiens de la base de données JIGSAWS (Section 3.3.1). Une fois cette classification effectuée, les cartes d'activation sont estimées et permettent de remonter aux instants qui ont été les plus importants dans la décision prise par le réseau, comme illustré Figure 4.5.

Bien que très intéressantes, ces méthodes sont exclusivement dédiées à des tâches de classification : une fois la classe d'un exemple identifiée, la carte d'activation est estimée uniquement pour cette classe et permet de déterminer les éléments qui ont été pris en compte pour réaliser cette classification. Il parait donc difficile d'utiliser ces approches pour des tâches de régression.

4.2.2.3 Approches Fondées sur les Perturbations

L'idée de ces approches est de perturber certaines parties de l'entrée et d'examiner leurs influences sur la sortie.

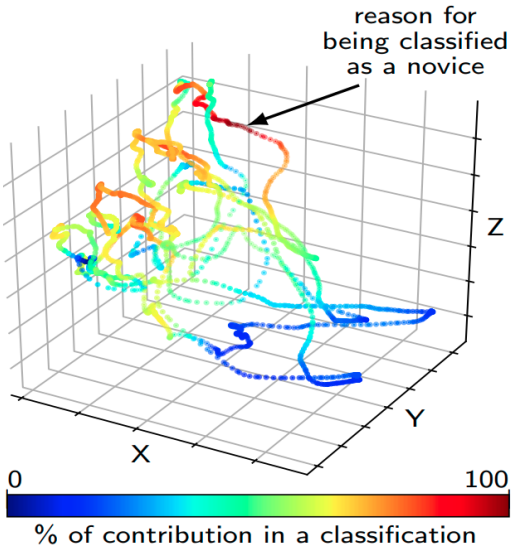


FIGURE 4.5: Illustration issue de [Fawaz et al., 2018] qui montre la méthode CAM pour des gestes chirurgicaux.

Zeiler *et al.* [Zeiler and Fergus, 2014] ont développé une méthode qui consiste à occulter successivement différentes portions de l'image d'entrée par un carré gris, et à regarder comment varie la sortie. Comme la probabilité de bonne classification diminue considérablement lorsque l'objet est occulté, cette technique permet à la fois de localiser des objets dans la scène et d'expliquer les décisions du réseau.

Une autre approche, fondée sur la perturbation, proposée par Ribeiro *et al.* [Ribeiro et al., 2016], est l'Explication Locale, Agnostique et Interprétable du Modèle (*Local Interpretable Model-Agnostic Explanation - LIME*). Avec cette approche un modèle linéaire local est créé autour de chaque exemple, en ajoutant des perturbations à l'exemple à expliquer. Ce modèle linéaire permet ensuite de déterminer les caractéristiques les plus importantes pour classer cet exemple. Ainsi, LIME fait des approximations locales de la surface de décision complexe.

4.2.2.4 Approches Fondées sur le Gradient

Par intuition, les valeurs de gradient importantes correspondent à des endroits de l'image qui ont une forte influence sur le résultat.
Ainsi, Simonyan et al. [Simonyan et al., 2013] ont proposé des cartes de sensibilité, en calculant le gradient de la sortie en fonction des pixels d'entrée dans une tâche de classification. Si $S_c(\mathbf{x})$ est la sortie du réseau de classification pour la classe c et $\mathbf{x}$ l'image d'entrée, alors les cartes de sensibilité sont définies par :

$$M_c(\mathbf{x}) = \frac{\partial S_c(\mathbf{x})}{\partial \mathbf{x}}. \tag{4.2}$$

En pratique, ces cartes de sensibilité sont très bruitées.
Une première solution pour les débruiter consiste à modifier l'algorithme de rétropropagation. Ainsi, les réseaux de déconvolution [Zeiler et al., 2011] ont été proposé par Zeiler *et al.*. Il va s'agir de projeter les activations des différentes couches dans l'espace de l'image, afin de pouvoir analyser quels ont été les pixels et régions importantes. Enfin, la rétropropagation guidée (*Guided Back-Prop*) [Springenberg et al., 2015] propose d'éliminer les valeurs de gradient négatives lors de l'étape de rétropropagation. L'idée est de conserver, à chaque étape, uniquement les entrées qui ont une influence positive sur le score.

Un autre problème des techniques fondées sur les gradients est que la fonction de score S_c peut être saturée pour des caractéristiques d'entrée ayant une valeur importante [Sundararajan et al., 2016]. Ainsi, la fonction peut être plate autour de ces entrées même si la caractéristique considérée est importante et donc, avoir un petit gradient. Certaines méthodes résolvent ce problème en calculant l'importance globale de chaque pixel. Ainsi, DeepLIFT (*Deep Learning Important FeaTures*) [Shrikumar et al., 2017] décompose la prédiction de sortie en rétro-propageant les contributions de tous les neurones du réseau à chaque caractéristique de l'entrée.

La méthode *Layer-Wise Relevance Propagation* (LRP) [Bach et al., 2015] utilise une décomposition au niveau du pixel pour comprendre la contribution de chaque pixel de l'image d'entrée x à la fonction de score $S_c(x)$. Une règle de propagation, appliquée de la sortie à l'entrée, distribue la pertinence de la classe trouvée à une couche donnée sur la couche précédente. Elle conduit à une carte qui met en évidence les pixels responsables de la classe prédite.

Trois autres méthodes, fondées sur l'algorithme classique de rétropropagation, existent pour expliquer les décisions des réseaux de neurones : *GradInput* [Shrikumar et al., 2017], *IntGrad* [Sundararajan et al., 2017] et *SmoothGrad* [Smilkov et al., 2017].

GradInput [Shrikumar et al., 2017; Bach et al., 2015] a été proposé pour améliorer les cartes de sensibilité. Ces cartes sont le produit entre les gradients de la sortie par rapport à l'entrée et l'entrée elle-même :

$$GradInput(\mathbf{x}) = M_c(\mathbf{x}) \times x \tag{4.3}$$

Au lieu de calculer les gradients de la sortie en fonction des pixels d'entrée x, Sundararajan et al. [Sundararajan et al., 2017] intègrent les gradients le long d'un chemin allant d'une base $\mathbf{x}'$ à l'entrée $\mathbf{x}$. Le gradient intégré, pour la dimension i de l'entrée x est défini comme suit :

$$IntGrad_i(\mathbf{x}) = (\mathbf{x}_i - x'_i) \times \int_1^{\alpha=0} \frac{\partial S_c(\mathbf{x}' + \alpha(\mathbf{x} - x'))}{\partial \mathbf{x} x_i} d\alpha \tag{4.4}$$

Lors du calcul, l'intégrale est approchée par une sommation : les gradients aux N points situés sur la ligne droite entre la base $\mathbf{x}'$ et l'entrée $\mathbf{x}$, sont ajoutés. Les

gradients intégrés s'ajoutent à la différence entre les sorties S_c à $\mathbf{x}$ et la base $\mathbf{x}'$. Ainsi, si la base a un score proche de zéro, les gradients intégrés forment une carte de sensibilité de la sortie de prédiction $S_c(\mathbf{x})$.

Étant donné les fluctuations rapides du gradient pour une image d'entrée x, les sauts de valeurs sont possibles ce qui risque de rendre les cartes de sensibilité illisibles. Ainsi, *SmoothGrad* [Smilkov et al., 2017] propose de créer une carte de sensibilité améliorée fondée sur un lissage de $\partial S_c(x)$ avec un noyau gaussien. Comme le calcul direct d'une telle moyenne locale dans un espace d'entrée à haute dimension est impossible, Smilkov et al. [Smilkov et al., 2017] calculent une approximation stochastique en prélevant des échantillons aléatoires dans un voisinage de l'entrée x et en faisant la moyenne des cartes de sensibilité résultantes :

$$SmoothGrad(\mathbf{x}) = \frac{1}{N}\sum_{i=0}^{N} M_c(\mathbf{x} + \mathcal{N}(0, \sigma^2)) \tag{4.5}$$

où N est le nombre d'entrées bruitées, et $\mathcal{N}(0, \sigma^2)$ est un bruit gaussien avec une moyenne de 0 et un écart-type de σ.

4.2.3 Bilan

Après avoir passé en revue les méthodes existantes pour l'explicabilité des réseaux de neurones, il convient maintenant de voir s'il est possible de faire une adaptation de certaines d'entre elles afin de voir si elles peuvent être appliquées à des tâches de régression.

Les cartes d'attention proposées aujourd'hui ne permettent pas d'obtenir une information conjointe, à la fois spatiale et temporelle. Tandis que certaines méthodes remontent uniquement aux aspects temporels (omettant donc les informations spatiales), d'autres méthodes proposent des cartes spatiales sans aucune informations temporelles. Il paraît difficile d'adapter la processus pour obtenir l'information conjointe recherchée.

Les approches utilisant les cartes d'activation permettent de déterminer quelles valeurs de l'entrée ont eu une influence pour classer un exemple dans une certaine catégorie. Elles partent donc de la décision de classification pour essayer de l'expliquer. Elles ne sont donc pas adaptables aux problèmes de régression. En effet, en calculant une carte d'activation sur un score, il serait impossible de déterminer si une entrée donnée a un impact positif ou négatif sur le score.

Les méthodes introduisant des perturbations semblent également difficiles à mettre en œuvre pour les problèmes de régression car il faudrait tester toutes les perturbations possibles, à la fois spatialement et temporellement, pour pouvoir fournir un retour d'information pertinent, ce qui n'est pas envisageable dans la pratique.

Les méthodes utilisant le gradient semblent être les plus adaptées à notre problème. Elles ont cependant toutes été utilisées dans un cadre de classification, ce qui va nécessiter des ajustements. Nous proposons donc une nouvelle approche où plutôt que d'estimer le gradient de la sortie par rapport à l'entrée, nous estimons le gradient de la sortie idéale (score maximal) par rapport à l'entrée. Ceci permet de déterminer comment l'entrée doit être ajustée de manière à optimiser le score. D'autre part, nous introduisons également une nouvelle approche permettant de faire face au problème de gradients bruités. Afin de valider ces apports et face au manque de bases de données adaptées, nous introduisons une base de données synthétique sur laquelle la vérité de terrain est connue.

4.3 Base de Données Synthétique et Régression

Dans cette section, nous présentons la base de données créée ainsi que l'estimation de la qualité de chaque signal par régression.

4.3.1 Base de Données

Les bases de données d'évaluation de la qualité du geste présentées Chapitre 3 fournissent seulement un score représentant la qualité du geste mais aucune vérité-terrain n'existe sur les défauts de celui-ci. Or afin d'évaluer et de comparer les méthodes fournissant un retour d'information, cette vérité terrain est essentielle. Ainsi nous proposons une base de données de signaux synthétiques, représentant des mouvements simples en 2D, où l'évaluation de la qualité, ainsi que la vérité-terrain du retour d'information sont disponibles.

Le but de cette base est de simuler des gestes. Chaque "*geste*", est représenté par 2 sinusoïdes (une pour chaque dimension) d'amplitude comprise entre 1 et 3, auxquelles un bruit gaussien de moyenne 0 d'écart type 0.05 est ajouté. Chaque signal est composé de 3 périodes et échantillonné à 60 Hz. Afin de simuler la variabilité de réalisation d'un geste, la fréquence de chaque sinusoïde est tirée aléatoirement entre 0.5 et 3.5 Hz. Afin de générer des erreurs sur ces sinusoïdes, des perturbations sont ajoutées de manière aléatoire. Leur nombre varie entre 0 et 8. Enfin leur position est tirée selon une loi uniforme. L'Algorithme 1 montre la génération de ces signaux et du score associé.

Algorithm 1 Création d'un signal de la base de données

$rand(a, b)$ renvoie une valeur tirée selon un loi uniforme entre a et b
$randn(m, \sigma)$ renvoie une valeur tirée selon une loi normale de moyenne m et d'écart-type σ
$g(x, m, \sigma)$ revoie la valeur de la gaussienne de moyenne m et d'écart-type σ, estimée en x

Ensure: Trial, GroundTruth, Score

$F_e = 60$
$A = rand(1, 3)$
$F = rand(0.5, 3.5)$
$Nb_{periode} = 3$
$Nb_{echantillon} = \frac{Nb_{periode} F_e}{F}$
$x = []$
$y = []$
for $t = 1$ *to* $Nb_{echantillon}$ **do**
  $x \leftarrow [x, A\sin(2\pi\frac{F}{F_e}t) + randn(0, 0.05)]$
  $y \leftarrow [y, A\sin(2\pi\frac{F}{F_e}t) + randn(0, 0.05)]$
end for
$GroundTruth = [x, y]$
$nb_{err_x} = rand(0 : 4)$
for $i = 0$ *to* nb_{err_x} **do**
  $time_{err_x} = rand(5, Nb_{echantillon} - 5)$
  $L_{error} = rand(1, 10)$
  $sign_{error} = sign(rand(-1, 1))$
  $\sigma = rand(1 : 5)$
  for $t = -\frac{L_{error}}{2}$ *to* $\frac{L_{error}}{2}$ **do**
    $x[time_{err_x} - t] \leftarrow x[time_{err_x} - t] + sign_{error} g(t, 0, \sigma)$
  end for
end for
$nb_{err_y} = rand(0 : 4)$
for $i = 0$ *to* nb_{err_y} **do**
  $time_{err_y} = rand(5, Nb_{echantillon} - 5)$
  $L_{error} = rand(1, 10)$
  $sign_{error} = sign(rand(-1, 1))$
  $\sigma = rand(1 : 5)$
  for $t = -\frac{L_{error}}{2}$ *to* $\frac{L_{error}}{2}$ **do**
    $y[time_{err_y} - t] \leftarrow x[time_{err_y} - t] + sign_{error} g(t, 0, \sigma)$
  end for
end for
$Trial \leftarrow [x, y]$
$Score \leftarrow 10 - (20 - norm2(GroundTruth - Trial))/2$

Un score est généré en calculant la distance euclidienne entre le signal parfait et le signal créé. 0 est attribué aux signaux idéaux tandis que le score avoisine

10, lorsque 8 perturbations sont présentes. 1000 signaux sont ainsi générés, 750 pour la base d'entraînement et 250 pour la base de test. Trois exemples de signaux extraits de la base de données sont présentés Figure 4.6.

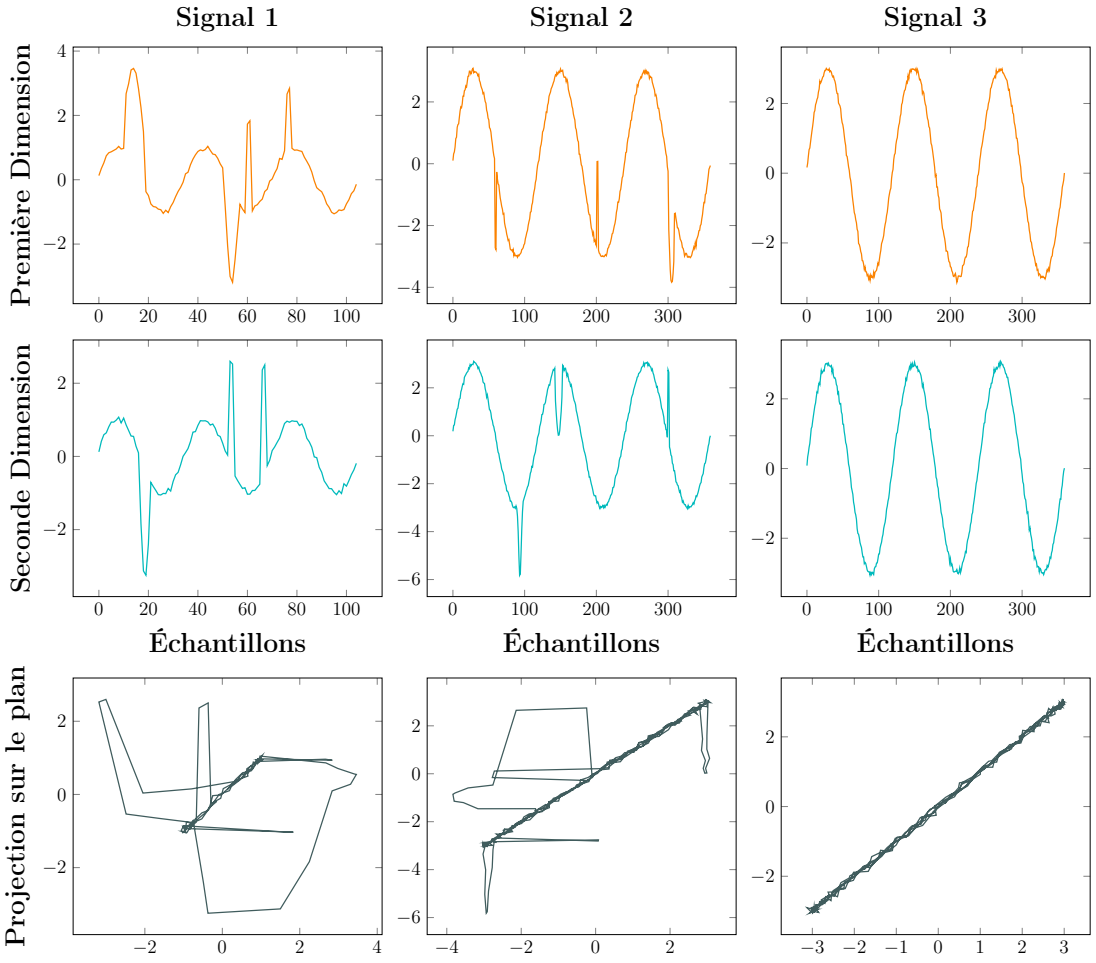


FIGURE 4.6: Exemples de signaux de la base de données. A gauche, un signal perturbé avec 5 erreurs, au milieu un autre signal perturbé avec 6 erreurs, et à droite un signal sans erreurs.

4.3.2 Modèle de Régression

De nombreuses architectures de réseau peuvent être utilisées pour la régression sur des séries temporelles, comme des réseaux récurrents ou des réseaux convolutionnels 1D ou 2D.
Dans notre cas, les meilleurs résultats de régression ont été obtenus en utilisant des convolutions temporelles (1D), c'est pourquoi seule cette architecture est présentée. En effet, si les scores sont mal régressés, il est impossible de fournir un retour d'information pertinent. Le réseau de régression est constitué de quatre couches de convolution temporelle avec des filtres $(8, 8, 16, 16)$ de taille $(25, 5, 5, 5)$, sans biais. Chacune d'elles est suivie d'une couche de *pooling* de taille 3. Deux couches de neurones connectés de taille 50 et 1, également sans biais, terminent le réseau.

Pendant l'entraînement, un taux de dropout de 0.5 est appliqué à la couche de 50 neurones, pour éviter un sur-apprentissage. Une illustration du modèle est présentée sur la Figure 4.7.

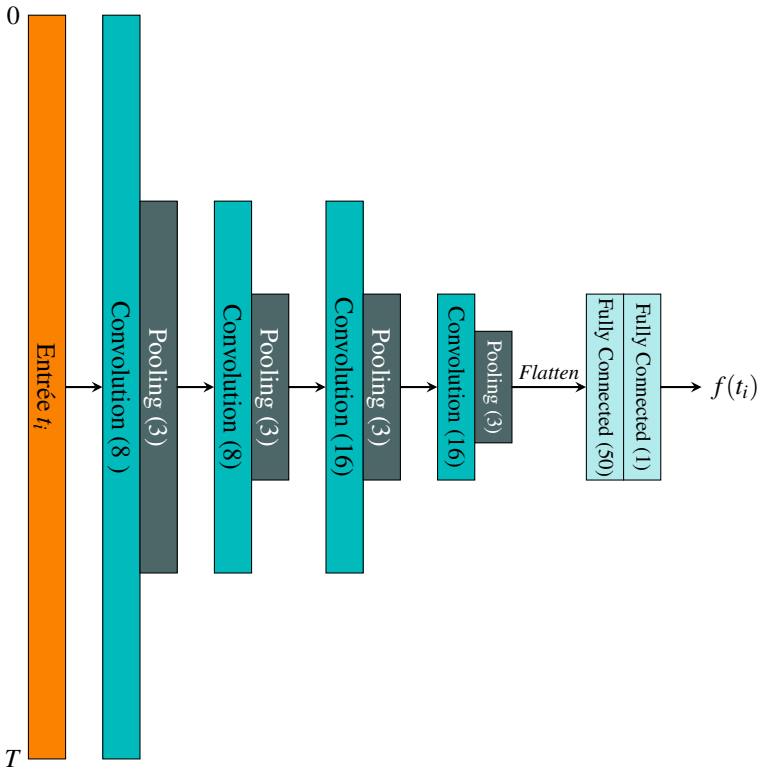


FIGURE 4.7: Illustration du modèle utilisé pour la regresssion des notes sur les signaux 2D.

L'apprentissage est réalisé en utilisant l'algorithme ADAM avec un taux d'apprentissage de 0.01, pendant 100 epochs. Pour obtenir 50 modèles différents, 50 entraînements sont réalisés. La MSE moyenne des 50 modèles, sur l'ensemble de la base de test, est de 0.619 avec un écart-type de 0.089. Ainsi, lors de la prédiction, ces modèles ont un comportement similaire.

4.4 Acccurate GRAdient - AGRA

Pour compléter les méthodes d'évaluation présentées Chapitre 3,nous avons décidé d'utiliser une méthode d'explication de réseau de neurones. En effet, souvent utilisé dans des problèmes de classification, ces méthodes ont montré de bons résultats qualitatifs. Comme le but de la méthode est d'évaluer la qualité d'un geste, nous nous plaçons donc dans un contexte de régression et non de classification, ce qui demande de changer le paradigme des méthodes actuelles de l'état de l'art qui fournissent, pour une classe donnée, les pixels contribuant à cette décision.

Comme évoqué Section 4.2.3, nous nous sommes tournés vers les méthodes utilisant le gradient et amenant à des cartes de sensibilité, comme proposé par Simonyan *et al.* [Simonyan et al., 2013] et défini Équation 4.2. L'application directe de ces méthodes amène à calculer le gradient de la fonction de coût par rapport à l'entrée. Celui-ci n'est cependant que très peu informatif puisqu'il indique les moments du signal qui ont une forte importance lors de la régression du score. Une modification doit donc être effectuée afin de pouvoir fournir un retour à l'utilisateur. Nous proposons donc de changer la fonction de coût du réseau utilisée pour calculer le gradient et de calculer le gradient, par rapport aux entrées du système, pour arriver à un score maximal. Ainsi, la nouvelle fonction de coût utilisée est :

$$l_2(\mathbf{x}) = (\hat{s}(\mathbf{x}) - score_{max})^2 \tag{4.6}$$

où $\hat{s}(\mathbf{x})$ est la prédiction du réseau et $score_{max}$, le score obtenu par les gestes sans erreurs. L'objectif de cette fonction de coût est de trouver les changements à apporter au signal d'entrée $\mathbf{x}$ pour que sa note soit maximale, ce qui amène directement aux défauts du geste.

Les gradients obtenus en utilisant la fonction de coût l_2, mettent en évidence des instants où des erreurs ont été commises, mais l'amplitude reste très loin de la vérité terrain. De plus, tous les instants erronés ne sont pas soulignés.

Afin de reconnaître au mieux les erreurs, une seconde modification est proposée. Elle consiste à réaliser plusieurs itérations de rétropropagation, dans le but de modifier l'entrée x pour qu'elle obtienne le meilleur score possible. Comme lors de l'entraînement d'un réseau où les poids w sont modifiés, il va s'agir ici d'appliquer la rétropropagation sur l'entrée $\mathbf{x}$, sans modifier les poids du réseau. La "correction" de l'entrée se fait en suivant l'Algorithme 2, où λ est le taux d'apprentissage et ϵ est la tolérance : la boucle s'arrête lorsque la différence entre la note maximum et $\hat{s}(\mathbf{x})$ est inférieure à ϵ.

Algorithm 2 Calcul du retour d'information

Require: $\mathbf{x}, \lambda, \epsilon$
Ensure: $GRAD(x)$
  $\mathbf{x}' = \mathbf{x}$
  while $l_2(\mathbf{x}') > \epsilon$ **do**
    $grad = \frac{\partial l_2(\mathbf{x}')}{\partial \mathbf{x}'}$
    $\mathbf{x}' \leftarrow \mathbf{x}' - \lambda grad$
  end while
  $Grad(x) = \mathbf{x} - \mathbf{x}'$

Comme indiqué précédemment, ce gradient $\frac{\partial l_2}{\partial \mathbf{x}}$ est très bruité [Smilkov et al., 2017; Kim et al., 2019]. De plus, au cours des expériences, nous avons observé qu'il dépend fortement de l'initialisation des poids et de l'entraînement du réseau.

Ainsi, même si deux entraînements différents conduisent à des scores de régression similaires, les gradients sont très variables. Deux exemples de gradients sont présentés Figure 4.8.

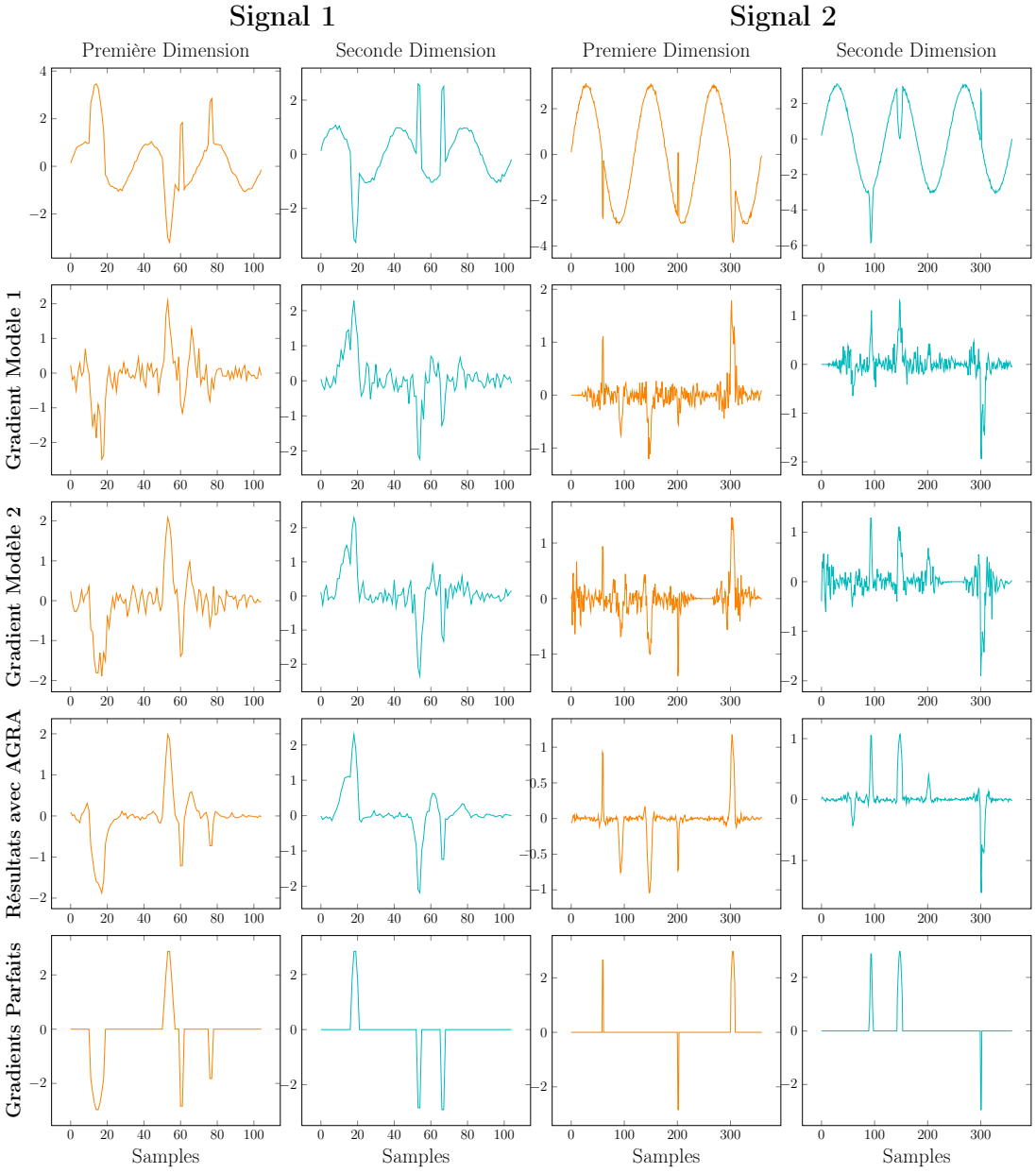


FIGURE 4.8: Gradients obtenus à partir de deux modèles avec des initialisations différentes et gradient obtenu avec la méthode AGRA.

Nous avons décidé de tirer parti de ces variations de gradients. En effet, si deux modèles peuvent souligner des instants différents avec des erreurs, alors en entraînant un grand nombre de modèles, il est possible de retrouver tous les instants erronés. La méthode *Accurate GRAdient* (*AGRA*) repose sur cette hypothèse. L'idée principale est d'entraîner N modèles M_i sur la même tâche d'évaluation de la qualité, mais en changeant l'initialisation des poids du modèle. Une fois ces N modèles entraînés, faire la moyenne des gradients obtenus par chaque modèle

permet d'obtenir un gradient mettant en évidence tous les instants erronés. La méthode AGRA est présentée plus en détails Algorithme 3.

Algorithm 3 Calcul du gradient suivant la méthode *Accurate Gradient - AGRA*

Require: $\mathbf{x}$
Ensure: $AGRA(\mathbf{x})$
  $grad = 0$
  for $i = 1$ to N **do**
    $grad = grad + Grad_{M_i}(\mathbf{x})$
  end for
  $AGRA(\mathbf{x}) = grad/N$

Comme le montre la figure 4.8, les gradients ainsi obtenus avec $N = 50$ modèles, sont plus nets et les instants erronés sont mieux mis en valeur. De plus, ils ne dépendent plus de l'entraînement et de l'initialisation, ce qui était le cas auparavant lorsque de bons ou de mauvais gradients étaient obtenus.

Cette méthode est coûteuse en termes de temps de calcul puisqu'elle nécessite l'apprentissage de N réseaux pour la même tâche de régression. Les résultats de régression $\hat{s}$ sont tous très similaires. Aussi, un tel apprentissage multiple pour la seule tâche de régression n'est pas optimal. Cependant, les gradients dépendent très fortement de l'initialisation du réseau, ce qui justifie l'apprentissage de plusieurs réseaux pour obtenir un gradient de bonne qualité et donc, un bon retour d'information sur les défauts du signal tel qu'illustré qualitativement Figure 4.8.

4.5 Résultats Expérimentaux

La méthode présentée Section 4.4 est testée sur la base de données synthétique présentée Section 4.3. Elle est également comparée à d'autres méthodes de l'état de l'art, afin de pouvoir constater son apport.
Dans un premier temps nous présentons le modèle de régression utilisé pour prédire les scores des signaux. Dans la suite, nous présentons tout d'abord des résultats qualitatifs des différentes méthodes, puis des résultats quantitatifs afin de voir quelle méthode donne le meilleur retour d'information. Enfin la méthode AGRA est testée avec des gradients calculés d'autres manières afin de montrer son efficacité et sa généricité.

4.5.1 Résultats Qualitatifs

Pour toutes les méthodes impliquées dans cette section, la fonction de perte $l_2(x)$ précédemment définie est utilisée pour calculer les gradients. Tout d'abord, nous présentons les résultats qualitatifs des cinq méthodes suivantes :

— Gradient *Grad* [Simonyan et al., 2013] calculé avec l'algorithme 2, un taux d'apprentissage de 0.1 et une tolérance ϵ de 0.015.

— *Grad*×*Input* $\mathbf{x}$ comme défini dans l'équation 4.3 et proposé par [Shrikumar et al., 2017; Bach et al., 2015].

— *SmoothGrad* [Smilkov et al., 2017] estimé comme la moyenne de 50 gradients obtenus avec l'Algorithme 2 en ajoutant un bruit gaussien de moyenne de 0 et d'écart-type 0.1 sur le signal d'entrée (équation 4.5).

— *IntGrad* [Sundararajan et al., 2017]. Comme le réseau proposé n'a pas de biais, la base $\mathbf{x}'$ est fixée à un signal composé uniquement de 0 de même longueur que $\mathbf{x}$. Dans ces conditions, le score de la base est $\hat{s}(\mathbf{x}') = 0$ et la méthode *IntGrad* peut être interprété comme une carte de sensibilité de la sortie de prédiction $\hat{s}(x)$.

— La méthode AGRA avec 50 modèles, avec un taux d'apprentissage de 0.1 et une tolérance ϵ de 0.015.

Pour toutes les méthodes, qui n'impliquent pas de faire la moyenne sur plusieurs modèles, un modèle aléatoire a été choisi parmi tous les modèles, et reste le même pour toutes les méthodes et tous les résultats présentés.

Comme présenté Figure 4.9, le gradient (*Grad*) est très bruité et ne permet pas d'obtenir des résultats clairs et faciles à interpréter, car les pics aux endroits de perturbation sont parfois trop fins et d'amplitude trop faible et peuvent être considérés comme du bruit.
La multiplication de ce gradient bruité par l'entrée ne fait qu'amplifier le bruit et rend les résultats encore moins interprétables. Les pics intéressants sont plus distincts, mais les résultats globaux semblent plus bruités qu'auparavant. De plus, le signe du gradient, qui donne des informations sur la direction de l'erreur, est perdu à cause de cette multiplication.
L'utilisation de la méthode *SmoothGrad* au lieu d'un gradient classique donne de meilleurs résultats qualitatifs avec un gradient moins bruité que précédemment. Cependant, le bruit est toujours présent et les résultats sont à nouveau difficiles à interpréter. De plus, l'amplitude du gradient sur les erreurs est souvent plus faible que la vérité-terrain.
La méthode *IntGrad* donne des gradients très bruités, qui présentent des pics à des endroits non perturbés, ce qui les rend très difficiles à interpréter.
Les résultats les moins bruités et les plus précis sont obtenus avec la méthode AGRA. La méthode met effectivement en évidence les échantillons correspondant aux perturbations, avec le bon signe, ce qui rend le retour d'information obtenu clair et facilement interprétable.

De plus, avec un retour d'information clair, il est possible de reconstruire ce que le réseau considère comme un signal idéal, en ajoutant le gradient calculé. Dans le cas idéal, ces signaux reconstruits sont supposés être une ligne, comme le dernier signal Figure 4.6.
La méthode AGRA réalise la meilleure reconstruction tel qu'illustré Figure 4.10.

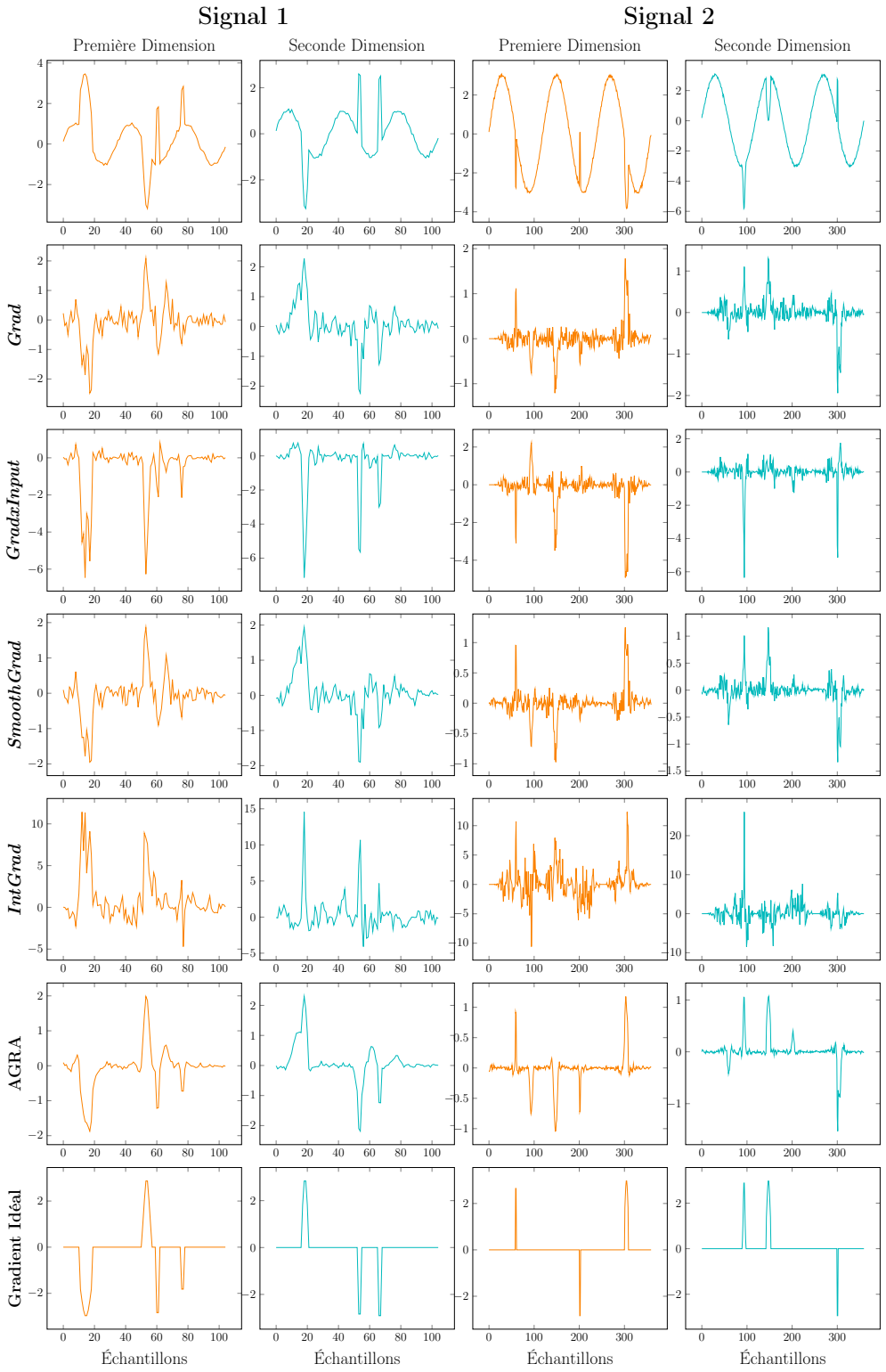


FIGURE 4.9: Résultats qualitatifs de toutes les méthodes pour 2 signaux avec des erreurs différentes.

Elle n'est pas parfaite car l'amplitude du gradient est souvent plus faible que la vérité-terrain.

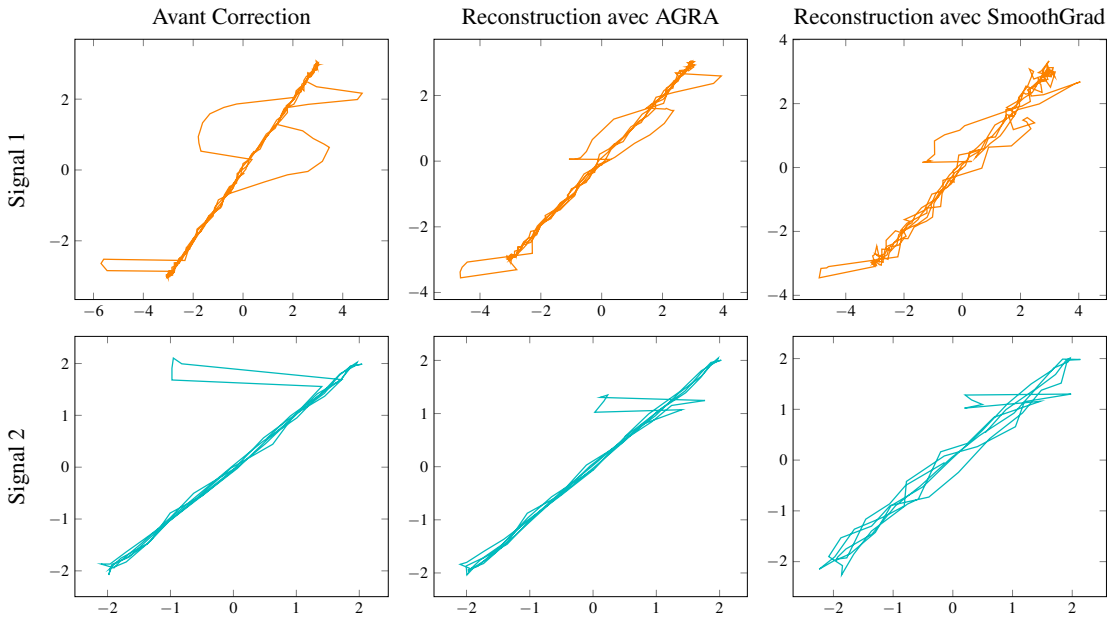


FIGURE 4.10: Correction obtenues avec les méthodes AGRA et *SmoothGrad* appliquées aux deux signaux tests.

4.5.2 Résultats Quantitatifs

Pour comparer les méthodes de manière plus approfondie, il est essentiel de donner des résultats quantitatifs. Comme la vérité terrain est disponible pour chaque exemple, il est possible de calculer le retour d'information idéal (la différence entre les signaux perturbés et les signaux idéaux) et de le comparer aux résultats obtenus avec les différentes méthodes. Deux mesures sont utilisées pour effectuer cette comparaison :

— Erreur Quadratique Moyenne (MSE) entre le signal sans erreur et le signal reconstruit obtenu grâce aux gradients. Cette métrique ne peut pas être utilisée pour des méthodes telles que *Grad*×*Input* ou *IntGrad*, car leur but est uniquement de mettre en évidence les étapes temporelles importantes et non de reconstruire un signal parfait.

— Le coefficient de corrélation de Pearson entre le gradient idéal et le gradient obtenu avec les différentes méthodes. Pour éviter de pénaliser les méthodes qui ne gèrent pas les signes (*Grad*×*Input* et *IntGrad*), ce coefficient est calculé entre les normes du gradient idéal et du gradient obtenu avec les méthodes.

Les 250 exemples de tests ont été utilisés pour ces 2 métriques et les résultats présentés sont les moyennes sur ces exemples.
De plus, pour les méthodes *Grad*, *Grad*×*Input*, *SmoothGrad* et *IntGrad*, le calcul des métriques à été fait sur les 50 modèles et ensuite moyenné afin de ne pas favoriser la méthode AGRA qui tire parti de ces 50 modèles.

Pour rappel, un gradient parfait amènerait à une MSE de 0. La Table 4.1 présente la MSE obtenu avec *Grad*, *SmoothGrad* et AGRA.

Méthodes	MSE
Grad [Simonyan et al., 2013]	5.81
SmoothGrad [Smilkov et al., 2017]	6.10
AGRA	**5.39**

TABLE 4.1: Erreur quadratique moyenne entre le signal d'origine et signal reconstruit. La MSE moyenne initiale, entre les signaux avec et sans erreur est de 7.74.

Les méthodes *Grad* et *SmoothGrad* sont toutes deux très bruitées. Même si la MSE diminue par rapport à celle des signaux initiaux, elle reste plus importante que celle de la méthode AGRA.

Méthode	Corrélation de Pearson
Grad [Simonyan et al., 2013]	0.85
GradxInput [Shrikumar et al., 2017; Bach et al., 2015]	0.86
SmoothGrad [Smilkov et al., 2017]	0.84
IntGrad [Sundararajan et al., 2017]	0.67
AGRA	**0.95**

TABLE 4.2: Coefficient de corrélation de Pearson pour différentes méthodes, estimé entre la norme des gradients.

Concernant la détection des défauts, les coefficients de corrélation de Pearson sont présentés Table 4.2.
Comme ces coefficients sont normalisés (la corrélation est divisée par l'écart type des deux gradients), ils peuvent être estimés pour chaque méthode, même lorsque le gradient est multiplié par l'entrée. Les meilleurs résultats sont obtenus avec la méthode que nous proposons, ce qui confirme l'étude qualitative précédente et prouve que cette méthode donne de meilleurs résultats que celles de l'état de l'art.

Le tableau 4.3 donne les coefficients de Pearson obtenus en gardant le signe des gradients lors du calcul de la corrélation : la corrélation est estimée pour chacune des deux dimensions et la moyenne est ensuite calculée. En utilisant cette métrique, seules les méthodes Grad et *SmoothGrad* peuvent être évaluées puisque pour les deux autres, la multiplication par l'entrée change les signes de gradient et les résultats ne seront pas exploitables. La méthode AGRA est à nouveau la méthode la plus efficace, même si les coefficients de Pearson ne tiennent pas compte de l'amplitude du gradient, ce qui ne pénalise pas la méthode *SmoothGrad* comme l'a fait le MSE.

Pour étudier le comportement de la méthode AGRA, il est intéressant de montrer l'évolution de la corrélation de Pearson et de la MSE, en fonction du nombre de modèles moyennés (Figure 4.11). Comme indiqué précédemment, les

Méthode	Corrélation de Pearson
Grad [Simonyan et al., 2013]	0.65
SmoothGrad [Smilkov et al., 2017]	0.65
AGRA	**0.74**

TABLE 4.3: Coefficient de corrélation de Pearson pour différentes méthodes, estimé sur toutes les dimensions du gradient.

gradients dépendent du modèle. Ainsi, la MSE et le coefficient de Pearson fournis par un réseau changent beaucoup selon le modèle.

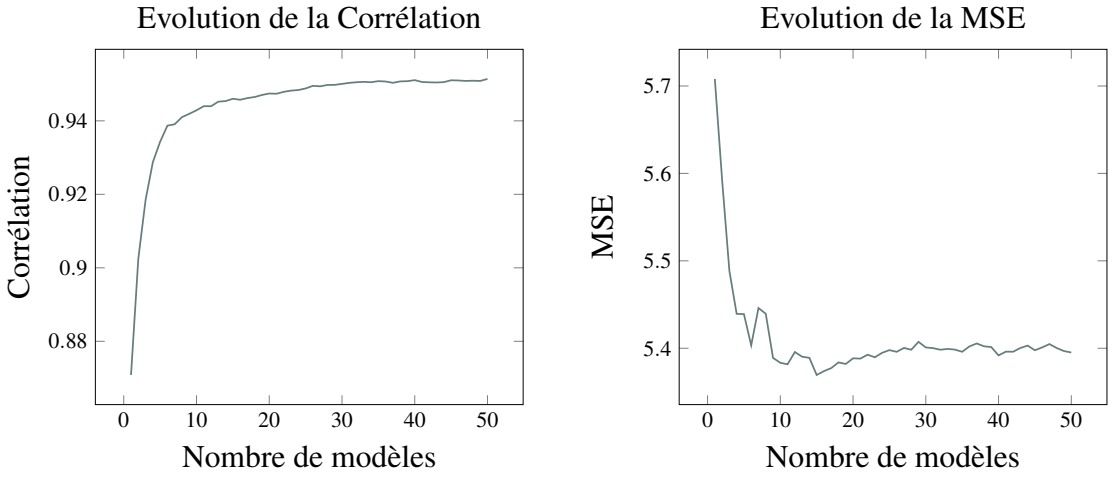


FIGURE 4.11: Évolution de la MSE et de la corrélation en fonction du nombre de modèles moyennés

Globalement, la corrélation et la MSE s'améliorent au cours des itérations pour se stabiliser vers 50.
Rappelons que les différents modèles ne changent que par l'initialisation des poids. Les scores de régression sont donc les mêmes, mais les gradients diffèrent fortement. Il est donc impossible de définir des modèles qui *a priori* conduiront à un retour d'information de bonne qualité.
Ainsi, Figure 4.11, la MSE est importante au début et diminue ensuite avant de se stabiliser. En faisant la moyenne des gradients obtenus par 50 modèles ou plus, les résultats d'explications sont bons indépendamment de l'apprentissage. Le même raisonnement peut être appliqué au coefficient de corrélation de Pearson.

4.5.3 AGRA pour Tous

Comme indiqué précédemment, il est possible de combiner notre approche avec différentes méthodes, telles que *Grad*×*Input*, *SmoothGrad* et *IntGrad*, afin d'améliorer les résultats tant qualitatifs que quantitatifs.

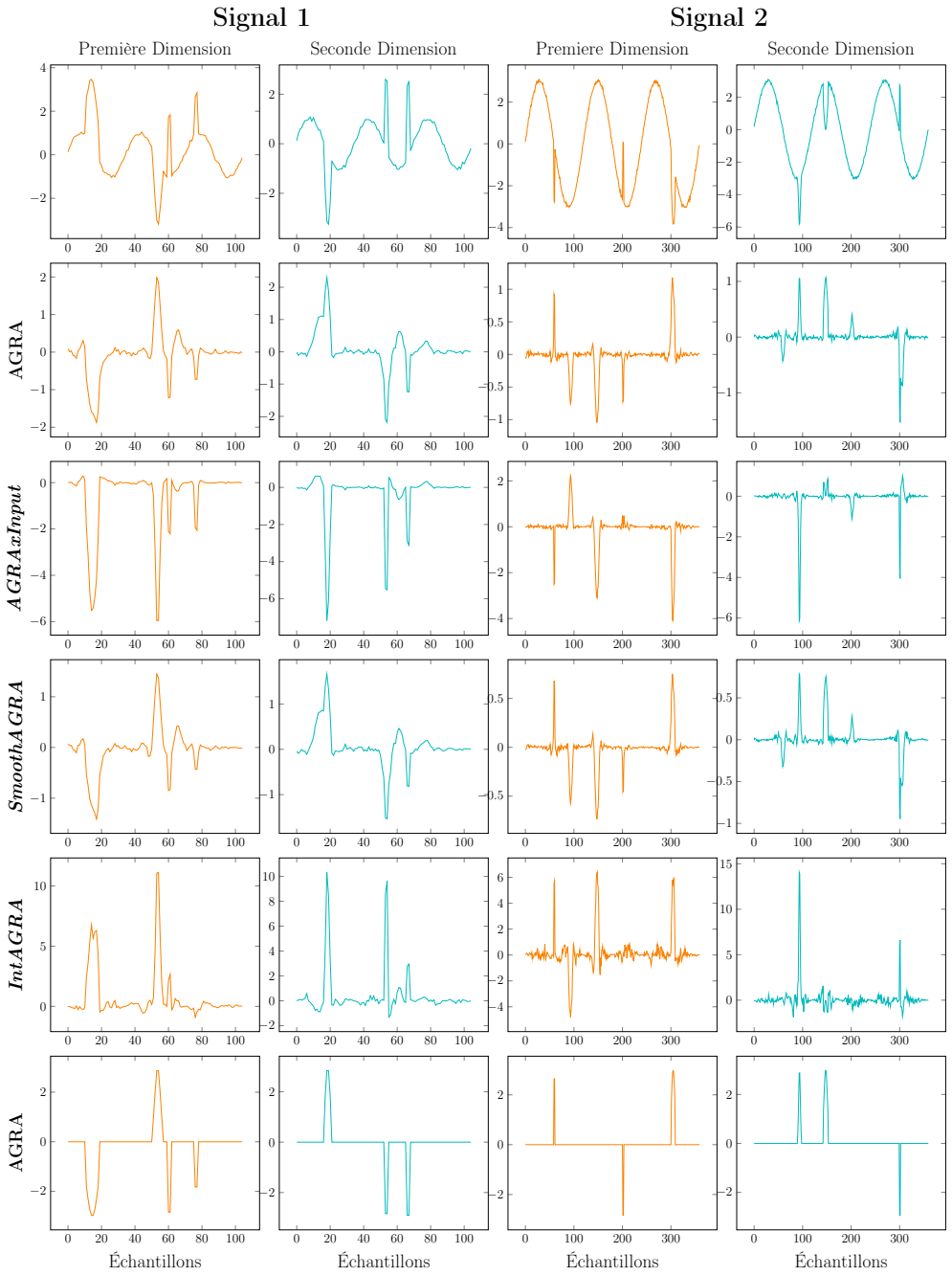


FIGURE 4.12: Résultat qualitatifs de toute les méthodes combinées avec AGRA pour 2 signaux avec des erreurs différentes.

Comme présenté Figure 4.12, l'utilisation de la moyenne de 50 modèles pour toutes les méthodes améliore grandement leurs performances. Cependant les méthodes multipliées par l'entrée (*AGRAxInput* et *IntAGRA*) ne permettent toujours pas de fournir un retour d'information de bonne qualité, car le signe des erreurs n'est pas pris en compte. De plus, *SmoothAGRA* a toujours des amplitudes faibles par rapport à la vérité-terrain. Donc même si la méthode AGRA permet d'améliorer les résultats, ceux-ci restent moins bons que la méthode AGRA.

En ce qui concerne les résultats quantitatifs, ils sont présentés Table 4.4.

Méthode	**Corrélation de Pearson**	**MSE**
Grad [Simonyan et al., 2013]	0.85	5.81
AGRA	**0.95**	**5.39**
Grad×*Input* [Shrikumar et al., 2017]	0.86	NA
AGRA×*Input*	0.91	NA
SmoothGrad [Smilkov et al., 2017]	0.84	6.10
SmoothAGRA	0.94	5.87
IntGrad [Sundararajan et al., 2017]	0.67	NA
IntAGRA	0.88	NA

TABLE 4.4: Coefficient de corrélation de Pearson et MSE pour différentes méthodes combinées avec la méthode AGRA.

Comme montré sur les résultats qualitatifs, la MSE et la corrélation de Pearson sont améliorées pour chaque méthode en utilisant AGRA. Cependant les meilleurs résultats pour les deux métriques sont toujours obtenus avec la méthode AGRA couplée avec un gradient simple.

4.6 Bilan

Dans ce chapitre, nous avons proposé une méthode permettant de donner un retour d'information automatique après une évaluation de gestes.
Cette méthode fait suite aux méthodes d'évaluation de la qualité de gestes présentées au chapitre précédent et vise à expliciter le score obtenu de manière à pouvoir remonter aux défauts de chaque réalisation de geste.
Parmi les méthodes d'explicabilité des réseaux de neurones, nous avons choisi d'utiliser celle fondée sur le gradient et les cartes de sensibilité. Celles-ci sont cependant très bruitées, peu précises et amène à un retour d'information difficilement compréhensible. Le gradient dépend de plus très fortement de l'initialisation des poids du réseau lors de l'apprentissage. C'est pourquoi, nous avons décidé de tirer parti de cette variation pour moyenner les gradients provenant de différents apprentissages et obtenir un gradient robuste, à même de détecter toutes les erreurs et moins bruité. Afin de tester cette approche, une base de données de synthèse, composée de signaux 2D, a été proposée. Elle présente l'avantage d'être associée à une vérité de terrain où les défauts des signaux sont connus.

En comparant la méthode proposée, AGRA, avec celles de l'état de l'art sur la base de données des signaux 2D, il apparaît qu'AGRA donne de meilleurs résultats à la fois de manière qualitative mais aussi quantitative.

Dans le prochain chapitre, nous proposons d'intégrer cette approche pour le rendu de retour d'information sur des gestes chirurgicaux. L'idée est de segmenter les gestes en tâches de manière à fournir un retour sur les tâches mal réalisées.

Chapitre 5

Segmentation Sémantique de Gestes

5.1 Introduction

Dans les chapitres précédents (Chapitre 4 et 3), nous avons vu comment évaluer un geste et proposer un retour d'information automatique, afin de créer sur le long terme un coach virtuel. Le retour d'information proposé actuellement renvoie les échantillons, et donc les instants à la seconde près, où une erreur a été commise. Cependant, cette précision peut rendre le retour d'information peu compréhensible pour un apprenant. En effet, indiquer les instants erronés à la seconde près donne un retour d'information précis mais peu d'indication quant à la partie du geste qui a été mal réalisée.
Si ce retour d'information est très important pour l'apprenant puisqu'il lui permet de connaître ses erreurs et donc, de s'améliorer lors de la réalisation suivante, il est difficile interprétable à un plus haut niveau, pour permettre le suivi progressif de

l'amélioration des compétentes. Ainsi être capable d'indiquer des sous-gestes mal réalisés plutôt que des instants est un pas vers la compréhensibilité de l'apprentissage.

Pour obtenir les sous-gestes composant un geste, il est essentiel de savoir segmenter sémantiquement le dit geste. La segmentation sémantique consiste à associer chaque pixel d'une image ou chaque instant d'un signal temporel à la classe correspondante.
Dans le cas d'une image, segmenter sémantiquement permet de reconnaître et de localiser des objets dans une scène. Dans le cas de signaux temporels, cette segmentation permettra de connaître la séquence de sous-gestes et donc par la suite d'utiliser cette segmentation à des fins d'évaluation de la qualité.

Dans ce chapitre, nous proposons tout d'abord de faire un état de l'art des méthodes de segmentation sémantique, que ce soit pour des images ou des signaux temporels. Dans la suite du chapitre, nous présentons une nouvelle méthode de segmentation sémantique de signaux temporels. Cette méthode est testée sur la base de données JIGSAWS, présentée Section 3.3.1. Enfin, nous proposons d'associer la segmentation sémantique à la méthode de retour d'information proposée précédemment (Chapitre 3), afin d'augmenter la compréhensibilité du retour d'information et de pouvoir fournir des statistiques sur le long terme de l'évolution des compétences de l'apprenant.

5.2 État de l'Art

La segmentation sémantique, est une étape naturelle pour arriver à une localisation fine des objets dans une image ou des sous-gestes dans le cas de signaux temporels. La segmentation sémantique regroupe deux tâches : la classification, qui prédit la classe de l'entrée, et la localisation, qui fournit des informations supplémentaires concernant l'emplacement spatial ou temporel. Ces deux tâches sont réalisées grâce à une prédiction dense, qui va pour chaque pixel et/ou échantillon prédire la classe à laquelle il appartient. Un exemple de segmentation sémantique est présenté Figure 5.1.

Dans cette section, nous introduisons différentes méthodes permettant la segmentation sémantique de scènes, que ce soit des images ou des signaux temporels. Nous présentons dans un premier temps des méthodes traditionnelles de segmentation avant d'introduire les méthodes plus récentes, fondées sur l'apprentissage profond.

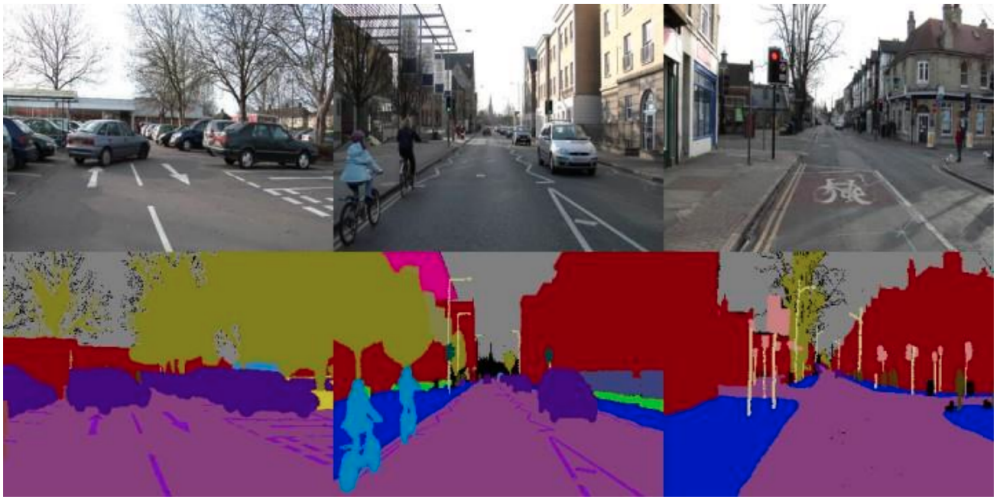


FIGURE 5.1: Illustration de la segmentation sémantique sur des images tirées de la base de données CamVid [Brostow et al., 2009].

5.2.1 Méthodes traditionnelles de segmentation

Avant l'arrivée massive des réseaux de neurones, plusieurs méthodes de segmentation ont été proposées dans la littérature, que ce soit pour des images ou pour des signaux temporels.

5.2.1.1 Segmentation d'Images

Dans le cas de segmentation sur des images, Konishi *et al.* utilisent les couleurs et les textures d'une image pour classifier chaque pixel de l'image dans 6 classes différentes [Konishi and Yuille, 2000]. Ils effectuent une classification bayésienne des pixels en utilisant les distributions des probabilités conjointes des filtres selon les classes. Schroff *et al.*, modélisent les classes d'objet grâce à des histogrammes de mots visuels [Schroff et al., 2006]. La classe de chaque pixel est ensuite trouvée en appliquant l'algorithme des KNN. Classifier pixel par pixel peut mener à des problèmes de classification, car les pixels autour et le contexte ne sont pas pris en compte. Pour palier ce problème, Tu *et al.* proposent d'utiliser les prédictions pixel par pixel comme une information de contexte et une supervision pour entraîner de manière itérative un autre classifieur [Tu and Bai, 2010]. Une autre méthode proposée par Kontschieder *et al.* utilise une supervision structurelle plutôt qu'une simple supervision de classe, afin d'introduire le contexte dans les prédictions par pixel [Kontschieder et al., 2011].
Afin d'exploiter au mieux les informations entre les pixels et les régions, il est intéressant d'utiliser des modèles tels que les Champs Aléatoires Conditionnels (*Conditional Random Field - CRF*) [Lafferty et al., 2001]. Cet algorithme prend en compte les interactions entre les variables proches les unes des autres, afin

de modéliser les probabilités d'appartenance à une classe. Les CRF sont considérés comme une variantes des champs aléatoires de Markov. Shotton *et al.* font l'hypothèse que certains objets sont susceptibles d'apparaître simultanément dans une zone, afin d'ajouter des informations contextuelles lors de l'entraînement d'un CRF [Shotton et al., 2006]. D'autres travaux ont préférés se tourner vers l'ajout d'information *a priori* pour rajouter du contexte. Ainsi, He *et al.* ont calculé les distributions de classes spécifiques à l'environnement afin de pouvoir par la suite développer un CRF dépendant de ces informations [He et al., 2006]. Silberman *et al.* [Silberman and Fergus, 2011] utilisent des informations 2D et 3D, afin de segmenter sémantiquement une scène.
Des variantes autour des CRF ont vu le jour également. Par exemple, les CRF hiérarchiques proposés par Ladick *et al.* [Ladicky et al., 2009]. Cette architecture particulière, est en fait composée de deux CRF différents qui vont agir sur deux types d'entrées différentes, comme illustré Figure 5.2.

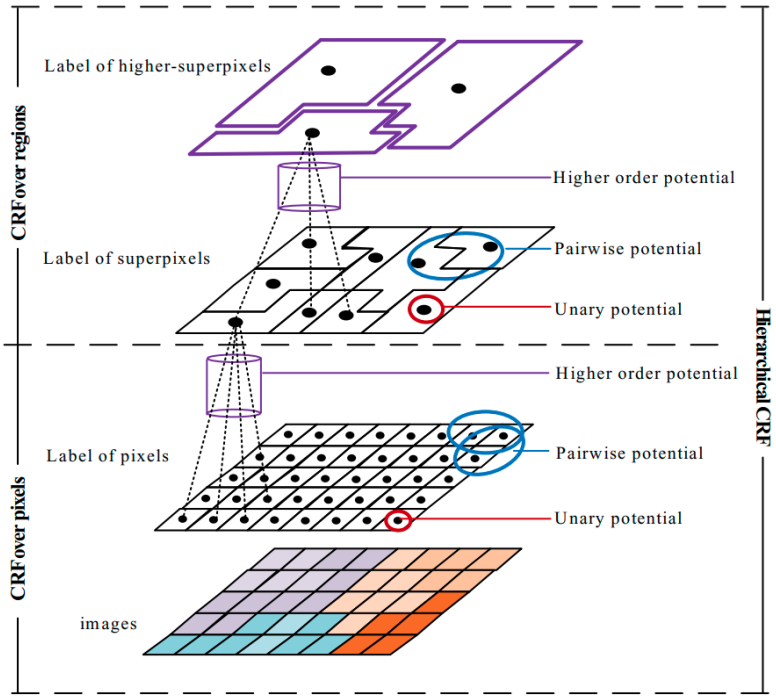
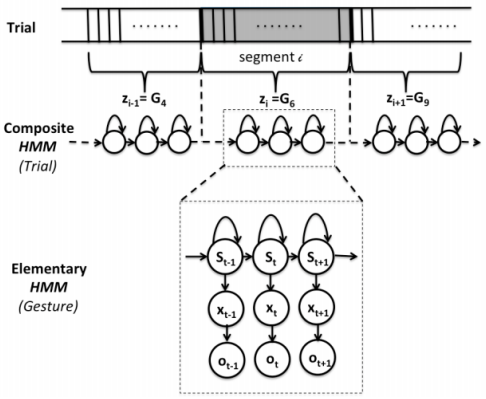


FIGURE 5.2: Illustration des CRF hiérarchiques pour la segmentation sémantique d'image, tirée de [Yu et al., 2018].

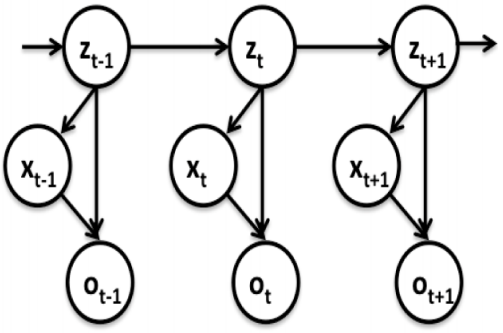
Enfin des méthodes non paramétriques ont également vu le jour, afin de pouvoir gérer l'ajout de nouvelles classes d'objet dans l'environnement. Ainsi, Liu *et al.* ont développé une méthode fondée sur le transfert de label afin de faire de la segmentation sémantique de scène [Liu et al., 2011]. L'idée est de trouver dans une base d'apprentissage, les scènes qui ressemblent le plus à la scène test. Par la suite, il faut trouver pour chaque pixel de la scène test, les pixels correspondant dans les scènes choisies dans la base d'apprentissage. Le label du pixel est ensuite estimé par un vote entre tous les pixels correspondants. Tighe *et al.*, ont également appliqué la même méthode de transfert, mais au niveau de super-pixels et non au niveau des pixels directement [Tighe and Lazebnik, 2010].

5.2.1.2 Segmentation de Signaux Temporels

Dans le cas de signaux temporels, d'autres stratégies ont été développées afin de segmenter sémantiquement, car les caractéristiques à extraire ne sont pas les mêmes que pour les images.
Ainsi, il est courant d'utiliser des HMM afin de segmenter des signaux temporels. En effet, Ahmidi *et al.* proposent deux méthodes fondées sur des HMM et une sur les CRF afin de segmenter des gestes chirurgicaux tirés de la base de données JIGSAWS (Section 3.3.1) [Ahmidi et al., 2017]. La première solution consiste en un HMM composite, illustré Figure 5.3A, où chaque état z_t d'un HMM global représente un sous-geste, et chaque sous-geste est lui-même représenté par un HMM a n état. Les observations du HMM, sont les données cinématiques de la base de données. La deuxième proposition est un HMM épars (*Sparse-HMM - S-HMM*), illustré Figure 5.3B. Comme pour le HMM composite, un sous-geste est représenté par un état caché z_t du HMM, cependant, le processus d'observation est modifié. En effet, l'observation o_t est modélisée par une combinaison linéaire d'éléments tirés d'un dictionnaire de micromouvements. Ainsi, l'observation va dépendre également d'un état latent x_t, comme illustré Figure 5.3B.



(A) Illustration du HMM composite proposé par Ahmidi *et al.*, extraite de [Ahmidi et al., 2017].



(B) Illustration du HMM épars proposé par Ahmidi *et al.*, extraite de [Ahmidi et al., 2017].

FIGURE 5.3: Illustration des deux méthodes proposées par Ahmidi *et al.* pour la segmentation sémantique de gestes chirurgicaux.

Enfin la dernière méthode proposée par Ahmidi *et al.* est fondée sur les CRF. Dans un CRF classique, les transitions sont considérées entre les échantillons, or un sous-geste peut durer plusieurs échantillons voir plusieurs secondes. L'algorithme développé tient compte de cette hypothèse et modélise les gestes à deux niveaux différents : un CRF va modéliser les transitions au niveau des échantillons et un autre les modélise au niveau de segment.

Ces méthodes statistiques montrent des résultats peu concluants et se montrent difficile à entraîner de manière efficace. Elles ont été surpassées ces dernières années par les méthodes utilisant l'apprentissage profond.

5.2.2 Méthodes fondées sur l'Apprentissage Profond

Les méthodes traditionnelles ont peu à peu été remplacées par des méthodes plus générales, qui peuvent s'adapter à tout type de base de données : les méthodes fondées sur l'apprentissage profond. Ces méthodes sont bien souvent fondées sur des réseaux de neurones, et donnent de bien meilleurs résultats de segmentation et de classification que les méthodes traditionnelles.

5.2.2.1 Segmentation d'Images

Plusieurs approches sont possibles pour segmenter des images avec des réseaux de neurones.
La manière la plus simple de faire une prédiction dense est de prendre des patches d'image et par la suite d'utiliser un CNN afin de prédire une valeur de classe pour le pixel central [Farabet et al., 2013; Couprie et al., 2013]. Cependant, prédire la classe de chaque pixel en observant uniquement une zone réduite est insuffisant. En effet, il manque des informations de contexte. Augmenter la taille du patch permet de pallier ce problème. Néanmoins, augmenter la taille du patch augmente le nombre de paramètres du CNN, et mène à une complexité de calcul beaucoup plus élevée. Ainsi pour ajouter du contexte dans la prise de décision, des méthodes multi-échelles ont vu le jour. Par exemple, Farabet *et al.* ont entraîné des CNN sur des patchs de l'image à plusieurs échelles différentes [Farabet et al., 2013]. Couprie *et al.* ont aussi adopté une approche multi-échelle, afin d'extraire des caractéristiques de données de profondeur, dont ils se servent pour segmenter sémantiquement une scène [Couprie et al., 2013].

Les dernières avancées pour la segmentation sémantique ont été réalisées grâce à la mise en place de réseaux de neurones complètement convolutionnels (*Fully Convolutionnal Networks - FCN*). Ces réseaux introduits par Long *et al.* [Long et al., 2015], ont pour particularité de ne posséder aucune couche de neurones complètement connectés, uniquement des couches de convolution de pooling et de sur-échantillonnage. Cela permet de réduire le nombre de paramètres du réseau et de traiter des images de taille variable. Une illustration est proposée Figure 5.4.

Cette méthode permet ainsi de segmenter sémantiquement des images de manière efficace et peu coûteuse en temps de calcul.
Cependant les cartes d'un FCN classique restent de résolution très basses comparées à l'entrée, et les sur-échantillonner causerait une perte d'information considérable. Afin de pallier ce problème, Badrinarayanan *et al.* proposent une architecture de type encodeur-décodeur [Badrinarayanan et al., 2017]. Le décodeur, afin de ne pas perdre d'information, récupère les indices des pixels ayant été choisi lors de l'opération de pooling correspondant dans l'encodeur afin de réaliser un sur-échantillonnage non linéaire. Une illustration est proposée Figure 5.5.

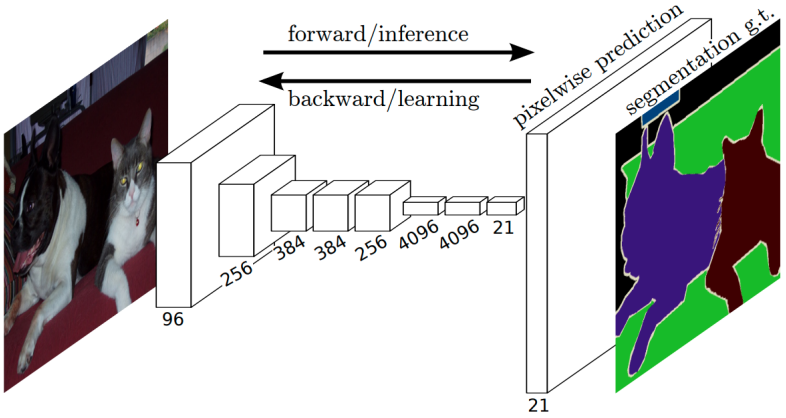


FIGURE 5.4: Illustration de la méthode FCN proposée par Long *et al.*. La figure est tirée de [Long et al., 2015].

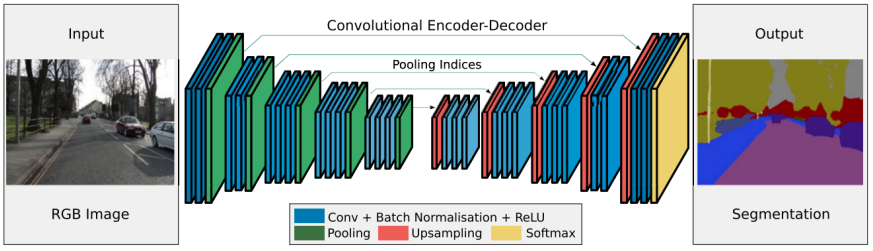


FIGURE 5.5: Illustration, tirée de [Badrinarayanan et al., 2017], du réseau SegNet proposé par Badrinarayanan *et al.*.

5.2.2.2 Segmentation de Signaux Temporels

Des méthodes fondées sur l'apprentissage profond ont également vu le jour pour la segmentation de gestes, et plus particulière la segmentation de gestes chirurgicaux.
La première est fondée sur des réseaux de neurones récurrents (*RNN*) [DiPietro et al., 2016]. En effet, en produisant une sortie y_t à chaque échantillon t, on retrouve une sortie possédant la même longueur que l'entrée. Il est donc possible de faire de la segmentation sémantique, grâce à des RNN. Les données utilisées par Di Pietro *et al.*, sont celles de la base JIGSAWS (Section 3.3.1) et chaque essai a une durée importante. Or les RNN classiques gèrent mal les séquences longues. C'est pourquoi ils ont préféré entraîner une variante des RNN : les LSTM présentés Section 2.2.2.3. En utilisant les données cinématiques disponibles dans la base et un réseau fondé sur des LSTM, Di Pietro *et al.* ont ainsi segmenté sémantiquement des gestes chirurgicaux.

Une autre méthode introduite par Lea *et al.*, illustrée Figure 5.6, utilise des convolution 1D pour segmenter des gestes [Lea et al., 2016b]. En s'inspirant des réseaux FCN créés pour la segmentation d'image, ils introduisent la même architecture pour la segmentation de séquences. Ce réseau plus léger en terme de

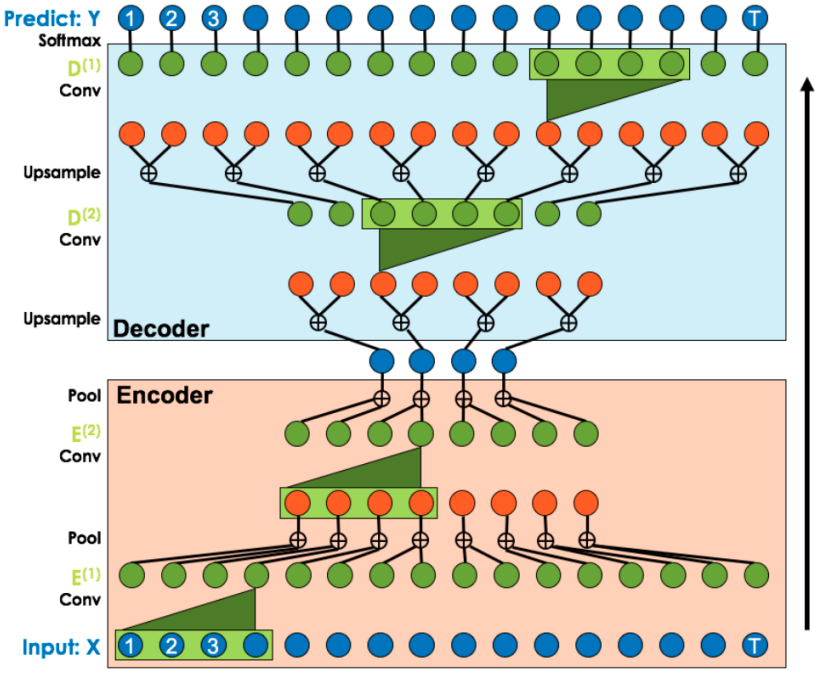


FIGURE 5.6: Illustration du réseau convolutionel introduit par Lea *et al.*, extraite de [Lea et al., 2017a].

paramètres que le réseau récurrent présenté précédemment, obtient également de meilleurs résultats de segmentation sur la base de données JIGSAWS.

5.2.3 Approche avec Apprentissage Semi-supervisées et Non supervisées

Les méthodes présentées précédemment nécessitent un grand nombre de données finement annotés (une étiquette par pixel de l'image ou par instant de la séquence). Or cette tâche d'annotation est fastidieuse et longue. C'est pourquoi des méthodes semi-supervisées voire non-supervisées ont vu le jour afin de rendre cette d'annotation plus simple, voire peut être un jour inutile.

5.2.3.1 Segmentation d'Images

Dans le domaine de la segmentation sémantique d'images, des méthodes semi-supervisées fondées sur des annotations fortes et faibles ont vu le jour.
Ainsi Papandreou *et al.* ont développé une méthode utilisant des étiquettes faibles, *i.e.* l'annotation n'est pas faite pixels par pixels, la seule information disponible est le nom des objets présents dans la scène [Papandreou et al., 2015]. Afin d'extraire des caractéristiques, un réseau convolutionnel est développé. Cependant lors de la phase d'apprentissage, comme les étiquettes des pixels ne sont pas disponibles, un algorithme d'espérance-maximisation (*Expectation-Maximization - EM*) est utilisé

[Dempster et al., 1977]. La méthode proposée, illustrée Figure 5.7, alterne entre l'estimation de la classe des pixels, et l'optimisation des paramètres du CNN.

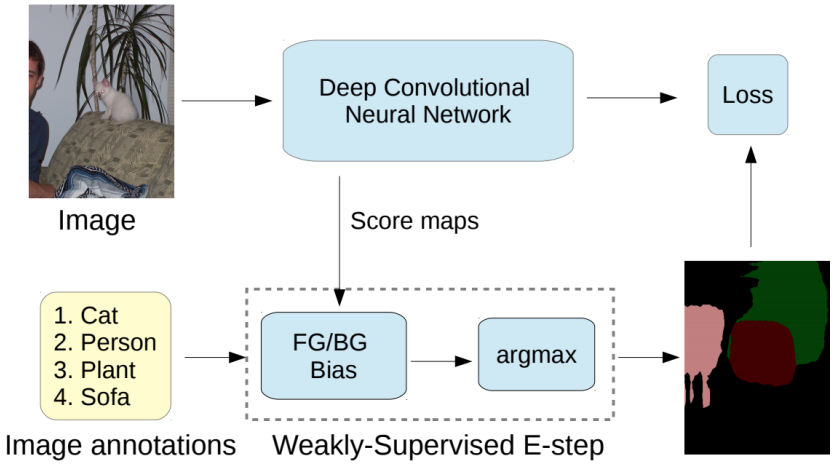


FIGURE 5.7: Illustration de la méthode d'apprentissage semi-supervisé proposée par Papandreou *et al.*, extraite de [Papandreou et al., 2015].

Afin d'obtenir des résultats plus précis, Luo *et al.* entraînent un réseau deux fois. Le premier apprentissage est effectué sur un petit jeu de données avec des annotations fortes (une étiquette par pixel). Une fois cette apprentissage terminé, le deuxième apprentissage se déroule de la même manière que pour le réseau de Papandreou *et al.*, un apprentissage semi-supervisé utilisant l'algorithme EM.
Hong *et al.* procèdent d'une façon différente [Hong et al., 2015]. Dans cette approche, les annotations faibles n'ont été utilisées que pour former un réseau de classification d'images. Par la suite, des cartes d'activations de classes, comme présentées Section 4.2, ont été calculées pour chaque image. Ensuite ces cartes d'activation ainsi que les cartes de caractéristiques de la dernière couche de convolution, sont utilisées en entrée d'un autre réseau convolutionnel entraîné à segmenter sémantiquement à partir d'annotations fortes. Cette méthode est très proche d'une méthode d'apprentissage supervisé, de par l'utilisation d'annotations fortes. Cependant il a été montré que le premier réseau de classification multi-label produit des caractéristique discriminantes pour la segmentation sémantique.

5.2.3.2 Segmentation de Signaux Temporels

Pour les signaux temporels, des méthodes proposent également des algorithmes non-supervisés afin de segmenter et de reconnaître la classe des segments.
Ainsi Despinoy *et al.* proposent une méthode non supervisée de segmentation pour la segmentation de gestes chirurgicaux réalisés à l'aide d'un robot [Despinoy et al., 2016], illustrée Figure 5.8.

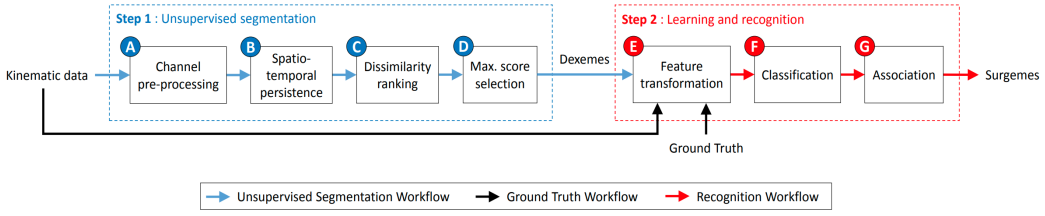


FIGURE 5.8: Illustration de la méthode proposée par Despinoy *et al.*, extraite de [Despinoy et al., 2016].

Des caractéristiques (la courbure κ, la torsion τ et leur dérivés du premier ordre en fonction de la longueur d'arc (κ_s et τ_s)) sont extraites des données cinématiques afin d'enrichir la représentation des gestes. Ces caractéristiques portent ainsi le nombre de variables représentant le geste à 24 (3 pour la position dans l'espace (x, y, z), 4 pour la représentation des rotations en quaternions, 4 caractéristiques présentées précédemment et 1 pour l'angle des pinces du robot). La segmentation est ensuite réalisée en appariant les instants critiques repérés grâce à l'algorithme introduit par [Weinkauf et al., 2010]. La reconnaissance des classes des segments est ensuite réalisée avec un avec un KNN ou un SVM.
Gao *et al.* ont également proposé une méthode non-supervisée pour la segmentation sémantique de gestes chirurgicaux [Gao et al., 2016]. La méthode extrait des caractéristiques pertinentes grâce à un réseau auto-encodeur (un réseau de neurone projetant les entrées dans un espace de plus petite dimension, et capable de reconstruire l'entrée à partir de cette représentation) et utilise l'algorithme du DTW (Section 2.2.2.1) pour aligner les différents exemples. Une fois l'alignement réalisé, un système de vote basé sur l'estimation de la densité par noyau, permet de transférer les labels d'une petite base d'essais annotés à des essais non annotés.

5.2.4 Bilan

Dans cette partie, nous avons présenté les méthodes existantes pour segmenter sémantiquement des images ou des signaux temporels. Il convient maintenant de voir quelle méthode est la plus adaptée à la tâche que nous souhaitons réalisée, qui est d'utiliser la segmentation sémantique afin de fournir un retour d'information plus global, permettant le suivi de l'acquisition des compétences

Les approches traditionnelles ont largement été dépassées ces dernières années par les approches utilisant l'apprentissage profond. Ces dernières nécessitent cependant d'avoir des grosses bases de données finement annotées, pixel par pixel, ce qui n'est pas toujours le cas. Ceci a amené aux approches semi voire non-supervisées, qui sont bien plus complexes à mettre en place. En effet utiliser des annotations faibles voire inexistantes demande de mettre en place des stratégies d'apprentissage complexes pour la segmentation sémantique. De plus, les résultats qualitatifs obtenus sont encore peu satisfaisants.

Se tourner vers des méthodes utilisant l'apprentissage profond est donc la solution que nous allons privilégier. En effet, ces méthodes amènent aux meilleurs résultats de segmentation, que ce soit pour les images ou pour les signaux temporels, dès lors que l'on possède assez de données. Parmi les méthodes existantes, les réseaux FCN ont montré les meilleurs résultats, avec peu de paramètres à apprendre et donc une puissance de calcul modérée. De plus, l'architecture introduite par Lea *et al.* [Lea et al., 2016b], qui est une variante des FCN pour les signaux temporels, amène aujourd'hui aux meilleurs résultats quantitatifs sur trois bases de données de séquences différentes. C'est donc l'approche que nous avons choisi de développer. Dans la suite du chapitre, nous présentons deux architectures fondées sur les FCN pour segmenter les gestes chirurgicaux. Ces deux approches sont par la suite testées sur la base de données JIGSAWS présentée Section 3.3.1. Enfin les segmentations obtenues sont combinées à la méthode produisant un retour d'information afin d'ajouter de la compréhensibilité.

5.3 Auto-Encodeur pour la Segmentation Temporelle

Au vu des bons résultats du réseau introduit par Lea *et al.* pour la segmentation sémantique [Lea et al., 2016a], nous avons décidé de nous en inspirer tout en rajoutant des modifications tirées de différents domaines (l'estimation de pose et la segmentation simple) afin d'améliorer les résultats de segmentation et fournir par la suite un retour d'information qui soit le plus compréhensible possible.

Dans cette section nous présentons d'abord les différents modèles qui ont servi d'inspiration. Par la suite nous présentons les deux nouveaux modèles, ainsi que les résultats de segmentation sur la base de données JIGSAWS (Section 3.3.1).

5.3.1 Auto-encodeurs de la littérature

Dans cette section, nous présentons les trois réseaux, dont nous nous sommes inspirés pour proposer deux réseaux réalisant la segmentation sémantique.

5.3.1.1 Encodeur-Décodeur avec des Convolutions Temporelles

Le réseau introduit par Lea *et al.* a pour vocation de remplacer l'approche consistant a décorréler les caractéristiques spatiale et temporelle [Lea et al., 2016a]. En effet, dans le cas de vidéos ou de séquence, il est possible d'utiliser un réseau convolutionnel afin d'extraire des informations spatiales et locales de chaque image et d'utiliser cette séquence de caractéristiques extraites en entrée d'un réseau RNN. Cette approche en plus de nécessiter l'apprentissage de deux modèles, ne permet

pas de capturer les mouvements de manière efficace, car elle ne prend pas en compte la notion de caractéristiques spatio-temporelles.
Pour palier cela, Lea *et al.* proposent donc d'utiliser un réseau fondé sur des convolutions temporelles (*Temporal Convolution Network - TCN*), illustré Figure 5.9.

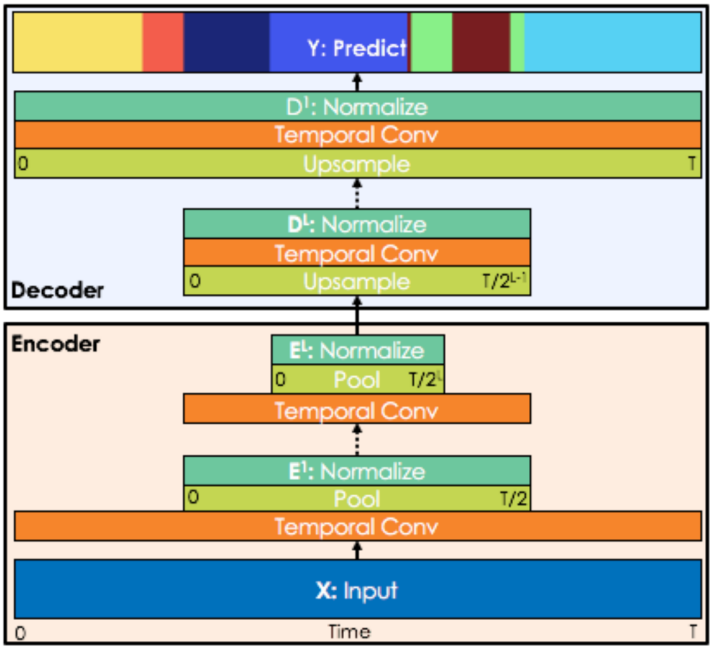


FIGURE 5.9: Illustration de l'encodeur-décodeur introduit par Lea *et al.*, extraite de [Lea et al., 2016a].

L'entrée X de ce réseau est un signal temporel, comme la sortie de capteurs, de longueur variable T. L'encodeur présenté Figure 5.9 est composé de convolution 1D suivi d'une couche de pooling et par la suite d'une normalisation des dimensions. Chaque enchaînement de ces trois couches représente un bloc, et l'encodeur est donc composé de L blocs. Le décodeur va être composé de la même façon, mais les couches de pooling sont remplacées par une couche de sur-échantillonnage, afin d'obtenir une sortie Y qui soit de longueur T.

5.3.1.2 *U-Net*

U-net est un réseau, inspiré des FCN, proposé par Ronneberger *et al.* pour segmenter des images biomédicales [Ronneberger et al., 2015]. Les FCN sont des réseaux entièrement convolutionnels mais contrairement aux architectures classiques (VVG, Resnet,...), la sortie n'est pas une classification globale mais possède la même dimension que l'entrée, ce qui veut dire que la classe de chaque pixel est prédite. Cela est rendu possible grâce à des opérations de sur-échantillonnage, comme pour le réseau Encodeur-Décodeur temporel présenté Section 5.3.1.1.

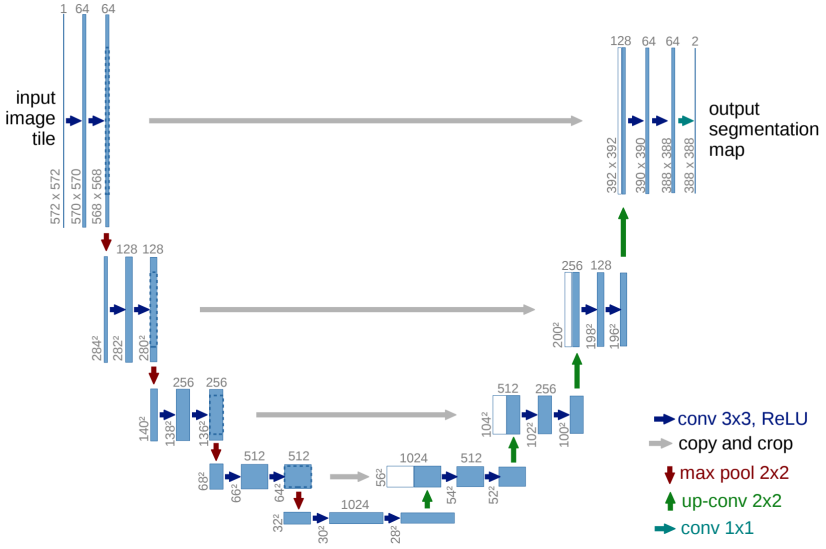


FIGURE 5.10: Illustration du réseau *U-net* développé par Ronneberg *et al.* pour la segmentation d'images biomédicales, extraite de [Ronneberger et al., 2015].

Cependant, le réseau *U-net* est un FCN modifié, illustré Figure 5.10. En effet, on retrouve comme dans les FCN, une partie encodant les informations, et une autre partie augmentant la résolution, jusqu'à arriver à la résolution de l'entrée. Cependant, contrairement au réseau FCN classique, la partie effectuant le sur-échantillonnage dispose de plus de couche, ce qui donne cette forme de "U" au réseau. De plus, des connections sont effectuées entre la moitié encodant et la partie sur-échantillonnant. En effet, pour chaque pallier de sur-échantillonnage, les cartes de caractéristiques correspondantes sont concaténées à l'entrée de cette couche afin d'obtenir une meilleure segmentation dans la couche de sortie.

5.3.1.3 Auto-Encodeur pour l'Estimation de Pose

Newell *et al.* ont introduit une nouvelle architecture pour estimer les positions 2D de squelettes à partir d'images [Newell et al., 2016]. Cette architecture en sablier, aussi appelé *Hourglass*, permet de conserver la résolution entre l'entrée et la sortie du réseau. Le réseau principal est présenté Figure 5.11.

Afin d'extraire des caractéristiques pertinentes à toutes les échelles, des connections parallèles sont introduites, de la même manière que pour *U-net* (Section 5.3.1.2). Un bloc est composé de couches de convolution et d'une couche de pooling. Avant chaque couche de pooling, une branche parallèle est créée. On trouve également des couches de convolution dans cette branche parallèle, mais pas de couche de pooling afin de conserver des informations spatiales à la résolution pré-pooling. Lorsque la résolution la plus basse est atteinte, le réseau commence à augmenter la résolution en sur-échantillonnant, et en réappliquant des convolutions. Pour assembler les caractéristiques des branches parallèles avec la branche principale, elles sont additionnées.

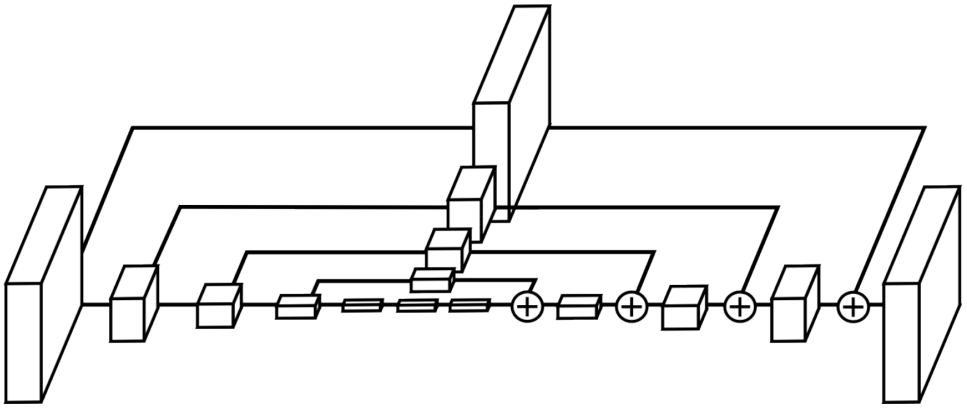


FIGURE 5.11: Illustration de l'architecture d'auto-encodeur proposé par Newell *et al.*, extraite de [Newell et al., 2016].

5.3.2 Modèles Proposés

Les architectures présentées précédemment ne servent pas toutes directement à segmenter sémantiquement, mais ont prouvé leur efficacité dans les domaines pour lesquels elles ont été développés. C'est pourquoi, nous avons décidé de combiner l'architecture TCN avec les deux modèles présentés précédemment (*U-net* et l'Auto-Encodeur) afin d'améliorer les résultats en segmentation du TCN.

5.3.2.1 Encodeur-Décodeur avec Connexions

Cette architecture est inspirée du modèle TCN proposé par Lea *et al.*. En effet, on retrouve l'architecture encodeur-décodeur qui permet de segmenter sémantiquement des signaux. Cependant, des connections entre l'encodeur et le décodeur ont été ajoutées, comme dans le réseau *U-net*. Le modèle est représenté Figure 5.12.

Ces connexions vont permettre au réseau d'améliorer les représentations dans les couches correspondant au décodeur, en apportant à la fois des informations globales (qui viennent de l'encodeur) et des informations de résolution fine (qui viennent de la connexion), ceci permet donc d'améliorer la segmentation finale. D'autre part lors de l'apprentissage, si le nombre de bloc L est important, le risque de disparition du gradient dans les premières couches est très fort, et elles seront donc mal entraînées. Avec ces connexions, le gradient circule dans le réseau par la voie principale mais également au travers des connexions. La valeur du gradient dans les premières couches a donc moins de risque d'être faible, ce qui permet un meilleur apprentissage.

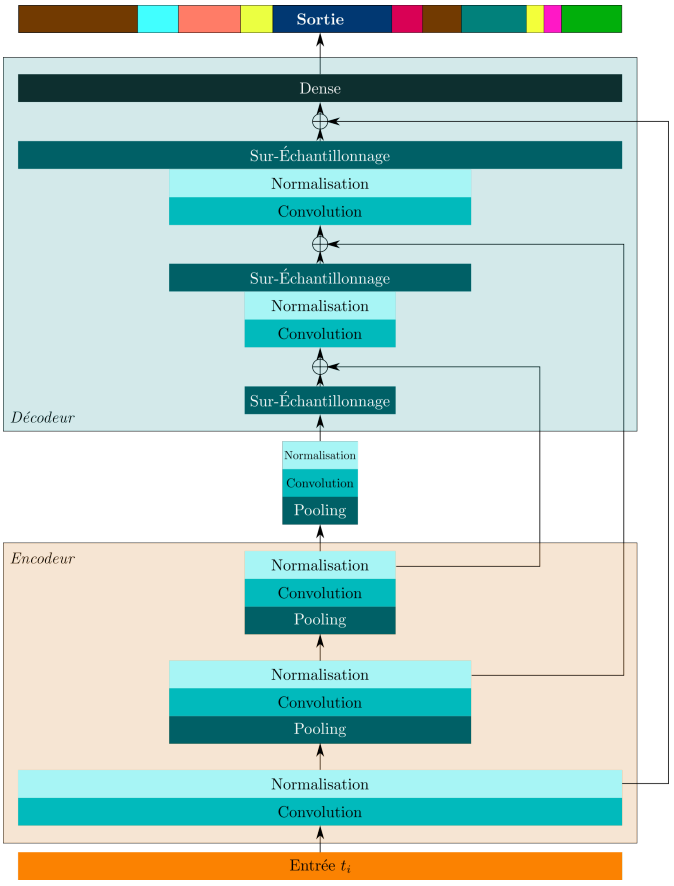


FIGURE 5.12: Modèle proposé d'encodeur-décodeur temporel avec des connexions.

5.3.2.2 Encodeur-Décodeur avec des Connexions Convolutionnelles

Cette autre architecture est inspirée du modèle TCN proposé par Lea *et al.*. De nouveau on retrouve l'architecture encodeur-décodeur pour segmenter, cette fois-ci couplée aux convolutions parallèles proposées par Newell *et al.* [Newell et al., 2016]. Lors des connections entre l'encodeur et le décodeur, une couche de convolution est ajoutée. Le modèle est représenté Figure 5.13.

Ces convolutions vont permettre de combiner des caractéristiques à différentes échelles temporelles. En effet, même si les informations fines, telles que les tremblements de la main, nécessitent des échelles temporelles fines, des mouvements plus lents et plus longs vont eux nécessiter une échelle temporelle plus larges. Donc combiner, les sorties de différentes couches de convolution permet d'extraire des informations à toutes les échelles temporelles.

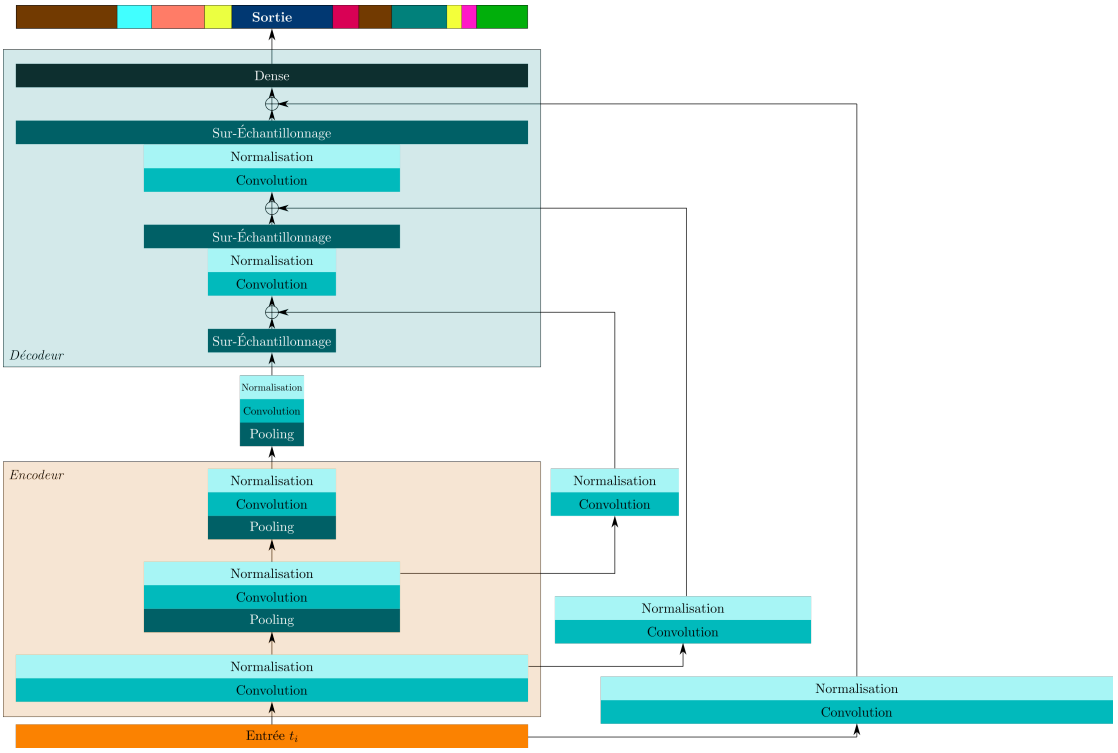


FIGURE 5.13: Modèle proposé d'encodeur-décodeur temporel avec des connexions convolutionnelles.

5.4 Résultats

Dans cette section, les résultats des architectures introduites précédemment sont présentés. Elles sont évaluées sur la base de données JIGSAWS (Section 3.3.1), en utilisant deux méthodes de validation : le *Leave-One-User-Out* (*LOUO*), où tous les essais dans un sujet sont enlevés de la base d'entraînement et constituent la base de test, et le *Leave-One-Super-Trial-Out* (*LOSO*), où un essai de chaque sujet est enlevé de la base d'apprentissage et ces essais vont constituer la base de tests. Afin d'évaluer les performances des modèles proposés, la métrique utilisée est la précision, qui représente le pourcentage d'échantillons bien classifiés. Si l'on introduit C, la matrice de confusion, alors ses éléments $C(i,j)$ avec $(i,j) \in 1...n$, où n est le nombre de classes, représentent les échantillons de la classe i reconnus comme étant de la classe j. La précision (*Accuracy*) est alors définie par :

$$Acc = \frac{\sum_{i=1}^{n} C(i,i)}{\sum_{i,j=1}^{n} C(i,j)} \tag{5.1}$$

5.4.1 Détails d'implémentation

Les deux réseaux présentés sont composés de $L = 3$ blocs, et les couches de convolution sont composées successivement de 32, 64 et 96 filtres pour l'encodeur et de 96, 64 et 32 filtres pour le décodeur. Concernant la durée des convolutions, il

a été déterminé empiriquement qu'une longueur de filtre de 30 échantillons amenait aux meilleurs résultats.
Chaque couche de pooling divise la longueur de la séquence par 2 et chaque sur-échantillonnage multiplie par 2 la longueur de l'entrée. De plus, pour éviter le sur-apprentissage un taux de dropout de 0.3 est appliqué à chaque couche de convolution, et une normalisation par batch est appliquée après chaque convolution.
En ce qui concerne les données d'entrée, Lea *et al.* ont montré que d'utiliser uniquement une sous-partie des données cinématiques donnait de meilleurs résultats de segmentation et de classification. Cette sous-partie est composée des positions (x, y, z), des vitesses de translation (v_x, v_y, v_z) et de l'angle de préhension (θ) des deux manipulateurs esclaves. De plus, le signal d'entrée dans le modèle de Lea *et al.* est sous-échantillonné : un échantillon sur trois est gardé dans le signal d'entrée du modèle. Après avoir réalisé des tests, utiliser uniquement une sous-partie des données et sous-échantillonner chaque signal donne effectivement de meilleurs résultats nous avons donc décidé d'utiliser la même approche.
Les paramètres de notre modèle ont été appris à l'aide de la fonction de coût d'entropie croisée, et l'algorithme ADAM est utilisé pour mettre à jour les poids. L'apprentissage dure 100 périodes et la taille des batchs est de 5 essais.

5.4.2 Résultats

Afin d'obtenir des résultats significatifs, chaque modèle a été entraîné 10 fois, et les résultats présentés sont les moyennes de ces 10 apprentissages.
Les premiers résultats présentés Table 5.1, sont ceux de la méthode de validation LOSO. Lea *et al.* [Lea et al., 2016b] présentant uniquement une moyenne dans leur article, nous avons relancé l'apprentissage de leur modèle pour pouvoir comparer tâche par tâche, l'apport des connexions sur l'architecture Encodeur-Décodeur.

	Suture	Parcours de l'Aiguille	Nœud	Moyenne
GMM-HMM (1)	82.22	70.55	80.95	77.91
KSVD-HMM (1)	83.40	73.09	83.54	80.01
SC-CRF (1)	85.18	77.30	80.72	81.07
ED-TCN (2)	84.22 ± 0.46	79.61 ± 0.26	82.02 ± 0.8	80.62 ± 0.51
ED-TCN-Link	88.14 ± 0.23	77.1 ± 0.96	85.36 ± 0.64	83.50 ± 0.51
ED-TCN-ConvLink	87.74 ± 0.28	76.87 ± 0.89	84.80 ± 0.65	83.14 ± 0.61

TABLE 5.1: Précision de la segmentation des deux méthodes présentées précédemment et ceux de l'état de l'art proposé par (1) [Ahmidi et al., 2017] et (2) [Lea et al., 2016b], pour la méthode validation LOSO.

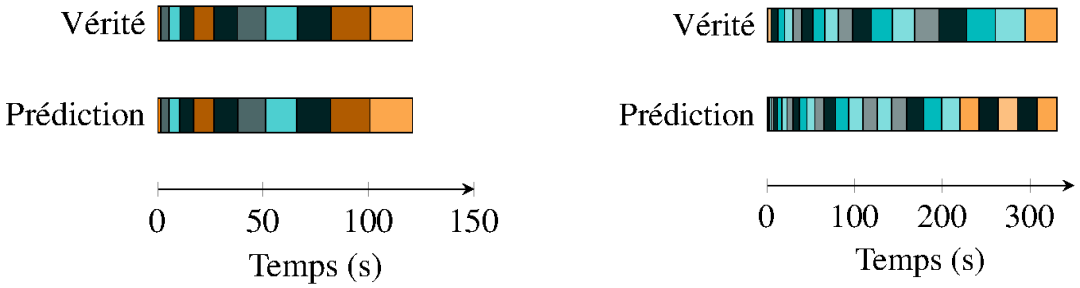
Comme on peut le voir Table 5.1, les deux modèles proposés obtiennent de meilleurs résultats en précision que les autres méthodes de l'état de l'art. Cependant on remarque que le modèle avec des connexions convolutionnelles amène à des résultats légèrement moins bons.
De plus la validation en LOSO, n'est pas très représentative de la réalité. En effet, pour une application réelle, la base d'apprentissage est exclusivement composée de gestes réalisés par des personnes, et par la suite, le modèle est testé avec des gestes réalisés par d'autres personnes. Pour cette raison, dans la suite du chapitre, les résultats présentés seront validés grâce à la méthode de validation LOUO, bien plus proche de la réalité.
La Table 5.2 présente les résultats des modèles et de l'état de l'art avec la validation LOUO.

	Suture	Parcours de l'Aiguille	Nœud	Moyenne
GMM-HMM (1)	73.95	64.13	72.47	70.18
KSVD-HMM (1)	73.45	62.78	74.89	70.37
SC-CRF (1)	81.74	74.77	78.95	78.49
LSTM (2)	80.5	N/A	N/A	N/A
BiLSTM (2)	83.3	N/A	N/A	N/A
ED-TCN (3)	80.89 ± 0.6	75.11 ± 0.45	79.49 ± 0.7	78.49 ± 0.58
ED-TCN-Link	82.97 ± 0.34	76.76 ± 0.45	83.01 ± 0.66	80.91 ± 0.48
ED-TCN-ConvLink	82.44 ± 0.48	76.37 ± 0.40	81.41 ± 0.78	80.07 ± 0.55

TABLE 5.2: Précision de la segmentation des deux méthodes présentées précédemment et ceux de l'état de l'art proposé par (1) [Ahmidi et al., 2017], (2) [DiPietro et al., 2016] et (3) [Lea et al., 2016b], pour la méthode validation LOUO.

Comme on le remarque Table 5.2, les meilleurs résultats en segmentation sont obtenus grâce aux deux architectures proposées précédemment. De la même manière, que pour la validation avec la méthode LOSO, les connexions convolutionnelles, bien que meilleures que l'état de l'art, amène à des résultats de précision plus faibles que la même architecture mais avec des connexions simples.
Au vu des deux Tables précédentes, il apparaît que les connexions simples sont suffisamment robustes pour segmenter de manière fiable et que les connexions convolutionnelles, bien qu'efficaces ne s'adaptent pas à notre problème. Ceci est sans doute dû au faible nombre d'exemples contenus dans la base de données : seuls 39 exemples sont utilisés pour entraîner le réseau. Ainsi, ajouter des couches convolutionnelles dans les branches parallèles augmente le nombre de paramètres à estimer et par la suite, peut nuire à la généralisation sur des bases de petite taille.

Après avoir regardé des résultats quantitatifs, des résultats plus qualitatifs sont présentés. On retrouve Figure 5.14, les résultats qualitatifs pour un exemple avec une précision forte et un exemple avec une précision faible.



(A) Essai avec les meilleurs résultats de segmentation.

(B) Essai avec les pires résultats de segmentation.

FIGURE 5.14: Résultats qualitatifs de l'architecture Encodeur-Décodeur avec des connexions avec la méthode de validation LOUO.

Comme on le remarque Figure 5.14, lorsqu'un exemple est bien segmenté, la suite de sous-gestes ainsi que les instants de transition correspondent parfaitement à la vérité-terrain. Cependant lorsque la précision baisse trop, la suite de sous-gestes prédite est loin de la vérité et les sauts entre les différents sous-gestes sont fréquents.

5.4.3 Comment utiliser la Segmentation pour donner un Retour d'Information ?

Maintenant que des méthodes de segmentation, ont été développées, il est intéressant de voir comment les coupler à l'évaluation de gestes et au retour d'information. En effet, ici, le but final de la segmentation sémantique est d'indiquer quel sous-geste a été mal réalisé, afin d'ajouter de la compréhensibilité à la méthode d'information. La segmentation sémantique permettra également de fournir des statistiques sur le long terme sur les compétences acquises.
Pour coupler la segmentation et le retour d'information, nous avons choisi de les faire travailler en parallèle. Le modèle de segmentation va classifier chaque échantillon d'une réalisation et en parallèle le modèle d'évaluation va fournir une note à cette même réalisation.
Une fois la note obtenue, la méthode AGRA est appliquée afin d'obtenir un retour d'information sur les instants erronés. Le retour d'information fourni, pour chaque échantillon et chaque dimension du signal d'entrée est une valeur proportionnelle à l'erreur commise (gradient). Les données cinématiques sont composées de 76 dimensions à chaque instant et donc le retour d'information fourni par AGRA est lui aussi composé de 76 dimensions. Il est difficile, à partir de ce signal multi-dimensionnel, de longueur variable, de remonter aux compétences acquises par

l'apprenant. Ce retour d'information brut est donc peu compréhensible. Une première simplification consiste à estimer la norme du gradient estimé par AGRA à chaque instant. La Figure 5.15 présente le couplage de la segmentation et de l'évaluation en utilisant le réseau siamois à sorties multiples présenté Section 3.4.2, pour une tâche de Suture. L'exemple choisi a obtenu un score de précision de segmentation de 92% et le score moyen prédit par les dix modèles d'évaluation est de 20.24 ± 1.76 pour une vérité terrain de 23. Rappelons que la note correspondant à une réalisation parfaite de gestes est de 30.

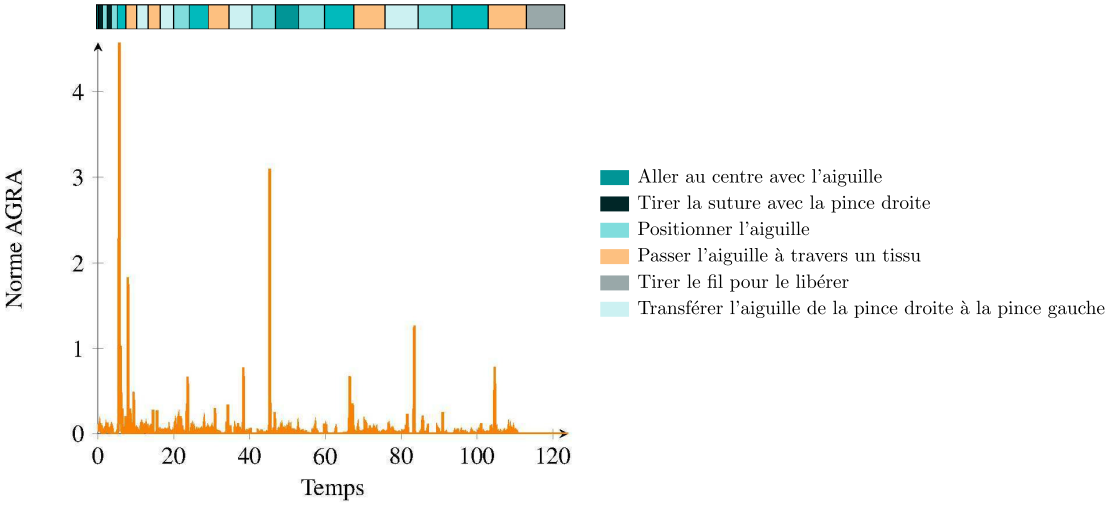


FIGURE 5.15: Couplage de l'évaluation et de la segmentation sémantique pour une tâche de Suture ayant obtenu un score de 23.

Comme on peut l'observer Figure 5.15, la norme du gradient générée grâce à la méthode AGRA est très peu bruitée, et des instants considérés comme erronés ressortent. Malheureusement, la vérité-terrain pour le retour d'information n'existant pas pour la base de données JIGSAWS, il est impossible de vérifier si les instants détectés sont effectivement erronés. Néanmoins, si l'on se fie aux résultats sur la base de données synthétique, les instants détectés ont de fortes chance d'être erronés. Ainsi, si l'on couple ces instants avec la segmentation de la tâche, on remonte aux sous-gestes correspondant aux instants détectés et il est donc possible d'indiquer à l'apprenant qu'une faute a sûrement été commise lors du premier transfert de l'aiguille (premier instant détecté), ou encore lorsque le fil a été tiré (dernier instant détecté).
La même procédure est effectuée sur un deuxième exemple supposé moins bon ayant une note prédite de 15.74 ± 0.71 pour une vérité terrain de 12 (Figure 5.16).

Pour l'exemple présenté Figure 5.16, il est également possible de remonter aux instants supposés erronés et donc aux sous-gestes correspondants.
Pour pousser l'apprentissage plus loin, il serait intéressant de faire ressortir des statistiques de ces erreurs, par exemple quels sous-gestes est le moins bien réalisé par quelqu'un, afin qu'il puise s'entraîner à refaire ce geste.
Comme expliqué précédemment, le gradient est lié, au niveau de l'erreur. Le score

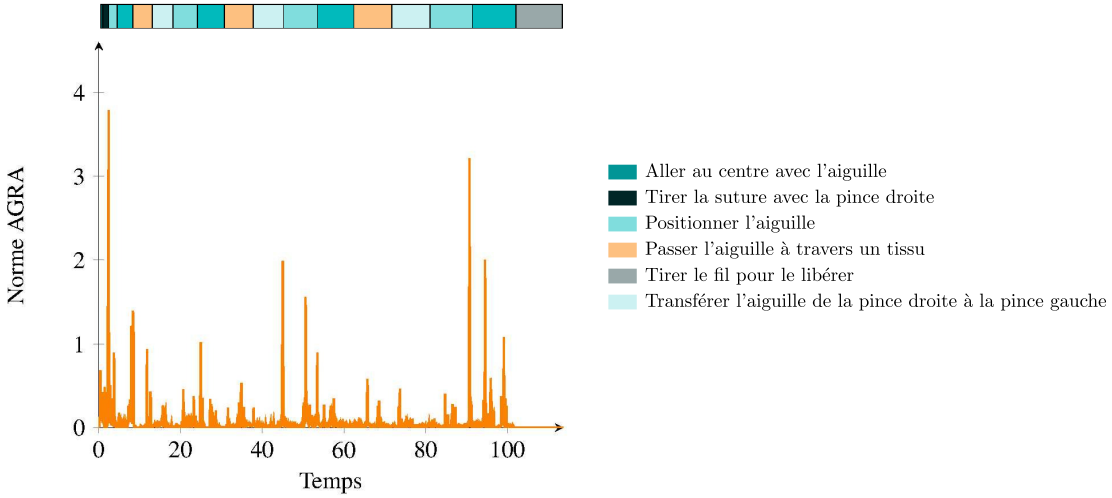
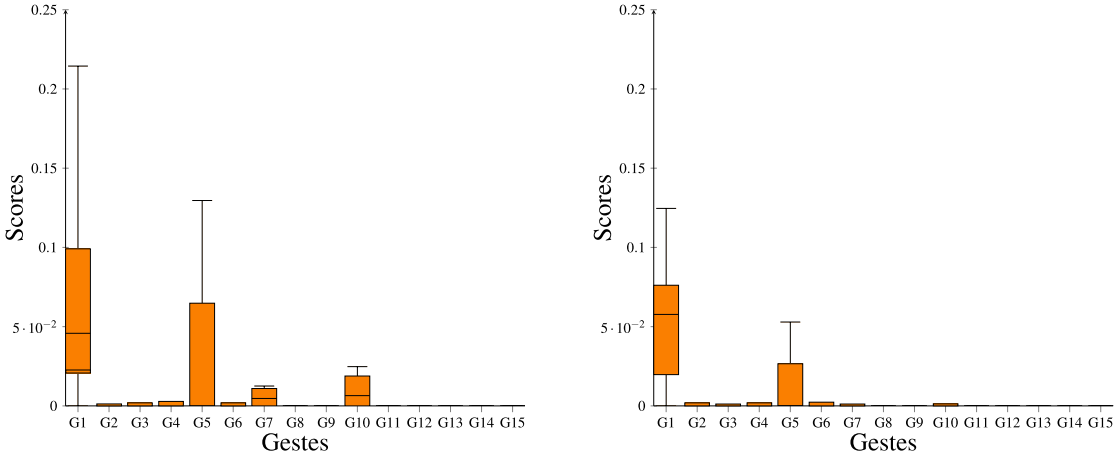


FIGURE 5.16: Couplage de l'évaluation et de la segmentation sémantique pour une tâche de Suture ayant obtenu un score de 12.

d'erreur proposé est la moyenne de la norme du gradient par sous-geste. Ainsi plus ce score est grand, plus le sous-geste est mal réalisé. Une fois les notes obtenues pour tous les essais d'une même personne, il est possible de faire des statistiques, pour lui indiquer les gestes les moins bien réalisés On retrouve ces statistiques pour deux personnes de la base de données, Figure 5.17.



(A) Statistique pour une personne ayant eu des notes entre 10 et 17.

(B) Statistique pour une personne ayant eu des notes entre 17 et 23.

FIGURE 5.17: Statistiques sur les notes obtenues par sous-gestes pour deux personnes.

Tous les sous-gestes n'étant pas présents dans la tâche de Suture, il est impossible de fournir des statistiques sur tous les sous-gestes. Cependant pour les sous-gestes retrouvés par l'algorithme de segmentation, on remarque qu'une personne ayant obtenues des meilleurs notes globales selon le modèle d'évaluation, obtient en général de meilleures notes par sous-gestes. Tout ces résultats restent à confirmer sur une base avec une vérité-terrain.

5.5 Bilan

En partant de la littérature existante sur la segmentation, nous avons pu explorer, dans ce chapitre, les différentes solutions existantes pour de la segmentation d'images ou de signaux.

Une architecture existante a retenu notre attention, l'encodeur-décodeur temporel proposé par Lea *et al.*, qui permet de segmenter des signaux et qui a déjà été testé pour de la segmentation de gestes. L'avantage des convolutions temporelles est qu'elles acceptent en entrée des signaux de toutes tailles et que lors de l'apprentissage des paramètres, les risques d'explosion de la valeur du gradient sont plus faibles que lors de l'apprentissage de réseaux récurrents.

En partant de cette architecture, deux nouvelles architectures sont proposées : l'encodeur-décodeur avec des connexions et l'encodeur-décodeur avec des connexions convolutionnelles. Ces deux contributions ont en commun d'utiliser les connexions entre les couches de l'encodeur et du décodeur.
Ces connexions impactent en premier l'apprentissage du réseau. En effet, le nombre de couche étant important, le risque d'avoir un gradient trop faible pour mettre à jour les poids des premières couches de l'encodeur est grand. Avec ces connexions, le gradient passe par deux chemins différents et réduit donc le risque de disparition.
La deuxième contribution des connexions est qu'elles permettent au réseau d'apprendre une représentation multi-résolution où à la fois les détails fins et des caractéristiques plus grossières sont représentées

Les deux propositions ont été testées sur une base de données de gestes chirurgicaux. Sur cette base de données, les résultats obtenus en précision sont meilleurs que ceux de l'état de l'art.
Par la suite, les segmentations obtenues sont couplées à une méthode d'évaluation proposée Chapitre 3 et à la méthode AGRA (Chapitre 4), afin de proposer un retour d'information qui soit le plus compréhensible possible.

Chapitre 6

Conclusion et Perspectives

Cette thèse vise à concevoir un système de coachs virtuels, capable d'aider et d'entraîner une personne à apprendre des gestes, que ce soit des gestes sportifs ou d'autres types de gestes, comme des gestes chirurgicaux. Un tel système soulève des problématiques auxquelles cette thèse a tenté de répondre.

Au cours de la thèse deux grandes problématiques ont été soulevées : comment évaluer automatiquement un geste et comment par la suite donner un retour d'information pertinent sur la réalisation du geste afin de l'améliorer.
Ces deux tâches étant différentes mais complémentaires, le choix a été fait de les traiter séparément afin d'obtenir les meilleurs résultats possibles. Ainsi après un état de l'art sur la problématique, notre choix s'est porté sur le développement d'une méthode fondé sur les réseaux de neurones. En effet, ces méthodes d'apprentissage statistique ont montré de très bonnes performances dans de nombreux domaines, comme la reconnaissance de scène et de gestes ou la segmentation sémantique d'images et de vidéos ou encore la synthèse d'images/de gestes.

Après avoir étudié les solutions spécifiques à l'évaluation automatique de gestes, notre choix s'est porté sur une architecture particulière des réseaux de neurones : l'architecture siamoise. Celle-ci permet de comparer deux réalisations de gestes mis entrée du système, notamment pour les classer selon leur niveau de réalisation. Une fois le classement de tous les couples de gestes réalisé, il est assez facile de remonter au classement global des gestes par qualité. L'architecture siamoise amène à de très bonnes performances. Intuitivement, il est plus facile de comparer deux gestes pour expliquer leur écart de note que d'expliquer les notes individuelles des gestes, surtout lorsque l'on a peu d'exemple pour généraliser. Cependant, un problème demeure : l'évaluation réalisée dépend des différentes comparaisons et donc des autres réalisations. Si tous les gestes sont mal réalisés, ce système trie les exemples par ordre de qualité mais à aucun moment ne renseigne sur la mauvaise qualité générale. D'autre part, ce système impose seulement le tri des exemples mais aucune contrainte n'est appliquée pour que l'écart de notes soit respecté entre les exemples. Nous avons donc introduit une nouvelle fonction de coût aux

réseaux siamois qui contraint le réseau à respecter l'écart de notes. Cette nouvelle fonction de coût améliore le classement des réalisations mais ne permet toujours pas de répondre au premier problème évoqué : elle ne permet pas de remonter au score absolu de chaque réalisation de geste. Pour palier ce problème, nous avons proposé deux solutions fondées sur les réseaux siamois qui amènent à une évaluation des réalisations selon une échelle de notation et non pas juste un classement des exemples. Ces deux solutions ont été testées sur deux bases de données publiques se concentrant sur l'évaluation de gestes : la base JIGSAWS, composée de gestes chirurgicaux et la base AQA-7 composée de gestes sportifs. Sur ces deux bases de données, les deux solutions proposées dans le Chapitre 3 ont amené à de très bonnes performances pour l'évaluation de la qualité des gestes. Elles se sont d'autre part avérées également plus efficaces pour le classement des gestes selon leur niveau de qualité.

Obtenir un score sur sa performance n'est pas suffisant pour l'apprentissage d'un geste et le retour d'information sur les erreurs commises est essentiel. Pour cela, il est nécessaire d'expliciter la décision – le score – fourni par le réseau de neurones. Après un état de l'art sur les méthodes d'explicabilité des réseaux, toutes utilisées lors de tâches de classification, notre choix s'est porté vers les méthodes fondées sur le gradient. En effet, ce sont les seules qui paraissent adaptables à un problème de régression. Le gradient de la sortie par rapport aux variables d'entrée amène à une carte de sensibilité représentant l'influence des variables d'entrée sur la sortie. Cette carte n'amène pas à l'information recherchée pour notre problématique car les variables d'entrée peuvent avoir une influence positive sur le score (endroits bien réalisés visant à augmenter le score) ou une influence négative. Une adaptation est ainsi nécessaire. Nous proposons de changer le calcul du gradient et d'utiliser une fonction qui compare la sortie du réseau au score parfait de la tâche réalisée. En utilisant ce gradient pour modifier les variables d'entrée, ceci permet de retoucher progressivement le geste de manière à le rapprocher d'un geste au score idéal. La comparaison de l'entrée et de l'entrée modifiée amène alors aux erreurs commises lors de la réalisation du geste. Un problème demeure : le gradient est très bruité et amène donc à des résultats également bruités. D'autre part, ils varient énormément en fonction de l'apprentissage du réseau et peuvent s'avérer plus ou moins pertinents quant au retour d'information fourni. Pour remédier à ces problèmes, nous avons introduit la méthode AGRA, qui consiste à entraîner un grand nombre de modèles de réseaux de neurones sur la même tâche et par la suite à moyenner les gradients obtenus. Afin de valider cette méthode, et face au manque de bases de données avec vérité de terrain sur les erreurs commises lors de la réalisation d'un geste, nous avons réalisé une base de données de signaux de synthèse sur lesquels nous avons introduit des perturbations. La méthode AGRA a montré des résultats très compétitifs sur cette base de données, comparée à d'autres méthodes de l'état de l'art.

Les résultats bruts fournis par la méthode sont assez difficilement compréhensibles pour une personne qui n'est pas du domaine de l'intelligence artificielle.

D'autre part, ils ne permettent pas de réaliser de statistiques, si ce n'est sur l'évolution du score global. Aussi, nous avons trouvé intéressant de réaliser une segmentation sémantique de gestes en sous-gestes. En effet, si le geste évalué est en parallèle segmenté et que chaque segment est reconnu, alors il est possible d'associer un score à chaque sous-geste et de donner les statistiques correspondantes pour avoir un retour plus fin sur les compétences acquises. Afin de valider cette démarche, nous avons utilisé une base de signaux chirurgicaux de la littérature qui a l'avantage de posséder des annotations de segmentation sémantique en gestes comme « Prendre l'aiguille avec la pince droite », « Tirer la suture avec la pince gauche » ou encore « Faire une boucle autour de la main droite ». Nous avons également proposé une nouvelle architecture pour la segmentation sémantique qui dépasse l'état de l'art sur cette base de données composée de très peu d'exemples.

Perspectives

Les travaux réalisés lors de cette thèse permettent de mettre en lumière plusieurs constats et d'envisager des perspectives à plus ou moins long terme.

La première perspective est de valider ces travaux, particulièrement ceux sur le retour d'information, sur une base de données réelle. Cela demande donc d'acquérir une nouvelle base de données de gestes mais surtout de l'annoter avec à la fois un score global de qualité, mais aussi les erreurs commises lors de la réalisation du geste, avec des annotations aussi bien temporelles que spatiales. Une telle base de données permettra de valider la méthode AGRA quant à sa capacité à fournir un retour d'information correct.

Une seconde perspective réside dans la façon de donner un retour pertinent à l'utilisateur. En effet, il n'est pas envisageable de fournir directement les données brutes résultant de l'explicabilité du réseau de neurones. Ainsi un post-traitement reste à faire sur ces données de manière à les rendre intelligibles, soit directement par l'utilisateur soit par un système de plus haut niveau fournissant par exemple, un retour tactile.

Une façon d'interpréter les résultats est de réaliser une segmentation sémantique, comme proposée Chapitre 5. Ceci ne se substitue pas au retour fin d'information qui permet de remonter aux vrais défauts du geste mais permet de tracer des courbes d'apprentissage de compétences, utiles pour mesurer l'apprentissage à long terme. Nous avons proposé une architecture permettant la segmentation sémantique sur une base particulière possédant très peu de signaux. L'approche proposée n'utilise donc que très peu d'information en entrée et l'architecture est peu profonde, de manière à amener à une bonne généralisation sur cette base. Nul doute que pour des bases de données de taille raisonnable, une nouvelle architecture, plus profonde, sera plus adaptée et menera à de meilleurs résultats de segmentation.

Bibliographie

Ahmidi, N., Tao, L., Sefati, S., Gao, Y., Lea, C., Haro, B. B., Zappella, L., Khudanpur, S., Vidal, R., and Hager, G. D. (2017). A Dataset and Benchmarks for Segmentation and Recognition of Gestures in Robotic Surgery. *IEEE Transactions on Biomedical Engineering*, 64(9) :2025–2041. *4 citations page xi, 85, 97, and 98*

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7). *3 citations page 63, 72, and 75*

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet : A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39 :2481–2495. *3 citations page viii, 86, and 87*

Bertasius, G., Stella, X. Y., Park, H. S., and Shi, J. (2016). Am i a baller ? basketball skill assessment using first-person cameras. *CoRR*. *Cited page 33*

Bialkowski, A., Lucey, P., Carr, P., Denman, S., Matthews, I. A., and Sridharan, S. (2013). Recognising team activities from noisy data. *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 984–990. *Cited page 8*

Bouma, H., Schutte, K., ten Hove, J.-M., Burghouts, G. J., and Baan, J. (2018). Flexible human-definable automatic behavior analysis for suspicious activity detection in surveillance cameras to protect critical infrastructures. In *Security + Defence*. *Cited page 8*

Breiman, L. (2001). Random forests. *Machine Learning*, 45 :5–32. *Cited page 12*

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1993). Signature verification using a siamese time delay neural network. In *NIPS*. *2 citations page 14 and 15*

Brostow, G., Fauqueur, J., and Cipolla, R. (2009). Semantic object classes in video : A high-definition ground truth database. *Pattern Recognit. Lett.*, 30 :88–97. *2 citations page viii and 83*

Burns, A.-M., Kulpa, R., Durny, A., Spanlang, B., Slater, M., and Multon, F. (2011). Using virtual humans and computer animations to learn complex motor skills : a case study in karate. In *BIO Web of Conferences*. *2 citations page 9 and 30*

Candalh-Touta, N. (2018). *Assistance à l'Apprentissage de la Dextérité en Laparoscopie*. PhD thesis, Sorbonne Université. *Cited page 57*

Caramiaux, B., Wanderley, M. M., and Bevilacqua, F. (2012). Segmenting and parsing instrumentalists' gestures. *Journal of New Music Research*, 41 :13 – 29. *Cited page 9*

Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition ? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. *2 citations page vii and 19*

Chechik, G., Sharma, V., Shalit, U., and Bengio, S. (2009). Large scale online learning of image similarity through ranking. In *IbPRIA*. *Cited page 16*

Chen, F.-S., Fu, C.-M., and Huang, C.-L. (2003). Hand gesture recognition using a real-time tracking method and hidden markov models. *Image Vis. Comput.*, 21 :745–758. *Cited page 18*

Chung, D., Tahboub, K., and Delp, E. J. (2017). A two stream siamese convolutional neural network for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1983–1991. *Cited page 15*

Coupeté, E. (2016). *Recognition of gestures and actions for man and robot collaboration on assembly line*. Theses, PSL Research University. *Cited page 8*

Couprie, C., Farabet, C., Najman, L., and LeCun, Y. (2013). Indoor semantic segmentation using depth information. *CoRR*, abs/1301.3572. *Cited page 86*

de Carvalho Correia, A. C., de Miranda, L. C., and Hornung, H. H. (2013). Gesture-based interaction in domotic environments : State of the art and hci framework inspired by the diversity. In *INTERACT*. *Cited page 8*

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em - algorithm plus discussions on the paper. *Cited page 89*

Despinoy, F., Bouget, D., Forestier, G., Penet, C., Zemiti, N., Poignet, P., and Jannin, P. (2016). Unsupervised Trajectory Segmentation for Surgical Gesture Recognition in Robotic Training. *IEEE Transactions on Biomedical Engineering*, 63(6) :1280–1291. *3 citations page viii, 89, and 90*

Diggin, D., O'Regan, C., Whelan, N., Daly, S., McLoughlin, V., McNamara, L., and Reilly, A. (2011). A biomechanical analysis of front versus back squat : Injury implications. In *International Conference on Biomechanics in Sports*. *Cited page 24*

DiPietro, R., Lea, C., Malpani, A., Ahmidi, N., Vedula, S., Lee, G. I., Lee, M., and Hager, G. (2016). Recognizing Surgical Activities with Recurrent Neural Networks. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, volume 9900, pages 551–558. *5 citations page xi, 9, 20, 87, and 98*

Doughty, H., Damen, D., and Mayol-Cuevas, W. W. (2018). Who's better ? who's best ? pairwise deep ranking for skill determination. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.* *3 citations page 32, 40, and 49*

Doughty, H., Mayol-Cuevas, W. W., and Damen, D. (2019). The pros and cons : Rank-aware temporal attention for skill determination in long videos. *Computer Vision and Pattern Recognition.* *6 citations page vii, 32, 33, 40, 41, and 59*

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., and Vapnik, V. (1996). Support vector regression machines. In *NIPS.* *Cited page 11*

Du, Y., Wang, W., and Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118. *Cited page 20*

Fang, S., Achard, C., and Dubuisson, S. (2016). Modeling the synchrony between interacting people : application to role recognition. *Multimedia Tools and Applications*, 77 :503–518. *Cited page 3*

Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2013). Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 :1915–1929. *Cited page 86*

Fard, M. J., Ameri, S., Chinnam, R. B., Pandya, A. K., Klein, M. D., and Ellis, R. D. (2016). Machine learning approach for skill evaluation in robotic-assisted surgery. *Proceedings of the World Congress on Engineering and Computer Science.* *Cited page 27*

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2018). Evaluating surgical skills from kinematic data using convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 214–221. Springer. *3 citations page 32, 61, and 62*

Feygin, D., Keehner, M., and Tendick, F. (2002). Haptic guidance : experimental evaluation of a haptic training method for a perceptual motor skill. *Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2002*, pages 40–47. *Cited page 57*

Forney, G. (1973). The viterbi algorithm. *Proceedings of the IEEE*, 61 :268–278. *Cited page 18*

Funke, I., Mees, S. T., Weitz, J., and Speidel, S. (2019). Video-based surgical skill assessment using 3d-convolutional neural networks. *International Journal of Computer Assisted Radiology and Surgery.* *Cited page 32*

Gao, Y., Vedula, S. S., Lee, G. I., Lee, M. R., Khudanpur, S., and Hager, G. D. (2016). Unsupervised surgical data alignment with application to automatic activity annotation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4158–4163. *Cited page 90*

Gao, Y., Vedula, S. S., Reiley, C. E., Ahmidi, N., Varadarajan, B., Lin, H. C., Tao, L., Zappella, L., Béjar, B., Yuh, D. D., Chen, C. C. G., Vidal, R., Khudanpur, S., and Hager, G. D. (2014). Jhu-isi gesture and skill assessment working set (jigsaws) : A surgical activity dataset for human motion modeling. *5 citations page vii, xi, 36, 39, and 44*

Gong, K., Guan, J., Liu, C., and Qi, J. (2019). Pet image denoising using a deep neural network through fine tuning. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 3(2) :153–161. *Cited page 43*

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2 :1735–1742. *Cited page 15*

Haralick, R., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. *IEEE Trans. Syst. Man Cybern.*, 3 :610–621. *Cited page 29*

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* *Cited page 14*

He, X., Zemel, R., and Ray, D. (2006). Learning and incorporating top-down cues in image segmentation. In *ECCV.* *Cited page 84*

Hebb, D. O. (1949). The organization of behavior : A neuropsychological theory. *New York, Wiley and Sons.* *Cited page 13*

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation.* *2 citations page 20 and 50*

Holden, D., Saito, J., and Komura, T. (2016). A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph.*, 35 :138 :1–138 :11. *Cited page 8*

Hong, S., Noh, H., and Han, B. (2015). Decoupled deep neural network for semi-supervised semantic segmentation. In *NIPS.* *Cited page 89*

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79 8 :2554–8. *Cited page 13*

Jie Huang, Wengang Zhou, Houqiang Li, and Weiping Li (2015). Sign language recognition using 3d convolutional neural networks. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. *Cited page 19*

John, F., Hipiny, I., Ujir, H., and Sunar, M. S. (2019). Assessing performance of aerobic routines using background subtraction and intersected image region. *2019 International Conference on Computer and Drone Applications (IConDA)*, pages 38–41. *Cited page 30*

Joshi, A., Monnier, C., Betke, M., and Sclaroff, S. (2015). A random forest approach to segmenting and classifying gestures. *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 1 :1–7. *Cited page 13*

Judkins, T. N., Oleynikov, D., and Stergiou, N. (2009). Objective evaluation of expert and novice performance during robotic surgical training tasks. *Surgical endoscopy*. *Cited page 27*

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*. *2 citations page 18 and 50*

Keogh, E. J. and Ratanamahatana, C. A. (2004). Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7 :358–386. *Cited page 17*

Kim, B., Seo, J., Jeon, S., Koo, J., Choe, J., and Jeon, T. (2019). Why are saliency maps noisy ? cause of and solution to noisy saliency maps. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4149–4157. *Cited page 69*

Kingma, D. P. and Ba, J. (2015). Adam : A method for stochastic optimization. *CoRR*, abs/1412.6980. *2 citations page 47 and 51*

Komura, T., Lam, B., Lau, R. W., and Leung, H. (2006). e-learning martial arts. In *International Conference on Web-Based Learning*. *Cited page 30*

Konishi, S. and Yuille, A. (2000). Statistical cues for domain specific image segmentation with performance analysis. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, 1 :125–132 vol.1. *Cited page 83*

Kontschieder, P., Bulò, S. R., Bischof, H., and Pelillo, M. (2011). Structured class-labels in random forests for semantic image labelling. *2011 International Conference on Computer Vision*, pages 2190–2197. *Cited page 83*

Kulpa, R., Multon, F., and Arnaldi, B. (2005). Morphology-independent representation of motions for interactive human-like animation. *Comput. Graph. Forum*, 24 :343–352. *Cited page 8*

Kyan, M., Sun, G., Li, H., Zhong, L., Muneesawang, P., Dong, N., Elder, B., and Guan, L. (2015). An approach to ballet dance training through ms kinect and visualization in a cave virtual reality environment. *ACM Transactions on Intelligent Systems and Technology (TIST)*. *Cited page 58*

Ladicky, L., Russell, C., Kohli, P., and Torr, P. (2009). Associative hierarchical crfs for object class image segmentation. *2009 IEEE 12th International Conference on Computer Vision*, pages 739–746. *Cited page 84*

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *ICML*. *Cited page 83*

Laptev, I., Caputo, B., Schüldt, C., and Lindeberg, T. (2007). Local velocity-adapted motion events for spatio-temporal recognition. *Comput. Vis. Image Underst.*, 108 :207–229. *Cited page 10*

Lea, C., Flynn, M., Vidal, R., Reiter, A., and Hager, G. (2017a). Temporal Convolutional Networks for Action Segmentation and Detection. pages 1003–1012. *2 citations page viii and 88*

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. (2017b). Temporal convolutional networks for action segmentation and detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. *Cited page 19*

Lea, C., Vidal, R., and Hager, G. D. (2016a). Learning convolutional action primitives for fine-grained action recognition. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1642–1649. *4 citations page ix, 8, 91, and 92*

Lea, C., Vidal, R., Reiter, A., and Hager, G. (2016b). Temporal Convolutional Networks : A Unified Approach to Action Segmentation. In *Computer Vision – ECCV 2016 Workshops*, volume 9915, pages 47–54. Springer International Publishing. *5 citations page xi, 87, 91, 97, and 98*

LeCun, Y., Haffner, P., Bottou, L., and Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*. *Cited page 14*

Lei, J., Li, G., Zhang, J., Guo, Q., and Tu, D. (2016). Continuous action segmentation and recognition using hybrid convolutional neural network-hidden markov model model. *IET Comput. Vis.*, 10 :537–544. *2 citations page 8 and 18*

Li, Z., Huang, Y., Cai, M., and Sato, Y. (2019). Manipulation-skill assessment from videos with spatial attention network. *ArXiv*. *6 citations page viii, 40, 41, 49, 59, and 60*

Liu, C., Yuen, J., and Torralba, A. (2011). Nonparametric scene parsing via label transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33 :2368–2382. *Cited page 84*

Liu, Y., Wang, X., and Yan, K. (2016). Hand gesture recognition based on concentric circular scan lines and weighted k-nearest neighbor algorithm. *Multimedia Tools and Applications*, 77 :209–223. *Cited page 10*

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440. *3 citations page viii, 86, and 87*

Maes, P.-J., Amelynck, D., and Leman, M. (2013). Dance-the-music : An educational platform for the modeling, recognition and audiovisual monitoring of dance steps using spatiotemporal motion templates. *EURASIP Journal on Advances in Signal Processing*. *Cited page 30*

Martin, J. A. G., Regehr, G., Reznick, R., MacRae, H., Murnaghan, J., Hutchison, C., and Brown, M. (1997). Objective structured assessment of technical skill (osats) for surgical residents. *Br J Surg*. *Cited page 37*

Mavroudi, E., Bhaskara, D., Sefati, S., Ali, H., and Vidal, R. (2018). End-to-end fine-grained action segmentation and recognition using conditional random field models and discriminative sparse coding. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1558–1567. *Cited page 8*

McCullogh, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activit. *Bull. of Math Biophysics*, 5 :115. *Cited page 13*

Mokhber, A., Achard, C., and Milgram, M. (2008). Recognition of human behavior by space-time silhouette characterization. *Pattern Recognit. Lett.*, 29 :81–89. *Cited page 10*

Moldagulova, A. and Sulaiman, R. B. (2017). Using knn algorithm for classification of textual documents. In *2017 8th International Conference on Information Technology (ICIT)*, pages 665–671. *Cited page 10*

Morel, M., Achard, C., Kulpa, R., and Dubuisson, S. (2017a). Automatic evaluation of sports motion : A generic computation of spatial and temporal errors. *Image Vis. Comput.*, 64 :67–78. *3 citations page 3, 9, and 17*

Morel, M., Achard, C., Kulpa, R., and Dubuisson, S. (2017b). Automatic evaluation of sports motion : A generic computation of spatial and temporal errors. *Image and Vision Computing*. *Cited page 58*

Morel, M., Achard, C., Kulpa, R., and Dubuisson, S. (2018). Time-series averaging using constrained dynamic time warping with tolerance. *ACLI*. *Cited page 58*

Mousas, C., Newbury, P. F., and Anagnostopoulos, C.-N. (2014). Analyzing and segmenting finger gestures in meaningful phases. *2014 11th International Conference on Computer Graphics, Imaging and Visualization*, pages 89–94. *Cited page 9*

Ménardais, S., Kulpa, R., Multon, F., and Arnaldi, B. (2004). Synchronization for dynamic blending of motions. In Boulic, R. and Pai, D. K., editors, *Symposium on Computer Animation*. The Eurographics Association. *Cited page 8*

Nagarajan, S. and Subashini, T. (2013). Static hand gesture recognition for sign language alphabetsusing edge oriented histogram and multi class svm. *International Journal of Computer Application*, 82. *Cited page 11*

Nakamura, T., Nagai, T., Mochihashi, D., Kobayashi, I., Asoh, H., and Kaneko, M. (2017). Segmenting continuous motions with hidden semi-markov models and gaussian processes. *Frontiers in Neurorobotics*, 11. *2 citations page 9 and 18*

Nasreddine, K. and Benzinou, A. (2016). Reconnaissance automatique de gestes manuels en langue des signes. *Cited page 8*

Newell, A., Yang, K., and Deng, J. (2016). Stacked Hourglass Networks for Human Pose Estimation. *computer vision - ECCV 2016*. *4 citations page ix, 93, 94, and 95*

Ninon, C.-T. and Szewczyk, J. (2017). How can we improve the training of laparoscopic surgery thanks to the knowledge in robotics ? In *5th International Conference on Education and InformationSystems, Technologies and Applications*. *Cited page 9*

Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.*, 29 :51–59. *Cited page 29*

Padoy, N., Blum, T., Feussner, H., Berger, M.-O., and Navab, N. (2008). On-line Recognition of Surgical Activity for Monitoring in the Operating Room. page 7. *Cited page 8*

Papandreou, G., Chen, L.-C., Murphy, K., and Yuille, A. (2015). Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1742–1750. *3 citations page viii, 88, and 89*

Parcheta, Z. and Martínez-Hinarejos, C. D. (2017). Sign language gesture recognition using hmm. In *IbPRIA*. *Cited page 18*

Park, H. S., Kim, E. Y., Jang, S. S., Park, S. H., Park, M. H., and Kim, H. J. (2005). Hmm-based gesture recognition for robot control. In *IbPRIA*. *Cited page 18*

Parmar, P. and Morris, B. T. (2017). Learning to score olympic events. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. *4 citations page vii, 33, 34, and 53*

Parmar, P. and Morris, B. T. (2018). Action quality assessment across multiple actions. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. *9 citations page vii, xi, 34, 35, 38, 39, 44, 50, and 53*

Parmar, P. and Morris, B. T. (2019). What and how well you performed ? a multitask learning approach to action quality assessment. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 304–313. *Cited page 34*

Patro, B. N., Lunayach, M., Patel, S., and Namboodiri, V. P. (2019). U-cam : Visual explanation using uncertainty based class activation maps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7444–7453. *Cited page 61*

Pham, M. T., Moreau, R., and Boulanger, P. (2010). Three-dimensional gesture comparison using curvature analysis of position and orientation. *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 6345–6348. *Cited page 17*

Pirsiavash, H., Vondrick, C., and Torralba, A. (2014). Assessing the quality of actions. In *ECCV*. *2 citations page viii and 58*

Poddar, P., Ahmidi, N., Vedula, S., Ishii, L., Hager, G., and Ishii, M. (2014). Automated objective surgical skill assessment in the operating room using unstructured tool motion. *International journal of computer assisted radiology and surgery*. *3 citations page vii, 9, and 28*

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77 :12. *Cited page 17*

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you ?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. *Cited page 62*

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net : Convolutional networks for biomedical image segmentation. In *MICCAI*. *3 citations page ix, 92, and 93*

Rosenblatt, F. (1958). The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6 :386–408. *Cited page 13*

Saha, S., Bhattacharya, S., and Konar, A. (2018). A novel approach to gesture recognition in sign language applications using avl tree and svm. pages 271–277. *Cited page 11*

Saha, S., Lahiri, R., Konar, A., Banerjee, B., and Nagar, A. K. (2017). Hmm-based gesture recognition system using kinect sensor for improvised human-computer interaction. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2776–2783. *Cited page 18*

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1) :43–49. *Cited page 17*

Schroff, F., Criminisi, A., and Zisserman, A. (2006). Single-histogram class models for image segmentation. In *ICVGIP*. *Cited page 83*

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam : Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626. *Cited page 60*

Sharghi, A., Haugerud, H., Oh, D., and Mohareri, O. (2020). Automatic Operating Room Surgical Activity Recognition for Robot-Assisted Surgery. *arXiv e-prints*. *Cited page 8*

Sharma, Y., Plötz, T., Hammerld, N., Mellor, S., McNaney, R., Olivier, P., Deshmukh, S., McCaskie, A., and Essa, I. (2014). Automated surgical osats prediction from videos. In *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*. *Cited page 29*

Shotton, J., Winn, J., Rother, C., and Criminisi, A. (2006). Textonboost : Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*. *Cited page 84*

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Not just a black box : Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org. *4 citations page 63, 72, 75, and 78*

Silberman, N. and Fergus, R. (2011). Indoor scene segmentation using a structured light sensor. *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 601–608. *Cited page 84*

Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks : Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034. *7 citations page 14, 62, 69, 72, 75, 76, and 78*

Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *NIPS*. *Cited page 19*

Singh, B., Marks, T., Jones, M. J., Tuzel, O., and Shao, M. (2016). A multi-stream bi-directional recurrent neural network for fine-grained action detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1961–1970. *Cited page 20*

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. (2017). Smoothgrad : removing noise by adding noise. *arXiv preprint arXiv :1706.03825.* *7 citations page 63, 64, 69, 72, 75, 76, and 78*

Sozykin, K., Khan, A. M., Protasov, S., and Hussain, R. (2018). Multi-label class-imbalanced action recognition in hockey videos via 3d convolutional neural networks. *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 146–151. *Cited page 8*

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. A. (2015). Striving for simplicity : The all convolutional net. *CoRR*, abs/1412.6806. *Cited page 63*

Sundararajan, M., Taly, A., and Yan, Q. (2016). Gradients of counterfactuals. *arXiv preprint arXiv :1611.02639.* *Cited page 63*

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org. *4 citations page 63, 72, 75, and 78*

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9. *2 citations page 14 and 60*

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface : Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. *Cited page 15*

Tao, L., Elhamifar, E., Khudanpur, S., Hager, G., and Vidal, R. (2012). Sparse Hidden Markov Models for Surgical Gesture Classification and Skill Evaluation. In *Information Processing in Computer-Assisted Interventions*, volume 7330, pages 167–177. Springer Berlin Heidelberg, Berlin, Heidelberg. *Cited page 18*

Tighe, J. and Lazebnik, S. (2010). Superparsing : Scalable nonparametric image parsing with superpixels. In *ECCV*. *Cited page 84*

Tran, D., Bourdev, L. D., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. *2015 IEEE International Conference on Computer Vision (ICCV).* *3 citations page 18, 33, and 50*

Tu, Z. and Bai, X. (2010). Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 :1744–1757. *Cited page 83*

Unuma, M., ichi Anjyo, K., and Takeuchi, R. (1995). Fourier principles for emotion-based human figure animation. In *SIGGRAPH '95*. *Cited page 8*

Vapnik, V. (1995). The nature of statistical learning theory. *Cited page 11*

Vedula, S. S., Malpani, A., Ahmidi, N., Khudanpur, S., Hager, G., and Chen, C. C. G. (2016). Task-level vs. segment-level quantitative metrics for surgical skill assessment. *Journal of surgical education*. *Cited page 28*

Veeriah, V., Zhuang, N., and Qi, G.-J. (2015). Differential recurrent neural networks for action recognition. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4041–4049. *Cited page 20*

Wang, H., Wang, P., Song, Z., and Li, W. (2017). Large-scale multimodal gesture segmentation and recognition based on convolutional neural networks. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 3138–3146. *Cited page 20*

Wang, J., Song, Y. J., Leung, T. K., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. *IEEE Conference on Computer Vision and Pattern Recognition*. *Cited page 41*

Wang, Z. and Fey, A. (2018). Deep Learning with Convolutional Neural Network for Objective Skill Evaluation in Robot-assisted Surgery. *International Journal of Computer Assisted Radiology and Surgery*. *2 citations page vii and 31*

Weinkauf, T., Gingold, Y. I., and Sorkine-Hornung, O. (2010). Topology-based smoothing of 2d scalar fields with c1-continuity. *Computer Graphics Forum*, 29. *Cited page 90*

Werbos, P. J. (1990). Backpropagation through time : what it does and how to do it. *Proceedings of the IEEE*, 78(10) :1550–1560. *Cited page 20*

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell : Neural image caption generation with visual attention. *ArXiv*, abs/1502.03044. *3 citations page viii, 59, and 60*

Yao, T., Mei, T., and Rui, Y. (2016). Highlight detection with pairwise deep ranking for first-person video summarization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 982–990. *Cited page 41*

Yu, H., Yang, Z., Tan, L., Wang, Y., Sun, W., Sun, M., and Tang, Y. (2018). Methods and datasets on semantic segmentation : A review. *Neurocomputing*, 304 :82–103. *2 citations page viii and 84*

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer. *Cited page 62*

Zeiler, M. D., Taylor, G. W., and Fergus, R. (2011). Adaptive deconvolutional networks for mid and high level feature learning. *2011 International Conference on Computer Vision*, pages 2018–2025. *Cited page 63*

Zhang, L., Zhu, G., Mei, L., Shen, P., Shah, S., and Bennamoun, M. (2018). Attention in convolutional lstm for gesture recognition. In *NeurIPS*. *Cited page 20*

Zhao, X., Song, Z., Guo, J., Zhao, Y., and Zheng, F. (2012). Real-time hand gesture detection and recognition by random forest. volume 289. *Cited page 13*

Zhong, D., Yang, Y., and Du, X. (2018). Palmprint recognition using siamese network. In *CCBR*. *Cited page 15*

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929. *2 citations page 60 and 61*

Zhou, D., Feng, X., Yi, P., Yang, X., Zhang, Q., Wei, X., and Yang, D. (2019). 3d human motion synthesis based on convolutional neural network. *IEEE Access*, 7 :66325–66335. *Cited page 8*

Zhou, Y., Li, Z., Xiao, S., He, C., Huang, Z., and Li, H. (2018). Auto-conditioned recurrent networks for extended complex human motion synthesis. *ICLR*. *Cited page 8*

Zhu, G., Zhang, L., Shen, P., and Song, J. (2017). Multimodal gesture recognition using 3-d convolution and convolutional lstm. *IEEE Access*, 5 :4517–4524. *Cited page 20*

Zia, A. and Essa, I. (2017). Automated Surgical Skill Assessment in RMIS Training. *International Journal of Computer Assisted Radiology and Surgery*. *2 citations page 11 and 28*

Zia, A. and Sharma, Y. (2017). Video and Accelerometer-Based Motion Analysis for Automated Surgical Skills Assessment. *Internation Journal of Computer Assisted Radiology and Surgery*, page 12. *Cited page 29*

Zia, A., Sharma, Y., Bettadapura, V., Sarin, E. L., Ploetz, T., Clements, M. A., and Essa, I. (2016). Automated video-based assessment of surgical skills for training and evaluation in medical schools. *International Journal of Computer Assisted Radiology and Surgery*, 11(9) :1623–1636. *2 citations page 9 and 29*