Détection automatisée du trouble du spectre de l'autisme via eye-tracking et réseaux de neurones artificiels : conception d'un système d'aide à la décision

Romuald Carette

Thèse de Doctorat

Mention Informatique

présentée à ***l'Ecole Doctorale en Sciences Technologie et Santé (ED 585)***

de l'Université de Picardie Jules Verne

par

Romuald Carette

pour obtenir le grade de Docteur de l'Université de Picardie Jules Verne

Détection Automatisée du Trouble du Spectre de l'Autisme via Eye-Tracking et Réseaux de Neurones Artificiels : Conception d'un Système d'Aide à la Décision

Soutenue le 25/09/2020, après avis des rapporteurs, devant le jury d'examen :

M. P. Villon, Professeur, Université de Technologie de Compiègne	**Président**
M. S. Canu, Professeur, INSA Rouen	**Rapporteur**
M^{me} S. Bringay, Professeur, Université de Montpellier	**Rapporteur**
M^{me} F. Levé, Maître de Conférence, Université de Picardie Jules Verne	**Examinateur**
M. G. Dequen, Professeur, Université de Picardie Jules Verne	**Directeur de thèse**
M. J-L. Guérin, Maître de Conférences, Université de Picardie Jules Verne	**Co-encadrant**
M. L. Vandromme, Professeur, Université de Picardie Jules Verne	**Invité**

Dédicace

A mes parents

Ce manuscrit de thèse est la conclusion de tous les efforts et sacrifices que vous avez fait pour me soutenir au cours de toutes ces années d'études, malgré les difficultés que cela a pu induire.
Ces travaux, ce manuscrit vous sont dédiés.
La fierté de cet accomplissement est également vôtre.
Je vous suis redevable et le serai toujours.

Déclaration

Cette thèse nommée "Détection Automatisée du Trouble du Spectre de l'Autisme via Eye-Tracking et Réseaux de Neurones Artificiels : Conception d'un Système d'Aide à la Décision" a été conduite au sein du Laboratoire MIS (Modélisation, Information & Systèmes - UR UPJV 4290) de l'UPJV (Université de Picardie Jules Verne située à Amiens, sous le co-encadrement de Messieurs Jean-Luc Guérin, Jérôme Bosche et Gilles Dequen.

Ces travaux ont été financés par la cadre d'un dispositif CIFRE (Convention Industrielle de Formation par la REcherche), financé par le Ministère de l'Enseignement Supérieur, de la Recherche et de l'Innovation. Ce financement est complété par la participation du Groupe Evolucare Technologies.

Je déclare que les travaux présentés dans ce manuscrit sont le résultat de mon propre travail, en collaboration avec mes collègues chercheurs. A moins qu'il n'en soit fait mention contraire, tout contenu de ce manuscrit est produit par mes travaux. Cette thèse n'a pas été acceptée, et n'a pas été soumise, pour une quelconque autre candidature à un autre diplôme.

Remerciements

Avant toute chose, je tiens à remercier Sandra Bringay et Stéphane Canu, rapporteurs de ce manuscrit pour le temps passé à la relecture et pour leur approbation quant à la tenue de la soutenance de cette thèse. Je remercie également Florence Levé et Pierre Villon en leur qualité d'examinateurs de cette soutenance, en particulier Pierre Villon pour voir accepté de présider ce jury. Enfin, je remercie Luc Vandromme, expert en psychologie, pour sa participation au jury en tant qu'invité. Merci bien sûr à Gilles Dequen et Jean-Luc Guérin, membres du jury en tant qu'équipe encadrante de ma thèse.

Ce manuscrit est le résultat d'années de travail avec le Laboratoire MIS de l'UPJV et le groupe Evolucare Technologies. L'implication du Laboratoire CRP-CPO de l'UPJV a également été fondamentale dans l'accomplissement de ce travail.

Je remercie ainsi Gilles Dequen, Jean-Luc Guérin et Jérôme Bosche pour leur soutien au cours de ces années de travaux, toujours présents malgré les difficultés d'agendas que peuvent impliquer une thèse CIFRE.

J'en profite pour remercier Anne Lapujade, qui a participé à cet encadrement la première année avant de devoir se concentrer sur la gestion du Master MIAGE, je tiens aussi à la remercier pour les opportunités d'enseignement qu'elle m'a proposé.

Je remercie Luc Vandromme et Federica Cilia, membres du CRP-CPO et de CHIMERE, qui m'ont permis de travailler sur ce sujet, apportant à la fois conseils, données et encouragements.

Je tiens à remercier les membres du laboratoire MIS, notamment l'équipe GOC. J'ai apprécié toutes les discussions, qu'elles soient scientifiques ou personnelles avec ses membres, merci notamment à Corinne, Laure, Céline et Yu.

Je tiens aussi à remercier mes collègues doctorants, toutes équipes confondues, avec un merci tout particulier à ceux qui sont devenus de proches amis : Olivier, Jordan et Clément.

Je remercie mes collègues d'Evolucare, membres de l'équipe R&D. Un merci tout particulier est dû à Vincent, qui m'a fait continuellement confiance depuis 7 ans, en tant que stagiaire, apprenti et thésard.

Merci à mes amis, amiénois et isariens, qui ont toujours été des soutiens moraux exceptionnels. En particulier, merci à Claire, Antoine, Jennifer, Mickaël et Vincent.

Merci à toute ma famille, oncles, tantes et cousins confondus. La famille m'importe plus que tout, et vous avez été essentiels à la poursuite de ce travail. Merci surtout à Mickaël et Céline qui ont montré beaucoup d'intérêt pour cette thèse.

Merci à tous ceux que je n'ai pas cité nommément, mais qui ont participé de près ou de loin à ces travaux ou à m'apporter un soutien, professionnel ou personnel.

Merci enfin à mes parents. Les mots manqueraient pour exprimer toute la gratitude que je leur porte. Je n'aurais pas pu accomplir une infime fraction de tout ceci sans leur aide, leur présence et leur soutien indéfectible.

Table des matières

Glossaire

Terme	**Signification**
ANN	Réseau de Neurones Artificiel
ACP	voir PCA
AUC	Aire sous la courbe (ROC) pour le calcul de la valeur d'efficacité d'un modèle
CARS	Score d'évaluation du TSA
CNN	ANN utilisant le principe de convolution pour accepter, notamment, des images en entrée
LSTM	Neurones (ou réseaux de neurones par abus de langage) avec fonctionnement séquentiel
MSE	Méthode de calcul de l'erreur d'évaluation
PC	Psychologie Cognitive
PCA	Méthode de réduction de dimensionnalité d'un problème
RGB	Canaux rouge/vert/bleu permettant l'affichage de la couleur sur des images
RNN	ANN permettant le traitement de données séquentielles
ROC	Courbe d'évaluation de l'efficacité d'un modèle
SRS	Score d'évaluation du TSA
TC	Enfant sans signe de TSA, d'âge chronologique équivalent
TSA	Trouble du Spectre de l'Autisme
TS	Enfant diagnostiqué TSA

Chapitre 1

Introduction

Au cours de cette thèse sous convention CIFRE, financée par le groupe Evolucare Technologies, en collaboration avec le Laboratoire MIS (Modélisation, Information, Systèmes) de l'Université de Picardie Jules Verne, j'ai travaillé sur le sujet de l'aide au diagnostic assisté par modèles obtenus via des méthodes informatiques, plus précisément des méthodes de Machine Learning.

Ces travaux conduits en Intelligence Artificielle visent le domaine de la Psychologie Cognitive, plus précisément, l'aide au diagnostic du Trouble du Spectre de l'Autisme (TSA). Ce trouble neurodéveloppemental, touchant environ 1% de la population mondiale, est notamment repérable d'une part par la présence d'un déficit persistant de la communication et des interactions sociales et d'autre part par le caractère restreint et répétitif des comportements, des intérêts ou des activités, dès la plus jeune enfance. Les interactions sociales des personnes atteintes par le TSA sont plus ou moins fortement altérées. Différents scores permettent de quantifier l'état de gravité du Trouble pour chaque cas. Les scores principaux sont le CARS (Childhood Autistic Rating Scale), le SRS (Social Responsiveness Scale). Ces scores se concentrent sur l'état des capacités cognitives de l'enfant, dans un cadre conversationnel ou dans l'étude des réactions à des stimuli particuliers. La compréhension de l'état émotionnel d'autrui, la faculté de communication verbale et non-verbale, la capacité de concentration cognitive face à une autre personne font partie des éléments d'observations pris en compte. Le quotient intellectuel (IQ) peut aussi être pris en considération.

Certains travaux utilisant un affichage graphique des données issues de relevés effectués par Eye-Tracking pour établir un diagnostic de l'état autistique d'un enfant, nous portons nos efforts sur ces mêmes données. L'Eye-Tracker est un outil de captation de la position du regard d'une personne sur un écran. Il permet de localiser le point de regard d'un participant à une fréquence dépendante du concepteur, cette fréquence pouvant varier de 60Hz à plus de 1000Hz. Après une phase de calibrage permettant d'adapter la mesure à la position relative de la tête du sujet observé, le point de regard est relevé avec précision sur l'écran à chaque instant d'enregistrement. D'autres informations peuvent être relevées, telles que les dimensions de la pupille ou la position relative des yeux par rapport à l'écran. L'Eye-Tracker permet de considérer des informations relatives à la dynamique oculaire, catégorisant le tracé du regard sur l'écran selon quatre classes : Fixation, Saccade, Blink et Post-Saccadic Oscillations (PSO). *Fixation* correspond à un ensemble de points successifs au sein d'une zone restreinte de l'écran, c'est un état stable du tracé oculaire. *Saccade* est l'opposé, un mouvement rapide entre des points éloignés. *Blink* correspond à un échec de mesure, ce qui peut correspondre à un clignement d'œil comme à un mouvement de tête du participant ou à une erreur de l'Eye-Tracker. *PSO* est l'état transitoire entre la Saccade et la Fixation, c'est la stabilisation du regard sur l'objectif du regard. Le repérage des PSO demande des mesures précises et à haute fréquence, l'appareil utilisé dans ce manuscrit, un SMI RED mobile IView XTM RED avec une fréquence de 60Hz, ne nous permet pas de les relever.

Au cours de nos travaux, nous avons utilisé différents outils de Machine Learning, en nous concentrant particulièrement sur les modèles du type Réseaux de Neurones Artificiels. En effet, nous avons également envisagé des approches relatives à la stabilité des systèmes physiques ou des approches sous forme d'algorithmes génétiques. L'une comme l'autre a produit des résultats nous ayant semblé insuffisants en termes de précision. Ainsi, elles ont été rejetés pour nous concentrer sur les Réseaux de Neurones Artificiels. Nos formats de données évoluant au cours des travaux, nous avons varié les types de Réseaux de Neurones exploités. Les Réseaux de Neurones Artificiels (RNA, dans la suite du document nous utiliserons l'acronyme ANN pour Artificial Neural Network) permettent le traitement de données de type numérique afin d'en obtenir un ou plusieurs résultats. Reprenant très schématiquement le fonctionnement du neurone biologique, le neurone artificiel reçoit un en-

semble de données d'entrée, les pondère, les traite puis les transforme en une sortie unique. Le Réseau de Neurones organise ces neurones généralement sous forme de couches successives de neurones, chaque couche connectant l'ensemble de ses neurones à ceux de la couche lui succédant. Initialement, le réseau est pondéré aléatoirement. Des données d'entrée sont fournies au réseau et ses sorties sont comparées aux valeurs attendues. L'erreur qui en ressort permet de modifier les valeurs des poids du réseau pour approcher au mieux la sortie attendue. Ce procédé est répété pour un jeu de données d'entrée dit d'entraînement, la durée de cette répétition dépendant de la difficulté du problème à résoudre. Une fois le modèle entraîné, il est supposé approximer efficacement la réponse au problème. Un jeu de données dit de validation permet alors de confirmer la généralisation du modèle entraîné.

Les Réseaux de Neurones Artificiels sont modifiés pour permettre le traitement de formats de données particuliers. Les Réseaux de Neurones Récurrents sont dédiés au traitement de séries de données numériques. Dans ces modèles, les couches reçoivent en entrée les sorties de la couche précédente, mais aussi ses propres sorties à l'instant précédent de la série de données. La connexion dite récurrente, d'une couche à elle-même, permet la simulation d'une "mémoire" du modèle. Ce moyen de conserver la mémoire pose des problèmes dits de diminution ou d'explosion du gradient dans le cas de longues séries de données à apprendre. Les Long Short-Term Memory (LSTM) et Gated Recurrent Units (GRU) ont été développés afin de pallier ce problème, le GRU étant une simplification des LSTM [Gers et al., 1999, Cho et al., 2014].

Les Réseaux de Neurones Convolutifs permettent, quant à eux, de prendre en compte les entrées présentées sous la forme d'images, en exploitant les valeurs RGB (canaux rouge, vert et bleu). Le modèle se présente en couches d'informations successives, commençant par une image. Lors du passage de l'information dans les couches, l'image subit des transformations. Pour commencer, les couches dites de convolution analysent l'image via une fenêtre glissante, permettant d'en extraire des informations locales. Le passage de cette fenêtre sur l'image entière permet la création d'une nouvelle "image" résultante. Aussi, les couches dites de pooling sont utilisées pour réduire les informations redondantes de l'image, localement. Cependant, ce traitement local n'est pas effectué par une fenêtre glissante, mais par un quadrillage de l'image en un ensemble de zones de taille identique. De chaque zone, la

couche de pooling isole une information importante (minimum, maximum ou moyenne, selon le type de pooling choisi). Ces deux types de couches sont alternées, un nombre de fois dépendant du concepteur du modèle. A ce moment, une couche de mise à plat permet de transformer une image de taille n par m en une liste de valeurs de taille nxm. Le résultat de cette couche devient l'entrée d'un ANN.

Pour certains travaux, nous avons utilisé des approches statistiques : la corrélation entre variables et l'Analyse par Composantes Principales (ACP, dans la suite du document nous utiliserons l'acronyme PCA pour Principal Component Analysis). La corrélation entre variables consiste en l'analyse du comportement de variables, deux à deux, dans le but de constater à quel point leurs tendances sont similaires. Deux variables sont corrélées si leur tendances sont identiques ou très similaires (corrélation tendant vers 1), mais aussi si ces tendances sont opposées (corrélation tendant vers -1). Au contraire, deux variables dont les comportements semblent aléatoires l'une par rapport à l'autre ne sont pas corrélées, et sont considérées indépendantes (corrélation tendant vers 0). La PCA exploite ces informations de corrélation pour réduire un ensemble de variables décrivant un problème en un nouvel ensemble de variables, réduisant la corrélation entre les variables résultantes. Après normalisation, il est recherché la projection orthogonale (donc la suppression d'une variable) permettant de maximiser la variance du jeu de données. Cette projection est ensuite appliquée, réduisant le nombre de variables de 1. Le procédé est alors répété jusqu'à atteindre le nombre de variables (ou composantes principales) souhaité.

Des travaux montrent l'exploitation du Machine Learning pour le TSA. Les études portent sur l'analyse de résultats d'Imagerie par Résonance Magnétique, ou d'éléments issus du calcul de score SRS [Zhou et al., 2014, Duda et al., 2016]. D'autres travaux lient Machine Learning et Eye-Tracking, notamment dans le cadre de la dissociation des états de la dynamique du regard, montrant des résultats supérieurs aux catégorisations proposées par les Eye-Trackers [Zemblys, 2016]. La reconnaissance d'émotions ou la réponse à des questionnaires comportementaux sont également exploitées par des procédés de Machine Learning. Enfin, des travaux mêlent l'Eye-Tracking et le Machine Learning pour différencier les participants TSA et non-TSA. Il s'agit notamment d'identifier l'autisme à partir d'un patern visuel particulier porté

à certaines zones de l'écran [Wan et al., 2018].

Nous avons porté nos efforts sur deux jeux de données provenant d'une équipe de chercheurs en PC partenaire de nos travaux. Chacun de ces jeux de données est issu d'une captation par Eye-Tracking. Lors d'une captation, l'enfant regarde une vidéo présentant des éléments visant à stimuler certaines réactions, et permet de dissocier enfants TSA et non-TSA. Le premier jeu de données comprend la position du regard, les dimensions de la pupille et la position du participant relativement à l'écran, en fonction du temps d'enregistrement, il est noté *RawData*. Le second jeu de données comprend l'ordre des événements du mouvement oculaire du participant, ainsi que des paramètres propres à ces événements (amplitude d'une saccade ou position d'une fixation par exemple), ce jeu de données est noté *EventData*. Les RawData et les EventData proviennent d'un ensemble d'enfants différent. Les RawData proviennent de 59 participants, des enfants de 8 ans de moyenne d'âge dont 29 sont des enfants TSA, les autres sont des enfants sans historique autistique. Les EventData proviennent de 22 enfants d'âges situés entre 8 et 10 ans, dont 17 enfants TSA et 15 sans historique autistique.

Contributions de cette thèse

Nous avons commencé nos travaux en exploitant le jeu de données EventData. Nous en avons extrait les événements de saccades et les informations associées. Ces informations comprennent notamment l'amplitude, la vitesse, l'accélération et la durée de la saccade. L'ensemble des saccades d'une captation est réuni en conservant leur ordre temporel. Nous utilisons un modèle de Réseaux de Neurones Récurrent constitué d'unités LSTM. Ce réseau, après entraînement, atteint une haute précision sur le jeu d'entraînement (erreur inférieure à 0.01). Nous utilisons 6 participants supplémentaires, non-inclus dans le jeu d'entraînement, pour validation. Nous observons une validation des classes (TSA ou non-TSA) de 5 des 6 enfants. Nous notons cependant le début d'un sur-apprentissage. Le jeu de données utilisé est très limité et son format ne permet pas d'être suffisamment représentatif pour l'entraînement visé, et par conséquent ne peut pas assurer la généralisation des résultats. Nous avons présenté ces travaux au cours de la conférence HealthyIoT 2017.

Nous avons ensuite considéré le suivi oculaire sous la forme de points de regard consécutifs, plutôt que les événements. Réduisant le suivi à des tranches

de 200 points consécutifs, nous avons extrait exclusivement les informations de position du regard de l'enfant (en abscisse et en ordonnée). Nous en avons calculé des valeurs de dynamique (vitesse, accélération et à-coup), en norme et en projections orthogonales, obtenant 9 valeurs de dynamique pour chaque instant de relevé. La représentation graphique de ces mouvements oculaire consiste à dessiner ces mouvements sur une image. Les points permettent de dessiner le tracé et la dynamique permet l'application d'une couleur à chaque ligne constituant le tracé. Pour une ligne donnée, la couleur du trait (sur les canaux RGB) dépend de la vitesse (rouge), de l'accélération (vert) et de l'à-coup (bleu) correspondant à l'instant de transition que représente la ligne. Par exemple, lors d'un mouvement oculaire entre différents points proches dans une zone (vitesse faible, accélaration et à-coup forts), la ligne tend vers le bleu cyan, quand un mouvement entre des points très éloignés (vitesse et accélération élevés, à-coup faible) induisent une ligne tendant vers le jaune. Un total de 547 images (640 par 480 pixels) a été généré avec les RawData à notre disposition, avec 59 participants. Ce jeu de données a fait l'objet d'une publication et est librement accessible pour comparaison de résultats. Il a été exploité lors d'une compétition nommée HackOver en Inde. Deux développeurs ont exploité ces données pour produire une application mobile de détection du TSA. Ce binôme a atteint la seconde place sur 56 participants, le choix de sujet était libre. Cela montre l'intérêt de notre modèle de données et son utilité comme outil de benchmark. Cependant, il faut remarquer que cette représentation permet d'observer un cheminement oculaire entre deux instants dans le temps, sans pour autant permettre de représenter l'ordre dans lequel ce chemin est parcouru.

Notre jeu d'images a été exploité via un Réseau de Neurones Convolutif à deux dimensions. Ces travaux visent à rechercher l'existence de modèles visuels du mouvement oculaire permettant de distinguer avec pertinence les enfants TSA et non-TSA. Une étude de la courbe ROC donne une aire sous la courbe de 0.71, soit un résultat légèrement meilleur en comparaison avec l'utilisation des données numériques par Réseau de Neurones Convolutif à une dimension. Par ailleurs, la représentation sous forme d'images étant inspirée de la représentation graphique que peuvent observer les experts en PC pour distinguer les enfants TSA, nous avons proposé un jeu d'images restreint à valider à quelques uns de ces experts. Ils obtiennent un taux de précision de 66% soit environ la performance moyenne de notre modèle lors de ces

entraînements. Cependant, sur le jeu de données en question, notre modèle atteint les 85% de précision. Les biais de notre manière de dessiner les images (image contre vidéo, fond noir contre vidéo diffusée) peut avoir influencé les experts dans leurs prises de décision.

Le principal défaut de l'approche précédente est dépendant de la présence trop faible d'informations pertinentes. Le fond noir constitue une très grande majorité de l'image et certaines zones ne sont jamais regardées. Cela implique le calcul de valeurs inutiles au sein de notre réseau de neurones. Nous cherchons donc à réduire le format de l'image pour réduire le besoin en mémoire et en calcul du réseau les exploitant. Les images au format 100 par 100 pixels sont encore réduites en passant en nuances de gris (soit un seul canal), sans pour autant perdre en informations concernant la dynamique. Les 10000 valeurs restantes pour nos images initiales sont enfin réduites à 50 valeurs par application d'une PCA. Ce jeu de données réduit est fourni à un Réseau de Neurones Artificiel, validant son entraînement avec une aire sur la courbe ROC de près de 0.92, ce qui représente une véritable avancée vers notre objectif.

Nous avons enfin choisi de représenter les informations des RawData en utilisant une étude statistique. Nous sélectionnons les 24 variables numériques à notre disposition et les exploitons pour obtenir une matrice de corrélation par captation selon les méthodes de Pearson, Kendall et Spearman[Benesty et al., 2009, Kendall, 1938, Spearman, 1904]. Nous obtenons donc 276 valeurs par méthode, soit 828 au total. Ces 828 valeurs sont calculées pour un total de 896 captations. Chaque participant est représenté entre 2 et 88 fois, en fonction du nombre de captations auxquelles il a participé, avec une moyenne de 15.19 captations par participant. Les données sont fournies à un modèle de Réseau de Neurones Artificiel. Il permet une détection bien plus efficace des enfants TSA, au détriment d'une baisse des performances pour les enfants non-TSA.

Ayant entraîné des modèles permettant des détections du TSA chez des enfants, nous cherchons à regrouper les résultats obtenus pour améliorer la performance globale du modèle. Nous basons cette hybridation sur trois de nos approches précédentes : le traitement de la dynamique oculaire au format numérique, le traitement de la dynamique oculaire au format image et le

traitement des données issues de la corrélation. Une méthode de moyennage pondéré des résultats obtenue pour ces trois approches permet d'atteindre une aire sous la courbe ROC de 0.93, et de 0.95 (excluant un cas aberrant).

Chacune des approches présentées dans ce manuscrit exploite des données issues d'une captation par Eye-Tracking. Les stimuli exploités ne sont pas fixes, mais cherchent tous à déclencher le même ensemble de réactions typiques chez les enfants. Nos modèles sont donc conçus pour fonctionner avec des données captées via le visionnage de toutes sortes de vidéos, tant que les stimuli visent à déclencher effectivement les mêmes réactions attendues.

Ce manuscrit décrit nos travaux, en débutant par une présentation du contexte du domaine de recherche, c'est-à-dire le Trouble du Spectre de l'Autisme et l'Eye-Tracking, suivi d'un état de l'art général, concentré sur les croisements entre Machine Learning, Trouble du Spectre de l'Autisme et Eye-Tracking. Ensuite seront présentées les données, élément central de ces travaux. Enfin, les travaux seront présentés en trois parties, chacune liée à une transformation spécifique : les données numériques au cours du temps, les données sous forme d'images et enfin les données statistiques. Un dernier point de discussions et de conclusion permettra de clore ce document.

Chapitre 2

Contexte

2.1 Trouble du Spectre de l'Autisme

Le Trouble du Spectre de l'Autisme (TSA), réduit par abus de langage à l'Autisme, est un trouble neurodéveloppemental touchant environ 1% de la population mondiale. Ce trouble est caractérisé par un décifit d'attention plus ou moins sévère ainsi que par un ensemble de défauts concernant les aptitudes sociales de la personne. En général, une personne souffrant de TSA montre plus de difficultés à maintenir un échange verbal et visuel avec une autre personne, notamment dans le cadre conversationnel. Nous parlons ici uniquement de la situation de TSA chez l'enfant.

L'expression Trouble du *Spectre* de l'Autisme est utilisée parce que ce que le grand public a tendance à nommer *Autisme* est en fait constitué d'une très grande variété de troubles, échelonnés en fonction de plusieurs grands éléments de diagnostic, répertoriés chez l'enfant dans le score CARS (Childhood Autism Rating Scale) [Schopler et al., 1980]. Ces éléments incluent la relation sociale, l'imitation, la réponse émotionnelle, l'utilisation du corps, l'utilisation des objets, l'adaptation au changement, les réponses visuelles, les réponses auditives, le goût-odorat-toucher, la peur-anxiété, la communication verbale, la communication non-verbale, le niveau d'activité, le niveau intellectuel et l'homogénéité du fonctionnement intellectuel et l'impression générale. Pour chacune de ces catégories, l'enfant correspond à un critère, de typique à sévère, on dit alors que l'enfant présente, ou non, certains traits autistiques. Toutes ces réponses permettent de poser un diagnostic concernant

l'enfant et sa place sur le Spectre, résumé à trois grandes classes de gravité du trouble : "Léger", "Moyen" et "Sévère".

Selon le résultat obtenu, l'enfant peut alors être suivi pour lui apporter l'aide nécessaire, ainsi qu'à son entourage, pour permettre d'adapter l'environnement à l'enfant et inversement. Ces diagnostics sont posés généralement entre 3 et 5 ans, âge déjà avancé pour que l'aide au développement de l'enfant soit véritablement efficace. Par ailleurs, au vu de la quantité de points à observer chez l'enfant, poser un diagnostic correct demande un investissement en temps assez important, sachant qu'il faut dissocier ce trouble d'autres troubles neurodéveloppementaux tels que l'hyperactivité ou d'autres troubles appartenant aux Troubles Envahissants du Développement (TED). Ces autres troubles n'étant pas la cible de nos travaux, seuls les enfants TSA et les enfants sans troubles neurodéveloppementaux ont été pris en compte.

Le TSA est un trouble qui génère une situation de handicap, en raison du déficit des facultés sociales notamment. Le TSA peut aussi parfois être la source de capacités supérieures à celles des enfants typiques, c'est-à-dire des enfants ne présentant aucun historique autistique. Ces facilités concernent les arts et la science notamment, et une faculté de mémorisation exceptionnelle. Le cas de l'autisme Asperger est représentatif de ces capacités.

Des outils ont été développés afin d'aider les personnes atteintes et leur entourage. [Daniels et al., 2018] a notamment conçu un outil, basé sur les Google Glass et l'OS Android, visant à aider les jeunes enfants TSA à améliorer leurs interactions sociales ainsi que leur capacité à reconnaître les émotions. A la fin de l'étude, 6 des 14 enfants se trouvent dans une catégorie de TSA moins sévère, tous ont un résultat du score SRS-2 (Social Responsiveness Scale - Second Edition [Bruni, 2014]) réduit. Pour des études plus complètes des méthodes d'aide au développement à destination d'enfants TSA, voir [Schreibman et al., 2015] et [Ingersoll and Schreibman, 2006].

[Vargas-Cuentas et al., 2016] a analysé la capacité de concentration d'enfants TSA et d'enfants non-TSA. Des vidéos incluant des animations attrayantes pour les enfants et présentant des situations sociales et abstraites sont diffusées aux enfants pendant 5 minutes, 1 minute par vidéo. La donnée reçue concernant chaque enfant est réduite à la quantité de vidéos visionnées

sans la moindre distraction. L'analyse des données sous une simple forme d'histogramme permet de montrer de fortes disparités, avec une plus faible capacité de concentration chez les enfants TSA.

Dans le cadre du diagnostic du Trouble du Spectre de l'Autisme, le niveau d'attention du participant et la capacité d'échanges et de contacts sociaux font partie des éléments considérés. Pour constater ce niveau d'attention, certains protocoles doivent être mis en place. En particulier, l'attention d'un participant concernant une discussion ou une demande de réaction à des instructions semble très pertinente. Afin d'assurer un ensemble de stimuli communs et invariants pour tous les participants, l'utilisation d'une vidéo pré-enregistrée, utilisant l'ensemble des éléments déclencheurs a été retenue.

Dans ce manuscrit, les enfants atteints d'autisme sont dits *enfants TSA*, *enfants atteints de TSA/d'autisme* ou *enfants TS* tandis que les enfants ne présentant par de traits autistiques seront nommés *enfants non-TSA*, *enfants non atteints par le TSA/l'autisme* ou *enfants TC*. Les dénominations TS et TC proviennent de la classification des données qui nous ont été fournies, limitée à deux caractères. TS provient logiquement de TSA, et TC correspond à "typique chronologique", c'est-à-dire un enfant d'âge équivalent aux enfants TS que nous observons. Les différentes appellations sont utilisées indifféremment pour chaque classe.

2.2 Eye-tracking

L'Eye-Tracking est un type d'outil permettant la mesure de la position du point de regard d'un participant sur un écran. D'autres paramètres sont également enregistrés, tels que la position des yeux par rapport à l'écran, la taille de la pupille, entre autres. Certaines données peuvent également être identifiées par l'Eye-Tracker, notamment la caractérisation du mouvement oculaire, le séparant entre Fixations, Saccades, Blinks et Oscillations Post-Saccade (PSO) :

- Une Fixation est un ensemble de points de relevé successifs concentrés dans une zone de taille limitée de l'écran. Il s'agit d'un état stable du regard.
- Une Saccade est un mouvement rapide entre deux zones éloignées de l'écran. Il s'agit généralement d'un état de transition entre deux points

FIGURE 2.1 – Un Eye-Tracker de marque SMI RED au centre, son placement sur une machine portable à gauche et sur un écran à droite [GmbH, 2014]

d'intérêt.
— Un Blink est une absence de détection de l'œil. Il peut s'agir d'un clignement, d'un œil hors de portée de l'Eye-Tracker, ou d'une erreur de captation.
— Une PSO est l'état de stabilisation du focus oculaire. C'est la transition suivant une Saccade et précédant une Fixation.

Pour chacune de ces classes de mouvement oculaires, différentes données sont calculées, par exemple pour une Saccade, l'amplitude, la vitesse ou la durée de la Saccade peuvent compléter l'information obtenue. Dans le cadre des travaux de cette thèse, l'Eye-Tracker utilisé est le SMI RED mobile IView XTM RED (voir Fig. 2.1), limité à une fréquence de captation relativement faible (60 Hz contre certains appareils haut-de-gamme pouvant atteindre les 1000 Hz). Cet appareil permet la mesure de toutes les informations citées précédemment dans ce paragraphe, à l'exception des PSO, pour lesquelles la fréquence est trop faible.

Contrairement à d'autres approches (génétique, IRM ou EEG), l'Eye-Tracking pose l'avantage d'être moins contraignant et surtout complètement non-invasif. En effet, l'utilisation d'un tel appareil repose sur l'exploitation de rayons infrarouges et peut ne durer que quelques minutes.

A partir de ce genre d'appareils, des jeux de données peuvent être générés. Ces jeux sont dépendants de l'ensemble des stimuli (des événements visant à déclencher une réaction spécifique du participant) utilisés lors de la captation, des participants et des conditions de captation, ce qui rend la normalisation de travaux différents et la reproductibilité de la captation particulièrement difficile à assurer. Pour cette raison, certains travaux se concentrent sur la

mise à disposition de jeux de données générés afin de pouvoir travailler sur des données communes. Par exemple, [Fang et al., 2014] se focalise sur l'établissement d'un jeu de données basé sur des vidéos stéréoscopiques en haute définition. Ces données sont constituées à partir des captations de 44 participants, sans trouble notable, sur 12 vidéos.

[Marighetto et al., 2017] s'oriente vers les variations du trajet oculaire en fonction de la présence, ou de l'absence, d'une piste audio, mais aussi en fonction du sujet de la vidéo, qu'il soit un paysage, des objets ou des personnes. Ces travaux ont également montré que le mouvement a une plus grande prégnance que l'audio. Aussi, dans le cas de paysages, l'audio a peu ou pas d'effet sur la concentration du sujet. Ces travaux proposent, en plus de la base de données de suivis oculaires créée, une boîte à outils permettant d'analyser l'importance des stimuli visuels et sonores dans la conception de scénarios vidéos, en se basant sur les tracés oculaires obtenus.

Par ailleurs, la précision de la reconnaissance des saccades a été étudiée. C'est le cas de [Salvucci and Goldberg, 2000], qui présente une vue d'ensemble des différentes méthodes utilisées pour la recherche de fixations dans un parcours oculaire, par seuils de dynamique oculaire et par définition de zones d'intérêt.

Chapitre 3

Etat de l'art

3.1 Réseaux de Neurones

3.1.1 Réseaux de Neurones Artificiels

Les Réseaux de Neurones Artificiels prennent exemple sur un modèle biologique : le cerveau et ses neurones, synapses et axones. Cette "copie" du monde biologique reste très simplifiée et se concentre sur le principe d'actions et de réactions propre au fonctionnement du cerveau. La Figure 3.1 présente les analogies entre les deux modèles de neurones utilisés.

Très schématiquement, leurs fonctionnements sont identiques. Dans le détail, le neurone (ou le corps cellulaire) est le point névralgique du modèle, regroupant les informations d'un certain nombre de neurones antérieurs. Ces informations passent à travers des poids (ou des synapses) et le neurone raffine ces informations en une seule sortie qui parcourt un nouveau poids (ou axone). Ces neurones peuvent s'organiser en couches, et transforment ainsi l'information un peu plus à chaque couche, selon la réaction attendue. La comparaison s'arrête cependant à ces points.

Chaque couche de neurones est complétée par un neurone de biais, qui a un effet comparable à la partie constante d'une équation linéaire, permettant d'éviter de produire une valeur nulle si toutes les variables sont nulles, et le calcul de l'entrée d'un neurone est une équation linéaire (voir 3.1.1). Aussi, la fonction de transfert peut varier, et en raison de la méthode de

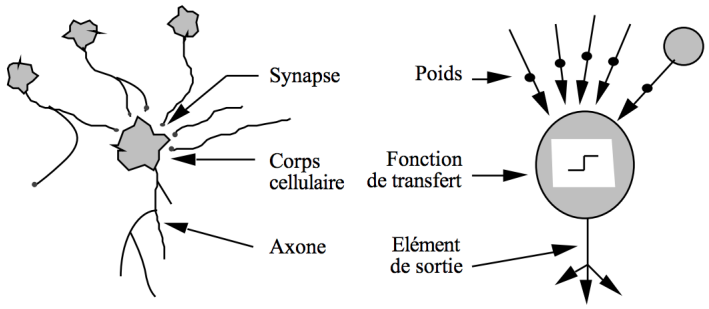


FIGURE 3.1 – Comparaison entre le neurone biologique et le neurone artificiel. [Chraibi Kaadoud and Vieville, 2017]

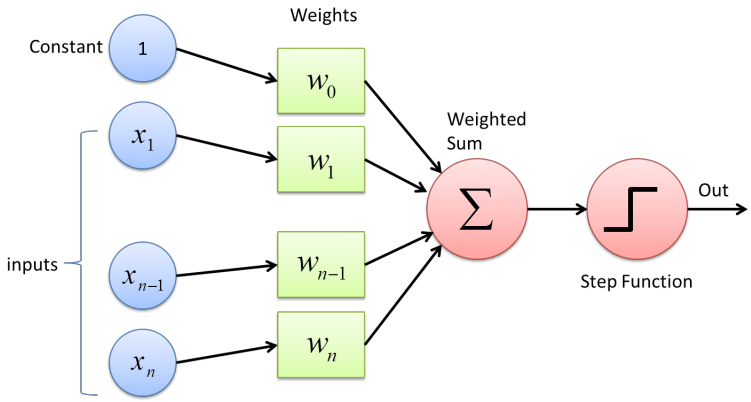


FIGURE 3.2 – Schéma représentant un perceptron [Sharma, 2018].

rétro-propagation [LeCun et al., 1988], qui permet de propager les erreurs du modèle pour corriger les poids qui le constituent, il sera préféré des fonctions facilement calculables et dérivables.

Architecture d'un réseau de neurones

Les débuts des réseaux de neurones prennent place en 1958 quand [Rosenblatt, 1958] propose le modèle du perceptron, directement inspiré du neurone formel de [McCulloch and Pitts, 1943]. Prenant en entrée une liste de valeurs booléennes, il produit une sortie unique, booléenne également. Un neurone logique à deux entrées peut ainsi être comparé à une porte logique. Tous

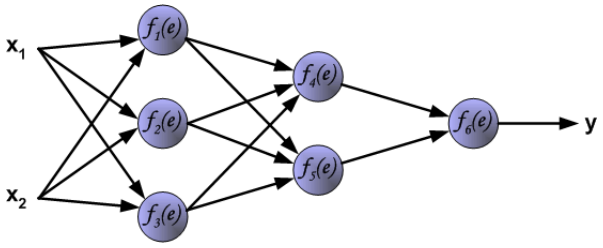


FIGURE 3.3 – Perceptron multicouche [Bernacki, 2004].

les modèles montrés dans cette Section sont des évolutions de ce modèle du neurone formel initial.

Ce modèle peut être comparé aux ALN (Réseaux de Neurones Logiques) qui organisent des portes logiques (donc équivalentes à des neurones formels à deux entrées) en couches. Chaque porte prend en entrée deux valeurs booléennes et propose comme réponse une valeur elle-aussi booléenne. Le modèle peut être représenté comme un arbre binaire dont les nœuds sont des portes logiques dont la valeur dépend de branches associées, les feuilles sont les valeurs binaires fournies au réseau et la valeur de la racine est la réponse du réseau. L'entraînement s'effectue en changeant les types de portes représentées dans l'arbre.

Le perceptron, quant à lui, peut être généralisé à des données numériques. Son fonctionnement est relativement simple. Les entrées sont fournies au perceptron, ces entrées sont pondérées et ensuite additionnées. La sortie dépend du signe de cette somme, 0 si la somme est négative ou nulle, 1 sinon. Le calcul de cette sortie, notée o, peut être résumé par la formule suivante :

$$o = \begin{cases} 1 & si \quad \sum_{i=1}^{n} w_i x_i > \theta \\ 0 & sinon \end{cases}$$

avec n le nombre d'entrées du perceptron, et pour toute entrée i, x_i sa valeur et w_i son poids. θ est un seuil d'activation, souvent fixé à 0.

Ce cas concerne un perceptron ne comptant qu'un neurone (voir Fig 3.2). Ce cas peut être généralisé à un nombre de neurones quelconque.

Initialement, en raison de limitations matérielles et de besoins en puissance de calcul importants, les réseaux de neurones étaient restreints à des modèles très simples, souvent réduits à quelques couches incluant peu de neurones. Le modèle le plus répandu est alors le perceptron multi-couches (MLP, voir Fig 3.3), dans lequel plusieurs perceptrons se succèdent. Le MLP constitue la base de la grande majorité des modèles de réseaux de neurones actuellement utilisés et développés.

Dans le cas général des réseaux de neurones artificiels, la quantité de sorties du modèle n'est pas limitée et leurs formats peuvent varier. Les modèles peuvent servir à la régression, à la classification binaire ou à la catégorisation. La régression vise à approximer une ou plusieurs valeurs numériques, comme pour la prédiction de températures dépendamment de l'ensoleillement, du taux de CO2 et de la vitesse des courants aériens. La classification binaire cherche à donner une réponse sous la forme Vrai/Faux, pour répondre à une question telle que "Cette image représente-t-elle un chat ?". La catégorisation a pour but de trouver la classe d'une entrée parmi un ensemble de classes possibles, il s'agit d'une généralisation de la classification binaire. La question pourrait devenir "Cette image représente-t-elle un chat, un chien ou un oiseau ?". Pour ces différents modèles, les fonctions d'activation peuvent varier, en particulier concernant la couche de sortie.

Fonctions d'activation

Tout neurone d'un réseau de neurones dispose d'une fonction d'activation. Cette fonction peut être comparée au point d'activation du neurone biologique. Le neurone biologique ne répond que si son potentiel est atteint. La fonction d'activation à base de seuil, vue lors de la présentation du perceptron, en est la transposition immédiate. La fonction s'applique à la sortie du neurone, après la somme des entrées pondérées. Elle altère la sortie du modèle, permettant par exemple de conserver les valeurs parcourant le réseau entre les bornes de fonctions d'activation choisies.

Les fonctions d'activation, qui remplacent dans la majorité des cas la fonction de seuil, doivent remplir, en général, certains critères. Avant toute chose, la fonction d'activation doit être définie sur l'ensemble des nombres réels, ou à défaut, sur l'ensemble des valeurs attendues dans le modèle utilisé.

Parmi l'ensemble des fonctions respectant ces critères, quelques unes sont couramment utilisées dans la littérature. Il y a la fonction de seuil, au final peu utilisée, tout comme la fonction identité. La sigmoïde et la tangente hyperbolique (tanH) [Hristev, 1998], ou bien la fonction Unité de Rectification Linéaire (ReLU) [Agarap, 2018] sont, elles, les plus exploitées. Les formules respectives de ces fonctions sont les suivantes :

$$f_{sigmoïde}(x) = \frac{1}{1 + e^{-x}}$$

$$f_{tanH}(x) = \frac{2}{1 + e^{-2x}} - 1$$

$$f_{ReLU}(x) = \begin{cases} 0 & si \quad x < 0 \\ x & sinon \end{cases}$$

La sigmoïde présente l'avantage de donner comme résultat une valeur comprise dans $(0, 1)$, utile pour l'estimation d'une probabilité ou d'une valeur comprise entre deux bornes définies (qui peuvent être rapportées à cet ensemble). La tangente hyperbolique donne un résultat compris dans $(-1, 1)$ et est facilement interchangeable avec la sigmoïde. Ces deux fonctions, proposant des résultats dont les valeurs sont strictement bornées, permettent d'assurer une certaine stabilité du gradient au cours de l'apprentissage. La fonction ReLU donne des résultats compris entre 0 et $+\infty$, dont l'étendue sans borne supérieure autorise des variations bien plus grandes du gradient et donc un apprentissage plus rapide, cette fonction d'activation est notamment préférée dans le cas de l'apprentissage profond (Deep Learning), des réseaux de neurones comprenant une quantité importante de couches, pour lesquelles la vitesse d'apprentissage est cruciale en raison du nombre de poids à optimiser.

Une dernière fonction d'activation, un peu particulière, est utilisée dans le cas de problèmes de catégorisation, en raison de sa capacité à normaliser les sorties, permettant de les traiter comme des probabilités de réponse. Cette fonction, nommée softmax (ou exponentielle normalisée), permet de transformer l'ensemble des sorties de telle sorte que chaque valeur soit incluse dans l'ensemble $[0, 1]$ et que la somme de ces sorties soit égale à 1. Pour chacune des sorties, l'exponentielle de la valeur de sortie est divisée par la somme des

exponentielles de toutes les sorties. La formule suivante permet de calculer la valeur de sortie $\hat{O}_i$ d'un neurone i :

$$\hat{O}_i = \frac{e^{c_i}}{\sum\limits_{j \in S} e^{c_j}}$$

avec S l'ensemble des neurones de sortie du modèle, et c_j la valeur calculée par le neurone j avant application d'une fonction d'activation.

Apprentissage d'un réseau de neurones

Initialement, les poids d'un réseau de neurones artificiels sont attribués aléatoirement. Cet aléatoire est en principe centré sur 0. De fait, le modèle produit alors des valeurs elles-aussi aléatoires.

Le réseau subit une phase d'entraînement qui vise à réduire l'erreur de sortie du modèle. Une entrée est l'ensemble des informations fournies au réseau pour un cas d'application, par exemple le taux d'ensoleillement, le taux de CO2 et la vitesse du vent en km/h. Pour toute entrée, le réseau produit une sortie, le résultat du modèle, par exemple, la température extérieure en degrés Celsius. Entrée et sortie du modèle sont donc fortement liés et ce lien représente le problème que le réseau doit apprendre. Le fait de fournir une entrée au réseau et d'en obtenir une sortie est nommé la propagation en avant du signal. Le signal (l'entrée) est altéré par les poids du réseau et les fonctions d'activations des neurones jusqu'à atteindre la couche de sortie.

Pour effectuer l'apprentissage, une entrée I est fournie au réseau, le modèle fournit alors une réponse à sa sortie, que l'on notera $\hat{O}_I$, la sortie est comparée à la sortie attendue, notée O_I. Ainsi, pour un neurone i de la couche de sortie S, l'erreur $\delta_{I,i}$ de i est calculée comme suit.

$$\delta_{I,i} = O_{I,i} - \hat{O}_{I,i}$$

avec $O_{I,i}$ la sortie attendue au neurone i et $\hat{O}_{I,i}$ la sortie obtenue au neurone i.

Le but est de trouver la tendance de l'erreur afin d'orienter vers son minimum, nous posons donc la fonction dérivée f' de la fonction d'activation f du neurone i. Ainsi, nous posons $\delta'_{I,i}$ tel que :

$$\delta'_{I,i} = f'(\hat{O}_{I,i}) - \delta_{I,i}$$

Ces erreurs sont propagées à la couche P précédant la couche de sortie S (donc dans le sens inverse de la propagation en avant), pondérées par les poids du réseau. Chaque neurone j de P reçoit $\delta'_{I,i}$ pondéré par le poids liant les neurones i et j, pour tout neurone i de S, puis somme les valeurs pondérées reçues. Cette somme devient l'erreur du neurone j. Le procédé de dérivation de l'erreur est répété sur P puis les nouveaux δ' sont transmis à la couche précédent P. Cela est répété jusqu'à avoir parcouru l'intégralité du réseau.

Il s'agit de la rétropropagation (ou backpropagation). Elle est appliquée pour chaque entrée I fournie au modèle lors de l'entraînement jusqu'à atteindre une erreur minimale. Puisque chaque itération demande le calcul d'autant de dérivées des fonctions d'activation que le réseau comprend de neurones, le fait d'utiliser des fonctions d'activation facilement dérivables permet de réduire fortement le temps de calcul nécessaire.

L'ensemble des erreurs est utilisé pour corriger les poids qui leurs sont liés dans la couche précédente (voir Figure 3.4). En considérant deux neurones i et j successifs, nous pouvons résumer cette correction à l'équation suivante :

$$w'_{i,j} = w_{i,j} - \lambda * \delta_j * o_j$$

avec $w_{i,j}$ et $w'_{i,j}$ le poids entre les neurones i et j respectivement avant et après correction, δ_j l'erreur calculée en j, o_j la sortie initialement obtenue en j et λ, une constante, représentant le taux d'apprentissage.

Ce taux d'apprentissage est essentiel dans la capacité d'un réseau à convenablement apprendre le modèle de données qu'il doit approximer. Il est en général compris entre 0 et 1, un taux approchant 1 permettant une forte variation des poids et un taux plus faible réduisant cette variation. La variation de ces poids, si elle est trop faible peut risquer de faire converger le

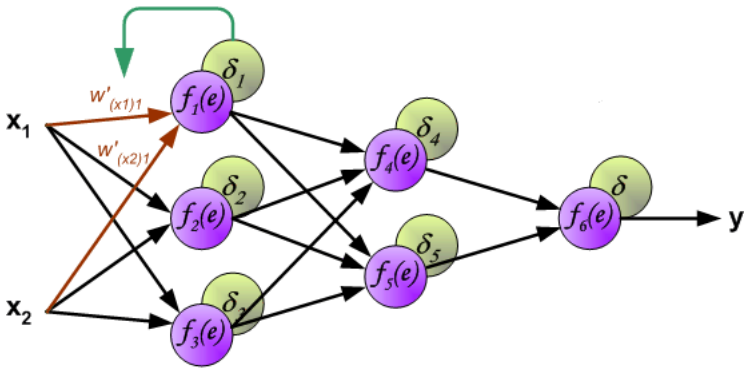


FIGURE 3.4 – Correction des poids. f_i est la fonction d'activation du neurone i et δ_i l'erreur pondérée sur ce neurone [Bernacki, 2004].

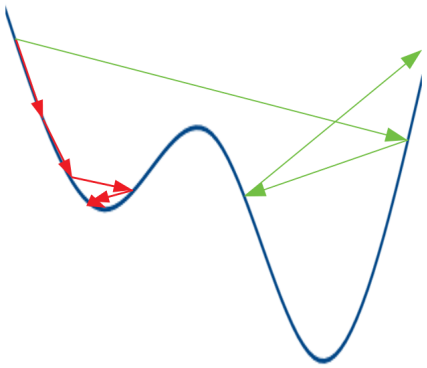


FIGURE 3.5 – Exemples de cas de taux d'apprentissages inadaptés. Les tracés rouge et vert correspondent respectivement à un taux d'apprentissage trop faible et un taux d'apprentissage trop élevé.

modèle vers un minimum local. Un minimum local est un état de l'apprentissage pour lequel la tendance de l'erreur tend vers 0 sans être pour autant la solution optimale du problème (il s'agirait alors d'un minimum global), La faible valeur du taux ne permettrait pas de sortir de celui-ci. Un taux faible peut aussi rendre l'apprentissage très lent. A l'inverse, quand le taux est trop élevé, il peut empêcher d'atteindre la solution souhaitée. La Fig 3.5 présente de manière très simplifiée ces problèmes de taux d'apprentissages inadaptés. L'apprentissage utilisant un taux faible tombe dans un minimum local du problème (creux de la courbe à gauche) tandis que le taux trop élevé manque le minimum global du problème, c'est-à-dire le point le plus bas de la courbe (ici, le creux de la courbe à droite).

Optimiseurs d'apprentissage et généralisation du calcul d'erreur

Au fur et à mesure que l'apprentissage avance, le besoin de modifier plus précisément et subtilement les poids du modèle augmente. Il existe des optimiseurs permettant de faire varier le taux d'apprentissage notamment, selon l'état du modèle actuel. La plupart de ces optimiseurs prennent en compte la tendance de variation des erreurs, nommée le momentum, et l'utilisent pour accélérer l'entraînement quand l'apprentissage converge tout en ignorant certains minimum locaux, comme le ferait un ballon sur une pente descendante dont la vitesse pourrait empêcher un arrêt sur une courte et faible pente ascendante. On parle ici d'opérateurs à inertie. Parmi ces optimiseurs, nous trouvons notamment RMSprop [Hinton et al., 2012], Adagrad [Duchi et al., 2011] et Adadelta [Zeiler, 2012], qui sont les optimiseurs les plus couramment utilisés.

Adagrad traite le réseau comme un ensemble. Ainsi, chaque poids du modèle dispose de son propre taux d'apprentissage et momentum, variant en fonction de l'élément que ces poids représentent. Cela permet de réduire rapidement la variation des poids dont la valeur est optimale, au risque de tendre vers un modèle figé (taux d'apprentissages de tous les poids très proches de 0) trop tôt dans certains cas. RMSProp et Adadelta sont des améliorations d'Adagrad pour lesquelles le momentum n'est calculé que sur une fenêtre limitée d'itérations de l'apprentissage.

Chacun de ces optimiseurs fonctionne en prenant en compte la valeur de l'erreur. Dans le cas général, l'erreur calculée n'est pas prise en compte pour

un seul jeu d'entrées, mais plutôt pour un lot de jeux d'entrées, voire l'intégralité du jeu d'entraînement. Afin de regrouper l'ensemble de ces erreurs, plusieurs méthodes sont utilisées, selon l'objectif d'entraînement fixé.

Cependant, il faut noter que l'objectif de l'entraînement d'un réseau de neurones n'est pas d'être parfaitement précis sur les données d'entraînement. Un modèle de réseau de neurones doit approximer la réponse à un problème, afin de pouvoir généraliser avec de nouvelles données, indépendantes du modèle initial. Or, dans le cadre d'un apprentissage, il est possible que le modèle devienne trop proche des données fournies pour l'entraînement et ne puisse répondre convenablement quand des données inconnues lui sont fournies. Nous pouvons comparer cela à l'approximation d'une fonction sinusoïde par une fonction carrée, en phase avec la sinusoïde. La fonction carrée approxime parfaitement sur les points d'amplitude maximale de la sinusoïde, sur ces points, les fonctions sont égales. Pour tous les autres points, c'est à dire la généralisation de la sinusoïde, les fonctions divergent. Dans le cas d'un apprentissage, nous parlons du sur-apprentissage. De ce fait, hors l'évitement de minima locaux évidents, gérés par les optimiseurs, parler de recherche du minimum global a très peu de sens, dans la grande majorité des cas.

Dans le cas d'un modèle visant à estimer des valeurs numériques (prédiction notamment), il est généralement fait usage de l'erreur quadratique (Mean Squared Error ou MSE). Elle vise à donner une estimation de la différence moyenne entre les valeurs obtenues et attendues, comparées deux à deux. Pour calculer l'erreur quadratique δ_i d'un neurone i, la formule prend la forme suivante :

$$\delta_{MSE_i} = \frac{1}{n}\sum_{j=1}^{n}(O_j - \hat{O}_j)^2$$

avec n le nombre de sorties du neurone i comparées, et O et $\hat{O}$, respectivement les ensembles des sorties attendues et obtenues pour le neurone i.

Bien que cette méthode puisse sembler efficace dans toutes les situations, dans les cas de classification, d'autres méthodes lui sont préférées.

En particulier, quand le problème demande à extraire une réponse juste parmi un ensemble de réponses possibles, il s'agit d'un problème de classi-

fication. Dans ces cas, la méthode utilisée pour le calcul de l'erreur privilégiée est l'entropie croisée (cross-entropie [De Boer et al., 2005]). Dans le cas d'un choix parmi deux réponses, on parlera d'entropie croisée binaire (binary cross-entropy). Dans le cas d'un choix parmi plus de deux réponses, il s'agira d'entropie croisée catégorique (categorical cross-entropy). La formule de calcul de cette entropie δ_i d'un neurone de sortie du réseau i est la moyenne des valeurs calculées de la manière suivante, avec O_i la sortie attendue et $\hat{O}_i$ la sortie obtenue :

$$\begin{cases} -log(\hat{O}_i) & si \quad O_i = 1 \\ -log(1-\hat{O}_i) & si \quad O_i = 0 \end{cases}$$

ce qui peut alors être résumé à :

$$\delta_i = -\frac{1}{n}\sum_{j=1}^{n} O_{i,j} * log(\hat{O}_{i,j}) + (1 - O_{i,j}) * log(1 - \hat{O}_{i,j})$$

avec n le nombre de sorties du neurone i comparées et $O_{i,j}$ et $\hat{O}_{i,j}$ respectivement les sorties attendues et obtenues du neurone i pour le $j$ième jeu d'entrées.

Ce calcul est répété pour chaque neurone de sortie.

Validation et utilisation d'un modèle

La progression de l'apprentissage est mesurée selon la qualité de prédiction du modèle. En général, le calcul de la moyenne des erreurs que le modèle produit est suffisant, une erreur moyenne de 0 correspondant à un modèle parfait. Le calcul de son complément à 1 permet d'obtenir sa précision :

$$précision = 1 - erreur\_moyenne$$

Dans les cas de catégorisation, seule une réponse est valide. Il est alors possible de calculer la précision en comptant le taux de réponses correcte du modèle. La classe de sortie du modèle la plus proche de 1 correspond à la réponse du modèle. La précision est alors calculée comme suit, avec *réponses* le nombre de réponses vérifiées et *bonnes_réponses* le nombre de réponses correctes :

$$précision = \frac{bonnes\_réponses}{réponses}$$

Le niveau de précision du modèle induit directement la qualité de réponse du modèle sur de nouvelles données. Pour cette raison, le calcul de précision est généralement effectué en utilisant un jeu de données distinct du jeu de données exploité pour l'entraînement du modèle.

Afin de vérifier la validité d'un modèle dans le cas général, ces nouvelles données dites de validation sont souvent issues d'un jeu initial duquel les données d'entraînement proviennent également. Une séparation du jeu initial est effectuée avant l'entraînement du modèle. Cependant, cette séparation, selon les données conservées pour l'entraînement, peut permettre un entraînement du modèle plus ou moins efficace. En particulier, les petits jeux de données ou ceux pour lesquels certains cas sont très peu représentés induisent une forte dépendance à la séparation du jeu de données initiale pour l'apprentissage efficace du modèle.

Pour ces petits jeu de données, il est possible d'utiliser le procédé de *cross-validation* [Arlot et al., 2010] (ou validation croisée), qui consiste à initialement séparer le modèle en n jeux de données de taille équivalente. Ainsi, n modèles sont entraînés, avec pour chaque modèle un jeu de validation différent issu de la séparation. Les modèles sont entraînés en utilisant l'ensemble du jeu de données initial, excluant le jeu de validation. Ainsi, l'efficacité de l'entraînement de tout cet ensemble de modèles est dépendant de l'efficacité de l'entraînement des n modèles, via la moyenne des précisions de validation de tous les modèles obtenus.

Pour certains cas, où la quantité de données est limitée, le *leave-one-out*, également inclus dans [Arlot et al., 2010], peut être utilisé. Ce procédé représente une exploitation à l'extrême de la cross-validation. Pour le leave-one-out, n vaut la taille du jeu d'entraînement initial. Le procédé d'entraînement est ensuite strictement identique.

3.1.2 Réseaux de Neurones Récurrents

Les réseaux de neurones artificiels présentés précédemment (voir Section 3.1.1) sont très efficaces pour l'obtention d'une estimation d'une valeur ou d'une classe selon une entrée fournie, mais sont mal adaptés à l'utilisation de données temporelles.

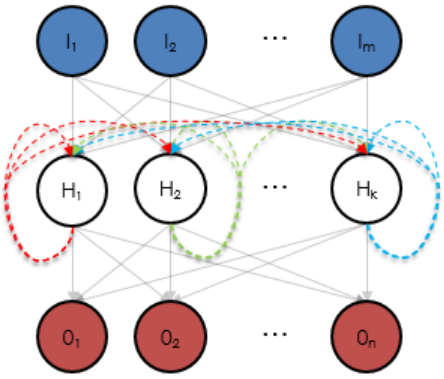


FIGURE 3.6 – Représentation des différences entre un RNN et un ANN. Les connexions récurrentes du modèle sont en pointillés.

Les données temporelles ou séquentielles sont des données présentant un ordre. Par exemple, les différentes positions successives sur une carte lors d'un déplacement en voiture constituent des données temporelles. Une seule de ces positions ne permet pas de décrire fidèlement le trajet effectué et seul l'ensemble des positions ordonnées peut le permettre.

Réseau de Neurones Récurrents

Pour résoudre ces problématiques, les réseaux de neurones récurrents (Recurrent Neural Network ou RNN) sont nécessaires. Ces réseaux de neurones reprennent en grande partie le fonctionnement des modèles d'ANN, qu'il s'agisse des fonctions d'activations, du principe entrée/sortie ou de la répartition par couches.

La différence entre un ANN et un RNN consiste en l'existence de poids supplémentaires dans ce dernier. Pour chaque couche intermédiaire du réseau, ces poids permettent une connexion complète des neurones entre eux, comme le montre la Figure 3.6. Quand une séquence d'entrées est fournie au réseau, ces nouvelles connexions permettent de donner aux couches intermédiaires une information concernant leur état précédent. En effet, sur une couche intermédiaire donnée, chaque neurone reçoit les résultats pondérés de la couche précédente, comme pour un ANN, mais aussi les résultats pondérés

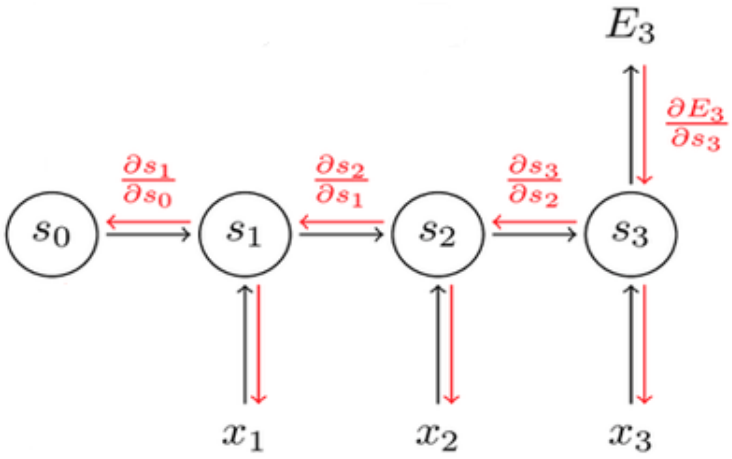


FIGURE 3.7 – Exemple de l'application de la rétro-propagation dans le temps. Les nœuds s_0 (initial) à s_3 sont quatre états successifs d'un modèle induits par les entrées successives de x_1, x_2 et x_3. La connexion reliant deux états successifs représente les poids récurrents du modèle. E_3 est l'erreur de l'état s_3 A l'opposé, les flèches rouges montrent les étapes de rétro-propagation, jusqu'à s_0, permettant la correction des poids, y compris récurrents. [Seo, 2018]

de la couche courante obtenus pour la donnée d'entrée précédente. Cela permet de simuler une mémoire. Ainsi, le modèle est en capacité de se "souvenir" des données lui ayant été fournies précédemment dans la séquence. L'exemple le plus parlant de ce genre d'application consiste en l'apprentissage de l'écriture en ce basant sur une œuvre. Le modèle apprend alors l'enchaînement des caractères, et ainsi la formation de mots, l'orthographe et la ponctuation. [Tanner, 2018] présente cet exemple de manière très didactique.

L'ajout de cette boucle de mémoire induit également une variation quant à la manière d'appliquer le processus de rétro-propagation. En effet, pour chaque instant supplémentaire de la séquence de données, un nouvel état du modèle est créé. Chaque état est lié au précédent par une pondération récurrente, ce qui induit également une erreur pouvant être remontée dans la séquence. La Figure 3.7 montre ces enchaînements. La retro-propagation

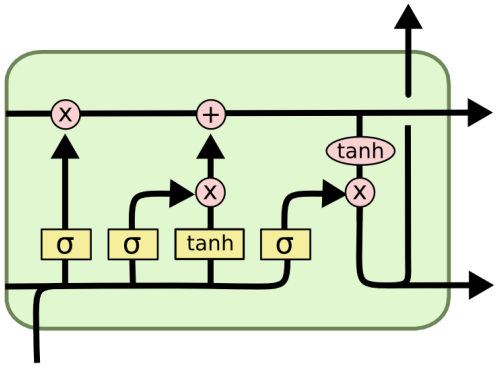


FIGURE 3.8 – Représentation d'une unité LSTM. L'entrée en bas de l'unité correspond aux valeurs d'entrée actuelles de l'unité. Les informations récurrentes (état et mémoire) proviennent des deux entrées supplémentaires, visibles à gauche. Les flèches sortantes à droite correspondent à l'état et la mémoire de sortie récurrente de l'unité, servant d'entrée lors du pas suivant de la séquence. La sortie de l'unité est la flèche en haut du schéma. σ représente la fonction sigmoïde et tanh la tangente hyperbolique. [Olah, 2015]

à travers le temps est ainsi limitée à la longueur de la séquence.

Long Short-Term Memory et Gated Recurrent Unit

Lors de l'apprentissage d'un RNN, les variations du modèle tendent à se stabiliser trop vite, empêchant l'évolution du modèle par la suite de l'entraînement, notamment dans le cas de longues séquences de données. Ce problème a mené à la création des unités LSTM (Long Short-Term Memory) [Gers et al., 1999]. L'unité étend le neurone tel que décrit lors de la présentation des ANN, lui ajoutant des portes affectant les valeurs en cours de calcul. Dans un LSTM (voir Figure 3.8), les portes d'entrée et de sortie permettent d'autoriser ou non l'arrivée d'une nouvelle entrée et la communication de la sortie aux neurones suivants, respectivement. La porte d'oubli, quant à elle, permet la remise à zéro de la mémoire de l'unité. Le GRU (Gated Recurrent Unit [Cho et al., 2014]) est une évolution du LSTM, dont le fonctionnement interne est simplifié. La Figure 3.9 montre la représentation d'une unité GRU, ainsi que les différences entre LSTM et GRU. La réduction d'éléments dans

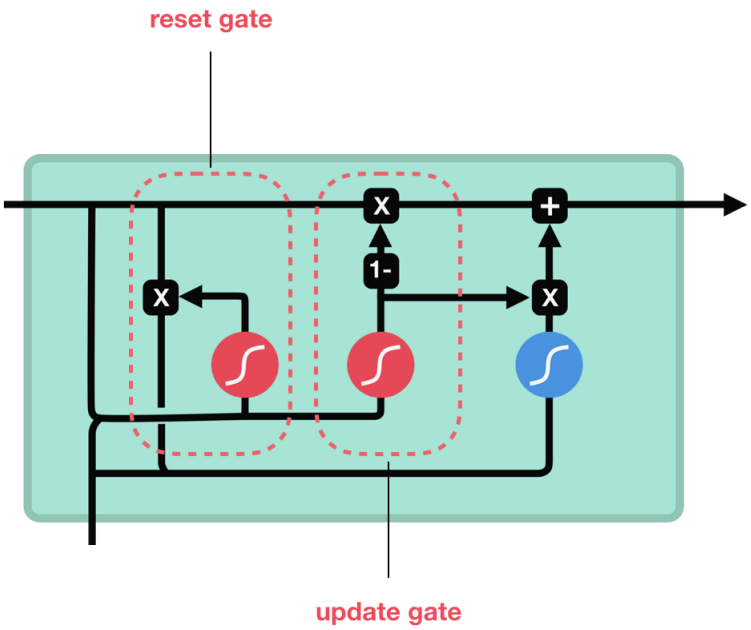


FIGURE 3.9 – Représentation d'une unité GRU. L'état de l'unité est le seul élément conservé pour la récurrence. Les portes sont réduites à la *reset gate* (porte de réinitialisation) qui définit la portée de la mémoire de l'unité et l'*update gate* (porte de mise à jour) qui réunit les effets de la porte d'entrée et d'oubli du LSTM. [Nguyen, 2018]

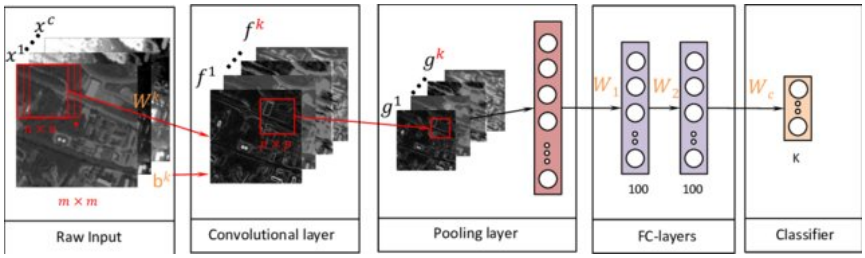


FIGURE 3.10 – Schéma réprésentant le fonctionnement d'un réseau de neurones convolutif [Längkvist et al., 2016].

le GRU induit un fonctionnement plus rapide.

3.1.3 Réseaux de Neurones Convolutifs

L'apprentissage d'un modèle de Machine Learning permet de repérer des schémas dans les données. Dans la plupart des cas, les liens entre les différentes entrées fournies ne sont pas évidents.

Les images présentent un cas particulier d'informations à traiter. Une image (une photographie ou un dessin) offre une information particulière dans le cas de données éparses : la proximité spatiale. De ce fait, bien qu'effectuer un entraînement via un ANN pourrait être viable, chacun des pixels de l'image, sur chacun de ses canaux de couleur, devrait être lié à chaque neurone de la couche suivante, ce qui engendrerait un coût en calcul très important.

Pour réduire les temps de calcul trop importants, les Réseaux de Neurones Convolutifs (CNN ou Convolutional Neural Network) ont été développés. Le modèle est représenté dans la Figure 3.10. Leurs couches principales sont nommées couches de convolution, en référence au produit de convolution. Le produit de convolution entre deux fonctions f et g noté $f * g$ se calcule de la manière suivante :

$$\begin{aligned}(f * g)(x) &= \int_{-\infty}^{\infty} f(x-t)g(t)dt \\ &= \int_{-\infty}^{\infty} f(x)g(t-x)dt\end{aligned}$$

car le produit de convolution est commutatif. Une manière de se représenter mentalement cette opération consiste à glisser la courbe représentant f de

$-\infty$ à $+\infty$ sur l'axe des abscisses et pour chaque instant de ce glissement, calculer l'aire commune entre les aires sous les courbes respectives de f et g. Ce principe de "fenêtre glissante" inspire le fonctionnement des couches de convolution.

L'image est parcourue par une fenêtre de taille prédéfinie, se déplaçant en général d'un pixel par étape. A chaque étape de ce parcours, chaque pixel de la zone couverte par la fenêtre est pondéré pour servir d'entrée à un pixel de la couche suivante. Comme pour un ANN, ce pixel dispose d'une fonction d'activation. Ce procédé est appliqué sur les différents canaux de l'image, ainsi la fenêtre définit une zone de l'image sur les trois canaux (rouge, vert et bleu) de l'image. Plusieurs résultats de la couche suivante sont ainsi produits, nous parlons de $filtres$. Cela peut être généralisé à tout autre nombre de canaux. De fait, la nouvelle image condensée, représentant la superposition des différents filtres, est constituée d'autant de canaux qu'elle comprend de filtres. Elle peut donc elle aussi se voir appliquer le procédé de convolution.

Cependant, le principe de la fenêtre glissante introduit une redondance de l'information lors du passage, en effet, si une fenêtre est de taille n par m, dans le pire des cas, un pixel peut être pris en considération $n * m$ fois. De la même manière, une zone de l'image de taille a par b, avec $a < n$ et $b < m$, est utilisée entièrement $(n - a) * (m - b)$ fois. Cette redondance non nécessaire peut être atténuée par l'utilisation d'un type de couche particulier dit de Pooling ou Subsampling. Le pooling consiste en une fenêtre parcourant l'image, dont les positions successives ne se chevauchent pas. Aussi, le résultat fournit par la fenêtre n'est pas pondéré mais est une opération statistique sur les pixels considérés. Cela peut être une moyenne, un minimum ou un maximum notamment. Les travaux présentés exploitant les CNN de ce manuscrit utilisent l'opération *maximum*.

Après un certain nombre de couches de convolution et de pooling, l'image est suffisamment traitée et identifie des éléments caractéristiques du problème posé. L'image résultante sortant de cette dernière couche est alors traitée comme un ensemble de valeurs numériques et sert d'entrée à un modèle de type ANN. L'ANN dispose alors en entrée des informations raffinées depuis l'image initiale (couche de Pooling du schéma en Figure 3.10).

Le modèle traitant les images est propre à l'utilisation de données sur deux dimensions. Il existe également des modèles analogues pour des données numériques temporelles, le réseau de convolution à une dimension, pour lequel la fenêtre parcourt le temps, les différentes variables jouent le rôle des canaux. Le modèle s'étend également à trois dimensions, avec une fenêtre également définie sur trois dimensions, pour représenter des éléments simulés dans l'espace, comme des molécules médicamenteuses par exemple. Le modèle est ainsi généralisable à N dimensions, bien que la plupart des travaux se limitent au plus à 3 dimensions, notamment pour des représentations 3D en Santé ([Kamnitsas et al., 2017] par exemple).

3.2 Prétraitement des données

Les données sont parfois fournies selon un format initialement inutilisable ou peuvent être trop brutes pour que leurs valeurs soient simplement exploitées. Par exemple, il existe des jeux de données d'une taille de l'ordre du téraoctet pour lesquels l'analyse complète des données et leur compréhension serait très fastidieuse et faiblement productive, pour cela les méthodes propres au Big Data sont utilisées (notamment le MapReduce [Dean and Ghemawat, 2004]). Aussi, des données peuvent être fournies peuvent être incomplètes ou redondantes. Dans ces cas, nous appliquons différentes méthodes d'altération de ces données, simples comme l'abandon de variables intuitivement inutiles ou plus avancées comme des analyses statistiques. Dans ce manuscrit, parmi les méthodes statistiques, nous n'exploitons que la corrélation entre variables [Benesty et al., 2009, Kendall, 1938, Spearman, 1904] et l'Analyse en Composantes Principales [Wold et al., 1987].

3.2.1 Matrices de Corrélation

La matrice de corrélation est une représentation permettant de noter la proximité entre les variables d'un jeu de données, deux à deux. Cette proximité est représentée par la corrélation, qui correspond à la similitude des variations de deux variables. Des variables identiques (ou égales modulo un certain nombre de transformations mathématiques) sont un exemple de variables corrélées. Moins il est possible de prévoir le comportement d'une variable selon une autre, moins ces deux variables sont corrélées. Deux variables ayant une tendance similaire ont une valeur absolue de leur corrélation proche

de 1, et plus deux variables sont de tendances distinctes, plus la valeur absolue de leur corrélation tend vers 0. La corrélation entre une variable et elle-même est toujours de 1. Par exemple, pour estimer le rythme cardiaque maximal d'une personne, chacun sait qu'il suffit de calculer $220 - \hat{a}ge$ avec $\hat{a}ge$ l'âge actuel de cette personne. Ainsi, l'âge et le rythme cardiaque maximal (simplifié) sont des valeurs très corrélées. La matrice de corrélation est symétrique, avec pour axe de symétrie la diagonale des variables avec elles-mêmes, l'ordre des labels étant, par convention identique entre lignes et colonnes pour le calcul d'une telle matrice. Cette symétrie est due au fait que la corrélation est commutative. Cette corrélation peut être mesurée selon trois approches : Pearson, Kendall et Spearman.

Corrélation (Pearson) La méthode de Pearson est la plus utilisée. Elle exploite la covariance qui montre les similarités de variation de deux variables. La corrélation en est une forme normalisée. La corrélation entre deux variables vaut le ratio entre la covariance des deux variables et le produit des écarts-types des variables. Cela donne la formule suivante :

$$corr_{Pearson}(X, Y) = \frac{Cov(X, Y)}{\sigma_X \sigma_Y}$$

avec, X et Y deux variables, Cov(X,Y) la covariance entre X et Y et σ_X et σ_Y les écarts-types des variables X et Y, respectivement.

Tau de Kendall Kendall considère la corrélation d'une manière différente. Pour tout couple i et j d'indices des listes de valeurs des variables X et Y, X_i et X_j représentent respectivement la $i^{\text{ème}}$ et la $j^{\text{ème}}$ valeur de la liste de X, similairement, nous obtenons Y_i et Y_j. Le signe de $X_i - X_j$ et celui de $Y_i - Y_j$ sont comparés, et le résultat inclut deux cas : soit le signe est le même dans les deux cas et il s'agit d'un cas positif, soit le signe diffère et le cas est négatif. Chaque cas positif vaut 1, chaque cas négatif vaut -1. La corrélation vaut alors la moyenne des valeurs des cas. En effet, les cas positifs montrent que les variables se comportent de manière similaire, et les cas négatifs montrent une opposition. Des variables complètement corrélées auront les mêmes variations et donc une corrélation de 1 (-1 si la similarité dépend d'un changement de signe). Plus simplement :

$$corr_{Kendall}(X, Y) = \frac{positives - negatives}{n(n-1)/2}$$

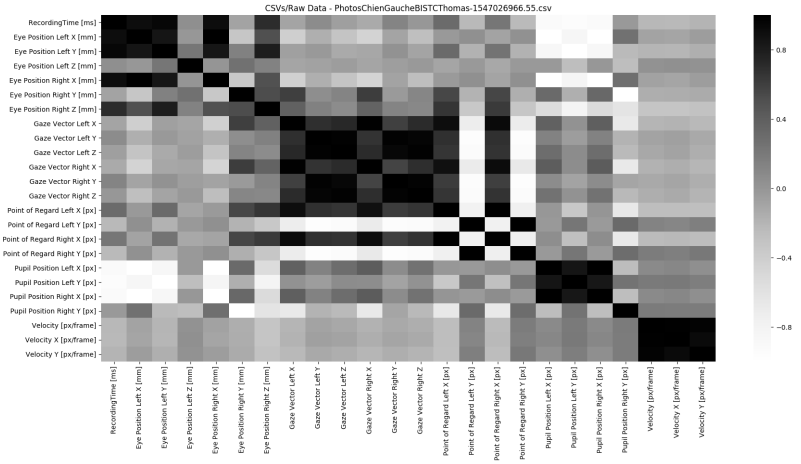


FIGURE 3.11 – Représentation d'une matrice de corrélation. Cette corrélation est établie pour 24 variables liées au relevé Eye-Tracking d'enfants regardant une vidéo. Seule la corrélation de Pearson est représentée. Les cases sont colorées en nuances de gris (de blanc=0 à noir=1).

avec *positives* et *negatives* le nombre de cas positifs et négatifs, respectivement et n le nombre de cas total.

Rho de Spearman La méthode de Spearman propose un autre fonctionnement. Les variables sont transformées pour garder un classement de leurs valeurs. Soit X une variable avec n valeurs. rg_X, sa transformation, est constituée de l'ensemble des entiers de 1 à n inclus, et est ordonnée de telle sorte que $corr_{Kendall}(X, rg_X) = 1$. rg_Y est conçu de la même manière à partir de la variable Y. Le calcul de la corrélation entre X et Y consiste alors en l'application de Pearson à rg_X et rg_Y. Cela équivaut à la formule suivante :

$$corr_{Spearman}(X, Y) = \frac{Cov(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}}$$

ou

$$corr_{Spearman}(X, Y) = corr_{Pearson}(rg_X, rg_Y)$$

La Figure 3.11 montre un exemple de représentation de la matrice de corrélation, limitée à la corrélation de Pearson.

3.2.2 Analyse en Composantes Principales

L'Analyse en Composantes Principales (ou PCA pour Principal Component Analysis) cherche à mettre en valeur les éléments d'un jeu de données les plus pertinents et permet une réduction de la dimension du problème en unissant les variables les plus corrélées entre elles (voir Section 3.2.1 pour le principe de corrélation). Il s'agit d'une méthode de réduction de la dimensionnalité de problèmes.

La simplicité du traitement de l'information dépend fortement de la quantité et de la redondance de l'information à traiter. De nombreux traitements automatisés seront plus efficaces si la quantité de données à traiter par individu est réduite au maximum. Par ailleurs, certaines variables peuvent être fortement corrélées et présentent de ce fait une redondance de l'information inutile. Ainsi, la PCA permet de réduire la quantité de variables, en réduisant la corrélation entre les variables produites.

Le procédé de la PCA commence par une normalisation de chacune des variables, de sorte que leurs moyennes et écart-types soient comparables. Il est alors recherché une variable dont l'axe permettrait une projection qui maximise la variance de l'ensemble des variables, une fois projetées. Cet axe trouvé, la projection de toutes les variables est effectuée selon cet axe. Le nombre total de variables est donc réduit de 1. Ce procédé est répété jusqu'à atteindre le nombre de variables choisi.

Le but recherché de maximiser la variance de l'ensemble de données vise à améliorer la capacité descriptive du jeu de données. Ainsi, pour chaque variable obtenue, l'individu est plus facilement distinguable des autres individus de l'ensemble de données. Le traitement des données est alors simplifié, notamment pour des tâches de classification. En effet, les individus sont représentés par une quantité d'information amoindrie, ce qui limite la capacité de ressources nécessaire à son traitement, mais aussi, chaque variable utilisée propose une pertinence accrue en raison d'une corrélation moindre.

3.3 Mesures statistiques

Dans le cadre des travaux présentés dans ce manuscrit, différentes méthodes statistiques sont exploitées. En particulier, nous devons estimer l'efficacité d'un apprentissage pour un réseau de neurones. Aussi, nous pouvons évaluer de diverses manières la précision d'un modèle de Machine Learning entraîné.

3.3.1 Evaluation de l'entraînement

Au cours de l'entraînement, diverses valeurs sont utilisées pour indiquer la précision et la tendance du modèle. En particulier, les fonctions de perte sont utilisées pour estimer la distance entre la réponse d'un modèle (*réponse*) et la réponse attendue (*attendue*), selon un ensemble de données de base. Les méthodes principalement utilisées sont présentées dans la Section 3.1.1, Partie "Optimiseurs d'apprentissage et généralisation du calcul d'erreur", notamment la MSE, qui sera utilisée comme exemple dans cette partie.

Ces calculs de perte permettent d'estimer l'instant à partir duquel l'apprentissage devient peu utile, mais surtout quand celui-ci tend vers le sur-apprentissage. Au cours de l'apprentissage, le modèle tend vers un état dans lequel il approxime au mieux le problème présenté, et répond donc au plus près de la réponse attendue. Ainsi, son erreur de réponse doit tendre vers 0. Or, le risque est d'entraîner un modèle de telle sorte qu'il soit parfaitement adapté aux données d'entraînement, sans toutefois avoir la possibilité de généraliser. Le calcul de la perte sur des données hors entraînement, dites de validation, permet de vérifier que l'entraînement ne présente pas ce problème. Au fur et à mesure que l'entraînement avance, la perte d'entraînement et de validation tendent vers une même valeur. Cela dit, quand le modèle commence à représenter trop fidèlement les données d'entraînement, il s'éloigne de la représentation de la validation, c'est à dire que la généralisation devient moins efficace. Ainsi, la perte de validation croît de nouveau et s'éloigne de l'objectif "parfait" (une perte nulle). Quand cette tendance divergente apparaît, le sur-apprentissage peut être constaté.

3.3.2 Matrices de confusion

Une fois un modèle de classification entraîné, il est possible de quantifier la précision du modèle (sur les données d'entraînement comme de validation) en calculant une matrice de confusion. Cela permet d'obtenir une représentation des quantités de cas selon quatre catégories : *Vrai-Positif* (*réponse* positive et *attendue* positive), *Faux-Positif* (*réponse* positive et *attendue* négative), *Vrai-Négatif* (*réponse* négative et *attendue* négative) et *Faux-Négatif* (*réponse* négative et *attendue* positive). Respectivement, ces catégories (et les quantités correspondantes seront notées *TP*, *FP*, *TN* et *FN*.

De ces catégories, nous pouvons obtenir certaines métriques utiles pour estimer la validité de modèles, en particulier la sensibilité et la spécificité, ainsi que la précision. Ces valeurs sont obtenues de la manière suivante :

$$sensibilité = \frac{TP}{TP + FN}$$

$$spécificité = \frac{TN}{TN + FP}$$

$$précision = \frac{TP + TN}{TP + FP + TN + FN}$$

La sensibilité permet de montrer la capacité d'un modèle à répondre positivement à un cas proposé positif. Son complément à 1 est le taux de cas positifs manqués par le modèle. La spécificité, quant à elle, montre le taux de cas négatifs reconnus comme tels. Enfin, la précision (*accuracy* en anglais) donne le taux de classification correcte sur l'ensemble des cas.

3.3.3 Courbes ROC

Dans le cas de classification utilisant des jeux déséquilibrés (ratio cas positifs/cas négatifs très inférieur ou supérieur à 1), les valeurs de la matrice de confusion peuvent présenter un biais. Aussi, la matrice de confusion se limite à une délimitation Vrai/Faux pour un seuil fixé arbitrairement (en général 0.5 pour des réponses allant de 0 pour Faux à 1 pour Vrai). Dans ces cas, nous utilisons le tracé d'une courbe dite *courbe ROC* (Receiver Operating Charateristic ou Caractéristique de Fonctionnement du Récepteur).

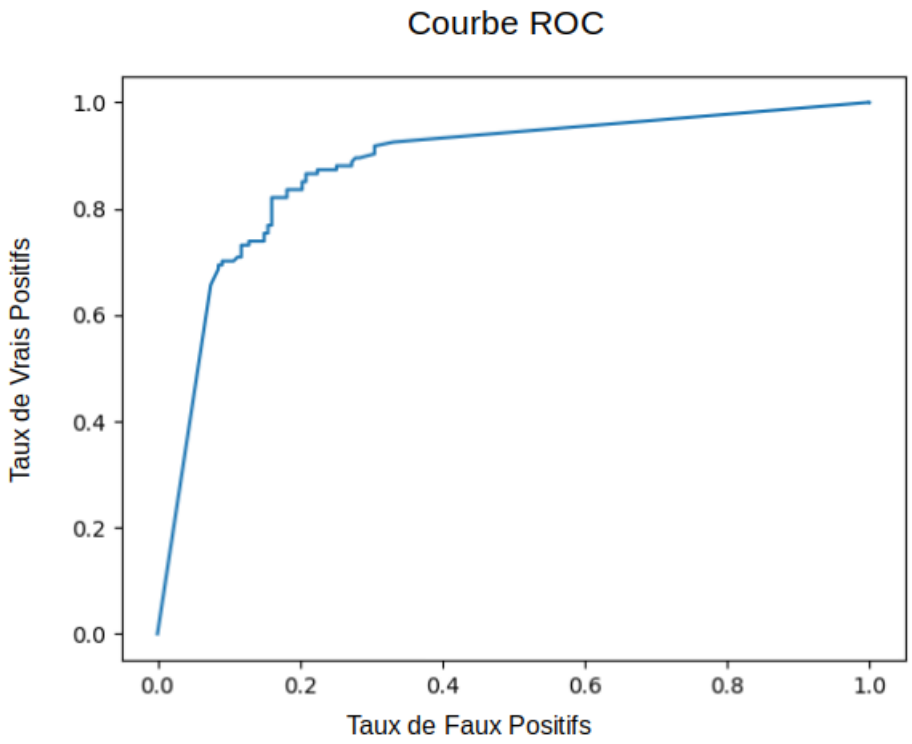


FIGURE 3.12 – Exemple de courbe ROC

Cette courbe est tracée en faisant varier le seuil entre la valeur choisie pour Vrai et celle choisie pour Faux, ici 1 et 0. Chaque seuil définit de fait une nouvelle matrice de confusion, avec notamment les valeurs TP et FP vues précédemment, en donc la sensibilité et la spécificité. La courbe ROC est alors obtenue en représentant la sensibilité en abscisse et le complément à 1 de la spécificité en ordonnée. Un exemple de courbe ROC est présenté dans la Figure 3.12. Il est simple d'observer que toute courbe ROC passe par les points (0,0) et (1,1), respectivement pour des seuils maximum (supérieur à 1, aucun cas positif possible) et minimum (inférieur à 0, tous les cas sont positifs).

Cette courbe seule ne permet pas de quantifier avec précision l'efficacité globale du modèle. En effet, une métrique supplémentaire, nommée *AUC* (Area Under the Curve ou aire sous la courbe). Cette valeur représente la superficie présente entre la courbe est l'axe des ordonnées. Un modèle hypothétique sans réponse (dont les seuls points seraient (0,0) et (1,1)) montre une AUC de 0.5. Le modèle parfait, dont le seul point intermédiaire serait (0,1) montre une AUC de 1. Entre ceux-ci, différentes AUC peuvent être calculées, et plus un modèle tend vers 1, plus il est considéré comme précis et efficace.

3.4 Utilisation du Machine Learning pour le Trouble du Spectre de l'Autisme

Le diagnostic du Trouble du Spectre de l'Autisme demande des rencontres entre le sujet et des experts en PC, qui doivent comprendre et analyser les réactions du sujet, ce qui peut être difficile dans les cas de TSA léger. Aussi, le faible nombre de centres de ressources Autisme (CRA) [1] ne permet pas d'assurer une orientation au plus tôt. En raison du caractère chronophage et tardif des méthodes de diagnostic actuelles pour le TSA, le monde scientifique cherche des méthodes d'automatisation de celles-ci.

Des travaux ont été menés concernant l'utilisation de données d'IRM (Imagerie par Résonance Magnétique) afin de différencier les enfants atteints par le

1. "En 2009, vingt quatre CRA sont implantés en France et dans les DOM. Ils se sont constitués en association nationale ANCRA." (http ://www.autismes.fr/fr/les-cra.html)

TSA et les enfants ne présentant pas de TSA. Notamment, [Zhou et al., 2014] a utilisé des méthodes basées sur des analyses simples (volumétrie cérébrale), la théorie des graphes (analyse basée l'activité des différentes régions cérébrales) et des techniques simples de Machine Learning (Machine à Vecteur de Support ou SVM notamment, une méthode de recherche de la meilleure séparation de l'espace de définition du problème pour différencier des classes choisies) pour permettre le repérage de marqueurs spécifiques au TSA. Aussi, les travaux présentés dans [Heinsfeld et al., 2018] montrent l'exploitation du jeu de données ABIDE (Autism Brain Imaging Data Exchange), un jeu de données d'imagerie cérébrale, pré-traité en utilisant une matrice de corrélation des activités de certaines zones du cerveau, à l'aide de méthodes de Machine Learning (Réseaux de Neurones et SVM en particulier). Ces résultats, encore peu avancés, proposent tout de même une précision d'environ 70% dans la différenciation entre les personnes souffrant de TSA et les personnes non atteintes.

D'autres travaux se sont concentrés sur la dissociation du Trouble du Spectre de l'Autisme d'autres troubles neurodéveloppementaux. [Duda et al., 2016] a, par exemple, montré que seuls 5 des 64 éléments nécessaires à l'établissement du score SRS (Social Responsiveness Scale), un score permettant d'évaluer l'état autistique d'une personne via ses réactions sociales, suffisent pour séparer les enfants TSA d'enfants atteints de TDAH (Trouble du Déficit de l'Attention avec ou sans Hyperactivité).

Par ailleurs, [Uluyagmur-Ozturk et al., 2016] traite du diagnostic du TSA et du TDAH (Trouble de Déficit de l'Attention avec ou sans Hyperactivité) en utilisant des techniques de Machine Learning chez des enfants. Les participants doivent reconnaître des émotions sur des visages qui leurs sont présentés. Deux valeurs sont extraites pour chaque visage : le temps de réponse et le fait que la réponse soit correcte ou non. Des modèles d'arbres de décision (ensemble de choix conditionnels présentés sous la forme d'un arbre) et AdaBoost (algorithme combinant plusieurs petits outils d'apprentissages, dits faibles [Freund et al., 1999]). Ces travaux permettent une dissociation relativement efficace (jusqu'à une précision de 85.56%) entre les enfants TSA et TDAH. Les enfants TSA sont également dissociés des enfants sans trouble avec cette approche (80%).

[Thabtah, 2019] s'est concentré sur des données de contexte pour apporter une solution d'aide au diagnostic du TSA. Le but de cette étude vise à développer un ensemble de règles par Machine Learning afin de permettre une distinction explicable entre les participants TSA et non-TSA. Les données exploitées incluent l'âge, le sexe, le pays de résidence et le fait qu'un membre de la famille soit atteint de TSA, entre autres. Elles incluent aussi 10 questions permettant de quantifier l'attention visuelle et auditive du participant. Les réponses sont fournies via une application développée précédemment ([Thabtah, 2018]). En exploitant les règles obtenues par deux des approches essayées (RML[Dimou et al., 2014] et RIPPER[Cohen, 1995]), les auteurs montrent que 4 des questions posées ont une forte influence sur le diagnostic final. Ces questions portent sur la communication non-verbale, la répétitivité du comportement et la capacité d'attention.

3.5 Exploitation de données d'Eye Tracking via ML

Les Eye-Trackers offrent une quantité impressionnante de données, sur des éléments assez variés du participant. Une quantité aussi importante intéresse de fait les chercheurs et concepteurs d'applications, en particulier quand leur domaine de recherche ou logiciel inclut du Machine Learning.

[Zemblys, 2016] propose un benchmark utilisant différentes approches de machine learning pour classer des données d'Eye-Tracking entre les quatre catégories d'événement possibles : Fixation, Saccade, Oscillation Post-Saccade et Blinks. Ces dernières incluent notamment les clignements. Les données sur lesquelles sont testées les approches sont des échantillons d'Eye-Tracking labellisées manuellement. Via l'application d'approches de Machine Learning, les auteurs montrent une précision de catégorisation croissante avec l'augmentation de la fréquence d'échantillonnage. Cela est cohérent avec nos remarques concernant l'Eye-Tracker que nous utilisons (voir Section 4.2). Avec sa fréquence limitée à 60Hz, il n'est pas en capacité de repérer les Oscillations Post-Saccade. Ces travaux ont été poursuivis avec [Zemblys et al., 2018], montrant des résultats meilleurs que ceux des pilotes intégrés aux outils d'Eye-Tracking du marché.

3.6 Eye-Tracking et Machine Learning pour l'aide au diagnostic du Trouble du Spectre de l'Autisme

La combinaison de l'Eye-Tracking et du Machine Learning semble utile pour l'aide au diagnostic du Trouble du Spectre de l'Autisme. En effet, des travaux montrent qu'il est approprié d'exploiter conjointement les systèmes d'Eye-Tracking et le Machine Learning afin de permettre le diagnostic du Trouble du Spectre de l'Austime. [Wan et al., 2018] a exploité les réactions, relevées par Eye-Tracking, pour déterminer l'état autistique de participants. Une vidéo sans son leur est diffusée, dans laquelle une femme s'adresse à eux. Leur tracé oculaire est enregistré, et l'information est résumée à la zone du visage de la femme que le participant regarde ("nez" ou "yeux" par exemple). Un SVM a été utilisé pour exploiter les temps de fixation sur les différentes zones du visage pour dissocier les participants TSA et non-TSA, avec une précision de 85%.

Des travaux similaires ont été conduits par [Liu et al., 2016]. Dans ce cas, les zones d'intérêt (ou AOI pour Areas Of Interest) ont été définies en analysant la concentration des participants sur certaines parties du visage. Les participants doivent mémoriser des visages puis les retrouver dans une liste proposée ultérieurement. Les auteurs proposent un découpage du visage selon les points les plus regardés grâce aux K-Moyennes. Elles permettent de trouver dans un espace un nombre prédéfini de points centraux d'attention, comme par exemple, si l'on considère des nuages de points, le centre de gravité de ces nuages. Les nuages de points sont ici constitués des points de fixation. Ces centres de gravité permettent alors de séparer l'image en autant d'AOI. Cherchant de 16 à 96 centres de gravité, la fréquence de regard des AOI obtenues est exploitée avec un SVM, donnant une classification précise à hauteur de 88%.

A l'opposé, [Yaneva et al., 2018] cherche à différencier les personnes TSA et non-TSA en observant une activité quotidienne : la visite de site-web. Différentes instructions, limitées dans le temps, sont données aux participants. Au cours de l'exécution de ces instructions, le parcours oculaire est relevé par un Eye-Tracker. Les AOI sont définies selon le code source de la page

consultée. Une régression logistique est utilisée, qui consiste à exploiter un ensemble de paramètres pour définir si un événement arrive, d'un point de vue probabiliste. Les paramètres utilisés sont les taux et nombre de fixation des AOI, le temps passé sur les pages, entre autres. La distinction entre TSA et non-TSA est d'une précision de 75%.

Dans chacun de ces cas les résultats sont d'une qualité remarquable. L'incertitude des résultats de ces travaux est justifiable par le fait que le diagnostic du Trouble du Spectre de l'Autisme n'est pas effectué, en pratique, avec les données seules de l'Eye-Tracking. La liste des éléments de notation du score CARS en est un parfait exemple (voir Section 2.1).

Il existe aussi des études orientées vers d'autres moyens de diagnostic tels que les EEG notamment [Murias et al., 2007, Kylliäinen et al., 2012, Jamal et al., 2014] qui montrent encore des résultats trop peu concluants (précision faible ou quantité de données restreinte) pour que de telles approches soient considérés dans nos travaux.

Les travaux décrits dans cette Section montrent la possibilité de combiner des résultats obtenus par Eye-Tracking à des modèles de Machine Learning pour permettre le diagnostic de l'autisme chez des enfants. En particulier, [Cilia et al., 2018] montre cet intérêt au cours d'une étude synthétique des approches utilisant les Eye-Trackers pour le diagnostic du TSA basé sur l'attention de l'enfant. Les travaux conduits au cours de la thèse présentée dans ce manuscrit sont tous basés sur des données provenant des auteurs de cette étude.

Chapitre 4

Données de travail

4.1 Protocole de collecte des données

Le travail de captation des données a été effectué par une équipde de chercheurs psychologues du Laboratoire CRP-CPO, nos référents en matière de connaissances concernant le Trouble du Spectre de l'Autisme.

Participants

Un groupe de 59 participants a pris part à un ensemble d'expérimentations liées à l'Eye-Tracking. Un psychologue a catégorisé les participants selon deux classes : TSA (atteint par le Trouble) et non-TSA (non touché par le Trouble).

Il est souhaité d'avoir des participants très jeunes, puisque l'objectif est le diagnostic précoce du TSA. Les participants représentent un groupe d'enfants d'âges compris entre 3 et 13 ans. La Figure 4.1 montre l'histogramme des âges des participants. Elle montre que près de la moitié des participants (environ 47%) sont âgés de 5 à 9 ans. Aussi, les participants sont tous des enfants, avec un âge moyen et médian d'environ 8 ans . Sur les 59 enfants, 38 sont des garçons, soit environ 64%. Sur ces mêmes 59 enfants, 29 sont diagnostiqués TSA, soit environ la moitié de la population. Le niveau autistique des enfants TSA a été validé par calcul du score CARS (voir Section 2.1), avec une moyenne d'environ 33 et une médiane de 34.5.

Le score CARS (Childhood Autism Rating Scale)[Schopler et al., 1980] est largement considéré comme une méthode standard pour représenter la

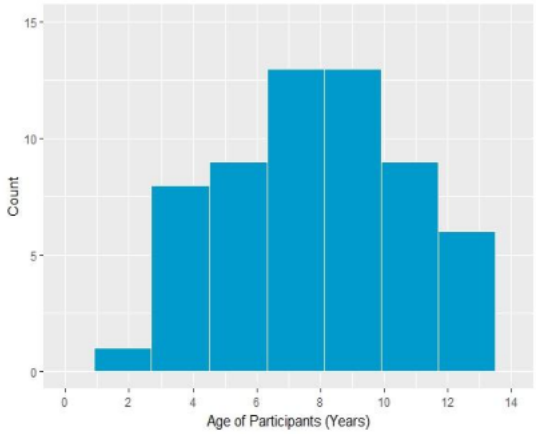


FIGURE 4.1 – Distribution des âges des participants.

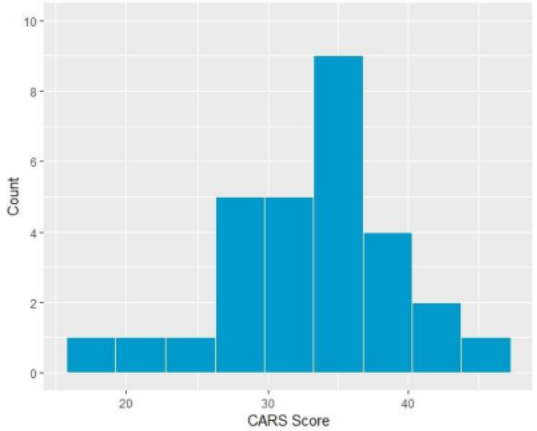


FIGURE 4.2 – Distribution des scores CARS des participants diagnostiqués TSA.

Nombre de participants	59
Distribution par Sexe (M/F)	38 / 21
Nombre de Non-TSA	30
Nombre de TSA	29
Age (Moyen/Médian)	7.88 ans / 8.1 ans
CARS (Moyen/Médian)	32.97 / 34.50

TABLE 4.1 – Statistiques de l'échantillon de participants.

sévérité des symptômes du TSA [Ozonoff et al., 2005]. L'échelle note subjectivement différents aspects du comportement, incluant les relations aux personnes, l'imitation, la réponse émotionnelle, la peur et la nervosité, la communication verbale, la communication non-verbale, le niveau d'activité, le niveau et la consistance de la réponse intellectuelle et enfin les impressions générales. La Figure 4.2 donne la distribution des valeurs CARS parmi les participants TSA. Il est à noter que sur ce schéma, 8 enfants présentent un score inférieur à 30, c'est-à-dire qu'ils ne devraient pas rentrer pas dans le TSA. Néanmoins, ces personnes ont eu un TSA diagnostiqué, ce qui peut influencer le résultat de l'évaluation, en raison d'une prise en charge ou d'un défaut de l'estimation du CARS.

Nous souhaitons avoir une répartition équilibrée entre nos deux catégories, de telle sorte que chacune représentent environ 50% des participants. La Table 4.1 donne une description statistique des participants.

Captation

Les captations ont été effectuées avec l'accord des parents, notamment pour le cas des enfants non concernés par le TSA, qui proviennent tous d'établissements scolaires de la région Hauts-de-France. Les parents ont été invités à indiquer leurs potentielles inquiétudes, toutes prises en considération avec précaution. L'assentiment de l'enfant a été reçu avant toute captation. Par ailleurs, toutes les procédures impliquant des participants humain ont été conduites en accord avec les principes éthiques de la recherche institutionnelle et nationale et avec la déclaration d'Helsinki de 1964 et ses amendements ultérieurs ou des standards éthiques comparables.

La captation est effectuée dans les locaux de l'Université de Picardie Jules Verne, dans une pièce calme, agrémentée de barrières visuelles blanche pour réduire les risques de distractions. Lors de la captation, chacun des participants est placé devant un écran auquel est fixé un Eye-Tracker. L'enfant est assis seul ou sur les genoux de l'un de ses parents. La distance avec l'écran est d'approximativement 60cm.

Avant toute mesure, une étape de calibrage permet d'assurer une qualité de mesure suffisante. Sur une série de points successifs prédéfinie, une petite image animée et attrayante pour des enfants est diffusée. L'enfant doit regarder ces différents points avant de passer au suivant. Cela permet d'associer des valeurs repères pour adapter les résultats de la captation précisément à la situation actuelle.

Les participants ont été invités à regarder une variété de vidéos, dans lesquelles des scénarios spécifiques sont inclus pour stimuler le mouvement oculaire sur l'écran. Les scénarios varient en contenu et en longueur pour permettre l'analyse de l'activité oculaire des enfants selon différentes perspectives.

Après cette étape de calibrage, une vidéo est diffusée. Cette vidéo contient un ensemble de stimuli spécifiques permettant de dissocier les comportements des enfants TSA de ceux des enfants non-TSA. Il existe plusieurs vidéos et chacune d'elles vise le même ensemble de stimuli. Une fois le calibrage effectué et au cours de la captation, plus aucune consigne n'est donnée à l'enfant. Il est seulement demandé à l'enfant de regarder l'écran, pour qu'il regarde les contenus le plus naturellement possible, sans biais.

Les scénarios vidéos ont été conçus pour inclure des éléments visuels, qui peuvent être spécialement attrayant pour les enfants (ballons colorés, dessins animés). Une partie de ces éléments est constituée de photos de visages de femmes et d'hommes, ainsi que des images d'objets attrayants. Certaines vidéos incluent également une actrice qui parle et tente d'attirer l'attention du participant à des éléments spécifiques, visibles ou invisibles. La position des éléments peut changer au cours des expérimentations. Ces positions et leurs ordres de présentation varient pour contrebalancer les effets du stimulus, et éviter des biais d'habitude. Ainsi, les éléments montrés (un ballon

par exemple) peuvent être à gauche ou à droite. Dans notre cas, toutes les conversations sont faites en français puisqu'il s'agit du langage maternel des participants.

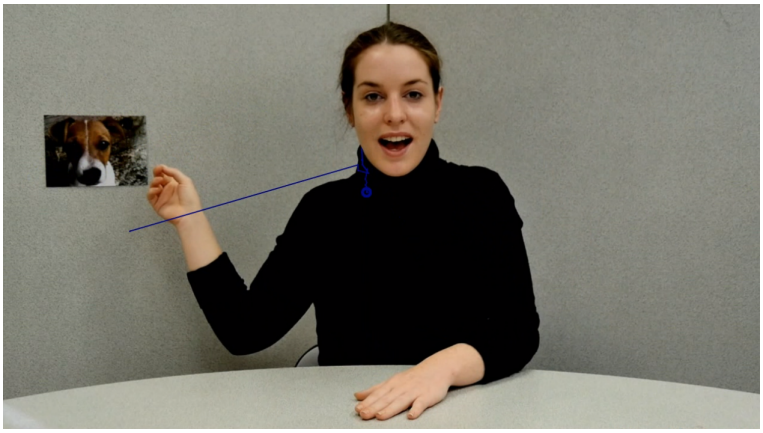
Un exemple de vidéo présente une actrice qui simule une conversation avec l'enfant. Elle s'adresse directement à lui, par la parole et le geste. Elle présente un objet ou une image apparaissant à l'écran, par le regard, par le geste, par la parole et par des mélanges de ces moyens de communication, continuant à s'adresser à l'enfant. Dans certains cas, l'un des objets ou images n'est pas visible, mais l'actrice s'y réfère comme s'il était effectivement visible. Les participants sont spécialement examinés concernant la qualité du contact visuel avec l'actrice et le niveau d'attention sur les autres éléments. La Figure 4.3 montre deux exemples instantanés de la vidéo, avec l'activité oculaire récente de participants, pour l'un des stimuli exploités (l'actrice montre un ballon visible). Il s'agit de la représentation du tracé oculaire que les psychologues experts sur le TSA peuvent obtenir d'un Eye-Tracker.

4.2 Données initiales

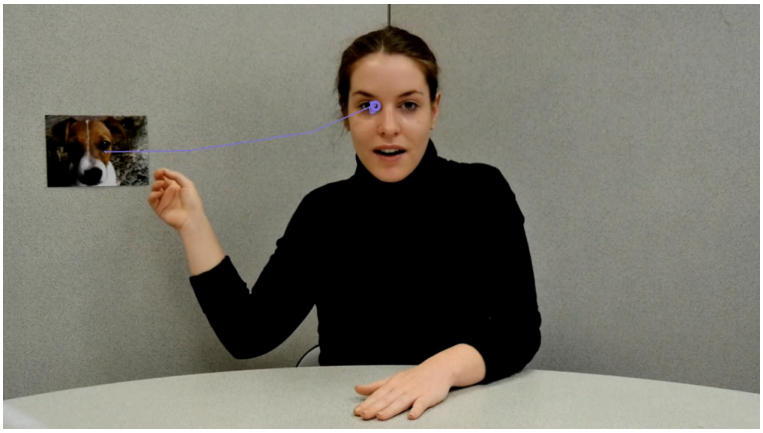
Au cours de ces travaux nous exploiterons deux types de données issues de l'Eye-Tracker : les *RawData*, données brutes issues de l'outil, et les *EventData*, données pré-traitées suivant les événements de la dynamique oculaire (voir [SMI, 2012] pour les détails de fonctionnement de l'appareil).

RawData

Au cours de la captation, l'Eye-Tracker relève des données à la fréquence de 60Hz, principalement les positions du regard de l'enfant sur l'écran, mais aussi d'autres informations telles que la position spatiale des yeux de l'enfant par rapport à l'appareil, la taille de sa pupille ainsi que son diamètre et sa position et le vecteur spatial d'orientation du regard, de l'œil au point de fixation sur l'écran. Chacune de ces mesures est effectuée pour l'œil gauche et l'œil droit séparément. Toutes ces informations sont par ailleurs localisées dans le temps, soit relativement à l'instant de début de la captation, soit dans l'absolu avec une date et/ou une heure précises à la milliseconde près.



(a) Participant TSA



(b) Participant TC

FIGURE 4.3 – Capture d'écran de la vidéo présentée avec le tracé oculaire récent du participant

D'autres valeurs peuvent être ajoutées à chaque enregistrement, comme par exemple l'attention portée à une zone d'intérêt (AOI). Une AOI est une zone de l'écran pour laquelle il est recherché des informations particulières, notamment le parcours oculaire sur cette zone. Dans notre cas, les valeurs issues des AOI sont fixées à 1 pour chaque instant d'enregistrement pour lequel le participant regarde l'AOI, elle est fixée à 0 sinon.

L'état du regard de l'enfant peut également être classé, à chaque instant de mesure, selon trois catégories en fonction de sa dynamique : la saccade, la fixation et le blink (ou clignement). La saccade permet d'indiquer un mouvement oculaire rapide tandis que la fixation concerne un mouvement oculaire lent, concentré sur une zone. Le blink indique le clignement de l'œil mais aussi un défaut de mesure. L'état indique un cas pour lequel l'œil n'est pas repérable.

Il existe un quatrième type d'état dynamique, l'oscillation post-saccade, représenté par la phase de stabilisation du mouvement oculaire après une saccade et avant une fixation, mais l'appareil utilisé lors de ces travaux ne permet pas, en raison de sa faible fréquence, de mesurer avec la finesse nécessaire pour repérer un tel événement.

L'identification des différents états de la dynamique du regard est effectuée par l'Eye-Tracker. Les travaux de cette thèse n'explorent pas la manière d'effectuer cette identification mais exploitent directement les données des événements.

L'ensemble de ces données est alors regroupé dans un fichier de type CSV, produit par le pilote fourni avec l'Eye-Tracker, incluant une ligne par mesure effectuée. Ces fichiers générés seront appelés par la suite les fichiers RawData. La Figure 4.4 montre visuellement le trajet suivi par le regard d'un enfant TSA sur une vidéo complète, indépendamment de toute notion dynamique. Les travaux présentés en Section 4.3 proposent, quant à eux une version colorée de ce trajet, permettant la visualisation de la dynamique pour chaque mouvement oculaire. Il faut noter que dans chacun de ces cas, la représentation sous forme d'image retire la dimension temporelle aux données (mis à part en suivant les points manuellement, ce qui est parfois très difficile considérant la quantité de lignes tracées).

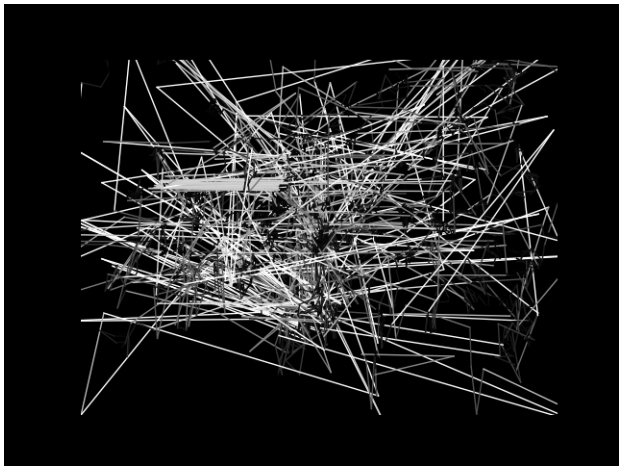


FIGURE 4.4 – Représentation graphique du tracé oculaire d'un participant au cours du visionnage d'une vidéo, dans un cadre défini. Chaque trait blanc constitue le déplacement du regard d'un instant d'enregistrement au suivant.

EventData

Alors que les RawData permettent le relevé d'informations à chaque instant de l'enregistrement, il est possible n'obtenir que les informations propres aux événements de dynamique du regard. Ainsi, si n instants consécutifs de Fixation se succèdent lors de l'enregistrement, l'Eye-Tracker peut produire un seul jeu d'informations, décrivant le comportement du regard au cours de ces n instants de Fixation. Cela s'applique également aux Saccades. En fonction de l'état regroupé, les informations calculées varient. Dans le cas d'une Fixation, sont ajoutées la position (en x et y sur l'écran), la taille moyenne de la pupille (en x et y, et son diamètre) et la dispersion, qui représente l'étendue du mouvement oculaire au cours de la Fixation (en x et y). Pour une Saccade, les calculs ajoutent les positions de début et de fin (en x et y), l'amplitude, l'accélération moyenne et maximale, la décélération maximale, la vitesse moyenne et maximale et le taux de mesures de la saccade pour lesquelles la vitesse est maximale. Dans les deux cas, la durée de l'événement, ainsi que son instant de début et de fin relativement à la durée de la captation est incluse. Les événements sont dissociés entre les yeux, dans le cas où un œil serait en situation de Blink (due à une potentielle erreur de mesure ou à une rotation de la tête sortant un œil du champ de vision de l'appareil).

Un fichier, produit par le pilote du constructeur, permet d'accéder à l'ensemble de ces données. Cependant, le constructeur ne partage pas les méthodes de calcul des valeurs produites. Chaque ligne du fichier correspond à un événement de la dynamique oculaire.

4.3 Notre contribution à la génération d'images

Contexte

Nous avons choisi de nous rapprocher de l'affichage du suivi oculaire proposé par l'Eye-Tracker, qui montre de fortes différences entre les enfants TSA et typiques. De ce fait, nous avons cherché à reproduire cet affichage, non extractible depuis l'outil, en le concentrant dans une image. Ce genre d'approche a déjà été entrepris par [Goldberg and Helfman, 2010]. Notre approche cherche à modéliser dans le tracé la dynamique du regard.

Les participants dont les données sont exploitées et le protocole de recueil des données sont présentés dans la Section 4.1.

Visualisation des enregistrements d'Eye-Tracking

L'idée clé de notre méthodologie est de construire une représentation visuelle de la dynamique oculaire. Une image est conçue comme suit :

- Une ligne est tracée pour chaque transition entre une position (x(t), y(t)) et (x(t+1), y(t+1)), où t est un instant défini de l'expérimentation.
- Le changement de couleur au cours de la ligne représente la dynamique des mouvements oculaires. Les valeurs RGB sont posées selon les trois dérivées de la position successives, c'est-à-dire la valeur de vitesse, d'accélération et d'à-coup à l'instant t. Pour le calcul de ces valeurs, voir les Equations 4.1 à 4.3, pour lesquelles *diagonale* vaut la taille de la diagonale de l'écran en pixels. L'équation 4.4 montre la manière dont sont ensuite bornées les valeurs de la vitesse (le traitement est analogue pour l'accélération et l'à-coup).

$$v(t) = \frac{\sqrt{(x(t) - x(t-1))^2 + (y(t) - y(t-1))^2}}{diagonale/4} \tag{4.1}$$

FIGURE 4.5 – Les gradients de couleurs représentant la dynamique des mouvements oculaire.

$$a(t) = \frac{v(t) - v(t-1)}{diagonale/4} - 0.5 \tag{4.2}$$

$$j(t) = \frac{a(t) - a(t-1)}{diagonale/4} - 0.5 \tag{4.3}$$

$$v_{utilisée}(t) = \begin{cases} 1 & si \quad v(t) > 1 \\ 0 & si \quad v(t) < 0 \\ v(t) & sinon \end{cases} \tag{4.4}$$

La Figure 4.5 montre les trois gradients de couleur utilisés pour représenter le mouvement oculaire. Par exemple, les valeurs de vitesse varient du noir (faible) à rouge (élevé). De cette manière, plus les valeurs de vitesse sont fortes, plus elles tendent, graduellement, vers des valeurs représentant un rouge plus prononcé. La même idée s'applique pour la représentation de l'accélération et de l'à-coup en utilisant les gradients de verts et de bleus respectivement.

Pour une meilleure illustration de notre méthode, la Figure 4.6 résume la procédure pas-à-pas de visualisation des enregistrements d'Eye-Tracking. Cette procédure est répétée pour chaque vidéo à traiter.

Un autre problème consiste à limiter les images de telle sorte qu'elles contiennent une quantité équivalente d'informations. Pour cela, nous avons limité le nombre de points consécutifs à dessiner selon les critères suivants :

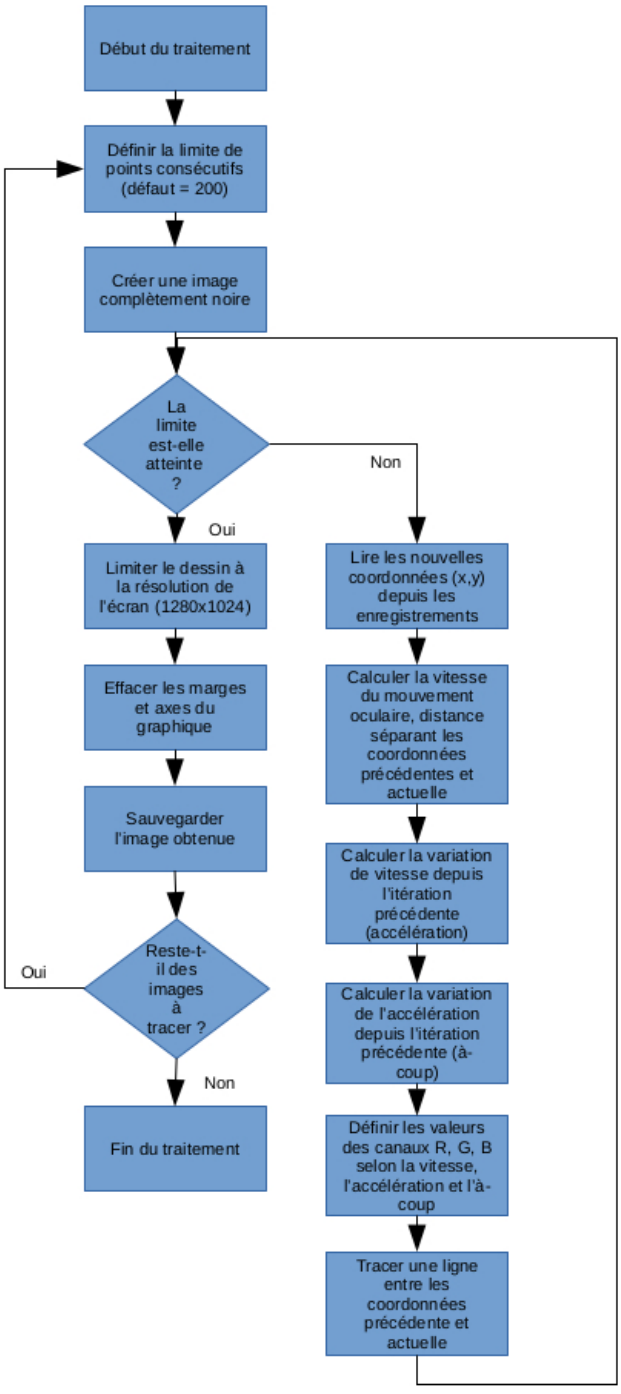
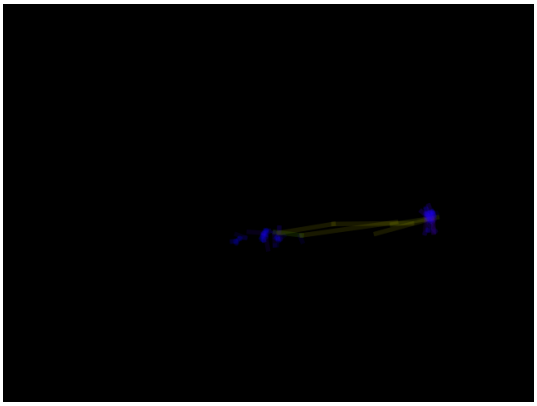
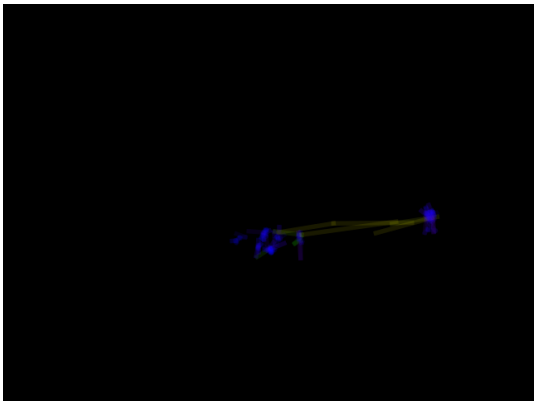


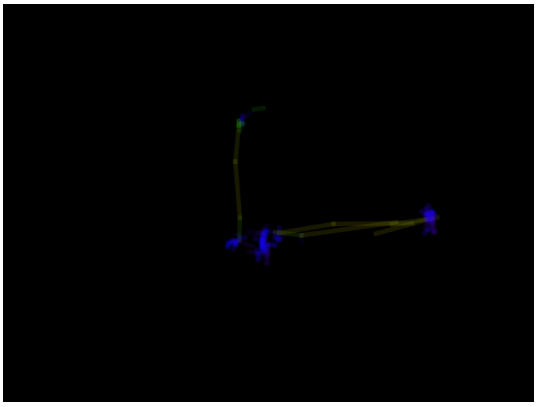
FIGURE 4.6 – Procédure de création des images.



(a)

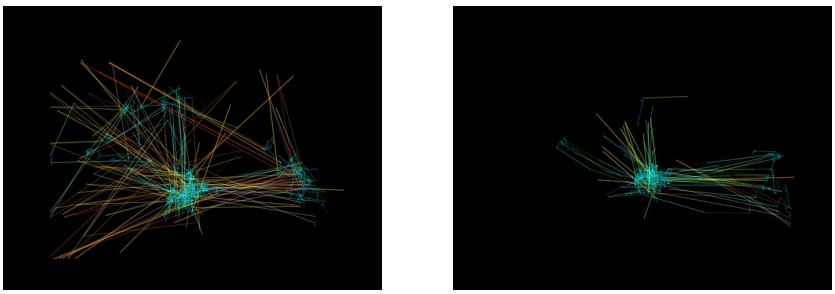


(b)



(c)

FIGURE 4.7 – Exemples d'images générées avec 100 (a), 150 (b) et 200 (c) points consécutifs.



(a) (b)

FIGURE 4.8 – Image générée, incluant la dynamique du regard pour un enfant TSA (a) et un enfant TC (b). Ces images comprennent l'intégralité du suivi oculaire d'un enfant chacune, pour une meilleure lisibilité.

cette limite doit être suffisamment élevée pour décrire correctement le comportement oculaire et de trop hautes valeurs posent le risque de produire des visualisations illisibles. Ainsi, plusieurs tests ont été conduits pour choisir une limite appropriée. Avec une limite fixée à 100 ou 150, le nombre de lignes représentant le chemin oculaire est trop faible et l'information présente sur l'image résultante semble insuffisante pour discriminer efficacement les catégories de participants. Finalement, nous avons choisi de fixer la limite à 200, ce qui règle correctement l'équilibre nécessaire à la reconnaissance des paramètres clés du mouvement oculaire (voir la Figure 4.7 pour des exemples d'images pour chaque limite).

La Figure 4.8 montre deux exemples de visualisation concernant un enfant TSA et un enfant non-TSA regardant la même vidéo. Cette figure permet de mettre en valeur la dispersion du regard des participants dans chaque cas.

Le procédé de transformation de données a été entièrement implémenté en utilisant Python. Les visualisations sont produites par le module Matplotlib [Hunter, 2007].

Dans le cas des vidéos proposées par l'Eye-Tracker, la couleur n'est pas utilisée pour montrer la dynamique du regard. Aussi, le tracé du regard est montré au fur et à mesure de l'avancée de la vidéo, sur un nombre de points limité.

Ce nombre de points consécutifs est insuffisant pour nos traitements, en raison de notre limite fixée à 200 points consécutifs. Aussi, nous ne visions pas le traitement de fichiers vidéo. Ce format fourni par l'Eye-Tracker n'était donc pas exploitable dans le cadre des travaux présentés dans ce manuscrit.

Cela étant, il est à noter que les psychologues, experts de l'autisme, peuvent utiliser ces tracés visuels en guise de soutien au diagnostic.

Jeu de données

Les 59 participants ont visionné les vidéos qui leur ont été proposées (2 à 88 par enfant, moyenne 15.19), nous générons des images suivant la limitation de 200 points d'enregistrement de la position du regard successifs. Le jeu de données obtenu selon la procédure présentée dans la Figure 4.6 contient 547 images, soit une moyenne d'environ 9.27 images générées par participant. 328 images représentent des tracés non-TSA, 219 des tracés TSA (moyennes par participant de 10.93 et 7.55 respectivement). Cependant, ce jeu de données peut être multiplié en appliquant des techniques typiques d'augmentation d'images (zoom, décalage, miroitement). Ces techniques sont utiles pour le développement d'applications de Machine Learning notamment.

La taille par défaut de ces images est fixée à 640x480 pixels. Le jeu de données est organisé dans deux dossiers principaux : Images et Metadata. Le dossier Images contient deux sous-dossiers : TCImages (Non-TSA) et TSImages (Diagnostiqués TSA). Ces sous-dossiers incluent l'ensemble des images générées pour les enfants non-TSA et TSA respectivement.

Exploitation

La collection d'images générées par ce procédé a fait l'objet d'une publication [Carette et al., 2018]. Le jeu de données est publiquement accessible sur Figshare. Les images et leurs métadonnées peuvent être directement téléchargées [1].

Ainsi, ce jeu d'images a été exploité lors d'une compétition nommée HackOver par Piyush Gupta et Gandhapu Kalyan[Gupta and Kalyan, 2019], en

1. https ://figshare.com/s/5d4f93395cc49d01e2bd

Inde. Les images ont été utilisées pour entraîner un CNN. Ce modèle montre des résultats en entraînement équivalents à ceux des travaux présentés en Section 6.1, soit une précision, sur les données d'entraînement, de 97%. Cela étant, ces travaux ne montrent pas l'efficacité de ce nouvel apprentissage en utilisant un jeu de validation, et ne peuvent pas être comparés à nos résultats avec précision.

Le résultat a conduit à la mise en place d'une application mobile permettant la dissociation entre TSA et non-TSA, sur la base d'images générées comme présenté dans cette Section. La thématique de la compétition n'est pas limitée au TSA, mais la préférence est de mettre en avant les thématiques sociales et environnementales. Les critères des juges de la compétition sont les suivants : le design, la qualité de l'idée, l'implémentation technique et l'impact potentiel. Le projet proposé a atteint la seconde place du classement sur 56 participants.

Un tel intérêt pose l'éventualité que cet ensemble d'images puisse être à nouveau utilisé, et devienne à terme un outil de benchmark de modèles de Machine Learning.

Format numérique

Ces données ont également été exploitées numériquement. Aux valeurs calculées, nous ajoutons les projections orthogonales des vitesses, accélérations et à-coups (v, a, j) sur les axes x (v_x, a_x, j_x) et y (v_y, a_y, j_y). v_x et v_y correspondent à la vitesse de déplacement selon l'axe x et l'axe y respectivement. Ainsi :

$$v_x(t) = x(t) - x(t-1)$$

pour tout t instant d'enregistrement.

Il en va de même pour v_y par rapport aux valeurs successives de y. a_x et a_y sont la variation des valeurs de vitesse projetées, et j_x et j_y la variation des accélérations projetées (voir les Equations 4.2 et 4.3 pour les méthodes de calcul).

4.4 Réduction de dimensionnalité

Motivations

Chaque image créée selon le procédé précédent comprend près d'un million d'informations (640 par 480 sur 3 canaux), et une grande partie de ces informations est inutile car de nombreuses zones des images sont complètement noires, de manière constante, sur chaque image. En effet, les stimuli utilisés sont principalement localisés à des zones précises de l'écran (pour analyser les fixations sur les différentes zones ciblées). Par ailleurs, la quantité limitée de traits tracés pour le chemin oculaire implique une majorité de pixels noirs dans l'image. Dans l'optique de simplifier les traitements sur ces images, nous avons choisi de limiter l'information aux éléments discriminants, tout en réduisant les canaux à une échelle de gris.

Procédé

La première étape consiste à réduire la taille totale de l'image, en la ramenant à une taille de 100 pixels par 100 pixels. La perte d'informations, bien qu'existante, a été considérée suffisamment faible pour assurer la poursuite des modifications. Ensuite, les trois canaux de couleur de l'image ont été ramenés à un seul, par calcul de luminance [ITU, 2017]. La formule utilisée pour le calcul de luminance pour chaque pixel est la suivante :

$$Luminance = 0.299 * R + 0.587 * G + 0.114 * B$$

avec R, G et B respectivement les valeurs des canaux R, G et B du pixel considéré. L'utilisation d'une telle formule conserve les distinctions entre saccades et fixations repérées précédemment, malgré la perte de deux des trois canaux. A ce point des transformations, la quantité d'information par image initiale et passée de près d'un million à 10000. Cela étant, il ne semble pas que tous les éléments de l'image soient aussi intéressants à traiter que les autres, certains ne montrant probablement pas suffisamment de variation entre les deux catégories d'enfants.

L'utilisation d'une Analyse par Composante Principale (PCA) permet de classer ces pixels en fonction de leur importance dans cet objectif de différenciation que nous nous sommes posés. Après divers essais, nous avons remarqué que réduire la dimensionnalité du problème, et donc le nombre de

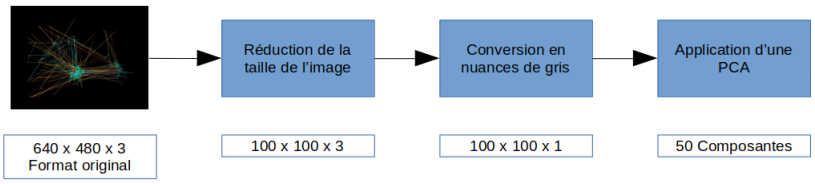


FIGURE 4.9 – Procédé de réduction d'une image de tracé oculaire en un ensemble de 50 valeurs les plus pertinentes.

valeurs traitées, à 50 permettait de convenablement classer des enfants selon le fait qu'ils souffrent de TSA ou non. Le procédé complet est résumé à la Figure 4.9. Ces résultats sont également inclus dans [Carette et al., 2019].

Ils sont exploités dans la Section 6.2 de ce manuscrit via l'utilisation de Réseaux de Neurones Artificiels, moins coûteux en espace mémoire et en temps de calcul que les Réseaux de Neurones Convolutifs, qu'il s'agisse de la phase d'apprentissage ou de la phase de validation et d'exploitation. En plus de ce gain en performance pour nos méthodes, la PCA permet de conserver la qualité des résultats de prédiction obtenus.

Chapitre 5

Données numériques et temporelles

5.1 Données d'événement

Les premières données utilisées sont les données des EventData présentées en Section 4.2. Par le fait qu'elles soient déjà pré-traitées par le pilote logiciel de l'Eye-Tracker, il nous a semblé évident de les exploiter, afin de réduire le travail sur les données à fournir.

Nous avons choisi de conserver l'ensemble du jeu de données généré pour chaque enfant. Aussi, nous avons dû normaliser ce jeu de données. Les informations fournies dans le cas des EventData étant variables selon la nature de l'événement observé, un choix devait être effectué entre ces événements du mouvement oculaire.

Les travaux en Psychologie montrent une variation importante des fréquences et intensités des saccades entre les enfants TSA (TS) et typiques (TC) [Kovarski et al., 2019]. Cette distinction importante entre les deux catégories d'enfants montre l'importance de ces événements oculaires dans l'optique de l'établissement d'un diagnostic, et par conséquent leur importance pour l'entraînement d'un modèle de Machine Learning visant à déterminer si un enfant souffre ou non de TSA. Par ailleurs, l'Eye-Tracker fournit plus d'informations décrivant les saccades que d'informations décrivant les fixations.

Ces deux raisons nous ont poussé à recueillir uniquement les données de saccades, en les réunissant par participant et par scénario visionné. Des différents fichiers EventData reçus, nous avons pu extraire le suivi de 28 enfants TC et 31 enfants TS uniques. Ces données sont traitées de la manière suivante : chaque suivi est constitué de tous les événements de type Saccade reconnus par l'Eye-Tracker. Chaque enregistrement de ce suivi inclut tous les éléments calculés par l'Eye-Tracker pour une saccade (durée en millisecondes, amplitude en degrés, accélération moyenne et maximale en degrés par seconde carrée, vitesse moyenne et maximale en degrés par seconde, décélération maximale en degrés par seconde carrée et taux de vitesse maximale en pourcents).

Nous avons établi une rapide analyse des données oculaires, desquelles découlent les états oculaires décrits dans la Section 5.2. Ces états oculaires décrivant de la dynamique du regard au cours du visionnage d'une vidéo, nous avons cherché à vérifier dans les données à notre disposition si les informations des deux yeux étaient absolument nécessaires, notamment pour des raisons d'optimisation liés à des contraintes matérielles. Nous avons donc utilisé la méthode de corrélation de Pearson sur les données disponibles. Nous avons ainsi analysé la corrélation pour chaque enfant entre les positions instantanées du regard sur l'écran de l'œil gauche et de l'œil droit. Ainsi, la position entre la position verticale de chaque œil est corrélée à 91.4% tandis que la position horizontale est corrélée à 89.9%. Ainsi, les informations concernant l'œil gauche permet de prédire assez fidèlement le comportement de l'œil droit, est inversement. Surtout, avec une corrélation aussi élevée, les catégorisations entre les états oculaires calculés par l'Eye-Tracker sont quasiment identiques, et la plupart des cas dissociés sont différenciés par un état de Blink, qui constitue un clignement, mais aussi un défaut de mesure ou une erreur. Ainsi, pour limiter la taille en entrée de notre modèle d'apprentissage, les données exploitées ne comprennent que l'information d'un seul œil. Ces données sont par ailleurs normalisées entre 0 et 1.

5.2 Détection automatisée du TSA par analyse de l'état oculaire via LSTM

Contexte

Dans les travaux présentés ici, nous utilisons un système d'Eye-Tracking. Le système considère trois états oculaires : fixation, saccade et clignement. Tandis que "fixation" et "saccade" représentent des données consécutives avec le focus oculaire restant dans une même zone pour le premier, avec le focus oculaire continuellement en mouvement pour le second, "clignement" est assez différent[Cilia et al., 2016]. Cet état indique la perte de localisation de l'œil, sans pouvoir savoir si le participant a effectivement cligné de l'œil ou seulement tourné la tête. Il a été remarqué que la dispersion oculaire d'un participant aide à l'indication de son état autistique.

Population

Les participants d'entraînement et de test sont âgés de 8 à 10 ans. Nous séparons leurs données en deux classes, nommément TSA (ou TS, classe 1) et non-TSA (ou TC, classe 2). Nous avons étudié les données d'un total de 17 enfants dans la classe 1 et de 15 enfants dans la classe 2.

Données et modèle

Les données sont réunies et utilisées anonymement. Le nom du participant est remplacé par une couleur. Pour être facilement utilisable avec un réseau de neurones LSTM (voir Section 3.1.2), les données sont normalisées de telle sorte que chaque valeur soit inclue dans [0,1]. Les données sont divisées en deux parties, données d'entraînement et de test (respectivement environ 75% et 25%). Les données de test comprennent 4 participants de la classe 1 et 3 participants de la classe 2.

Nous implémentons notre solution via le module Pybrain [1], utilisant les couches de neurones LSTM qu'il propose. En raison de l'aspect séquentiel de nos données, nous travaillons avec un réseau constitué de couches cachées de type LSTM. Ces couches sont au nombre de 2, comprenant 20 neurones

1. http://pybrain.org/ version 0.3

chacune. Ces quantités de neurones, déterminées empiriquement, sont suffisamment hautes pour permettre d'obtenir des résultats valides, tout en restant assez faibles pour être facilement entraînés avec une machine personnelle, de puissance limitée. Les couches d'entrée et de sortie sont des couches linéaires. Considérant les données à notre disposition, 7 neurones sont utilisés pour l'entrée, tandis que nous utilisons 3 neurones pour la sortie : classe 1, classe 2 et indéterminé. Indéterminé n'est pas utilisé par l'entraînement puisque la classe de chacun des participants du jeu d'entraînement est connue par avance, mais permet de révéler l'indécision du modèle, et semble améliorer la phase d'entraînement, autorisant le modèle à indiquer un doute dans le cas de données d'entrée inattendues. Un fonctionnement similaire, utilisant une seule sortie, a été proposé par [Campagner et al., 2019]. Par exemple, avec une borne de 0.2, le modèle répond "oui" entre 0.8 et 1.0, "non" entre 0.0 et 0.2 et "incertain" entre 0.2 et 0.8.

Approche

Ce réseau de neurones est entraîné en utilisant le jeu de données d'entraînement. Nous vérifions la précision du modèle sur un jeu de validation quand l'erreur de celui-ci baisse sous des paliers espacés de 0.001. La présentation des résultats se limitera aux paliers compris entre 0.008 et 0.004 inclus. Le jeu de validation est composé de 6 participants, non inclus dans les jeux d'entraînement et de test. Pour chaque palier et chaque participant, nous vérifions s'il est classé TC, TS, ou indéterminé. Aussi, pour chaque palier, nous calculons une valeur dite de *confiance*, calculée comme suit :

$$confiance = \frac{\sum_{i=1}^{6} diagnostic_i}{6}$$

avec $diagnostic_i$ la valeur prédite par le modèle pour le participant de validation i.

Résultats expérimentaux

Nous extrayons les états du réseau ayant une fitness comprise entre 0.008 et 0.004. Nous fournissons à ces états du réseau 6 nouveaux participants pour vérifier sa validité.

TABLE 5.1 – Résultats des six participants de validation. Pour chaque seuil de fitness atteint, la classe prédite (TS ou TC) par le modèle est indiquée, avec la probabilité prédite. Les "Enfants TS" ont pour classe attendue "TS", les "Enfants TC" ont pour classe attendue "TC".

	Enfants TS			Enfants TC		
Fitness	CyanTRIS	DarkBlue	DarkCyan	Cyan	CyanBIS	DarkRed
0.008	TS (88%)	TS (79%)	TS (94%)	TC (70%)	TS (92%)	TC (77%)
0.007	TS (88%)	TS (86%)	TS (96%)	TC (73%)	TS (93%)	TC (87%)
0.006	TS (89%)	TS (92%)	TS (98%)	TC (78%)	TS (94%)	TC (94%)
0.005	TS (92%)	TS (97%)	TS (56%)	TC (79%)	TS (95%)	TC (90%)
0.004	TS (95%)	TS (98%)	TC (89%)	TC (81%)	TS (95%)	TC (87%)

TABLE 5.2 – Confiance du modèle. Pour chaque seuil de fitness atteint, la confiance est calculée comme la moyenne des scores pour les classes attendues des six participants supplémentaires (scores TS pour les enfants TS et scores TC pour les enfants TC).

Fitness	Avg. Conf.
0.008	67.3%
0.007	66.3%
0.006	65.7%
0.005	60.1%
0.004	54.6%

TABLE 5.3 – Matrice de confusion pour les valeurs de fitness entre 0.008 et 0.004

	Prédiction TS	Prédiction TC
Enfant TS	3	0
Enfant TC	1	2

Les participants sont classés dans différentes classes apprises : tandis que DarkBlue, DarkCyan et CyanTRIS sont TS (TSA diagnostiqué, classe 1), Cyan, CyanBIS et DarkRed sont TC (enfants typiques avec le même âge chronologique sans TSA). Le calcul d'une probabilité pour les catégories TS et TC sont calculées de la manière suivante : pour un participant donné, notre modèle produit une valeur pour les classes 1 (TS), 2 (TC) et 3 (indéfini). Chacune de ces valeurs est comprise dans [0,1]. Ensuite, la probabilité d'une classe i peut être retrouvée en divisant la sortie i par la somme de toutes les sorties. Par exemple, si les résultats sont 0.2, 0.55 et 0.05, le participant est reconnu TS et TC à respectivement 25% et 68.75% et il existe une incertitude de 6.25%.

Pour rendre les résultats plus lisibles, seule la plus haute probabilité est montrée dans la Table 5.1. Aussi, la Table 5.2 indique la probabilité (confiance) moyenne concernant le résultat valide pour chaque participant, il ne s'agit donc pas forcément d'une moyenne des valeurs indiquées dans le tableau. En effet, dans le cas d'erreurs, la confiance moyenne peut être très faible. Enfin, la table 5.3 montre la matrice de confusion du modèle concernant les données de test.

Nous observons que, entre les valeurs de fitness 0.008 et 0.005, 5 des 6 participants voient leur diagnostic confirmé. Quand la fitness atteint 0.005 le réseau souffre de sur-apprentissage. En effet, nous observons que les données de test de DarkCyan donnent des résultats plus incertains à 0.005, puis des résultats erronés à 0.004. Nous remarquons que la plus haute confiance du réseau atteint 90% en moyenne, 98% au meilleur (pour une valeur de fitness à 0.006 et 5 résultats valides sur 6), tandis que le résultat erroné (CyanBis) est fort (94%). Aussi, nous notons, dans le meilleur cas, une sensibilité de 0.75 et une spécificité de 1.

5.3 Analyse dynamique du regard pour l'aide au diagnostic du TSA

Origine des données

Après avoir traité les données d'événement, nous utilisons le modèle numérique des données de dynamique (voir Section 4.3, partie *Format numérique*).

Ces données sont la représentation des vitesses, accélérations et à-coups, les trois dérivées de la position, sur l'écran, et les projetées des vecteurs le représentant sur les axes des abscisses et des ordonnées, soit un total de 9 valeurs par instant d'enregistrement.

Différence de cette approche

Contrairement à l'approche présentée dans la Section 5.2, ces travaux sont basés sur les RawData (voir Section 4.2). L'avantage de ces données consiste en l'absence de pré-traitements au-delà de la mesure initiale, pré-traitement qui est inhérent à la création des EventData. Ainsi, nous disposons de la donnée brute, et les modifications qui leurs sont apportées sont toutes en notre contrôle. Par ailleurs, ce jeu de données propose une quantité d'enfants, et de visionnage de scénarios par enfant, bien supérieur, passant de 32 enfants (plus 6 de validation) sur une vidéo à 59 enfants avec en moyenne 15.19 vidéos visionnées.

Expérimentation

Notre modèle comprend trois couches de convolution à une dimension consécutives puis une couche de Pooling à une dimension. Ensuite, deux couches d'ANN sont utilisées, puis une dernière couche, de sortie, de taille 2. Les couches de convolution comprennent une fenêtre de taille 20, et, dans l'ordre, 16, 32 et 64 filtres de sortie. Ces couches sont activées par une fonction ReLU. Les couches ANN sont de taille 128 et 64 avec une activation sigmoïde. La couche de sortie est activée par une fonction softmax, qui permet de normaliser l'ensemble des sorties (ici nos deux sorties) de telle sorte que la somme des valeurs de cette couche soit égale à 1 et que chaque valeur soit comprise entre 0 et 1.

Nous avons entraîné ce modèle par cross-validation (voir Section 3.1.1 partie *Validation et utilisation d'un modèle*). Cette cross-validation est établie en 10 parties, séparant les enfants de manière indifférenciée en 10 jeux de validation de taille équivalente, notés de V_1 à V_{10}. 10 modèles M_1 à M_{10} indépendants, de topologies équivalentes sont entraînés. Pour tout modèle M_i, l'entraînement est effectué sur l'ensemble des données disponibles en excluant V_i. La précision du modèle est alors vérifiée avec les données de V_i. Nous répétons ce procédé de cross-validation 10 fois, pour réduire le caractère aléatoire

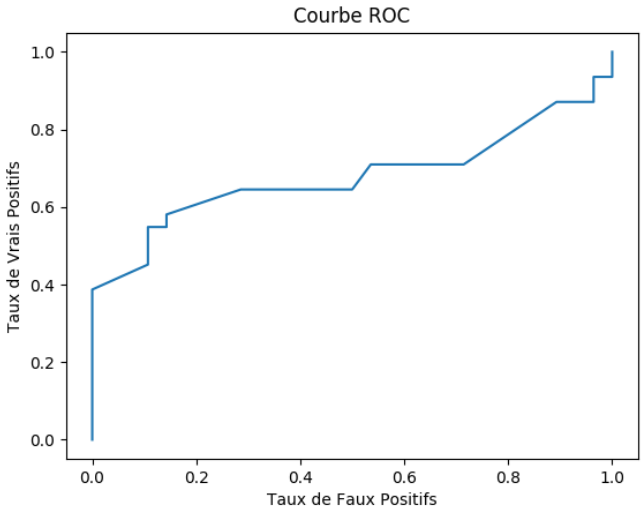


FIGURE 5.1 – Courbe ROC obtenue par apprentissage sur les données de dynamique.

de la séparation en 10 parties. La précision du modèle est calculée selon le taux de réponse juste des modèles sur leurs jeux de validation respectifs. Le moyenne des taux obtenus lors des 10 répétitions de la cross-validation sert alors de valeur de précision globale de notre modèle.

Un second traitement a été appliqué, visant à préciser la granularité de la séparation de la cross-validation. Nous suivons alors le principe du leave-one-out, et entraînons un modèle par enfant. Ainsi, pour chaque enfant, nous isolons les données le concernant et entraînons un modèle sur le reste des données disponibles. La capacité de réponse du modèle sur l'enfant nous donne l'efficacité du modèle. Nous obtenons les réponses du modèle, dont le nombre dépend du nombre de vidéos regardées par l'enfant, et la moyenne de ces réponses donne un score permettant de classer cet enfant dans une catégorie d'enfant ou l'autre.

Résultats

Les résultats de la cross-validation sont plutôt bons, avec une précision aux alentours de 85%, dans la tranche haute des résultats montrés dans la

Section 3.6 compris entre 75 et 88%. Cependant, un biais est envisageable, puisque pour un enfant donné, une séparation aléatoire ne peut pas garantir que l'enfant ne soit pas présent à la fois dans le jeu d'entraînement et dans le jeu de validation, ce qui est plus incertain quand l'enfant a été suivi pour le visionnage de nombreux scenarii.

L'entraînement utilisant le leave-one-out montre en effet ce biais. Afin de mieux de représenter les résultats, ici non binaires, mais basés sur un score moyen par enfant, nous exploitons la courbe ROC de ce modèle. Cet entraînement donne des résultats moyens, avec une AUC d'environ 0.68. La courbe présentée en Figure 5.1 présente la courbe ROC des résultats. Nous voyons notamment qu'environ 40% des enfants TSA sont toujours correctement diagnostiqués et environ 5% des enfants TSA échappent systématiquement à la détection. Il est aussi visible que le modèle a des difficultés à reconnaître les enfants TSA dans la globalité.

Ainsi, l'étude basée sur la représentation numérique de la dynamique montre quelques défauts. Au mieux, ces résultats sont dans l'intervalle de précision de travaux de la littérature. Cette approche montre ainsi des résultats acceptables, qui pourraient être améliorés par l'utilisation d'une représentation graphique de ces données, représentation *a priori* plus intuitive de celles-ci.

Conclusions

Nous avons établi qu'un réseau de neurones peut différencier le statut autistique de jeunes participants grâce à l'utilisation des mouvements oculaires de l'enfant en se basant sur des données numériques issues du suivi oculaire.

Les EventData posent une limitation en raison de son prétraitement, et de la faible quantité de données disponibles. Les RawData, quant à elles, peuvent être manipulées à souhait, et nous permettent de conserver le contrôle sur les opérations effectuées.

La précision des deux modèles sont concluantes. L'utilisation des EventData est limitée à la confiance en les méthodes de création des valeurs d'événement par l'Eye-Tracker. Le modèle utilisant les RawData est meilleur sur les données non triées par enfant. Cela indique que l'entraînement sans trier

les données par participant induit un biais de l'entraînement. Les résultats sur les données issues d'un tri par participant restent corrects mais ne suffiront pas à établir un diagnostic efficace du TSA.

Il est nécessaire d'envisager des traitement plus avancés des données EventData ou RawData. Les RawData, données brutes, seront privilégiées pour cela.

Les travaux sur les EventData ont fait l'objet d'une présentation à la conférence HealthyIoT 2017, suivie d'une publication [Carette et al., 2017].

Chapitre 6

Données graphiques

6.1 Détection du TSA via analyse graphique du tracé oculaire par CNN

Contexte

Nous travaillons ici avec une représentation des données sous forme d'image. Les images générées contiennent des informations de qualité similaire à ce qu'un expert humain pourrait consulter via les vidéos proposées par l'Eye-Tracker. Les images utilisées et la méthode de génération de celles-ci est présentée dans la Section 4.3.

Données et modèle

Afin d'exploiter ces données sous forme d'images, l'utilisation d'un modèle de réseaux de neurones tels qu'un ANN ou un RNN n'est pas appropriée et il nous est nécessaire de concevoir un modèle de type CNN, capable de traiter des données de type image.

Les 547 images fournies par notre processus de transformation sont de taille 640 par 480 pixels. 328 de ces images proviennent de suivi d'enfants TC et 219 de suivis d'enfants TS. Les techniques d'augmentation d'images proposées par Keras nous permettent d'augmenter significativement ces nombres. Notamment, les augmentations incluent le zoom, la rotation, le miroitement

(liste complète dans la documentation de Keras [1]).

Le module Keras inclut un ensemble de méthodes permettant de faciliter le traitement d'images. Il est simplement demandé d'indiquer les localisations des dossiers d'images d'entraînement et de validation, chaque classe ayant son sous-dossier. Keras peut alors appliquer les modifications d'images choisies de manière aléatoire pour atteindre le nombre d'images souhaitées.

Après ces transformations supplémentaires, nous concevons notre CNN. Il est composé de trois couches de convolution successives de fenêtres (5,5), chacune suivie d'une couche de MaxPooling de taille (2,2). Après le dernier pooling, les données sont fournies à des couches simples et enfin une couche à 2 neurones permettant la classification entre les deux catégories (TS et TC). Toutes les couches sont activées avec la fonction ReLU, à l'exception des couches de pooling (sans fonction d'activation) et de la couche de sortie, qui utilise la fonction Softmax, permettant de normaliser les sorties et de proposer une information pouvant être interprétée comme une probabilité.

Entraînement du modèle et premiers résultats

A chaque étape d'entraînement, nous générons 1000 images aléatoirement à partir du jeu d'images initial. L'entraînement se déroule sur 250 étapes successives. Chaque étape est complétée par une validation sur 50 images. Les images d'entraînement doivent être les plus variées possibles et notre jeu d'entraînement est restreint, ainsi elles sont augmentées en utilisant les techniques proposées par Keras. Ce n'est pas le cas des images de validation pour lesquelles nous cherchons simplement à savoir si le modèle fonctionne sur des images complètement nouvelles. Une variation de ces images serait donc inutile car elle entraînerait une redondance. Ces images de validation sont exclues de l'ensemble servant à la génération. Afin d'assurer que le caractère aléatoire ne mène pas à un cas d'entraînement fortuitement avantageux, ce procédé est reproduit 50 fois.

50 modèles ont été entraînés et présentent des résultats variables, avec en moyenne environ 65% de précision. La précision sur les 50 images de

1. https ://keras.io/preprocessing/image/

validation est calculée selon la formule suivante :

$$précision = \frac{TS_{TS} + TC_{TC}}{50}$$

avec TS_{TS} le nombre d'enfants autistes correctement classés et TC_{TC} le nombre d'enfants non autistes correctement classés. Dans le meilleur des 50 modèles, nous atteignons une précision de 88%, avec 5 enfants autistes et 1 non-autiste mal classés.

Cependant, dans notre cas, les images pourraient être plus clairement dissociées. En effet, bien que les images du jeu de validation soient absentes du jeu d'entraînement, les images d'un même enfant peuvent être réparties entre les deux jeux, ce qui inclut alors un biais, l'entraînement et la validation pouvant être considérés comme étant mêlés.

Entraînement "par participant"

Partant de cette observation concernant la séparation des images, nous avons abordé le problème différemment, selon le principe du *leave-one-out* [Arlot et al., 2010]. Pour chaque participant p, nous créons un modèle CNN à entraîner. Ce modèle est entraîné en utilisant tous les autres participants et est ensuite validé sur les images représentant p. Ainsi, nous obtenons un total de 59 modèles, dont les résultats respectifs moyens permettent d'estimer la précision pour chacun des enfants, et la précision du modèle générale. Pour éviter le cas éventuel d'un hasard avantageux, les modèles sont entraînés 10 fois consécutives, soit un total de 590 entraînements et scores de validation.

Pour chaque modèle, après avoir été entraîné, l'ensemble des images de validation est fourni afin d'obtenir une valeur d'appartenance aux classes TC et TS. Chaque image contenant la même quantité d'information de par le procédé de création utilisé, la moyenne de ces valeurs fournies par le modèle est utilisée pour permettre un classement du participant entre enfant souffrant d'autisme et enfant typique. Ces scores sont résumés sous la forme d'un nuage de points dans la Figure 6.1. Le score correspond à la valeur obtenue pour la classe TS multiplié par 100, pour une meilleure lecture. Pour une image d'entrée I :

$$\begin{aligned} score_I &= sortieTS_I * 100 \\ &= (1 - sortieTC_I) * 100 \end{aligned}$$

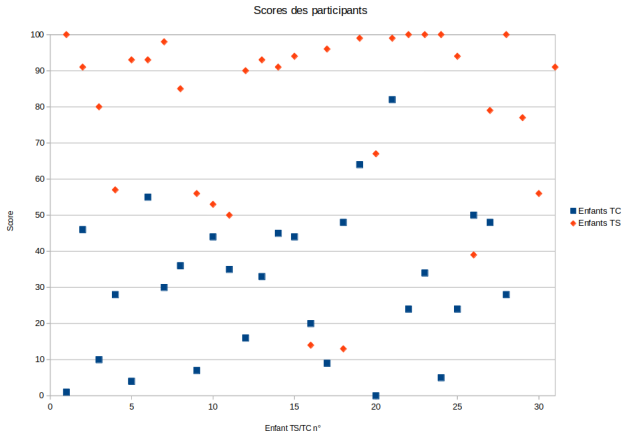


TABLE 6.1 – Scores obtenus par le modèle de chaque participant. Les enfants TC sont représentés en bleu et les enfants TS en rouge.

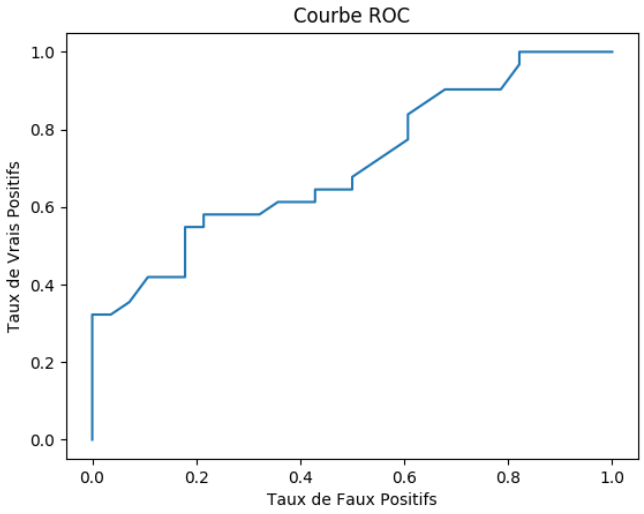


FIGURE 6.1 – Courbe ROC des modèles appris par CNN.

avec $sortieTS_I$ et $sortieTC_I$ les sorties du modèle respectives pour la classe TS et la classe TC, avec l'image I en entrée.

Le score d'un participant correspond à la moyenne des scores des images le représentant. Un participant TS doit tendre vers un score de 100 tandis qu'un participant TC doit tendre vers 0.

Résultats

La valeur de score permettant de dissocier les enfants TS et TC, si elle est fixée à 50, n'est pas la plus adaptée à notre problème. Il en va de même pour tout seuil arbitrairement fixé. De ce fait, nous validons notre modèle en nous servant d'une courbe ROC (Receiver Operating Characteristic, ou Caractéristique de Fonctionnement du Récepteur). Cette courbe permet de dessiner la précision d'un modèle en utilisant toute valeur de seuil (entre 0 et 100 dans notre cas).

Pour chaque valeur s possible de seuil, nous plaçons un point de la courbe. Sa valeur en abscisse correspond au taux de Faux Positifs induit par le seuil s, et sa valeur en ordonnée correspond au taux de Vrais Positifs induit par le seuil s. Ici, nous considérons comme cas Positif le cas d'un enfant TS.

L'ensemble des points permet le tracé de la courbe ROC, et l'aire entre celle-ci et l'axe des abscisses (AUC ou Area Under the Curve) indique la précision générale du modèle. Le modèle parfait a une AUC de 1.0 et plus un modèle tend vers cette valeur plus il est proche de l'optimal. La Figure 6.1 présente la courbe ROC obtenue. Nous utilisons les scores (Table 6.1) pour définir les classes obtenues.

L'AUC de la courbe, d'environ 0.71, montre un résultat assez moyen, mais incite à d'autres essais, utilisant d'autres méthodes, notamment en recherchant avec précision les éléments de l'image à traiter.

Comparaison avec des experts

Les données sous forme d'images générées sont proches des affichages visuels que peuvent proposer le pilote de l'Eye-Tracker. Ainsi, nous avons sélectionné 50 images de tracés oculaire de notre ensemble (25 TS et 25 TC).

Ces images ont été fournies à 5 psychologues experts de l'autisme, qui ont dû reconnaître l'état autistique des enfants correspondant à ces images.

Les seuls choix possibles sont TS et TC. Le choix indécis a été exclu. Le principe du vote a été appliqué à leurs avis. Tout enfant présentant au moins trois avis TS est classé TS, sinon il est classé TC. Les experts ont reconnu correctement 16 enfants TS et 17 enfants TC, soit une précision de 66%, ce qui place notre modèle au niveau de l'expertise humaine, en moyenne. Notre modèle avait obtenu une précision d'environ 85%, soit notre meilleur résultat, en prenant ces images comme jeu de validation.

Cette précision inférieure reste à relativiser, les experts ayant en général d'autres informations à disposition. Aussi, notre représentation est similaire à ce que peuvent proposer les pilotes des outils d'Eye-Tracking, mais montre quelques différences qui peuvent induire un biais.

6.2 Utilisation d'images réduites au format numérique pour la détection du TSA via ANN

Contexte

Comme expliqué dans la Section 4.4, les images contiennent trop d'éléments inutiles (tous les pixels noirs) qu'un CNN n'a pas besoin d'utiliser pour son apprentissage. Le modèle est donc complexe alors que des techniques de réduction de la complexité des données permettrait de réduire également la complexité du modèle à apprendre, et par la même occasion augmenter l'efficacité en temps et en performance de l'apprentissage de celui-ci.

Données et modèle

Afin de réduire la quantité d'information et de n'exploiter que de l'information pertinente, nous utilisons les données réduites proposées dans la Section 4.4.

Nb neurones	Score d'AUC
50	0.8985
100	0.9124
150	0.9166
200	**0.9171**
250	0.9207
300	0.9220
350	0.9226
400	0.9250
450	0.9256
500	0.9264
50,25	0.9007
100,50	0.9160
150,75	0.9197
200,100	0.9202
250,125	0.9212
300,150	0.9219
350,175	0.9231
400,200	0.9243
450,225	0.9259
500,250	0.9270

TABLE 6.2 – Scores d'AUC obtenus pour chaque taille de couche cachée testée. La première partie concerne les scores pour les réseaux à une couche cachée. La seconde partie indique les résultats des réseaux à deux couches cachées, les quantités sont indiquées au format*taille_couche_1*,*taille_couche_2*.

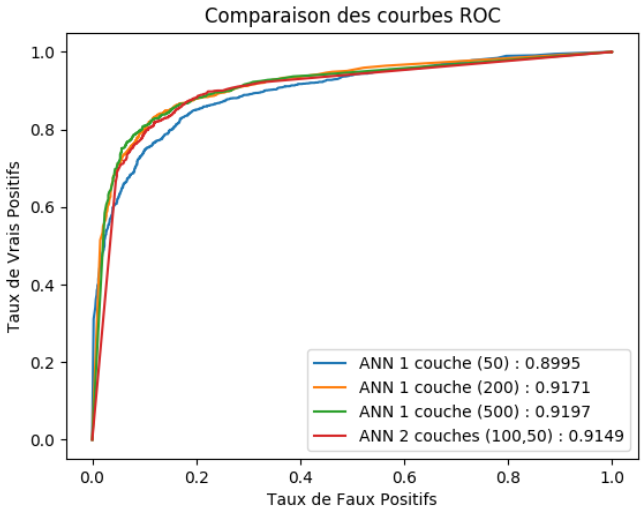


FIGURE 6.2 – Courbe ROC de chaque modèle appris via les données réduites.

La simplicité de ce nouveau modèle de données permet d'envisager l'utilisation d'un modèle d'apprentissage lui-aussi simplifié. En effet, un simple réseau de neurones peut aisément traiter ce genre de données d'entrée. Nos essais se sont donc concentrés sur des ANN à une seule couche cachée. Nous entraînons ce modèle en lui attribuant 50, 200 et 500 neurones sur la couche cachée. Nous avons tenté des approches avec 50 neurones minimum (taille du jeu d'entrée) et avons observé que les résultats tendaient à ne plus s'améliorer sensiblement dès 200 neurones, la tendance semblant atteinte dès 500 neurones. Nous avons également effectué un essai avec un modèle à deux couches cachées avec différentes combinaisons de tailles des couches cachées, de 50 à 500 neurones pour la première, 25 à 250 pour la seconde. La Table 6.2 présente les scores d'AUC obtenus.

Expérimentation

L'entraînement est effectué suivant le principe de la *10-cross validation* (ou validation croisée [Kohavi et al., 1995]). Le jeu de données est aléatoirement réparti en 10 sous-jeux de données disjoints. Pour chacun de ces sous-jeux de données v_i, le reste du jeu de données initial est noté t_i. Nous entraînons 10 modèles et pour chaque modèle i, l'entraînement est effectué sur t_i et sa

validation obtenue en utilisant les données de v_i.

Nous prenons l'ensemble des résultats de validation des v_i pour obtenir un résultat global. La Figure 6.2 présente les courbes ROC de différentes tailles de réseau de neurones, tailles choisies comme étant représentatives de nos essais, et indique en légende la valeur de l'AUC pour chacun.

Résultats

Des essais ont été conduits avec des approches plus simples telles que de la Régression Logistique, des Machines à Vecteur de Support (SVM) ou des Forêts Aléatoires. Ces essais ont mené à des résultats bien moindres avec des AUC allant de 0.67 à 0.77. Dans le cas des ANN, une seule couche cachée est suffisante, notamment si nous comparons l'efficacité des modèles et leurs temps d'entraînement respectifs. En guise de compromis entre le temps et la précision, le modèle à une couche cachée et 200 neurones sur celle-ci est le plus efficace atteignant une AUC de 0.9171. Ces travaux ont fait l'objet d'une publication [Carette et al., 2019]

Conclusions

Nous avons montré qu'il est possible d'identifier l'état autistique d'un enfant en visualisant le tracé de son regard pendant le visionnage d'une vidéo, avec une précision de 88%. Il s'avère que des psychologues, experts du TSA, montrent des résultats inférieurs à notre modèle sur un jeu à valider commun. L'approche utilisant les données réduites par PCA montrent des résultats très concluants également, avec 92% d'AUC.

Classer les enfants est cependant, et logiquement, plus difficile quand nous séparons le jeu de données de telle sorte que les images représentant un même enfant ne puissent être réparties entre les jeux d'entraînement et de validation. La baisse de diversité, ou le retrait d'un biais, fait chuter les résultats.

Les résultats obtenus restent cependant prometteurs. Il est nécessaire de trouver une meilleure représentation des données disponibles pour permettre une amélioration des résultats. Nous devons également chercher d'autres moyens d'exploiter les données issues de l'Eye-Tracking pour trouver une

approche permettant de combler les erreurs que montrent ces modèles. Une étude statistique pourrait notamment identifier les éléments importants d'une image ou les valeurs mesurées par Eye-Tracking les plus pertinentes.

Chapitre 7

Données statistiques

7.1 Approche statistique

Motivations

Les approches précédentes ont montré des failles, notamment l'exclusion de nombre de données fournies par l'Eye-Tracker dans les RawData. En effet, la représentation du tracé oculaire sous forme d'images, et ses transformations ultérieures, ne se sont concentrées que sur les points de regard du suivi. Cependant, inclure toutes les informations fournies reste difficile à mettre en œuvre en raison de la masse d'informations à réunir, ce qui inclut alors des problèmes de vitesse de traitement notables.

Partant de ce fait, et des différences de dynamisme et de réactions aux stimuli soumis aux participants selon qu'ils soient atteints de TSA ou non, nous avons émis une hypothèse selon laquelle la corrélation entre les variables mesurées différait en fonction de l'état du participant. Il a alors été choisi de générer des matrices de corrélation sur un ensemble des variables disponibles.

Données

Nous avons donc sélectionné des données en suivant un certain nombre de critères. Premièrement, nous avons limité les données à exploiter aux données numériques. Ensuite, nous avons exclu les données faiblement représentées dans l'ensemble des données RawData, ces fichiers étant partiellement hétérogènes. En résultat de ce tri de variables, 24 d'entre-elles ont été conservées :

— le temps écoulé depuis le début de la captation.
— la position de l'œil dans l'espace (en x, y et z, pour chaque œil, en millimètres)
— les vecteurs de regard (en x, y et z, pour chaque œil, sans unité)
— le point de regard (en x et y, pour chaque œil, en pixels)
— la position de la pupille sur l'écran (en x et y, pour chaque œil, en pixels)
— la vélocité du regard (et ses projections sur les axes x et y)

Concernant l'intérêt d'ajouter le temps écoulé, des travaux ont montré des différences notables de capacité à maintenir une concentration sur des vidéos en fonction du fait qu'un enfant soit atteint de TSA ou non [Vargas-Cuentas et al., 2016].

Nous avons choisi d'utiliser les trois méthodes de calcul de la corrélation existantes, c'est-à-dire Pearson, Kendall et Spearman (voir Section 3.2.1). Ces méthodes sont proposées par le module Pandas [1] de Python [2], permettant une gestion simplifiée des calculs à effectuer.

Jeu de données

Pour chaque suivi vidéo, nous calculons la corrélation entre les 24 variables selon les méthodes de Pearson, Kendall et Spearman. Chaque méthode permet de calculer 276 valeurs de corrélation. Chaque suivi vidéo est donc représenté par 828 valeurs.

Une fois l'ensemble des suivis vidéo convertis en valeurs de corrélation, certains suivis sont exclus. En effet, certaines variables font défaut lors de certains suivis, influant sur les résultats de corrélation obtenus. Nous les avons exclus afin d'éviter le risque d'induire de l'erreur dans l'entraînement. Après ce tri, les corrélations de 896 suivis vidéos sont conservés.

La Figure 7.1 montre la représentation en nuances de gris de la corrélation d'un suivi vidéo. L'ordre des labels, en abscisses (de gauche à droite) comme en ordonnées (de haut en bas) est identique à la liste détaillée dans la partie *Données*.

1. https ://pandas.pydata.org, version 0.23.4
2. https ://www.python.org/, version 2.7.13

FIGURE 7.1 – Représentation d'une matrice de corrélation d'un enfant TC. Les corrélations de Pearson, Kendall et Spearman sont représentées respectivement sur les canaux rouge, vert et bleu. Chaque canal prend ses valeurs entre 0 (pour une corrélation nulle) et 1 (pour une corrélation maximale, en valeur absolue).

Chaque enfant est représenté sur plus d'un suivi. Le nombre de suivi par enfant est compris entre 2 et 88 inclus (moyenne 15.19, écart-type 20.23). Le nombre total d'enfants est de 59, dont 31 atteints de TSA.

7.2 Application pour la détection du Trouble du Spectre de l'Autisme

Contexte

Les données obtenues par analyse statistique des RawData (voir Section 7.1) permettent une amélioration des traitement d'apprentissage automatisé, similairement à la réduction dimensionnelle présentée dans la Section 6.2. En effet, chaque suivi oculaire d'un enfant est réduit à 828 valeurs, permettant un traitement accéléré et plus léger, qu'il s'agisse de l'entraînement du modèle ou de sa validation et de son application en conditions réelles.

Modèle et protocole expérimental

A partir des données de corrélation, nous entraînons un modèle sous la forme d'un réseau de neurones artificiel. Le modèle est formé d'une entrée de taille 828, puis de 2 couches de tailles respectives 128 et 64 utilisant la fonction d'activation ReLu, complétées par une couche composées d'un simple neurone avec la fonction sigmoïde comme fonction d'activation. La valeur attendue de cet unique neurone de sortie du réseau est de 0 pour les enfants TC et de 1 pour les enfants TS.

Nous avons entraîné notre modèle selon la technique du *leave-one-out* [Arlot et al., 2010], qui demande à extraire un élément de l'ensemble d'entraînement pour validation. Dans notre cas, retirer un seul élément pourrait entraîner un biais d'entraînement puisque chaque enfant est représenté plusieurs fois. De ce fait, nous avons choisi de retirer toutes les données d'un enfant pour chaque essai. Nous avons donc obtenu 59 modèles, que nous avons évalué, selon qu'il soit validé par les données d'un enfant TC ou d'un enfant TS. Chacun des essais a été reproduit 10 fois, afin de pallier les risques de biais liés à l'aléatoire.

Class	Image	Numerical	Correlation
TC	0.96	**1.00**	0.79
TS	0.32	0.35	**0.87**
Total	0.63	0.67	**0.83**

TABLE 7.1 – Taux de bons résultats de classification entre les enfants TC et TS selon les méthodes à base d'images, de données numériques, et des valeurs de corrélation (avec utilisation d'un seuil à 0.5).

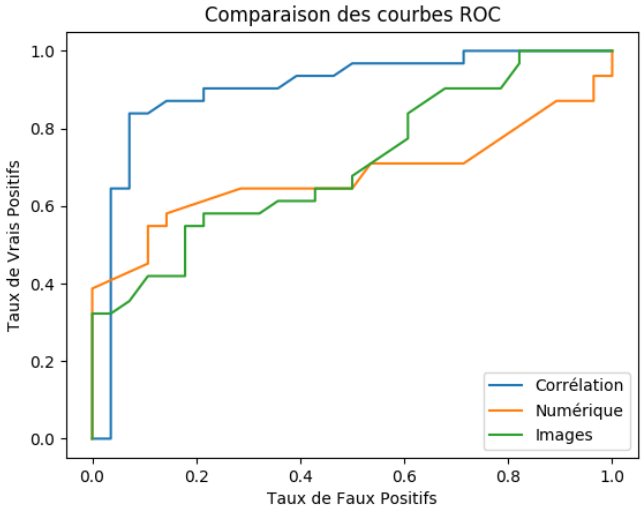


FIGURE 7.2 – Tracé des courbes ROC pour les approches Image, Numérique et Statistique.

Résultats

Après entraînement, le résultat de validation moyen des 10 modèles liés à l'enfant prend sa valeur entre 0 et 1, nous visons alors 0 pour les enfants TC et 1 pour les enfants TS. Nous commençons par lire naïvement les résultats en posant un seuil de discernement entre les deux classes à 0.5. Chaque enfant sous ce seuil sera diagnostiqué TC, un enfant au-dessus sera diagnostiqué TS. Les résultats de ce classement naïf est présenté en Table 7.1.

Ces résultats basés sur un seuil arbitraire sont difficilement interprétables. Or, l'utilisation d'une courbe ROC permet un affichage plus clair et sans le besoin de poser arbitrairement un seuil, qui pourrait ne pas avoir d'explication logique. La Figure 7.2 présente cette courbe, comparée aux courbes ROC obtenues pour certains des travaux précédents : ANN utilisant les données numériques de la dynamique du regard (voir Section 5.3) et CNN avec des images représentant le tracé oculaire et sa dynamique (voir Section 6.1). La supériorité de la méthode utilisant les corrélation y est visible, avec une AUC de 0.901. Par ailleurs, la Table 7.1 permet également de comparer les taux de diagnostics corrects en posant un seuil de 0.5 arbitraire. Il peut y être noté qu'au détriment d'un taux d'erreur légèrement accru dans la détection des enfants non atteints par le TSA (environ 20% d'erreurs), le taux de validation des enfants TSA prend des valeurs bien plus élevées (87%, soit plus de 2 fois les taux des deux autres approches). Enfin, la précision globale, de 83% pour un seuil arbitraire, est très satisfaisante.

Conclusions

Les résultats de ces travaux montrent l'intérêt d'une étude statistique des variables mesurées par l'Eye-Tracker pour l'aide au diagnostic du TSA. Ces résultats dépassent notamment ceux de nos approches passées, concentrées sur l'analyse d'images.

Comme pour les approches précédentes, nos résultats ne sont pas parfaits. Il serait intéressant de constater les avantages et inconvénients de nos approches et de tenter de les combiner pour améliorer les performances.

Chapitre 8

Hybriation des approches

8.1 Motivations

Différentes observations de nos résultats nous ont montré que les enfants mal diagnostiqués ne sont pas les mêmes selon les approches utilisées. De ce fait, nous avons pensé à mettre en commun les résultats de validation de ces modèles pour permettre un meilleur diagnostic du Trouble du Spectre de l'Autisme.

Nous avons exploité les résultats de certaines de nos approches : l'approche CNN basée sur des Images (Section 4.3, notée Image), l'approche CNN basée sur les données numériques de la dynamique (Section 5.3, notée Numérique) et l'approche ANN basée sur les données statistiques du tracé oculaire (Section 7.2, notée Statistique). Ces approches sont issues de traitements légers des données brutes : l'analyse graphique, numérique et statistique des RawData.

Nous avons exclu les résultats issus de l'analyse des événements oculaires (Section 5.2) en raison de la faible intersection entre les enfants issus des EventData utilisés et ceux provenant des RawData, utilisés dans toutes les autres approches. Aussi, les résultats basés sur les traitements via PCA (Section 6.2) sont exclus, étant un second traitement de données déjà altérées, la conception d'une image sous forme de pixels forçant une première approximation des tracés oculaires.

Nous avons testé différentes manières de comparer nos résultats afin d'atteindre un modèle global efficace. Pour commencer, nous étudions le principe du vote, qui consiste à choisir la classe d'enfant selon la classe la plus prédite parmi nos approches. Ensuite nous établissons la classe d'un participant selon la moyenne des résultats obtenus, puis en pondérant cette moyenne.

8.2 Méthodes utilisées et résultats

Données

Les données de travail sont l'ensemble des réponses des trois modèles. Comme montré dans de précédentes Sections, les valeurs sont comprises entre 0 et 100, valeur de la prédiction de la classe TS pour chaque enfant, multipliée par 100. Un enfant TS doit tendre vers 100 tandis qu'un enfant TC doit tendre vers 0. Dans cette Section, toute référence à un enfant pourra être retrouvée dans l'Annexe. Les trois modèles seront ici nommés "sous-approches".

Dans cette Section, nous avons tenté d'utiliser différentes approches de combinaison des résultats obtenus : Le vote, la moyenne et la moyenne pondérée.

Exception

Les données d'un enfant, noté TS18 par la suite, donne des résultats erronés dans chacune de nos approches. En effet, il obtient un score de 23, 26 et 13 avec Numérique, Image et Statistique respectivement, contre un score attendu de 100. Les Psychologues experts sur l'autisme nous ont informé que cet enfant est diagnostiqué d'un Trouble Envahissant du Développement (TED), et non TSA.

Par conséquent, les travaux de cette section ont été conduits avec et sans les données de TS18 pour constater leur impact sur les résultats. Les résultats excluant cet enfant ont été traités à part pour éviter la confusion (voir Section 8.3).

Approches

Approche du vote La première solution étudiée est de choisir la classe faisant consensus entre les sous-approches. Cette solution permet d'avoir une réponse claire du modèle. L'enfant est ou n'est pas atteint de TSA, il n'existe pas de valeur intermédiaire. Cette absence de nuance est cependant problématique dans le cas de résultats initiaux indécis.

Cette approche permet de reconnaître l'ensemble des 28 enfants non autistes dans la bonne classe. Dans le cas des enfants TS, seuls 14 sur 31 sont correctement classés. Nous obtenons une précision globale du modèle d'environ 71%, ce qui est assez médiocre.

Approche de la moyenne Pour cette approche, nous avons tenté de réduire la valeur de chaque enfant à la moyenne des trois sous-approches considérées. Ici, la nuance est possible, mais cela pose un autre problème dans le cas ou l'une des sous-approches est très fortement erronée. Dans ces cas, le modèle peut pencher dans le sens d'une classe erronée si les autres sous-approches ne permettent pas de contrebalancer cette erreur.

Ces réponses nuancées permettent la vérification des performances via une courbe ROC. Pour cette approche, nous obtenons une AUC de 0.906, ce qui est légèrement supérieur aux résultats de la sous-approche Statistique (présentée dans la Section 7.2).

Approche de la moyenne pondérée Une moyenne appliquée à l'ensemble de nos résultats semble fournir des résultats intéressants. De ce fait, nous cherchons à pondérer les résultats de nos trois sous-approches. Des triplets de poids proportionnels ayant un effet équivalent sur le calcul d'une moyenne pondérée, nous pouvons les normaliser et donc considérer chaque poids comme étant compris entre -1 et 1. Nous cherchons donc les poids avec des valeurs comprises entre -1 et 1, ce qui peut être représenté par un cube d'arête 2, centré sur l'origine d'un repère orthonormé.

Afin de rechercher la meilleure valeur possible d'AUC (à 0.001 près), nous étudions l'ensemble des triplets possibles, avec un pas de variation des poids de 0.025. La représentation de cette étude est visible dans la Figure 8.1. Dans

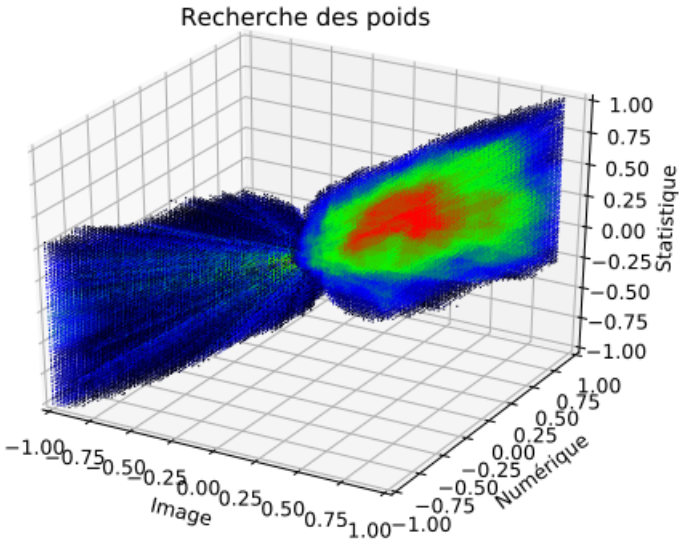


FIGURE 8.1 – Représentation de la précision des poids selon la validation par AUC.

Sous-Approche	Poids
Image	0.925
Numérique	-0.675
Statistique	1

TABLE 8.1 – Meilleure combinaison de poids pour les trois sous-approches.

cette représentation, nous ne représentons, pour des raisons de lisibilité, que les combinaisons de poids dont la valeur d'AUC dépasse 0.90. Un point prend sa couleur en fonction de son AUC, selon un spectre de couleur spécifique. A 0.90, la couleur est noire, à 0.91 bleue, à 0.92, verte et à 0.93 (et au-dessus), rouge. Toute valeur intermédiaire prend une couleur selon un dégradé linéaire entre les deux bornes les plus proches.

De cette étude, nous extrayons les valeurs optimales (voir Table 8.1). La valeur d'AUC obtenue est de 0.933, ce qui représente un pas de progression plus pertinent que dans le cas de la moyenne simple.

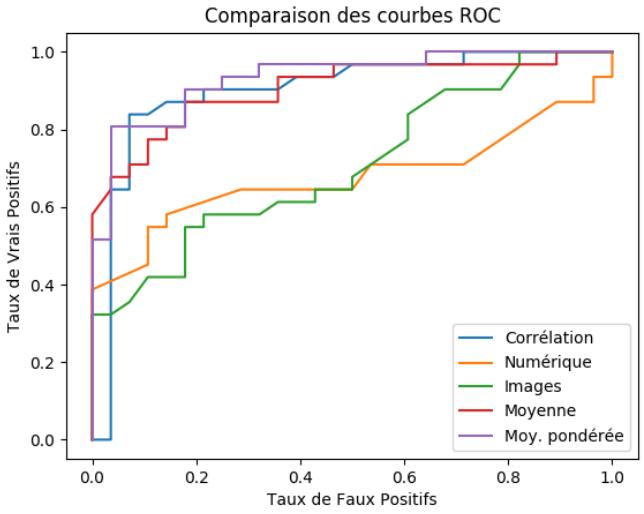


FIGURE 8.2 – Représentation de la précision des poids selon la validation par AUC.

Observations

Aussi, nous observons que le poids de l'approche Numérique est négatif. Cela étant, nous avons essayé de rechercher les meilleurs poids pour toutes les combinaisons de nos sous-approches, en n'en sélectionnant que deux sur trois, mais aucune de ces combinaisons n'a atteint les performances en termes d'AUC de ce modèle à trois sous-approches.

La représentation des courbes ROC de nos trois sous-approches, de l'approche moyennée et de l'approche moyennée pondérée est disponible dans la Figure 8.2.

8.3 Données aberrantes

Motivations

Une analyse rapide sur les valeurs des résultats de nos sous-approches a montré un important défaut. Un des enfants, TS 18, est mal classé, pour toutes les sous-approches, avec des scores de 13 à 26 (un enfant TS doit s'approcher de 100), et donc toutes les approches d'hybridation.

Sous-Approche	Poids
Image	0.915
Numérique	-0.661
Statistique	1

TABLE 8.2 – Meilleure combinaison de poids pour les trois sous-approches en excluant le participant TS 18.

Modèle	AUC avec TS18	AUC sans TS18	Variation
Image	0.709	0.721	0.021
Numérique	0.678	0.701	0.023
Statistique	0.901	0.922	0.021
Moyenne	0.906	0.932	0.024
M. Pondérée	**0.933**	**0.952**	**0.019**

TABLE 8.3 – Evolution de l'AUC en fonction de la présence de TS18.

Or, TS18 est dans une condition "atypique" de TSA, en effet, les Psychologues experts en TSA nous ont indiqué que cet enfant est atteint de Trouble Envahissant du Développement (TED). Nos modèles basés sur l'Eye-Tracking l'ont tous exclu des enfants TSA, ce qui montre l'efficacité de ces modèles à reconnaître spécifiquement les cas "typiques" de TSA. L'efficacité des résultats est évaluée en excluant cet enfant.

Méthodes

Les procédés sont strictement identiques à ceux exploités dans la Section précédente. Les poids de la moyenne pondérée varient très légèrement (voir la Table 8.2).

Observations

De la même manière, les courbes ROC profitent de ce retrait de l'enfant TS18, de manière marginale. La Table 8.3 présente les anciennes et nouvelles valeurs d'AUC, ainsi que l'évolution entre les deux situations.

Conclusions

Après avoir étudié trois méthodes pour l'hybridation de nos résultats, nous obtenons des résultats supérieurs à nos sous-approches traitées seules. Nous obtenons notamment une AUC supérieure à la meilleure d'entre elles (0.933 contre 0.901 pour Statistique).

Cette réunion des résultats présente des valeurs très fortes de précision, mais il faut considérer que les poids ont été cherchés sur des résultats déjà obtenus. Il serait nécessaire d'obtenir de nouvelles données de validation pour définitivement confirmer, ou bien infirmer, la validité de ces poids.

Les résultats privés des données de TS18 sont légèrement supérieurs. Il s'agit de l'effet logique du retrait d'une erreur. Cela étant, il reste à noter que les résultats deviennent particulièrement précis, avec une AUC atteignant 0.952.

Chapitre 9

Conclusion

Au cours de nos travaux, regroupés dans ce manuscrit, nous avons pu montrer l'intérêt de l'application de procédés de Machine Learning, en particulier les Réseaux de Neurones, dans le cadre du diagnostic du Trouble du Spectre de l'Autisme. En particulier, nous avons surtout prouvé que l'étude des mouvements oculaires dans le cadre du diagnostic peut être automatisée et donne des résultats probants.

Alors que les domaines de l'Intelligence Artificielle et de la Psychologie Cognitive sont peu liés, nous avons mené des travaux visant à dresser un pont entre eux. La littérature comprenait déjà des travaux visant ces domaines et nous nous sommes basés sur ceux-ci pour comprendre l'aide que nous pouvions apporter. En parallèle, les travaux en Psychologie Cognitive, précisément pour le Trouble du Spectre de l'Autisme, sur l'analyse du regard et les différences de dynamique entre les enfants touchés par le Trouble et les enfants non touchés, nous ont indiqué une voie assez peu explorée en Intelligence Artificielle.

Un laboratoire partenaire, le CRP-CPO (Centre de Recherche en Psychologie, Cognition, Psychisme et Organisations), a mis à notre disposition différents jeux de données. Ces données étaient de deux formes : brutes et classées par événements. La récolte de ces données a été effectuée par les chercheurs de ce laboratoire. Des enfants avaient été placés devant un écran sur lequel sont diffusés des événements visant à déclencher des réactions sociales (communication verbale et non-verbale notamment). Leurs réactions

visuelles étaient enregistrées par un Eye-Tracker, qui relève la position du regard sur l'écran à une fréquence de 60Hz.

Les données brutes fournissaient l'ensemble des points de la position du regard sur l'écran relevés par l'appareil d'Eye-Tracking, avec quelques autres données telles que les dimensions des pupilles ou la position des yeux par exemple. Les données d'événements, quant à elles, listent l'enchaînement des événements de dynamique du regard du participant, entre la concentration du regard sur une zone restreinte, un mouvement rapide et la perte du suivi (clignement ou erreurs de mesure).

Le choix de nos travaux s'est rapidement porté sur l'étude de ces données par l'utilisation de différentes méthodes de Réseaux de Neurones Artificiels. Selon les types de données exploités (valeurs numériques, images, séries temporelles), nous avons utilisé différents types de réseaux de neurones (Réseaux de Neurones Artificiels, Récurrents, Convolutifs).

Dans nos approches, les données ont subi un pré-traitement. Qu'il s'agisse d'un tri pour conserver un type spécifique d'événement, d'un tracé d'image représentative du chemin de l'œil sur l'écran ou de l'application de méthodes statistiques pour réduire la quantité d'informations ou permettre une représentation des liens entre variables, ces pré-traitements ont tous permis de considérer une représentation des données particulière.

Nous avons débuté nos travaux par l'utilisation des données d'événement, nous concentrant sur les informations propres aux saccades, des mouvements oculaires rapides, dont la durée et la fréquence ont été repérées comme un élément dissociatif entre enfants atteints d'autisme et enfants sans le trouble [Carette et al., 2017]. Exploitant les saccades successives de chaque enfant en séquence de données, nous avons entraîné un réseau de neurones récurrent utilisant des neurones de type LSTM, capables de conserver une mémoire des informations sur des séquences. Ces travaux ont permis de montrer, malgré une faible quantité de données, l'intérêt des données de suivi oculaire dans le cadre de la détection automatisée du Trouble du Spectre de l'Autisme chez des enfants. En raison du manque de données et de l'incertitude quant à la précision des algorithmes utilisés pour les calculs des données d'événement

proposés par le constructeur de l'Eye-Tracker, nous avons choisi de porter notre attention sur des données brutes.

L'exploitation des données numériques brutes a permis de considérer les suivis des enfants selon une approche dynamique. Pour chaque instant de mesure, nous sommes en capacité de calculer la vitesse du mouvement oculaire, son accélération et l'à-coup, dérivée de l'accélération. Ces données calculées sont exploitées elles-aussi sous le format de données séquentielles, avec l'utilisation d'un réseau de neurones convolutif à une dimension. Ces travaux ont montré l'intérêt de l'exploitation de la dynamique du regard pour le diagnostic du Trouble, mais les résultats demandent à être améliorés, notamment en exploitant ces informations de dynamique sous un format graphique.

Nous avons représenté le tracé oculaire graphiquement, colorant les lignes du tracé selon l'état des trois valeurs de dynamique du regard (vitesse, accélération et à-coup). Ces images ont été exploitées avec un réseau de neurones convolutif à deux dimensions, montrant des résultats avec une aire sous la courbe ROC (AUC) de 0.71. Ces résultats manquant de précision, nous avons traité nos images avec une PCA (Analyse par Composantes Principales) pour réduire la quantité d'informations à traiter et conserver les données les plus pertinentes. Ces images, réduites à 50 valeurs numériques ont été traitées par un réseau de neurones artificiel, pour permettre d'atteindre une AUC de 0.91, soit une forte progression. Nous cherchons alors une approche demandant moins de pré-traitements afin de différencier les catégories d'enfants. Aussi, le jeu de données d'image a fait l'objet d'une publication [Carette et al., 2018] et est disponible en ligne. Ce jeu de données a notamment été exploité pour un concours en Inde. L'ensemble de ces procédés à base d'images et de PCA a fait l'objet d'une publication [Carette et al., 2019].

Afin de traiter au maximum les informations des données brutes, nous avons réduit les valeurs relevées par l'Eye-Tracker en fonction de leurs corrélations. Ainsi, des jeux de données très grands sont réduits à seulement 828 valeurs, exploitables avec un réseau de neurones artificiel, bien moins gourmand en ressources qu'un réseau de neurones récurrent ou convolutif. Exploitant ces nouvelles données, nous obtenons des résultats élevés (AUC de 0.90).

Nos différentes approches montrant des résultats valides pour des enfants différents, nous cherchons à trouver une manière de les combiner pour obtenir un modèle hybride. Nous exploitons directement les scores de sortie de trois de nos modèles (basés sur les données au format image, les données au format numérique de représentation de la dynamique et les données de l'étude statistique). La solution retenue, une moyenne pondérée, permet d'obtenir une AUC de 0.93. Par ailleurs, cette étude croisée a permis de repérer un enfant dont le profil particulier (Trouble Envahissant du Développement, et non Trouble du Spectre de l'Autisme). Ses résultats ont donc montré une erreur de classification dans chaque approche. Le retrait de cet enfant dont les données sont aberrantes a permis d'atteindre une AUC de 0.95.

L'ensemble de ces approches a permis d'avancer les travaux dans le cadre de l'aide au diagnostic du Trouble du Spectre de l'Autisme. Peu de ces travaux exploitaient les données d'Eye-Tracking. Ces travaux utilisaient des approches assez différentes des nôtres (concentration sur l'écran, observation de zones d'intérêt de l'écran) avec des résultats moindres (jusqu'à 88% de précision).

L'Eye-Tracker utilisé pour les relevés des données exploitées dans cette thèse reste cependant limité. En effet, sa fréquence d'échantillonnage n'est que de 60Hz, loin des possibilités que peuvent offrir les appareils haut de gamme. Par ailleurs, cette limitation nous empêche de pouvoir constater l'un des éléments propres à la dynamique du regard : les Oscillations Post-Saccades (PSO). L'exploitation de ces PSO, ainsi que l'étude des valeurs spécifiques à cet état du suivi oculaire, pourrait intégrer des éléments-clé pour le perfectionnement d'un système d'aide au diagnostic.

Par ailleurs, nos approches se résument à une réponse binaire quant à l'état autistique d'un enfant. Bien que cela puisse être suffisant pour permettre la prise en charge des enfants concernés, il serait également pertinent de continuer ces travaux dans le sens de l'estimation du score CARS de l'enfant. Catégoriser l'état autistique de l'enfant entre les classes de gravité que le CARS permet d'obtenir ("Sans", "Léger", "Moyen" et "Sévère").

Aussi, d'autres données, non exploitées dans le cadre de cette thèse, telles que des données d'EEG pourraient être utilisées dans de futurs travaux, pour

permettre un diagnostic plus précis. En effet, le diagnostic de Trouble du Spectre de l'Autisme manuel n'est pas posé uniquement par observation du suivi oculaire de l'enfant, et si nous souhaitons poser un diagnostic efficace, il est nécessaire d'utiliser l'ensemble des informations à disposition des experts du domaine.

Enfin, le diagnostic de ce Trouble est lié à des paramètres humains, et donc variables selon les individus à diagnostiquer. Ainsi, il est peu probable que soit trouvé un modèle infaillible, notamment dans les cas d'autisme léger. Augmenter la précision des modèles devrait tendre vers un maximum, et l'intérêt serait alors de pouvoir établir un diagnostic avec un détail de la gravité du trouble, c'est-à-dire la position de l'enfant sur le spectre.

Bibliographie

[Agarap, 2018] Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv :1803.08375*.

[Arlot et al., 2010] Arlot, S., Celisse, A., et al. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, 4 :40–79.

[Benesty et al., 2009] Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer.

[Bernacki, 2004] Bernacki, M. (2004). Principles of training multi-layer neural network using backpropagation.

[Bruni, 2014] Bruni, T. P. (2014). Test review : Social responsiveness scale–second edition (srs-2).

[Campagner et al., 2019] Campagner, A., Cabitza, F., and Ciucci, D. (2019). Exploring medical data classification with three-way decision tree. In *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2019)-Volume*, volume 5, pages 147–158.

[Carette et al., 2017] Carette, R., Cilia, F., Dequen, G., Bosche, J., Guerin, J.-L., and Vandromme, L. (2017). Automatic autism spectrum disorder detection thanks to eye-tracking and neural network-based approach. In *International Conference on IoT Technologies for HealthCare*, pages 75–81. Springer.

[Carette et al., 2019] Carette, R., Elbattah, M., Cilia, F., Dequen, G., Guerin, J.-L., and Bosche, J. (2019). Learning to predict autism spectrum disorder based on the visual patterns of eye-tracking scanpaths. In *12th International Conference on Health Informatics*.

[Carette et al., 2018] Carette, R., Elbattah, M., Dequen, G., Guérin, J.-L., and Cilia, F. (2018). Visualization of eye-tracking patterns in autism spec-

trum disorder : method and dataset. In *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, pages 248–253. IEEE.

[Cho et al., 2014] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv :1406.1078*.

[Chraibi Kaadoud and Vieville, 2017] Chraibi Kaadoud, I. and Vieville, T. (2017). Reprenons les bases : Neurone artificiel, neurone biologique.

[Cilia et al., 2016] Cilia, F., Deschamps, L., and Vandromme, L. (2016). Investigation of interactive visual patterns during semi-structured joint attention sequences in children with autism spectrum disorder. In *RIPSY-DEVE Conference, Louvain-la-neuve*.

[Cilia et al., 2018] Cilia, F., Garry, C., Brisson, J., and Vandromme, L. (2018). Attention conjointe et exploration visuelle des enfants au développement typique et avec tsa : synthèse des études en oculométrie. *Neuropsychiatrie de l'Enfance et de l'Adolescence*, 66(5) :304–314.

[Cohen, 1995] Cohen, W. W. (1995). Fast effective rule induction. In *Machine learning proceedings 1995*, pages 115–123. Elsevier.

[Daniels et al., 2018] Daniels, J., Schwartz, J. N., Voss, C., Haber, N., Fazel, A., Kline, A., Washington, P., Feinstein, C., Winograd, T., and Wall, D. P. (2018). Exploratory study examining the at-home feasibility of a wearable tool for social-affective learning in children with autism. *npj Digital Medicine*, 1(1) :32.

[De Boer et al., 2005] De Boer, P.-T., Kroese, D. P., Mannor, S., and Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of operations research*, 134(1) :19–67.

[Dean and Ghemawat, 2004] Dean, J. and Ghemawat, S. (2004). Mapreduce : Simplified data processing on large clusters.

[Dimou et al., 2014] Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., and Van de Walle, R. (2014). Rml : A generic language for integrated rdf mappings of heterogeneous data. *Ldow*, 1184.

[Duchi et al., 2011] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul) :2121–2159.

[Duda et al., 2016] Duda, M., Ma, R., Haber, N., and Wall, D. (2016). Use of machine learning for behavioral distinction of autism and adhd. *Translational psychiatry*, 6(2) :e732.

[Fang et al., 2014] Fang, Y., Wang, J., Li, J., Pépion, R., and Le Callet, P. (2014). An eye tracking database for stereoscopic video. In *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*, pages 51–52. IEEE.

[Freund et al., 1999] Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780) :1612.

[Gers et al., 1999] Gers, F. A., Schmidhuber, J., and Cummins, F. (1999). Learning to forget : Continual prediction with lstm.

[GmbH, 2014] GmbH, S. I. (2014). Sensomotoric instruments unveils smi red-n consumer eye control technology for gaming and computing.

[Goldberg and Helfman, 2010] Goldberg, J. H. and Helfman, J. I. (2010). Visual scanpath representation. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, pages 203–210. ACM.

[Gupta and Kalyan, 2019] Gupta, P. and Kalyan, G. (2019). Autism detection : Learning to predict autism spectrum disorder based on the visual patterns of eye-tracking scanpaths.

[Heinsfeld et al., 2018] Heinsfeld, A. S., Franco, A. R., Craddock, R. C., Buchweitz, A., and Meneguzzi, F. (2018). Identification of autism spectrum disorder using deep learning and the abide dataset. *NeuroImage : Clinical*, 17 :16–23.

[Hinton et al., 2012] Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14 :8.

[Hristev, 1998] Hristev, R. (1998). The ann book. *GNU public license*, 71.

[Hunter, 2007] Hunter, J. D. (2007). Matplotlib : A 2d graphics environment. *Computing in science & engineering*, 9(3) :90.

[Ingersoll and Schreibman, 2006] Ingersoll, B. and Schreibman, L. (2006). Teaching reciprocal imitation skills to young children with autism using a naturalistic behavioral approach : Effects on language, pretend play, and joint attention. *Journal of autism and developmental disorders*, 36(4) :487.

[ITU, 2017] ITU (2017). Studio encoding parameters of digital television for standard 4 :3 and wide screen 16 :9 aspect ratios. International Telecommunication Union-Radiocommunication Sector, Geneva.

[Jamal et al., 2014] Jamal, W., Das, S., Oprescu, I.-A., Maharatna, K., Apicella, F., and Sicca, F. (2014). Classification of autism spectrum disorder using supervised learning of brain connectivity measures extracted from synchrostates. *Journal of neural engineering*, 11(4) :046019.

[Kamnitsas et al., 2017] Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. P., Kane, A. D., Menon, D. K., Rueckert, D., and Glocker, B. (2017). Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36 :61–78.

[Kendall, 1938] Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2) :81–93.

[Kohavi et al., 1995] Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

[Kovarski et al., 2019] Kovarski, K., Siwiaszczyk, M., Malvy, J., Batty, M., and Latinus, M. (2019). Faster eye movements in children with autism spectrum disorder. *Autism Research*, 12(2) :212–224.

[Kylliäinen et al., 2012] Kylliäinen, A., Wallace, S., Coutanche, M. N., Leppänen, J. M., Cusack, J., Bailey, A. J., and Hietanen, J. K. (2012). Affective–motivational brain responses to direct gaze in children with autism spectrum disorder. *Journal of Child Psychology and Psychiatry*, 53(7) :790–797.

[Längkvist et al., 2016] Längkvist, M., Alirezaie, M., Kiselev, A., and Loutfi, A. (2016). Interactive learning with convolutional neural networks for image labeling. In *International Joint Conference on Artificial Intelligence (IJCAI), New York, USA, 9-15th July, 2016*.

[LeCun et al., 1988] LeCun, Y., Touresky, D., Hinton, G., and Sejnowski, T. (1988). A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pages 21–28. CMU, Pittsburgh, Pa : Morgan Kaufmann.

[Liu et al., 2016] Liu, W., Li, M., and Yi, L. (2016). Identifying children with autism spectrum disorder based on their face processing abnormality : A machine learning framework. *Autism Research*, 9(8) :888–898.

[Marighetto et al., 2017] Marighetto, P., Coutrot, A., Riche, N., Guyader, N., Mancas, M., Gosselin, B., and Laganiere, R. (2017). Audio-visual attention : Eye-tracking dataset and analysis toolbox. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 1802–1806. IEEE.

[McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4) :115–133.

[Murias et al., 2007] Murias, M., Webb, S. J., Greenson, J., and Dawson, G. (2007). Resting state cortical connectivity reflected in eeg coherence in individuals with autism. *Biological psychiatry*, 62(3) :270–273.

[Nguyen, 2018] Nguyen, M. (2018). Illustrated guide to lstm's and gru's : A step by step explanation.

[Olah, 2015] Olah, C. (2015). Understanding lstm networks.

[Ozonoff et al., 2005] Ozonoff, S., Goodlin-Jones, B. L., and Solomon, M. (2005). Evidence-based assessment of autism spectrum disorders in children and adolescents. *Journal of Clinical Child and Adolescent Psychology*, 34(3) :523–540.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6) :386.

[Salvucci and Goldberg, 2000] Salvucci, D. D. and Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, pages 71–78. ACM.

[Schopler et al., 1980] Schopler, E., Reichler, R. J., DeVellis, R. F., and Daly, K. (1980). Toward objective classification of childhood autism : Childhood autism rating scale (cars). *Journal of autism and developmental disorders*, 10(1) :91–103.

[Schreibman et al., 2015] Schreibman, L., Dawson, G., Stahmer, A. C., Landa, R., Rogers, S. J., McGee, G. G., Kasari, C., Ingersoll, B., Kaiser, A. P., Bruinsma, Y., et al. (2015). Naturalistic developmental behavioral interventions : Empirically validated treatments for autism spectrum disorder. *Journal of autism and developmental disorders*, 45(8) :2411–2428.

[Seo, 2018] Seo, J. D. (2018). [back to basics] deriving back propagation on simple rnn/lstm (feat. aidan gomez).

[Sharma, 2018] Sharma, S. (2018). What the hell is perceptron ?

[SMI, 2012] SMI (2012). Red-m eye tracking system manual.

[Spearman, 1904] Spearman, C. (1904). The proof and measurement of association between two things. *American journal of Psychology*, 15(1) :72–101.

[Tanner, 2018] Tanner, G. (2018). Generating text using a recurrent neural network.

[Thabtah, 2018] Thabtah, F. (2018). An accessible and efficient autism screening method for behavioural data and predictive analyses. *Health informatics journal*, page 1460458218796636.

[Thabtah, 2019] Thabtah, F. (2019). Machine learning in autistic spectrum disorder behavioral research : A review and ways forward. *Informatics for Health and Social Care*, 44(3) :278–297.

[Uluyagmur-Ozturk et al., 2016] Uluyagmur-Ozturk, M., Arman, A. R., Yilmaz, S. S., Findik, O. T. P., Genc, H. A., Carkaxhiu-Bulut, G., Yazgan, M. Y., Teker, U., and Cataltepe, Z. (2016). Adhd and asd classification based on emotion recognition data. In *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*, pages 810–813. IEEE.

[Vargas-Cuentas et al., 2016] Vargas-Cuentas, N. I., Hidalgo, D., Roman-Gonzalez, A., Power, M., Gilman, R. H., and Zimic, M. (2016). Diagnosis of autism using an eye tracking system. In *Global Humanitarian Technology Conference (GHTC), 2016*, pages 624–627. IEEE.

[Wan et al., 2018] Wan, G., Kong, X., Sun, B., Yu, S., Tu, Y., Park, J., Lang, C., Koh, M., Wei, Z., Feng, Z., et al. (2018). Applying eye tracking to identify autism spectrum disorder in children. *Journal of autism and developmental disorders*, pages 1–7.

[Wold et al., 1987] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3) :37–52.

[Yaneva et al., 2018] Yaneva, V., Ha, L. A., Eraslan, S., Yesilada, Y., and Mitkov, R. (2018). Detecting autism based on eye-tracking data from web searching tasks. In *Proceedings of the Internet of Accessible Things*, page 16. ACM.

[Zeiler, 2012] Zeiler, M. D. (2012). Adadelta : an adaptive learning rate method. *arXiv preprint arXiv :1212.5701*.

[Zemblys, 2016] Zemblys, R. (2016). Eye-movement event detection meets machine learning. *Biomedical Engineering 2016*, 20(1).

[Zemblys et al., 2018] Zemblys, R., Niehorster, D. C., Komogortsev, O., and Holmqvist, K. (2018). Using machine learning to detect events in eye-tracking data. *Behavior research methods*, 50(1) :160–181.

[Zhou et al., 2014] Zhou, Y., Yu, F., and Duong, T. (2014). Multiparametric mri characterization and prediction in autism spectrum disorder using graph theory and machine learning. *PloS one*, 9(6) :e90405.

Les domaines de l'Informatique et de la Psychologie sont très éloignés. Cependant, certains besoins en Psychologie Cognitive (PC) peuvent être satisfaits par l'exploitation de l'Intelligence Artificielle (IA), en particulier ses approches connexionnistes. Il existe quelques applications des principes d'IA en PC, mais elles sont relativement discrètes. En particulier, le cas de l'aide à la détection du Trouble du Spectre de l'Autisme (TSA) est un domaine vierge, à quelques exceptions près. Les travaux de cette thèse ont porté sur l'application de diverses modifications sur les données (présentation sous forme d'images, réduction de dimensionnalité, approches statistiques). Ils ont exploité divers modèles d'IA (Réseaux de neurones artificiels, récurrents et convolutifs) afin de produire des techniques d'aide à la détection. Nous avons ensuite appliqué une moyenne pondérée des résultats de nos approches pour obtenir une technique d'aide à la détection encore plus précise. Le meilleur de nos résultats permet d'obtenir une courbe ROC dont l'AUC atteint les 95%, ce qui permet d'envisager, avec l'ajout de données supplémentaires utilisées en diagnostic manuel, une aide presque parfaite et une libération complète du temps de l'expert pour permettre une totale concentration sur la mise en place d'une aide pour l'enfant et le suivi au plus près de celui-ci. Par ailleurs, l'opportunité d'un diagnostic au plus tôt de l'enfant permet de réduire au mieux le retard neurodéveloppemental de l'enfant.

***

The Computer Sciences and Psychology fields are very far from each other. However, some needs in Cognitive Psychology (CP) can be satisfied through the use of Artificial Intelligence (AI), in particular its connectionist approaches. There are a few uses of AI principles in CP, but they are quite inconspicuous. In particular, the case of the diagnosis support applied to Autism Spectrum Disorder (ASD) is a blank slate, with a few exceptions. The work in this thesis have focused on the application of various data modifications (presented as images, with a dimension reduction or statistics). It used various AI models (Artificial, Recurrent and Convolutional Neural Networks) to produce detection support techniques. Then, we have applied a weighted mean over these results to increase the precision of this detection support technique. The best of our results allowed to get a ROC curve with an AUC reaching 95%, which allow to think about an almost perfect support, given we could add some data used in manual diagnosis, and a complete freeing of the experts' time to enable a total focus on the setup of the child's support and his/her closer following. Moreover, the opportunity of an earlier child diagnosis can help better reducing the child's neurodevelopmental delay.