

UNIVERSITÉ DU QUÉBEC

**RECONNAISSANCE D'OBJETS TRIDIMENSIONNELS
À PARTIR DE DONNÉES TACTILES**

COMME EXIGENCE PARTIELLE DE LA MAÎTRISE EN SCIENCE ET TECHNOLOGIE
DE L'INFORMATION

PAR NICOLAS PEDNEAULT

DIRECTRICE DE RECHERCHE : ANA-MARIA CRETU
PRÉSIDENTE DU JURY : ROKIA MISSAOUI
MEMBRE DU JURY : MAREK ZAREMBA

SEPTEMBRE 2017

Remerciement

Je tiens à remercier Alain Dessureault et Luc Beaudoin, des anciens patrons et des amis pour toujours, pour leur support tout au long du périple académique qui a mené à ce mémoire.

Un immense merci à Dre Ana-Maria Créto, professeure et superviseuse comme il s'en fait peu, pour son aide tout au long de ce travail. Ce travail, c'est aussi beaucoup le vôtre.

Et au final, un incommensurable merci à la femme et l'amour de ma vie, Sylvie Desrosiers, pour m'avoir littéralement enduré quand je n'étais même plus en mesure de m'endurer moi-même au cours des six dernières années. Sans toi, il n'y aurait pas eu de maîtrise.

Résumé

Les percées technologiques du dernier siècle en matière d'électronique et de miniaturisation ont véritablement permis le développement d'appareils et d'instruments ne permettant pas seulement de décupler l'acuité des sens humains, mais de les reproduire de manière synthétique. Les capteurs développés pour reproduire le toucher demeurent tous néanmoins limités en matière de reconnaissance d'objets, d'où la nécessité de développer de nouveaux types de capteurs d'un côté, et de l'autre, d'améliorer de manière intelligente le processus d'acquisition, d'interprétation et d'intégration des données collectées.

Ce projet vise, en premier lieu, l'amélioration du processus d'acquisition des données tactiles sous forme « d'empreintes » tactiles. Pour identifier les endroits sur la surface d'un objet 3D (les points d'intérêt), sur lesquels seront collectées les empreintes tactiles, nous utiliserons une approche informée, sous forme d'algorithmes visuels de détection de points d'intérêt. L'approche informée présuppose l'utilisation de connaissances extraites par d'autres sens, la vision dans le cas présent, et ayant comme objectif la maximisation du gain d'information sur l'objet pour guider le processus d'acquisition, au lieu de toucher l'objet à l'aveuglette (approche non-informée). En s'inspirant de l'expérience humaine en matière du toucher et de la vision qui démontrent l'utilité de la vue pour saisir et manipuler des objets, tous les algorithmes utilisés, sauf un, sont donc inspirés directement de l'attention visuelle humaine pour identifier les régions qui attirent l'attention.

En deuxième lieu, puisque l'acquisition de données réelles aux endroits identifiés nécessiterait énormément de temps, nous simulerons, dans une première étape, celle-ci à l'aide d'objets virtuels, sur lesquels nous récolterons des empreintes en croisant un plan virtuel, représentant la surface d'un capteur tactile, avec leurs surfaces. Ce plan subira ensuite de légères translations successives vers l'intérieur de l'objet pour simuler l'interaction avec un capteur réel, et les points d'intersection de ce plan avec l'objet virtuel seront enregistrés, créant une série de courbes de niveaux. Chacune des séries de courbes de niveaux deviendra une empreinte locale sur l'objet virtuel.

En troisième lieu, nous tenterons de reconnaître de manière autonome les objets pour lesquels nous aurons obtenus les empreintes. Diverses techniques d'apprentissage, telles que les réseaux de neurones, l'apprentissage profond, les réseaux bayésiens, les arbres, les forêts d'arbres et le *boosting* de arbres de décision, les machines de vecteurs de support (*support vector*

machines) ainsi que la technique de K plus-proches voisins seront mises à profit et comparées pour atteindre ce but. Finalement, des données réelles (profils locaux de pression) sont également récoltées sur un nombre réduit d'objets réels en utilisant un capteur résistif sensible à la force pour valider les résultats obtenus.

Table des matières

Remerciement	2
Résumé.....	3
Table des matières.....	5
Table des figures	8
Chapitre 1 : Introduction.....	10
Chapitre 2 : État de l’art.....	13
2.1. Toucher humain	13
2.2. Les capteurs robotiques tactiles	14
2.2.1 L’haptique	14
2.2.2 Caractéristiques des capteurs tactiles.....	15
2.2.3 Capteurs tactiles actuels.....	16
2.3. Stratégies d’exploration robotique.....	20
2.3.1 Stratégies d’exploration locales et globales.....	20
2.3.2 Stratégies d’exploration informées et non-informées	22
2.3.3 Autres stratégies d’exploration de surface.....	22
2.3.4 Les textures et leur importance	23
2.4. Méthodes d’analyse, d’interprétation et de classification des données tactiles	24
2.5. Conclusions sur l’état de l’art	26
Chapitre 3. Objectifs	29
Chapitre 4 : Méthodologie	31
4.1. Capteurs de profils de pression <i>FSR</i>	32
4.2. Détection des points d’intérêt	34
4.2.1. L’algorithme “Points Saillants” (PS, <i>Salient points</i>)	35
4.2.2. L’algorithme “Saillance du Maillage” (SM, <i>Mesh Saliency</i>)	35
4.2.3. L’algorithme basé sur l’attention visuelle (AV, AVO)	36
4.2.4. L’algorithme basé sur la signature thermique (HKS).....	36
4.3. Objets virtuels	37
4.4. Simulation du processus de collecte des données tactiles	37
4.5. Extraction de caractéristiques à partir des données tactiles	43
4.5.1. RapidMiner	43
4.5.1.1. Présentation de <i>RapidMiner</i>	43
4.5.1.2. Module d’analyse d’images pour <i>RapidMiner</i>	44
4.5.2. Extraction des caractéristiques.....	44
4.6. Classification des objets.....	46
4.6.1. Les réseaux de neurones et l’apprentissage profond.....	47
4.6.2. Les réseaux bayésiens	50
4.6.3. Les arbres de décision.....	51
4.6.4. Forêt d’arbres décisionnels (<i>Random Forest</i>).....	52
4.6.5. Boosting d’arbres de décision (<i>Gradient Boosted Tree</i>).....	52
4.6.6. <i>K</i> plus proches voisins (<i>k-Nearest Neighbours, k-NN</i>)	53
4.6.7. Machines à vecteurs de support (<i>SVM</i>).....	53

4.6.8. L'évaluation de la classification	53
Chapitre 5. Résultats expérimentaux	55
5.1. Données simulées de tests.....	55
5.2. Résultats expérimentaux pour la classification des données simulées en utilisant les caractéristiques IMMI pour les six objets de test.....	58
5.2.1. Résultats expérimentaux utilisant l'apprentissage profond	58
5.2.2. Résultats expérimentaux utilisant le module <i>Naïve Bayes</i>	59
5.2.3 Résultats expérimentaux utilisant les arbres de décision.....	60
5.2.4 Résultats de classification obtenus avec la forêt d'arbres décisionnels.....	60
5.2.5 Résultats de classification avec le <i>boosting</i> d'arbres de décision.....	61
5.2.6 Résultats de classification avec les machines de vecteurs de support	61
5.2.7 Résultats de classification avec les <i>K</i> plus proches voisins	62
5.3. Résultats expérimentaux pour la classification des données simulées en utilisant les caractéristiques IMMI pour quatre objets de test.....	64
5.3.1. Résultats expérimentaux utilisant l'apprentissage profond	64
5.3.2. Résultats expérimentaux utilisant le module <i>Naïve Bayes</i>	65
5.3.3 Résultats expérimentaux utilisant les arbres de décision.....	65
5.3.4 Résultats de classification obtenus avec la forêt d'arbres décisionnels.....	66
5.3.5 Résultats de classification avec le <i>boosting</i> d'arbres de décision.....	66
5.3.6 Résultats de classification avec les machines de vecteurs de support	66
5.3.7 Résultats de classification avec les <i>K</i> plus proches voisins	67
5.4. Résultats expérimentaux pour la classification des données simulées en utilisant le sous-échantillonnage	69
5.5 Utilisation de la corrélation croisée normalisée pour améliorer les résultats de la classification	70
5.6 Résultats expérimentaux obtenus sur les objets réels	72
Chapitre 6. Contributions et conclusions	78
Bibliographie.....	81
Annexes.....	88
Annexe A – code pour générer les intersections.....	88
Annexe B – code pour générer les empreintes.....	92
Annexe C – matrices de confusion	98
Apprentissage profond, six objets virtuels.....	98
Naive Bayes, six objets virtuels	99
Arbres de décision, six objets virtuels	100
Forêt d'arbres, six objets virtuels.....	103
<i>Boosting</i> d'arbres de décision, six objets virtuels.....	104
Evolutionary SVM, six objets virtuels.....	106
K plus proches voisins, six objets virtuels	107
Apprentissage profond, quatre objets virtuels	108
Naive Bayes, quatre objets virtuels.....	110
Arbres de décision, quatre objets virtuels	112
Forêt d'arbres, quatre objets virtuels	113
<i>Boosting</i> d'arbres de décision, quatre objets virtuels.....	115
Evolutionary SVM, quatre objets virtuels	116
K plus proches voisins, quatre objets virtuels.....	118

Objets réels..... 120

Table des figures

Figure 1 : Liens entre l'exploration et la manipulation (boucle haptique d'Okamura, adapté de [10]).....	14
Figure 2 : Processus de capture et classification d'empreintes. À gauche, le processus pour les objets virtuels, et à droite, celui pour les objets réels.	31
Figure 3 : Diverses représentations d'un capteur FSR et ses caractéristiques [37].	33
Figure 4 : (a) Profil de pression, et (b) empreinte sous forme d'image obtenus à l'aide d'un capteur FSR.....	34
Figure 5 : Points d'intérêt, en rouge, obtenus sur la surface d'un même objet, en utilisant les algorithmes : (a) objet virtuel obtenu par maillage, (b) points saillants (Salient Points) [71], (c) HKS [71], (d) attention visuelle (AV) [66] et (e), saillance du maillage [71].	38
Figure 6 : Maillage de triangles représentant la surface d'un objet virtuel ainsi que les points d'intérêt, en rouge.	39
Figure 7 : Relation entre le point d'intérêt, les points adjacents, l'estimation d'un plan tangent à la surface et le vecteur orthogonal servant d'axe de translation du plan.	40
Figure 8 : (a), (b) Plans successifs sur l'objet virtuel ; (c) courbes d'intersection obtenues sur l'objet ; et (d) image de l'empreinte obtenue après toutes les étapes précédentes.	41
Figure 9 : Exemple de : (a) empreinte invalide, causée par un axe de translation des plans désaxé par rapport à la normale à la surface; et (b) empreinte valide - l'axe de translation des plans est parallèle à la normale à la surface.	42
Figure 10 : (a) Illustration en jaune de la figurine de vache, ses points d'intérêt détectés par l'algorithme d'attention visuelle en rouge et les empreintes obtenues aux points de contact du capteur en bleu et (b) la série d'empreintes générées pour la figurine de la vache.	42
Figure 11 : Interface graphique de RapidMiner Studio.	43
Figure 12 : Capture d'écran : caractéristiques extraites des images dans RapidMiner.	45
Figure 13 : Exemple d'un réseau neuronal simple.	47
Figure 14 : Fonction d'activation sigmoïde.	48
Figure 15 : Étapes nécessaires à la rétro-propagation.....	49
Figure 16 : Structure d'un réseau bayésien simple [75].	50
Figure 17 : Exemple d'arbre de décision pour classification d'empreintes.....	51
Figure 18 : Exemple de matrice de confusion.	54

Figure 19 : Objets réels et modèles (maillages) correspondants, avec les points d'intérêt marqués en rouge.....	55
Figure 20 : Série des empreintes générées pour trois algorithmes d'identification de points d'intérêt pour l'objet tasse ("cup") : (a) "points saillants", (b) HKS et (c) "attention visuelle".....	56
Figure 21 : Nombre d'empreintes par objet, pour chacun des algorithmes.....	57
Figure 22 : Capture d'écran du sous-processus de classification utilisant l'algorithme Deep Learning.....	59
Figure 23 : Résultats obtenus pour les données simulées sur les six objets avec tous les classificateurs en utilisant le module IMMI.....	63
Figure 24 : Exemple de similarité (calculée à l'aide de la corrélation croisée normalisée) entre les empreintes de "cow" et "dromedary".....	64
Figure 25 : Résultats obtenus pour les données simulées sur les quatre objets avec tous les classificateurs en utilisant le module IMMI.....	68
Figure 26 : Résultats obtenus avec tous les classificateurs, pour quatre objets virtuels, avec la méthode de l'attention visuelle optimisée.....	69
Figure 27 : Graphique illustrant les résultats obtenus avec tous les classificateurs, pour les objets virtuels, avec la méthode de l'attention visuelle optimisée et les empreintes les plus dissimilaires, comparativement à l'attention visuelle et toutes les empreintes.....	71
Figure 28 : Graphique illustrant les résultats obtenus avec tous les classificateurs, pour six objets virtuels avec les méthodes IMMI, pour les vecteurs 16x16 et pour les vecteurs 32 x32, ainsi qu'avec la corrélation croisée normalisée.....	72
Figure 29 : (a) Empreinte originale et (b) empreinte finale. On y constate l'adoucissement des contours de (a) vers (b).	73
Figure 30 : Résultats obtenus avec tous les classificateurs, pour les quatre objets réels pour les caractéristiques extraites par le module IMMI et obtenues par sous-échantillonnage.	74
Figure 31 : Résultats obtenus avec tous les classificateurs, pour les quatre objets réels, en utilisant la similarité.....	75
Figure 32 : Meilleurs résultats sur les données simulées et réelles.	76

Chapitre 1 : Introduction

Les humains sont dotés de cinq sens qui leur permettent de percevoir le monde qui les entoure. Ces sens ont toutefois leurs limites : on ne peut voir s'il n'y a pas de lumière ; on ne peut toucher si les objets sont physiquement hors de portée. L'idée de projeter, d'améliorer et de décupler les sens humains est donc une réponse logique et un désir légitime par rapport à ces limites.

Les percées technologiques du dernier siècle en matière d'électronique et de miniaturisation ont véritablement permis le développement d'appareils et d'instruments permettant non seulement de décupler l'acuité des sens humains, mais de les reproduire de manière synthétique. À titre d'exemple, on n'a qu'à penser aux systèmes de reconnaissance auditive utilisés par les fournisseurs de systèmes téléphoniques. Or, si la technologie a permis l'émergence de systèmes fiables permettant la reproduction des sens de la vue, de l'ouïe et même de l'odorat, le sens du toucher semble avoir été laissé pour compte. Ainsi, il existe peu de systèmes opérationnels, en dehors des cadres académiques, reproduisant le sens du toucher, du contact avec l'objet jusqu'à l'interprétation cognitive de ce contact.

Néanmoins, la dernière décennie a vu l'apparition de nombreux capteurs tactiles composés de matériaux divers, du module le plus simple à la peau artificielle englobant une matrice de petits capteurs tactiles. La plupart des solutions proposées n'utilisent qu'une seule technologie, telle que les capteurs à effet Hall, les baromètres, les matrices de capteurs tactiles, les capteurs optiques et les accéléromètres pour obtenir des données tactiles. Plusieurs systèmes reproduisant le toucher humain ont été aussi proposés, incluant ceux qui captent des empreintes grâce à la pression ou les ombres causées par la déformation d'une membrane illuminée de l'intérieur, alors que d'autres reproduisent le mouvement de la main pour détecter la texture d'un objet. D'autres capteurs sont placés au bout de dispositifs reproduisant grossièrement l'aspect des doigts d'une main humaine ; ce qu'offre la nature en matière d'organes sensoriels est d'ailleurs souvent imité, mais rarement égalé.

Un large éventail de techniques d'analyse de données, incluant les arbres de décisions, les sacs de caractéristiques, les réseaux neuronaux, les machines de vecteurs de support, etc. ont été testées afin d'interpréter les données tactiles dans des tâches telles que la reconnaissance de textures, l'identification de matériaux de surfaces, la détection de glissement pour des applications robotisées et l'identification d'objets. Les capteurs développés pour reproduire le

toucher demeurent tous néanmoins limités en matière de reconnaissance d'objets, d'où la nécessité de développer de nouveaux types de capteurs d'un côté, et de l'autre, d'améliorer de manière intelligente le processus d'acquisition, d'interprétation et d'intégration des données collectées.

Un défi important à relever pour la collecte des données est lié au fait que l'acquisition de données tactiles nécessite un contact direct avec l'objet. En ce qui a trait à la reconnaissance d'objets, le processus de positionnement du capteur s'avère long et fastidieux. La reconnaissance aveugle est possible mais généralement peu efficace pour comprendre la structure tridimensionnelle d'un objet [1]. L'expérience humaine en matière de toucher et de vision démontre l'utilité de la vue pour saisir et manipuler des objets. L'intégration de la rétroaction visuelle à des stimuli sensoriels peut compenser pour l'imprécision des systèmes visuels seuls, due parfois à des occlusions, et pour le fait qu'ils ne peuvent transmettre d'information concernant la force appliquée. De plus, l'utilisation de données visuelles peut guider le capteur tactile vers des régions caractéristiques de l'objet afin de réduire le temps d'exploration qui peut parfois être long à cause des mouvements complexes nécessaires pour obtenir des données tactiles.

La biologie offre de nombreuses pistes de solution ; en s'inspirant des mouvements exploratoires de la main que les humains utilisent fort efficacement, une procédure ressemblant à un tapotement a été intégrée dans des systèmes robotiques pour différencier différentes textures sur des surfaces [2], alors que des mouvements de grattement effectués par une main robotique ont été utilisés pour estimer la rugosité d'une surface [3]. Ces accomplissements récents illustrent l'influence et l'apport de la biologie sur le développement et le perfectionnement des méthodes sélectives existantes, ce que cherche également à accomplir le présent projet.

Ce projet vise, en premier lieu, l'amélioration du processus d'acquisition de données sous forme « d'empreintes » tactiles d'objets virtuels, et ensuite, la reconnaissance automatique de ces objets virtuels à partir de leurs empreintes à l'aide de techniques d'apprentissage. Pour identifier les endroits sur la surface d'un objet tridimensionnel sur lesquels seront collectées les empreintes tactiles, nous utiliserons une approche informée, sous forme d'algorithmes visuels de détection de points d'intérêt. Tous ceux utilisés, sauf un, sont inspirés directement de l'attention visuelle humaine pour identifier les régions qui attirent l'attention. Puisque l'acquisition de données réelles aux endroits identifiés nécessiterait énormément de temps, nous simulerons, dans une

première étape, celles-ci à l'aide d'objets virtuels, sur lesquels nous récolterons des empreintes en croisant un plan virtuel, représentant la surface d'un capteur tactile, avec leurs surfaces.

Diverses techniques d'apprentissage, telles que les réseaux de neurones, l'apprentissage profond, les réseaux bayésiens, les arbres de décision ainsi que d'autres techniques seront mises à profit pour classifier les objets à partir des empreintes collectées. Une fois ce processus bien établi, nous testerons l'approche avec des empreintes prises sur des objets réels.

Les applications relatives à l'identification d'objets à l'aide de capteurs tactiles sont diverses et variées. En premier lieu, les capteurs tactiles permettent d'obtenir de l'information dans des situations où les autres types de capteurs ne sont pas en mesure de le faire; citons notamment des conditions d'éclairage difficiles, où un système optique ne pourrait voir un objet, mais où le capteur tactile pourrait néanmoins le toucher afin de l'identifier. Les capteurs tactiles pourront également fournir de l'information à des systèmes d'analyse, tributaires de plusieurs autres types de capteurs, tels des capteurs visuels, sonores et olfactifs, afin de mieux cerner le contexte et l'environnement de situations particulières. En deuxième lieu, ce type de capteur pourra améliorer les capacités sensorielles de la prochaine génération de robots personnels. Ainsi, une personne pourra demander à un robot d'aller lui chercher un beau fruit mûr, et celui-ci saura faire la différence entre un fruit prêt à être consommé par rapport à un autre qui le sera moins en utilisant le sens de la toucher. Le robot saura aussi comment manipuler l'objet demandé afin de ne pas l'endommager. Finalement, la reconstruction d'objets et leur identification successive à l'aide de capteurs tactiles pouvant reconnaître les objets de manière autonome facilitera également la recherche d'objets, telles des boîtes noires d'aéronefs tombés au fond des océans, ou d'autres objets dangereux ou localisés dans endroits difficilement accessibles à l'être humain.

Le présent travail est organisé comme suit : le chapitre 2 fait un survol de l'état de l'art actuel en matière de capteurs tactiles et de classification automatique d'objets ; le chapitre 3 présente les objectifs de ce mémoire, à la lumière des travaux récents présentés dans la littérature sur le sujet ; le chapitre 4 décrit la méthodologie de la présente recherche ; le chapitre 5 présente les résultats obtenus ; et le chapitre 6 présente les conclusions et détaille les contributions de ce mémoire, en plus de décrire les travaux futurs relatifs au présent effort de recherche.

Chapitre 2 : État de l'art

Cette section présente l'état actuel des technologies et des techniques utilisées pour reconnaître des objets à partir d'empreintes générées sur leurs surfaces par des capteurs tactiles. Elle présente également, en premier lieu, une brève description du toucher humain.

2.1. Toucher humain

Le toucher synthétique s'inspire du toucher humain. Une bonne connaissance de celui-ci est donc essentielle pour réussir à le reproduire. Oliveira et al. [4] identifient les structures adaptatives mécano-réceptrices les plus importantes du toucher humain, à savoir :

- les corpuscules de Meissner pour la détection de la vitesse et du mouvement sur la peau; ceux-ci sont également utilisés pour établir la distinction entre deux points ;
- les disques de Merkel pour ressentir les textures et les formes lors de touchers superficiels ;
- les corpuscules de Pacini pour ressentir les changements de pression, les pressions fortes et les vibrations à plus de 250 Hz ; et
- les corpuscules de Ruffini pour détecter les étirements de la peau, et les glissements lors de touchers et de pressions continus.

Hannaford, Allison et Okamura [5] indiquent également que les organes des tendons Golgi fournissent des informations importantes en ce qui a trait au positionnement et aux forces déployées par les muscles.

Des psychologues expérimentaux [6], cités dans [7] et [5] ont identifié six types de mouvements exploratoires permettant de distinguer les propriétés des objets : la fermeture (*enclosure*) pour déterminer la forme générale et le volume d'un objet ; la pesée (*hefting*) pour déduire le poids ; la pression pour déterminer la consistance et la dureté ; le contact statique pour déterminer la température ; la poursuite des contours et les glissements latéraux pour déterminer la texture. Il est attendu que le développement d'algorithmes intelligents reproduisant ces types de mouvements, pourra augmenter la capacité des robots à reconnaître des objets en les touchant, à la manière des humains.

Schopfer et al. [8] citent [9], pour démontrer que les humains ne reconstruisent pas les objets mentalement pour les reconnaître, mais font plutôt appel à d'autres caractéristiques, telles que la texture ou la conductivité thermique pour le faire. Ainsi, il n'y a pas vraiment lieu de reconstruire un objet pour le reconnaître ; il s'agit plutôt d'en dégager des caractéristiques uniques et discriminantes.

Okamura [10] considère que la manipulation et l'exploration tactile sont des processus itératifs qui mènent éventuellement à la reconnaissance d'objets. À mesure que l'exploration progresse, le capteur est en mesure de déterminer certaines caractéristiques de l'objet, telles que sa forme, son inertie, sa température, les propriétés de ses surfaces, etc., tout cela menant éventuellement à la reconnaissance de l'objet.

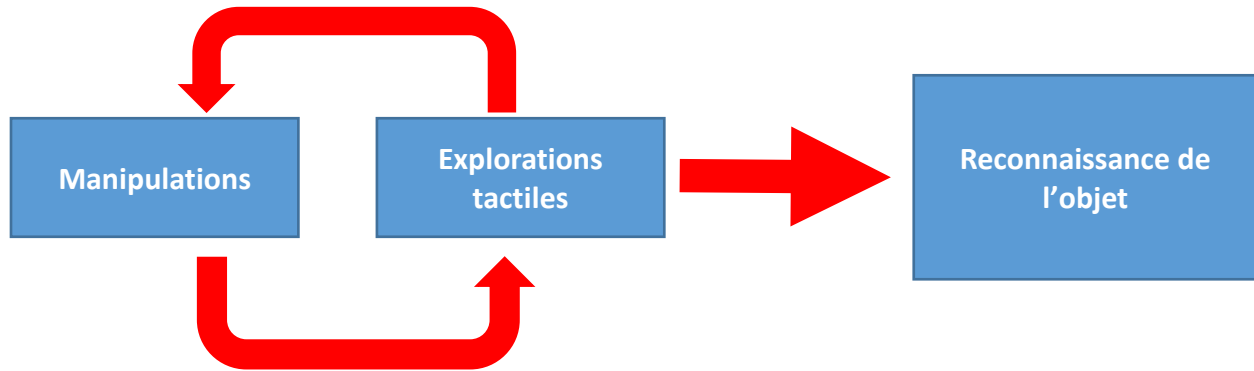


Figure 1 : Liens entre l'exploration et la manipulation (boucle haptique d'Okamura, adapté de [10]).

Ces dernières deux recherches suggèrent l'idée que, lors de l'exploration tactile robotique, il serait intéressant, au lieu de toucher la surface d'un objet aveuglement ou de manière aléatoire pour reconnaître ses propriétés et ainsi son identité, de toucher à des endroits précis, identifiés comme uniques et discriminants pour la forme de l'objet. Cette idée sera explorée sous une certaine forme dans le cadre du présent mémoire.

2.2. Les capteurs robotiques tactiles

2.2.1 L'haptique

Hannaford et Okamura [5] décrivent l'haptique comme étant la science et la technologie relatives à la production et la perception de sensations tactiles par un opérateur. Les capteurs

tactiles, quant à eux, répliquent ou améliorent ces sensations tactiles à l'aide d'appareils mécaniques et d'ordinateurs. L'haptique est principalement associée à deux fonctions du système nerveux : la kinesthésie, qui est la perception consciente de la position et des mouvements des différentes parties du corps, et les sensations tactiles. La première est généralement reproduite par les capteurs de force rétroactifs (*force feedback sensors*), et les secondes, par les capteurs tactiles.

Winiarski and Zielinski [11] donnent une explication alternative quant aux interfaces haptiques : selon eux, elles offrent aux humains un sens du toucher à distance. Le but des capteurs tactiles est donc de synthétiser le toucher humain et de le reproduire à l'aide de machines. Les objectifs peuvent toutefois varier : on peut vouloir reconnaître des surfaces et des textures afin d'éventuellement pouvoir identifier des objets, voire même générer des représentations virtuelles de ceux-ci, ou rapporter à un utilisateur à distance les sensations perçues par un capteur tactile.

Le présent mémoire ne s'intéressera qu'à l'acquisition de données par l'entremise des capteurs tactiles ; il ne s'attardera pas au retour haptique des sensations tactiles vers un opérateur.

2.2.2 Caractéristiques des capteurs tactiles

Les capteurs robotiques tactiles possèdent tous des caractéristiques semblables. Antonya [12] a décrit ces caractéristiques de la manière suivante :

- faible inertie ;
- rigidité ;
- peu de friction ;
- plan de travail relativement grand ;
- plusieurs niveaux de liberté ; et
- dotés d'une force semblable à celle des humains.

Un élément important à considérer dans le cas des capteurs tactiles est le choix des transducteurs, qui traduisent un type d'énergie en un autre, tel qu'une différence de potentiel,

dans le but de mesurer une variation. Giguère et Dudeck [13] définissent les caractéristiques d'un bon transducteur de la manière suivante :

- ne perturbe pas la quantité physique mesurée ;
- très haute sensibilité ;
- bonne résolution ;
- interférence faible ;
- précis ; et
- possède une large bande.

2.2.3 Capteurs tactiles actuels

La dernière décennie a vu l'apparition de nombreux capteurs tactiles composés de matériaux divers, du module le plus simple à la peau artificielle englobant une matrice de petits capteurs tactiles. La plupart des solutions proposées n'utilisent qu'une seule technologie, telle que les capteurs à effet Hall [14], [15], les baromètres [16], [17], les matrices de capteurs tactiles [18]-[25], les capteurs à fibre optiques [26],[27], d'autres types de capteurs optiques [28]-[32], et les accéléromètres [33],[2],[34] pour obtenir des données tactiles.

Yousefian et al. [14] ont conçu un capteur tactile déformable en forme de dôme. Un aimant dans le haut de la structure et quatre capteurs à effet Hall permettent d'évaluer la normale et la force appliquée sur le capteur. Les auteurs de [15] utilisent, quant à eux, deux capteurs à effet Hall et un aimant pour mesurer l'angle de déplacement des articulations des doigts d'une main robotisée, les données obtenues permettant ensuite de classifier des objets. Faria et al. [35] utilisent des capteurs électromagnétiques pour suivre les contours d'un objet. Ces capteurs, placés sur la main d'un manipulateur qui suit le long des contours de l'objet, enregistrent à intervalles réguliers des données relatives à la forme de l'objet.

Le capteur multimodal de [36] utilise un capteur de profils de pression, sous forme d'une matrice de résistances sensibles à la force (matériaux aux propriétés piézoélectriques « *Force Sensing Resistors* » ou « *FSR* ») reproduisant la fonction de la peau humaine pour raffiner des mesures prises sur des objets tridimensionnels dans des zones pré-identifiées, à l'aide d'un télémètre (*laser range finder*). Le système peut reconnaître de petits objets non-identifiables par

le télémètre en plus d'estimer leur forme et leur orientation. Les auteurs de [19] abordent le problème de reconnaissance d'objets à l'aide de profils de déplacement obtenus par des transducteurs tactiles sensibles à la force, et la reconnaissance de textures en utilisant un doigt robotisé, dont le bout est recouvert d'une surface tactile dynamique et qui exécute un mouvement de frottement sur l'objet. Des capteurs tactiles de type FSR sont utilisés pour traduire la pression sur une surface en des données de texture et de contours par Pasca [37] et Pasca et al. [36].

Pezzementi et al. [22] simulent la collection aveugle d'images tactiles telles qu'elles auraient été prises par un capteur tactile à profils de pression et proposent une solution d'analyse sac-de-caractéristiques (*bag-of-features*) pour estimer la probabilité de l'identité d'un objet. Schneider et al. [23] utilise une approche similaire, basée sur des images tactiles obtenues grâce à un capteur tactile matriciel de 6x14 éléments fixés au doigt d'un robot pour classifier des objets industriels. Dans [21], les données tactiles générées par des matrices tactiles de 3x8 sur chacun des trois doigts d'une main robotisée sont vues séquentiellement et utilisées pour classifier un ensemble d'objets, soit en mouvement ou immobile. Dans [16], la matrice tactile englobe trois puces électroniques entourées de polymère souple et équipées de capteurs barométriques. Les auteurs de [20] présentent une matrice tactile basée sur un caoutchouc piézo-résistant, pris entre deux couches de PVC et deux couches de fils conducteurs. Une pince robotisée avec des capteurs installés sur chacun des doigts effectue un mouvement de palpation pour obtenir de l'information tactile et pour classer les objets palpés. Dans [25], les changements de caractéristiques des résistances piézo-électriques sont proportionnels au stress longitudinal appliqué sur un capteur ayant pour fonction d'évaluer la rugosité. Sturm [52] emploie un capteur en forme de pince qui, en plus de capter des empreintes sur un champ de 6 x 14 sur chacune de ses pinces, capte également la distance entre les deux parties de celle-ci ainsi que leur hauteur par rapport à la base du capteur pour la reconnaissance d'objets touchés. Dans la même catégorie technologique, Dattaa et al. [38] utilisent un capteur composé d'une membrane capacitive de 32x32 cellules sensibles à la force pour extraire les caractéristiques discriminantes des objets. Les auteurs de [24] emploient des capteurs placés sur l'avant-bras d'un robot, sous la forme d'une matrice de 384 capteurs de type pression-capacitance pour collecter des données en vue de la reconnaissance des différentes catégories d'objets.

Fishel et Loeb [7] utilisent le doigt robotisé BioTac[®] composé d'un bloc solide recouvert d'une peau élastomérique, remplie de liquide. Lors du contact avec l'objet, des vibrations se propagent dans le liquide et sont ensuite interprétées par des capteurs de pression.

Une autre catégorie de capteurs tactiles utilise la fibre optique et des capteurs optiques. Park et al. [26] intègrent quatre fibres optiques dans l'exosquelette d'un doigt-robot pour détecter le contact et contrôler la force pendant la manipulation fine. Les réseaux de Bragg sur fibres optiques assurent une grande sensibilité et immunisent contre les interférences électromagnétiques. Le capteur tactile de De Maria et al. [28] contient une matrice de phototransistors-DEL intégrée à une coupole élastique et déformable. Celle-ci traduit les forces et les torsions appliquées sur elle en des déformations locales qui elles, produisent une variation de l'intensité de la lumière réfléchiée et du courant lumineux captés par les photo-détecteurs. Dans [29], une autre version du capteur précédent, basé toutefois sur le filtre étendu de Kalman, permet de retrouver la position et l'orientation de la surface en contact avec un objet rigide. Le coefficient de friction est ensuite évalué en tenant compte de la position et de l'orientation du plan de la surface de contact, couplées à la force du vecteur de contact mesurée par le capteur. Un algorithme de contrôle du glissement est proposé par [30] pour les tâches de manipulation. Corradi et al. [39] emploient un capteur tactile ayant des similitudes avec le capteur de [28], composé d'une membrane qui, en se déformant, créent des zones d'ombres, qui sont ensuite captées par une caméra qui les transforme finalement en impressions tactiles. Aggarwal et Kirchner [40] utilisent des capteurs à fibres optiques, placés deux par deux dans de la styromousse, dont l'une émet de la lumière, et l'autre, mesure l'intensité lumineuse. Lorsque le capteur établit le contact avec un objet, la styromousse se déforme, ce qui modifie l'intensité lumineuse lue par la deuxième fibre optique. Les lectures simultanées de plusieurs couples de fibres optiques permettent d'obtenir des empreintes tactiles d'objets.

Les auteurs de [31] ont développé un capteur basé sur la morphologie de la peau sur le bout des doigts humains et sur le rôle des corpuscules de Meissner dans l'encodage de l'information tactile pour les contours et les aspérités. Le capteur est composé d'un mince film de caoutchouc présentant des détails structuraux et intègre une couche de polymère souple et transparente. Une caméra CCD (*charge coupled device*) intégrée permet de suivre des marqueurs placés sur la paroi interne du capteur, illuminée par quatre DEL infrarouges, le tout permettant

de détecter des réflexions sur cette paroi. Ce capteur est utilisé par [32] pour détecter les contours d'un objet dans le contexte d'un robot ayant pour tâche de suivre des contours.

Dallaire et al. [33] utilisent un stylet métallique attaché à un accéléromètre à triples axes digitales pour obtenir de l'information de texture grâce à un mouvement de frottement, alors que la surface tourne sur une table tournante. Une approche non-supervisée est ensuite proposée pour l'identification de la surface (e.g. linoleum, cuirette, tapis-gazon, etc.). Sinapov et al. [34] utilisent également un accéléromètre à triples axes pour capter la rétroaction tactile à mesure qu'un robot gratte 20 surfaces différentes afin de reconnaître le matériau à l'aide d'algorithmes d'apprentissage.

Toutes les technologies sont assujetties à des contraintes spécifiques. L'utilisation de capteurs magnétiques est réservée aux environnements non-magnétiques. Les matrices tactiles sont généralement reconnues pour leur incapacité à mesurer les forces de contact [41], et leur rigidité nuit à la collecte de données sur des surfaces courbes. Les capteurs à fibres optiques sont dispendieux et difficiles à intégrer dans des modules tactiles puisque toute courbe entraîne des pertes de transmission lumineuse [28]. Les solutions basées sur des caméras CCD sont également dispendieuses et lourdes, alors que la perte de luminosité causée par des micro-plies et l'interférence distordent les données [41]. Afin de partiellement éliminer les désavantages causés par l'utilisation d'une seule technologie, quelques solutions utilisent plusieurs capteurs pour obtenir des données tactiles tels [2], [33], [42], [43]. Chathuranga et al. [42] ont développé un capteur en forme de bout de doigt qui intègre sept mini-capteurs flexibles de forces et cinq accéléromètres dans une couche de polyuréthane, le tout enveloppé dans une couche de caoutchouc striée et sillonnée. Ce capteur a été utilisé pour classifier des échantillons de bois en frottant leur surface. Les auteurs de [43] présentent une approche basée sur l'analyse des données pour résoudre le problème de la sélection et de la discrimination des contours d'un objet par un capteur. Les données obtenues des moteurs, également par des unités de mesure d'inertie, et un magnétomètre attaché à un doigt robotisé à quatre degrés de liberté, durant l'exploration de sept figures virtuelles, sont analysées par analyse de composantes principales ; un perceptron multi-niveau est ensuite entraîné pour classer les figures. Le capteur de Kromer et al. [2] est composé d'une aiguille flexible qui entre en contact avec la surface ainsi que d'un microphone à condensateur qui détecte les vibrations de l'aiguille. Un algorithme de fusion multi-sources de

données combine les enregistrements de l'aiguille avec des données visuelles obtenues avec une caméra pour classer des surfaces texturées.

Bordegoni et al. [44] ont mis au point un capteur haptique déformable, intégrant plusieurs technologies, dont un mécanisme servo-contrôlé pouvant suivre les contours d'un objet et permettant à l'utilisateur de sentir la surface de l'objet avec sa main libre. Le but du capteur est d'être plus intuitif que les autres afin de pouvoir créer des objets virtuels « avec ses mains ».

2.3. Stratégies d'exploration robotique

La perception tactile n'est pas passive [45]. Le capteur doit explorer, toucher, chercher et aller au-devant des objets pour les percevoir. Selon les auteurs de [45], la « *perception active est un problème pour un processus intelligent d'acquisition de données* » (traduction libre). De plus, l'acquisition de données tactiles nécessite un contact direct avec l'objet. En ce qui a trait à la reconnaissance d'objets, les mouvements et le positionnement du capteur est long et fastidieux, d'où la nécessité d'une exploration intelligente et informée de l'objet.

2.3.1 Stratégies d'exploration locales et globales

Luo et al. [50] établissent la distinction entre deux types d'exploration : locale et globale. L'exploration globale s'étend normalement au-delà de la pointe du capteur. L'exploration locale se réfère aux parties d'un objet qui entrent complètement sur le bout des capteurs tactiles, aussi appelées information fractionnaire des objets. Aggarwal et Kanpmann [46] définissent une exploration sommaire (« *coarse exploration* »), qui permet d'emblée la reconnaissance de certains objets. Ils définissent également l'exploration locale comme étant la « *fine exploration* », l'exploration fine, où le capteur prend des mesures sur la surface même de l'objet, pour en déterminer les contours, voir même la texture. Les données sont prises très près l'une de l'autre. Les auteurs de [46] précisent également que, pour l'exploration globale, le capteur enregistre des données aléatoirement et de manière non-continue, à l'intérieur d'un espace de travail connu préalablement, pour délimiter grossièrement les limites physiques de l'objet.

Pezzementi et al. [22] décrivent une stratégie d'exploration où le capteur échantillonne globalement un objet à l'aide d'un bras robotique articulé virtuel, auquel on a associé une caméra orthographique tridimensionnelle dont le volume de visualisation est composé d'une couche

déformable dans laquelle les objets virtuels peuvent pénétrer. Un contrôleur est utilisé afin d'orienter le capteur pour maximiser sa surface de contact avec l'objet. Le but de la fonction d'orientation est de placer le capteur de la manière la plus coplanaire possible afin de maximiser la qualité et l'information de l'empreinte. Des séries d'empreintes locales sont enregistrées avec ce dispositif. Les auteurs ont également démontré que l'utilisation combinée de stratégies globales et locales permettent d'obtenir de bons résultats.

Okamura [47] évoque, dans le cadre de l'exploration, la nécessité d'un système dit « intelligent », mis à jour continuellement, semblable à l'exploration bayésienne [48], [7]. Fishel et Loeb [7] ont proposé l'idée d'exploration bayésienne pour déterminer le bon mouvement à effectuer pour mieux distinguer la texture. Avec cette méthode, les résultats préalablement obtenus par le capteur sur l'objet sont utilisés pour déterminer le prochain mouvement à effectuer afin d'obtenir l'information la plus discriminante possible en ce qui a trait à la nature de l'objet. L'exploration bayésienne a également été mise à contribution dans [48]. Le but est de trouver des mouvements qui sont peu susceptibles d'engendrer de la confusion entre deux ou plusieurs objets. Les auteurs ont donc utilisé des techniques de renforcement et une équation pour distinguer les mouvements discriminants de ceux qui le sont moins. Ils ont également introduit les notions de conformité (*compliance*), c'est-à-dire le ratio déformation-force appliquée, de texture, soit la capacité à ressentir des micro-vibrations causées par l'objet sur la surface du capteur, et de conductivité thermique, qui permet de distinguer la température d'un objet pour identifier la composition matérielle de l'objet. Leur taux de reconnaissance avoisinait les 99% pour 10 objets.

Lepora et al [49] ont utilisé la perception bayésienne active pour déplacer leur capteur. Ils ont utilisé une boucle de rétraction positionnelle pour réaligner le capteur après chaque prise d'empreinte. Ils n'ont toutefois pas fourni de résultats en matière de reconnaissance d'objet.

Luo et al. [50] tentent d'orienter selon la normale la surface plane de leur capteur afin d'obtenir des empreintes les plus évocatrices possibles.

Aggarwal et Kampann [46] ont obtenu de bons résultats en faisant travailler les deux stratégies d'exploration (globale et locale) dans les mêmes séquences d'exploration, en alternance. Dans leur cas, ils ont défini les actions locales comme étant celles effectuées alors que le poignet de leur robot était fixe. Ces actions étaient, par exemple, un resserrement ou un remplacement des trois doigts du capteur pour mieux toucher l'objet.

Yeung et al. [51] décomposent le toucher humain en deux modes : un mode passif et un mode actif ; le mode passif détecte les contours, les textures et la température, tandis que le mode actif intègre des données relatives à la position des articulations aux données tactiles. Le mode actif est donc une stratégie globale, et le mode passif, une stratégie locale.

2.3.2 Stratégies d'exploration informées et non-informées

Sturm [52] compare des stratégies d'exploration informée et non-informée ; pour cette dernière, Sturm se fie aux repères visuels du manipulateur humain pour intuitivement déterminer des sites de prélèvement d'images significatifs et évocateurs. Dans le cas des stratégies informées, l'objectif est de déterminer l'action qui maximise le gain d'information. Dans le cas mentionné ci-haut, les stratégies informées se sont avérées supérieures. Le présent travail utilise des stratégies informées puisqu'il détermine au départ les points d'intérêt où obtenir des empreintes sur les objets.

2.3.3 Autres stratégies d'exploration de surface

Dans [53], Phung et al. décrivent deux techniques d'exploration de surfaces bidimensionnelles ; la première analyse la surface au complet, tandis que la deuxième est en mesure de suivre les crêtes (*edges*). Ces deux algorithmes utilisent un capteur tactile simple de 2x2 éléments. Celui-ci s'avère d'ailleurs plus grand que les surfaces qu'ils explorent. Ces deux techniques tiennent compte des données de l'empreinte présente (empreinte n) afin de déterminer le mouvement qui mènera vers la position de la prochaine empreinte (empreinte $n+1$). Pour ces deux techniques, le parcours et les mouvements subséquents du capteur sur l'objet sont déterminés par les combinaisons des signaux reçus ou non par les quatre cellules de sa paroi. Par exemple, dans le cas de l'algorithme de reconnaissance, si les cellules 2 et 4 n'ont pas de signal après un mouvement vers la gauche, le capteur ne continuera pas vers la gauche, mais se dirigera plutôt vers le haut.

Schopter et al. [8] font état d'une stratégie inflexible et unique pour des objets tridimensionnels. Le parcours du capteur est déterminé d'avance selon quatre angles différents, et les objets sont retournés sur eux-mêmes pour engendrer de nouvelles captures.

Odiase et Richardson [54] utilisent des équations polynomiales permettant de calculer les angles et accélérations nécessaires pour passer d'une position d'acquisition à l'autre dans un temps prédéfini.

Nous sommes d'avis que les points sur lesquels les empreintes des objets sont prises doivent avoir une signification ; ils ne peuvent être complètement aléatoires et doivent avoir une valeur intrinsèque pouvant éventuellement mener à l'identification de l'objet. Nous proposons donc d'utiliser une approche informée, sous forme d'algorithmes visuels de détection de points d'intérêt mis à profit afin d'identifier au préalable les endroits où seront prises les empreintes tactiles. En s'inspirant de l'expérience humaine en matière de toucher et de vision qui démontre l'utilité de la vue pour saisir et manipuler des objets, les algorithmes utilisés dans le cadre de ce travail, sauf un, sont inspirés directement de l'attention visuelle humaine pour identifier les régions qui attirent l'attention sur la surface des objets.

2.3.4 Les textures et leur importance

En matière de toucher synthétique, deux axes importants se dessinent : d'un côté, les empreintes issues de contacts statiques, et de l'autre, le toucher dynamique, qui génère des profils de contact permettant de discerner des textures. Ainsi donc, la notion de texture implique nécessairement un mouvement du capteur sur l'objet (Katz, 1925, dans Fishel et Loeb [7]). Cutkosky et al. [55] indiquent d'ailleurs que le mouvement fait partie intégrante de l'exploration et de la reconnaissance sensorielle.

Dallaire et al. [33] et [4] ont tous deux développés des dispositifs expérimentaux qui créent un mouvement du capteur par rapport à l'objet, à la manière de doigts qui bougent sur une surface, pour en distinguer la texture. Dans [33], les auteurs utilisent une table tournante sur laquelle diverses surfaces sont déposées. Le capteur fixe est ainsi stimulé par les disques de divers matériels qui passent sous son aiguillon, à la manière d'un tourne-disque conventionnel. Dans [4], ils utilisent un bras robotisé qui bouge le long de la surface pour enregistrer ces mêmes données. Dans ces deux cas, c'est la réponse engendrée par la surface à mesure qu'elle défile sur le capteur qui engendre la notion de texture.

Le présent mémoire entend se concentrer sur l'exploration statique et séquentielle, et non pas sur l'exploration dynamique, pour identifier des objets à partir d'empreintes obtenues à l'aide d'un capteur tactile.

2.4. Méthodes d'analyse, d'interprétation et de classification des données tactiles

Un large éventail de techniques d'analyse de données, incluant les arbres de décisions, les sacs de caractéristiques, les réseaux neuronaux, les machines de vecteurs de support (*SVM*), etc. ont été testées afin d'interpréter les données tactiles dans des tâches telles que la reconnaissance de textures, l'identification de matériaux, la détection de glissement pour des applications robotisées et l'identification d'objets. Les prochains paragraphes décriront les principales d'entre elles.

Bien qu'il soit possible de classifier les objets en analysant directement les taxels (*tactile pixels*, pixels des empreintes), il s'avère généralement plus avantageux d'en extraire des caractéristiques intermédiaires et de créer le modèle de classification à partir de celles-ci. Au lieu de travailler directement avec les taxels, la méthode sac-de-choses (*bag-of-things*) extrait d'abord des caractéristiques intermédiaires spécifiques de celles-ci avant de procéder à la classification [56]. Luo et al. [50] ont fait usage de la technique sac-de-choses, mais en la modifiant pour utiliser également la technique appelée *Ensemble Manifold Regularization (EMR)*. Rui et al. [57] ont eux aussi utilisé une approche sac-de-choses, ou sac-de-systèmes dans leurs cas, en la couplant à d'autres techniques. Ils ont d'abord extrait des caractéristiques des objets et les ont modélisés en systèmes linéaires dynamiques ; ils ont ensuite utilisé la distance de Martin pour créer des grappes et générer un sac-de-systèmes envoyé à une Machine Apprenante Extrême (*Extreme Learning Machine ou ELM*). Au final, ils ont prouvé que les résultats de classification de l'ELM sont supérieurs à ceux d'une machine à vecteurs de support (*SVM*).

Dattaa et al. [38] utilisent des ondelettes pour extraire des caractéristiques discriminantes des objets, suivies des algorithmes tels que *K* plus proches voisins, *Naïve Bayes* et l'analyse discriminante linéaire (*LDA*) pour leur identification. Corrdi et al. [39] utilisent la dérivée discrète pour extraire des caractéristiques des empreintes tactiles, bien qu'une fois celles-ci analysées, elles s'avèrent moins efficaces que les données brutes. Dans ce travail, plusieurs manières sont explorées pour l'encodage de données, incluant l'analyse en composantes principales, les moments Hu et Zernike ainsi que plusieurs techniques supervisées pour la classification des données (p.ex. *K* plus proches voisins, réseaux de neurones, *SVM* et *Naïve Bayes*). Selon les résultats obtenus, l'analyse des composantes principale appliquée sur les

moments Zernike, suivi de l'algorithme K plus proches voisins pour la classification des données donne le meilleur taux de classification (96.4%).

Les arbres de décision sont utilisés par [8] pour classifier et identifier des objets. Les auteurs utilisent préalablement l'algorithme C4.5 pour regrouper les objets, avant de tenter de les identifier. Carotenuto et al. [58] ont défini un classificateur flou pour classifier des données tactiles primitives. Au final, cette technique s'apparente aux arbres de décision.

Sturm [52] base sa classification d'objets sur l'extraction de deux images tactiles sous forme des matrices $6 \times 14 \in \{0,1\}$ prises aux antipodes avec une pince. La méthode sac-de-caractéristiques (*bag-of-features*) est ensuite employée pour classifier les objets. La performance obtenue est meilleure pour des objets familiers (96,2%) que pour des objets industriels (58%).

Dans [59], les auteurs extraient premièrement des zones de hautes pressions à partir des données tactiles; la classification est ensuite faite par des séparateurs à vastes marges linéaires. D'autres caractéristiques sont extraites de ces données pour faire de la reconstruction 3D. Les deux types de données sont utilisés ensemble pour retenter une classification (modèle hybride).

Babu et al. [60], extraient à partir des données tactiles des caractéristiques à l'aide de la distance moyenne entre les centroïdes et leurs variances, ainsi que l'excentricité. Au lieu de ne simplement utiliser qu'une seule méthode d'analyse à la fois, ils utilisent plusieurs techniques dont un arbre de classement (C4.5 ou J48) et un classificateur *Naïve Bayes*, et unissent ensuite leurs efforts respectifs avec un classificateur votant. De cette manière, ils obtiennent une performance de 96% pour la reconnaissance des objets (2D seulement et sur des formes très rudimentaires) à partir d'empreintes tactiles.

Les auteurs de [24] emploient l'analyse en composantes principales (*PCA*) et ensuite le regroupement en grappes k -moyennes (*k-means clustering*) pour la reconnaissance des catégories d'objets (rigides fixes, rigides amovibles, mous fixes, mous amovibles ; ensuite, amovibles et fixes) en utilisant des données tactiles sous la d'une matrice de 384 capteurs de type pression-capacitance pour collecter des données en vue de la reconnaissance des différentes catégories d'objets ainsi que des données relatives à l'acquisition de ces données. Les performances obtenues varient entre 82% pour 4 catégories et 92% pour 2 catégories. Chebotar [61] a également mis à profit la formation de grappes (*clustering*) et l'analyse en composantes principales dans ses travaux, sans toutefois fournir de résultats.

L'utilisation des réseaux neuronaux pour classifier les objets à partir de leurs empreintes a également donné de bons résultats, et ce, à plusieurs reprises. Giguère et al. [13], Pasca et al. [36] et de Oliveira et al. [4] ont tous rapporté des performances de plus de 90% de précision pour la classification supervisée des objets. Dallaire et al. [33] propose une approche non-supervisée pour l'identification de la surface (e.g. linoleum, cuvette, tapis-gazon, etc.). Castillo-Muniz et al. [62] ont aussi démontré que les réseaux neuronaux pouvaient être utilisés pour reconstruire des objets tridimensionnels, à l'aide de réseaux neuronaux de type GRNN (*General Regression Neural Network*). Dans leur cas, ils ont extrait les caractéristiques de masse, rigidité et absorption, sans toutefois fournir de données en matière de reconnaissance des objets.

Aggarwal et Kirchener [40] ont démontré avec succès une approche nommée *Batch Ransac and Iterative Closest Point augmented Particle Filter (BRICPPF)* pour explorer et reconnaître des objets, ainsi qu'une deuxième approche utilisant la première, mais axée sur les recherches de contours et les sous-parties des objets, inspirée de la biologie. Les taux de reconnaissance de cette recherche sont de 80 à 100%, et ce, pour les deux techniques, avec un léger avantage pour *BRICPPF*.

Certains auteurs ont utilisé des approches aux antipodes des techniques d'apprentissage. Parmi ceux-ci, notons Keren et al. [63]. Les auteurs ont utilisé une approche algébrique en définissant des invariants différentielles représentant les surfaces courbes d'objets simples (ex. sphère, cylindre, cube). Les auteurs de [64] ont utilisé une approche similaire pour déterminer la courbature d'un objet simple, avec des résultats mitigés. Odiase et Richardson [54] ont déterminé les périmètres d'objets simples (cubes et sphères) pour tenter de les reconnaître. Dans [69], les auteurs extraient des facteurs de Fourier, qui sont ensuite utilisés avec divers classificateurs (p.ex. *Naïve Bayes*, arbres de décision) pour la reconnaissance des textures, pour laquelle ils ont obtenu un taux de réussite de 94%. Kim et al. [65] ont obtenu un taux de réussite de 64% en matière de reconnaissance de textures, en utilisant un capteur de 4X4 et l'estimation de la probabilité maximale.

2.5. Conclusions sur l'état de l'art

Malgré toutes les avancées technologiques récentes, les capteurs tactiles actuels et les techniques associées pour la collecte et l'interprétation des données récupérées par ces capteurs restent limités en matière de reconnaissance d'objets, d'où le besoin d'en développer de

nouveaux ainsi que d'améliorer de manière intelligente le processus d'acquisition, d'interprétation et d'intégration des données collectées. Les capteurs actuels les plus répandus pour la reconnaissance d'objets à partir des données tactiles sont les capteurs résistifs sensibles à la force (*force sensing resistors*, ou *FSR*). Ceux-ci permettent d'évaluer les forces appliquées sur une surface, et de déterminer des profils de pression. À partir des *FSR*, un grand nombre de capteurs ont été développés, tel que le BioTac®, un doigt artificiel recouvert de *FSRs* et d'une couche de gel, le tout englobé dans une membrane de caoutchouc. Lorsque cette membrane touche un objet, le gel transmet des vibrations aux *FSR*, ce qui permet d'établir un profil de pression caractéristique.

La fibre optique est également utilisée pour concevoir des capteurs. Ceux-ci peuvent détecter des variations lumineuses alors qu'ils s'approchent d'objets. Les caméras CCD (*charged coupled devices*) sont également utilisées pour concevoir des capteurs. Celles-ci peuvent suivre des marqueurs placés sous la surface d'un dôme se déformant au contact d'un objet. Les capteurs à effet Hall fonctionnent de manière semblable, mais les caméras CCD sont ici remplacées par des détecteurs à effet Hall, et les marqueurs, par des aimants placés sous la surface du dôme.

La détection de texture s'avère également importante. Des capteurs utilisant des accéléromètres ont été utilisés pour effectuer cette tâche.

L'exploration tactile est un processus actif qui nécessite un contact direct avec l'objet. Les stratégies d'exploration tactile peuvent être informées ou non-informées. Les stratégies informées ou intelligentes (p.ex. l'exploration bayésienne, [48], [7]) évaluent la progression du capteur à la surface de l'objet au fur et à mesure que celui-ci génère des empreintes. Elles peuvent également utiliser des algorithmes de détection de points d'intérêt, tout cela afin de maximiser le gain d'information. Dans le cas des stratégies non-informées, elles s'en remettent au point de vue du manipulateur pour planifier le parcours du capteur sur la surface de l'objet. L'exploration peut aussi être locale quand elle s'en remet aux zones situées tout prêt de la pointe du capteur, et globale quand le capteur bouge bien au-delà de sa pointe pour aller chercher de nouvelles empreintes.

La classification des objets peut se faire directement à partir des empreintes brutes, ou après avoir extrait certaines caractéristiques bien précises de celles-ci. Les techniques les plus utilisées sont les arbres de décision, les sacs-de-caractéristiques, la technique *Naïve Bayes*,

l'algorithme K plus proches voisins, l'analyse en composantes principales et les réseaux neuronaux.

Au final, nous sommes d'avis que les futurs capteurs tactiles et leurs techniques associées pour la collecte et l'interprétation des données devront s'inspirer davantage de la biologie humaine, et intégrer des flux de données provenant d'autres types de capteurs, notamment des capteurs visuels. À ce titre, dans le contexte du présent travail, nous explorerons la capacité des informations visuelles, en particulier des informations extraites par des algorithmes inspirés de l'attention visuelle humaine, afin d'aider le processus d'acquisition des données tactiles et ainsi améliorer la capacité des algorithmes intelligents de reconnaître des objets à partir de ces données.

Chapitre 3. Objectifs

Le but du présent mémoire est de proposer une nouvelle technique permettant d'identifier et de reconnaître des objets tridimensionnels à partir de données recueillies par un capteur tactile qui, en touchant ces objets, génère une série d'empreintes caractéristiques.

La démarche à suivre pour atteindre ce but est divisée en plusieurs étapes : d'abord, il faut établir un contact physique avec l'objet. Le déplacement et le positionnement du capteur pour réussir à établir ce contact sont généralement coûteux en temps d'exécution ; il importe donc d'optimiser la stratégie de déplacement de celui-ci afin d'obtenir des données de qualité. Conséquemment, un soin particulier est accordé à l'étape d'acquisition, afin de la raffiner de façon à ce qu'elle dirige, automatiquement et de manière indépendante, le capteur vers les zones les plus significatives de l'objet. À ce titre, divers algorithmes de détection des points d'intérêt inspirés de l'attention visuelle humaine [66], [67] seront testés afin d'identifier les points d'intérêts sur des objets, pour ensuite diriger le capteur vers ces points. Cette stratégie d'acquisition des données est dite informée. Un autre algorithme de détection de points d'intérêt, HKS [67], sera également testé à titre comparatif.

Puisque l'acquisition de données réelles nécessiterait énormément de temps, nous simulerons dans une première étape celles-ci à l'aide d'objets virtuels, sur lesquels nous récolterons des empreintes en interceptant un plan virtuel, représentant la surface d'un capteur résistif sensible à la force (*FSR*), avec leurs surfaces. Ce plan subira ensuite de légères translations successives vers l'intérieur de l'objet, et pour chaque translation, nous récolterons les points d'intersection de ce plan avec l'objet virtuel, créant une série de courbes de niveaux. Chacune des séries de courbes de niveaux deviendra des empreintes locales de nos objets virtuels. Cette approche est similaire à celle de Luo et al. [50] et de Pezzementi et al. [22] dans le sens où on tente d'orienter la surface plane du capteur virtuel selon la normale et de la manière la plus coplanaire possible par rapport à la surface de l'objet afin de maximiser la qualité et l'information de chaque empreinte.

Dans une étape ultérieure, nous récolterons également des données sur un nombre réduit d'objets réels en utilisant un capteur résistif sensible à la force.

Une fois les données acquises, nous en extrairons certaines caractéristiques, telles que: des statistiques globales (moyenne, médiane, déviation standard, etc.) ; des histogrammes, la

distance logarithmique et la couleur des blocs; ou encore, nous réduirons la taille des données pour chaque empreinte en utilisant la technique du sous-échantillonnage; nous proposerons ensuite des techniques d'apprentissage pour la reconnaissance des objets à partir de ces caractéristiques.

En résumé, les objectifs de ce projet sont les suivants :

- l'étude de l'impact de l'utilisation des points d'intérêt obtenus par des algorithmes de détection inspirés de l'attention visuelle sur la collecte des données tactiles et la reconnaissance des objets à partir des données acquises ;
- la conception et l'implémentation d'une technique pour la simulation de l'acquisition des données tactiles sous forme d'empreintes (profils de pression) aux endroits identifiés par les points d'intérêt sur des objets 3D ;
- la conception et l'implémentation d'une technique (ou une combinaison des techniques) d'apprentissage pour la reconnaissance d'objets à partir des empreintes obtenues ;
- la validation de la solution proposée sur une base de données virtuelle ; et
- la validation de la solution proposée sur des données réelles.

Nos tests, analyses et calculs seront effectués à l'aide de la plateforme Matlab pour la génération des empreintes sur les objets visuels, et avec le logiciel RapidMiner pour la reconnaissance d'objets.

Chapitre 4 : Méthodologie

Le processus de capture et d'analyse d'empreintes est divisé en plusieurs étapes, tel qu'illustré à la figure 2.

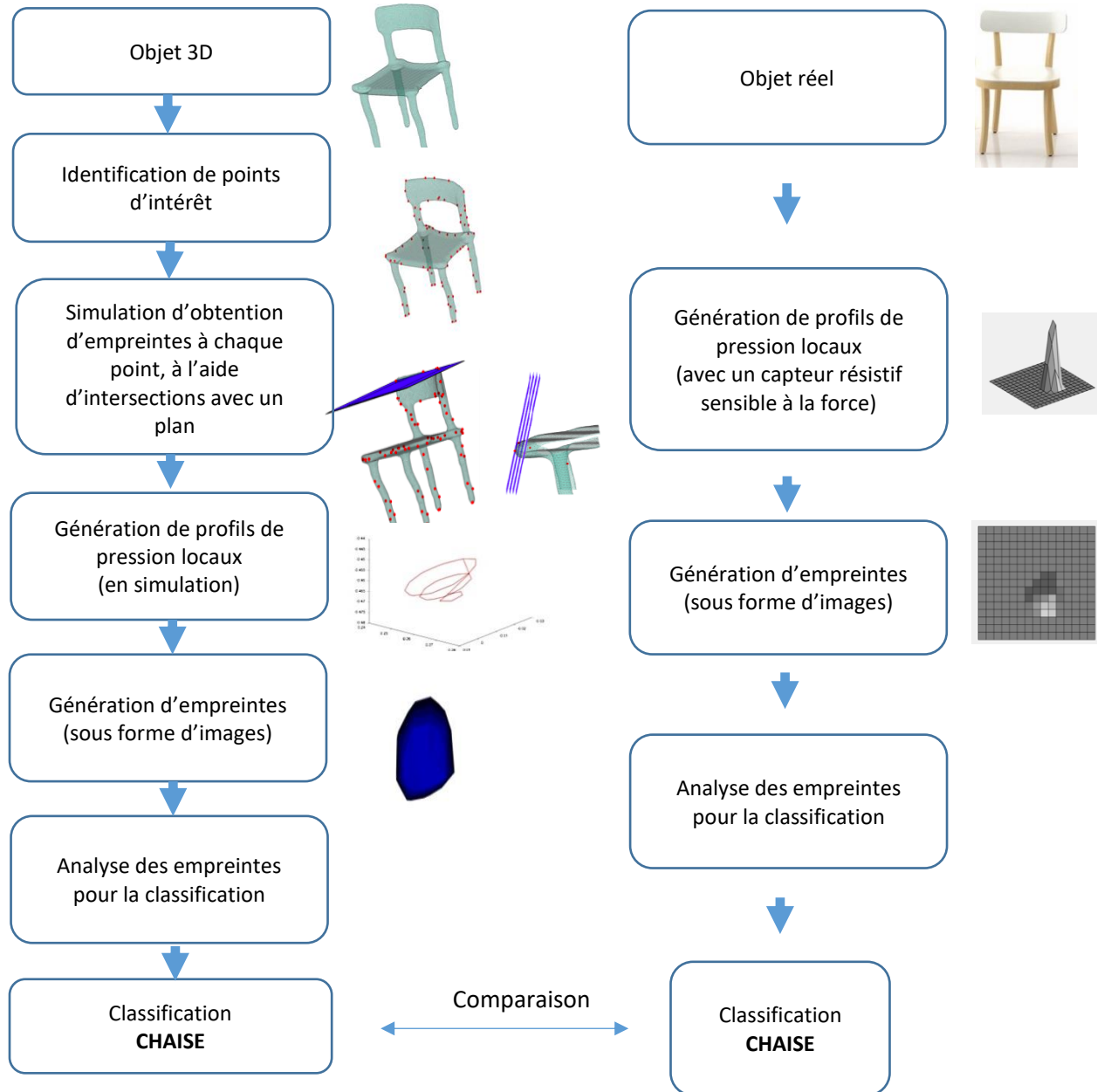


Figure 2 : Processus de capture et classification d'empreintes. À gauche, le processus pour les objets virtuels, et à droite, celui pour les objets réels.

Notre approche de reconnaissance d'objets utilise des données tactiles. À partir d'un objet virtuel tridimensionnel, des points d'intérêt sont identifiés à sa surface. Pour des objets réels, un capteur de couleur et profondeur de type Kinect peut être employé pour la collecte 3D. Des logiciels tels que Kinect Fusion ou Skanect peuvent ensuite être utilisés pour reconstruire la surface entière d'un objet par l'intégration des vues multiples obtenues par la Kinect. Le maillage de la surface résulte en un modèle semblable aux modèles virtuels utilisés. Les algorithmes d'identification de points d'intérêt à la surface d'un objet 3D s'inspirent, en tout ou en partie, de l'attention visuelle humaine. L'utilisation de données visuelles guidera le capteur tactile vers des régions caractéristiques de l'objet afin de réduire le temps d'exploration, qui peut parfois être long à cause des mouvements complexes nécessaires pour obtenir des données tactiles. Pour chacun des points d'intérêt, on capte l'intersection de plans quasi-tangents à l'objet et équidistants les uns des autres. On oriente la surface plane du capteur virtuel (et du capteur *FSR* réel) perpendiculairement à la normale afin d'obtenir des empreintes les plus évocatrices possibles. Les courbes de niveaux générées par l'intersection des plans avec la surface d'un objet sont ensuite projetées en 2D pour créer des empreintes sous la forme des images, desquelles on extrait des caractéristiques particulières. Ces dernières sont finalement nettoyées afin d'éliminer les caractéristiques avec des valeurs constantes qui nuisent à l'apprentissage et utilisées dans une série de classificateurs afin d'identifier le meilleur d'entre eux.

Un processus semblable, montré de côté droite de la figure 2, est ensuite appliqué pour une série d'objets réels qui seront également classés à l'aide des mêmes classificateurs.

4.1. Capteurs de profils de pression *FSR*

Le capteur que nous simulerons et utiliserons par la suite dans le cadre de ce mémoire est un capteur résistif sensible à la force. Ce type de capteur est un des plus développés et des plus documentés parmi tous les capteurs tactiles. Son principe est simple : il est composé d'un groupe d'éléments sensibles à la force appelés taxels ; chaque taxel renferme deux électrodes superposées. Quand ces électrodes, placées de chaque côté d'un matériau sensible à la pression et conducteur, se rapprochent à cause d'une pression appliquée sur les plaques réceptrices, la résistance électrique à l'intersection des électrodes décroît. Ce changement de résistance est proportionnel à la force normale appliquée sur les plaques. Historiquement, les capteurs résistifs

sensibles à la force sont utilisés pour faire du profilage tactile statique [36], et plus récemment pour du profilage dynamique [20] [21] pour classifier des objets en reproduisant un mouvement de palpation.

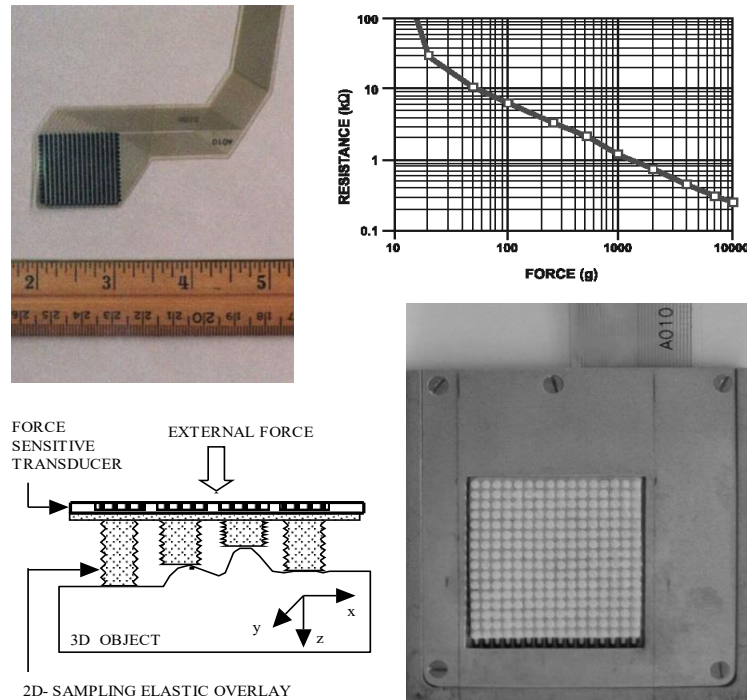


Figure 3 : Diverses représentations d'un capteur résistif sensible à la force (FSR) et ses caractéristiques [37].

Dans le cadre du présent mémoire, nous utiliserons le capteur *FSR* développé par Pasca [37] (figure 3) dans le cadre d'une application de profilage local tactile statique. Muni d'un circuit à interface d'analyse de signal, celui-ci est en mesure de capter et de décoder l'information tactile. Le capteur possède deux multiplexeurs analogues, le premier servant à déterminer la position des rangées, et le second, la position des colonnes, tout cela en mesurant la différence de potentiel à la sortie des colonnes. Celle-ci est mesurée par un convertisseur A/C placé sur un microcontrôleur PIC de 16 bits. En plus de la sélection des adresses du *FSR*, celui-ci effectue plusieurs fonctions de contrôle, telles que l'entreposage de données en mémoire et la communication avec l'ordinateur sur un port série. L'empreinte tactile telle que décodée par l'interface, est une matrice 16x16 de données représentant un déplacement selon la normale d'un profil géométrique tridimensionnel. Un exemple de profil local est montré à la figure 4a. Dans le cadre de ce travail, les empreintes seront projetées le long de l'abscisse pour obtenir un encodage

de l’empreinte sous la forme d’une image (figure 4b). Les images résultantes pourront ensuite être analysées afin d’identifier l’objet ayant été mis en contact avec la surface du capteur.

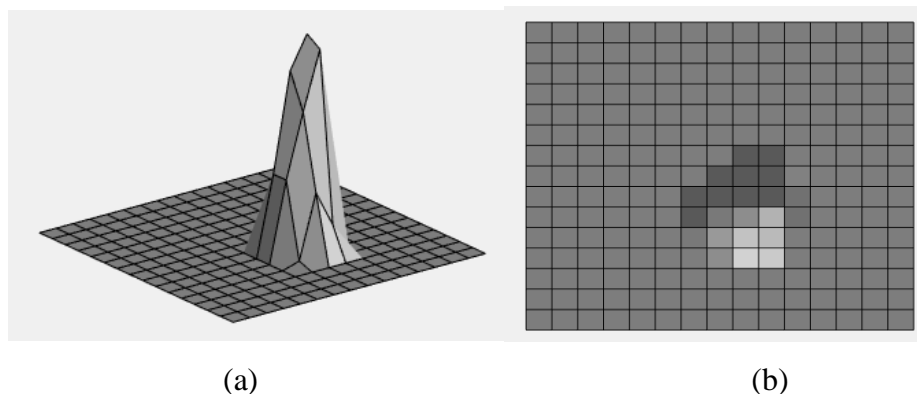


Figure 4 : (a) Profil de pression, et (b) empreinte sous forme d'image obtenus à l'aide d'un capteur FSR.

4.2. Détection des points d'intérêt

Les expériences tactilo-visuelles humaines démontrent l'importance de la vision pour faciliter la préhension et la manipulation. L'intégration de la rétroaction visuelle avec les stimuli tactiles peut compenser pour l'imprécision des systèmes visuels causée par les occlusions et pour la difficulté qu'ont ceux-ci à juger la force appliquée. De plus, l'utilisation de stimuli visuels permet de guider la collecte tactile vers des zones caractéristiques afin de minimiser le temps nécessaire à l'exploration, qui tend à être long puisque le manipulateur doit exécuter de multiples mouvements complexes afin d'obtenir des données tactiles. Ces aspects nous ont incités à explorer l'utilité de l'attention visuelle pour détecter des points d'intérêt sur lesquels nous avons ensuite obtenu des empreintes susceptibles d'attirer l'attention d'un observateur, c'est-à-dire des zones caractéristiques d'un objet donné.

Des modèles informatisés de l'attention visuelle ont été au cœur de bon nombre de recherches dans plusieurs champs d'étude [17], de la vision et du graphisme informatique (pour des tâches telles que la segmentation d'image, la détection de la qualité, la compression vidéo, la localisation d'objets saillants, la détection et la reconnaissance ainsi que la détermination de points d'intérêt), jusqu'à la robotique (pour la localisation robotique, la navigation et la vision active). Le rôle de l'attention visuelle est de segmenter le problème de compréhension d'une scène en une série de problèmes informatiques visuels, localisés et nécessitant moins de puissance de calculs [37], et de décider l'ordre dans laquelle les régions les plus intéressantes

doivent être analysées. Ce processus définit donc l'ordre dans lequel les yeux se positionnent [37], quand la fovéa se fixe sur des régions particulières de l'objet, maximisant le focus sur ces régions, rendant les régions centrales de la vision plus claire (c'est-à-dire le mécanisme concentrique du champ réceptif visuel humain).

La plupart des modèles numériques de l'attention visuelle sont basées sur des approches du bas vers le haut (*bottom-up*), concentrées sur les caractéristiques d'une scène qui captent l'attention [17] lors de conditions de visionnement libres si elles sont suffisamment distinctives par rapport à leur environnement.

Nos travaux ont fait usage de cinq algorithmes d'identification de points d'intérêt, soit l'algorithme "points saillants" [68], l'algorithme de "sailance du maillage" [70] (les deux inspirés de la vision humaine), deux algorithmes basés sur des modèles numériques de l'attention visuelle [66], [82], ainsi que l'algorithme "signature thermique" (*Heat Kernel Signature, ou HKS*, [69]). Les prochaines sections donneront de brèves descriptions de chacun d'entre eux.

4.2.1. L'algorithme "Points Sillants" (PS, *Salient points*)

Développé par Castellanni et al. [68], cet algorithme identifie les points saillants d'un objet à partir de différentes perspectives de celui-ci. En s'inspirant du mécanisme concentrique du champ réceptif visuel humain utilisé par la fovéa, les auteurs de [68] détectent des points d'intérêt basés sur un filtre de la différence de gaussienne qui simule ce mécanisme. Cet algorithme est divisé en trois étapes. À l'étape 1, la détection s'effectue à l'aide d'une mesure de sailance tridimensionnelle permettant d'identifier quelques points saillants sur des zones caractéristiques de l'objet. À l'étape 2, une approche statistique, faisant appel à des modèles de Markov (*Hidden Markov Models, ou HMM*) pour décrire les points saillants, est employée pour entraîner un modèle non-supervisé en utilisant l'information contextuelle de voisinage tridimensionnel. Finalement, lors de la 3^e étape, une réconciliation est effectuée entre les points saillants obtenus sous chacune des différentes perspectives. On nomme cette méthode "points saillants".

4.2.2. L'algorithme "Sailance du Maillage" (SM, *Mesh Saliency*)

Lee et al. [70] utilisent certains aspects de l'attention visuelle, particulièrement la courbure, une caractéristique typique de l'attention visuelle, pour identifier des points d'intérêt

dans la carte de courbure d'un objet tridimensionnel, en utilisant une correspondance entre chaque sommet et sa courbure moyenne. On nomme cet algorithme "saillance du maillage".

4.2.3. L'algorithme basé sur l'attention visuelle (AV, AVO)

L'algorithme d'attention visuelle (AV) utilisé dans le cadre de nos travaux est le fruit du travail de Chagnon-Forget et al. [66]. Avec cet algorithme, des points d'intérêt sont identifiés grâce à un modèle numérique de l'attention visuelle, soit le modèle d'attention visuelle de Itti, basé sur la couleur, l'intensité et l'orientation. Les points sont initialement identifiés à partir des images captées de plusieurs points de vue d'un objet tridimensionnel et ensuite transposés en trois dimensions en adaptant l'algorithme d'intersection rayon-surface.

Nous avons également utilisé une version améliorée de ce modèle, soit l'attention visuelle optimisée (AVO) [82]. Celle-ci utilise des caractéristiques additionnelles pour construire le modèle d'attention visuelle, soit le code de couleur DKL, la courbure, le contraste, la symétrie, l'entropie en plus d'utiliser une nouvelle méthode d'extraction des points.

4.2.4. L'algorithme basé sur la signature thermique (HKS)

À titre de comparaison, l'algorithme HKS, qui retourne un petit nombre de points d'intérêt, est aussi utilisé aussi. Celui-ci n'est pas inspiré par l'attention visuelle. Présenté par Sun et al. [69], cette méthode fait appel aux propriétés de diffusion de la chaleur à la surface d'un objet. HKS, qui signifie, *Heat Kernel Signature* (signature thermique), capte toutes les informations contenues dans la signature thermique et les utilise pour caractériser la forme de l'objet.

Il est important de mentionner que les algorithmes ne sont pas implémentés dans ce travail, mais qu'on utilise plutôt les points d'intérêt sous forme de coordonnées x , y et z ; celles-ci sont disponibles pour les solutions "points saillants", "saillance du maillage" et *HKS* dans [71] et sont mises à notre disposition pour les modèles d'attention visuelle par les auteurs de [66] et [82]. Une fois les points d'intérêt identifiés, on procède dans une première étape à la collecte d'empreintes virtuelles à ces points.

4.3. Objets virtuels

Les objets virtuels utilisés dans le cadre du présent travail sont formés d'un assemblage de maillages triangulaires (figure 5a). Un maillage est une collection de points (sommets), décrits par des coordonnées x , y et z , reliés entre eux par des arêtes formant des polygones. Dans Matlab, les objets sont représentés par deux matrices: la première est une série de points ayant des coordonnées x , y et z (les coordonnées de tous les sommets composant le modèle), et la deuxième, une matrice où chacune des lignes représente un polygone. Pour former ce polygone, on mentionne quels sont les sommets de la première matrice qui y font partie (les indices des sommets composant chaque face du maillage). Dans le cas présent, chacun de ces polygones un triangle.

4.4. Simulation du processus de collecte des données tactiles

La méthode de prise d'empreintes proposée dans le présent mémoire s'inspire des capteurs résistif sensible à la force *FSR*, de la cartographie (principalement des courbes topographiques que les géographes utilisent depuis longtemps), et également des empreintes digitales humaines, qui tracent des formes courbes et concentriques à la surface des doigts. Le principe est simple : capter les contours sur la surface de l'objet afin d'obtenir des courbes de niveaux topographiques, semblables aux montagnes sur les cartes des géographes. Ainsi, toutes les caractéristiques locales (aspérités) d'un objet deviennent autant de « petites montagnes », qui seront ensuite utilisées pour caractériser celui-ci, et éventuellement, l'identifier. Les courbes de niveaux obtenues ont, de plus, l'avantage d'être équidistantes, ce qui permet d'avoir un aperçu de la géométrie locale de l'objet, à l'endroit où l'empreinte est générée. Cette méthode est semblable à l'obtention de données avec un capteur *FSR* (figure 3), mais s'en distingue néanmoins sous un aspect: alors que le *FSR* identifie des points de pression sur une matrice carrée, notre méthode identifie des points d'égale hauteur sur l'objet et retourne une série de lignes de contour équidistantes. Celles-ci sont ensuite converties en images, toutes comme les données provenant du capteur *FSR* afin d'être analysées.

Afin d'être en mesure de produire des données expérimentales dans un laps de temps raisonnable, et puisque l'acquisition d'empreintes sur des objets est un exercice long et

fastidieux, nous simulerons dans une première étape le processus sur des objets virtuels, obtenus du projet *Shape Analysis Research : A Benchmark for 3D Interest Points Algorithms* [71]. Ceux-ci étaient également fournis avec divers points d'intérêts, obtenus avec divers algorithmes, notamment "points saillants" (section 4.2.1) et "saillance du maillage" (section 4.2.2) et *HKS* (section 4.2.4). Tel que mentionné dans la section précédente, nous effectuerons également des tests avec l'algorithme de détection des points basé sur l'attention visuelle proposé dans [66] ainsi qu'avec un modèle amélioré [82] (section 4.2.3). À titre d'exemple, la figure 5b-e montre les points d'intérêt pour le même modèle, obtenus par la méthode "points saillants" (figure 5b), la méthode *HKS* (figure 5c), la méthode de l'attention visuelle (figure 5d) et la méthode "saillance du maillage" (figure 5e).

Ces différents ensembles de points d'intérêt seront également mis à contribution afin de déterminer expérimentalement, lequel d'entre eux, lorsque couplé avec notre méthode de prise d'empreintes, performe le mieux en matière de reconnaissance d'objets.

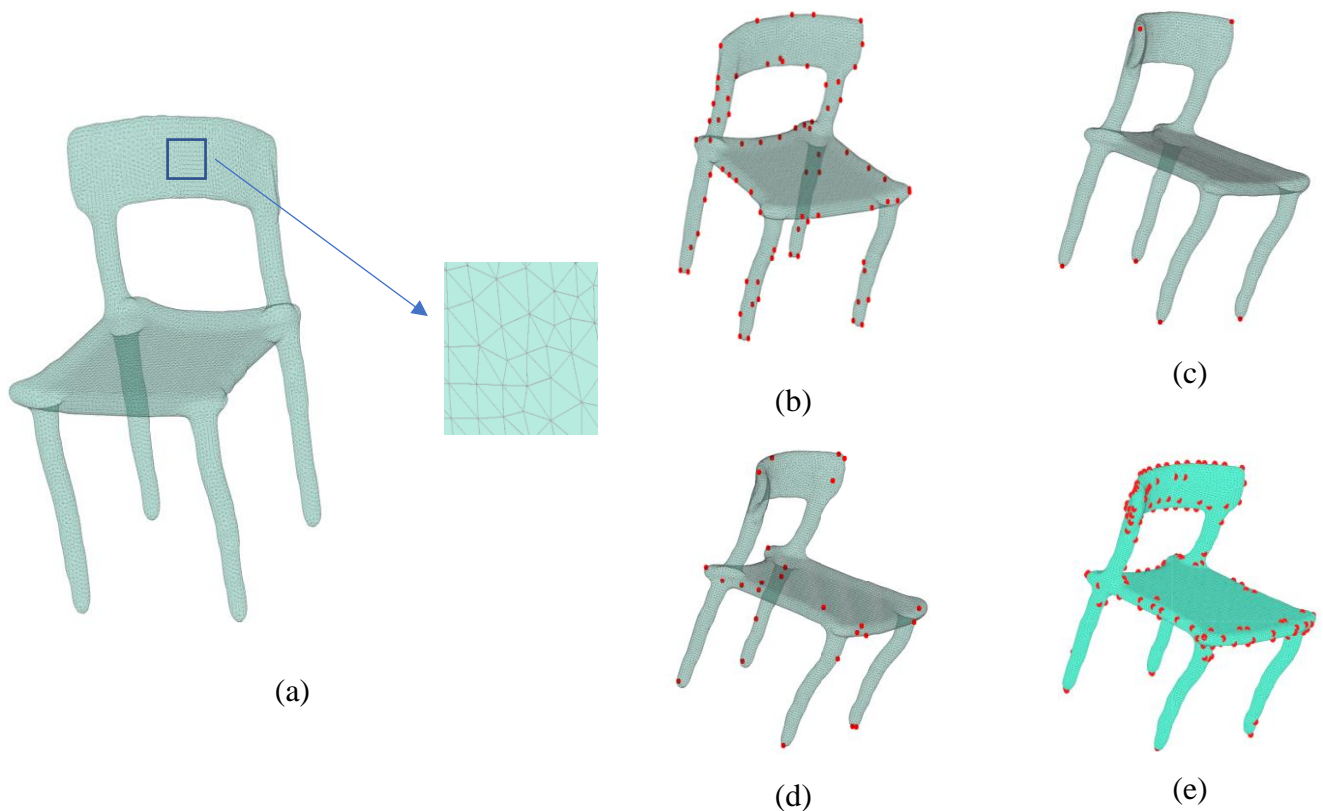


Figure 5 : Points d'intérêt, en rouge, obtenus sur la surface d'un même objet, en utilisant les algorithmes : (a) objet virtuel obtenu par maillage, (b) points saillants (Salient Points) [71], (c) *HKS* [71], (d) attention visuelle (AV) [66] et (e), saillance du maillage [71].

Afin de simuler la prise d’empreinte sur une surface virtuelle, nous modélisons le corps du capteur virtuel comme un volume rectangulaire, d’une certaine épaisseur simulant la partie déformable du capteur *FSR* en contact avec l’objet (figure 3, en bas, à gauche). Pour obtenir un profil local de pression (déformation locale), l’intersection avec le capteur est modélisée comme une série des courbes de différentes profondeurs. À cette fin, nous avons d’abord dû obtenir des plans tangents à celle-ci à des points d’intérêt précis. Or, notre objet virtuel est représenté par un maillage de triangles reliés entre eux par des arêtes, tel qu’illustré par les figures 5a et 6.

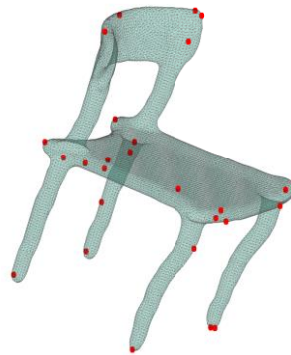


Figure 6 : Maillage de triangles représentant la surface d’un objet virtuel ainsi que les points d’intérêt, en rouge.

Pour déterminer ces plans, nous avons dû mettre au point une nouvelle technique de capture de plans quasi-tangents aux points d’intérêt, déterminés préalablement par des algorithmes tels que “points saillants”, “saillance du maillage”, les algorithmes d’attention visuelle et *HKS*.

L’approche proposée pour l’estimation du plan tangent est la suivante :

1. Pour chaque point d’intérêt donné de coordonnées $P_1 (X, Y, Z)$ sur la surface d’un objet, déterminer les points adjacents à celui-ci (P_1, P_2 et P_3) (figure 7).
2. S’assurer qu’il y a au moins trois points adjacents. Les points adjacents sont choisis aléatoirement s’il y en a plus que trois. S’il y a moins que trois points adjacents, le point d’intérêt est ignoré.
3. Calculer un plan à partir de trois points adjacents P_1, P_2 et P_3 au point d’intérêt.
4. Déterminer un vecteur orthogonal au plan en calculant le produit vectoriel des vecteurs P_1-P_2 et P_1-P_3 . Ce vecteur deviendra l’axe de translation selon lequel le plan sera déplacé pour obtenir des intersections successives avec la surface de l’objet virtuel.

5. Obtenir les points d'intersection entre le plan généré à l'étape 4, et l'objet.
6. Glisser le plan original le long du vecteur normal, et obtenir les points d'intersection de ce nouveau plan avec l'objet.
7. Recommencer les étapes 5 et 6 autant de fois que nécessaire (p.ex. pour nos travaux, nous avons effectué 4 intersections).

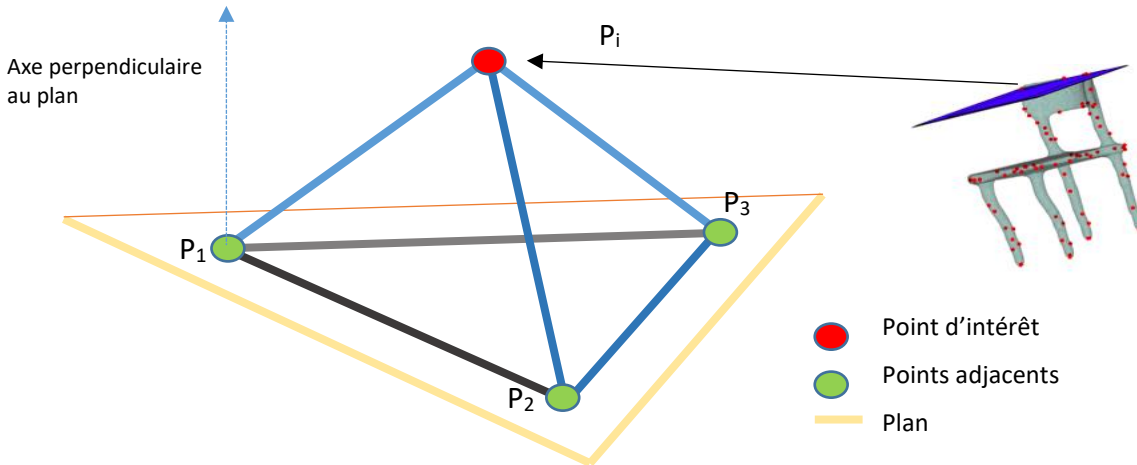


Figure 7 : Relation entre le point d'intérêt, les points adjacents, l'estimation d'un plan tangent à la surface et le vecteur orthogonal servant d'axe de translation du plan.

Afin d'obtenir plusieurs courbes à différents niveaux, nous effectuons une translation le long de l'axe perpendiculaire au plan, qui simule la partie déformable du capteur en contact avec l'objet, et nous traçons quatre plans qui coupent l'objet à quatre endroits de plus en plus profonds le long de cet axe, tel qu'illustré à la figure 8b. Chacun de ces plans est espacé de 0,0005 unités des plans voisins. Dans la figure 8, une chaise virtuelle est utilisée, et on trace des plans quasi-tangents à l'objet à un de ses points saillants (figure 8a). Pour chacun des plans, les points d'intersection avec l'objet sont reliés entre eux pour former une courbe de niveaux, tel qu'illustré à la figure 8c.

Une fois obtenues, les courbes de profil de pression (figure 8c) sont traduites en empreinte (image, figure 8d) en projetant chacun des points de ces courbes sur un plan unique. Une fois projetés sur un plan unique, les régions concentriques qui forment les projections de chacune des courbes de profil sont colorées de bleu pâle à bleu foncé, selon qu'elles étaient près ou éloignées de la surface du capteur virtuel. Ce procédé permet de conserver, en dépit de la perte de la troisième dimension, les informations reliées à la profondeur de l'empreinte.

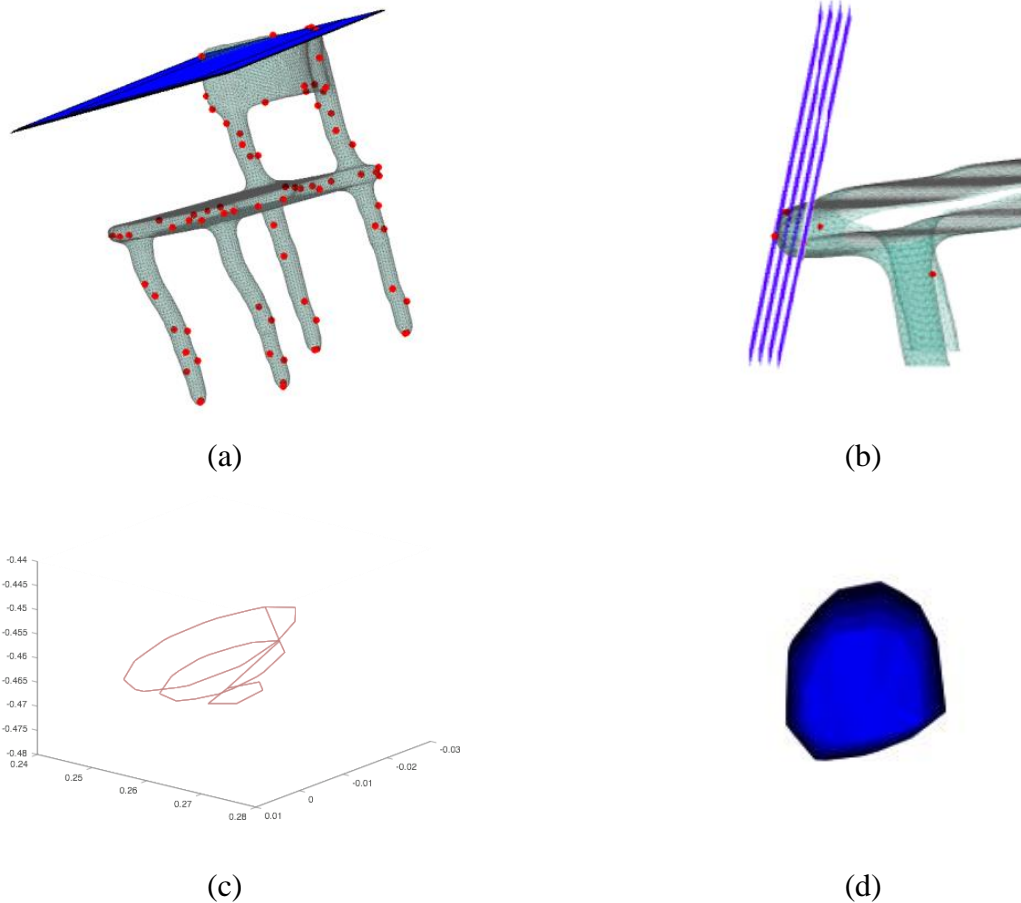


Figure 8 : (a), (b) Plans successifs sur l'objet virtuel ; (c) courbes d'intersection obtenues sur l'objet ; et (d) image de l'empreinte obtenue après toutes les étapes précédentes.

Bien que l'approche proposée puisse résulter, dans certains cas, en des plans qui ne sont pas parfaitement tangents à la surface, nous estimons que ceux-ci conviennent à nos besoins pratiques. L'approche proposée permet le calcul des plans même dans les cas où des points d'intérêt se retrouvent sur des crêtes pour lesquelles il serait impossible de calculer un plan tangent, tel que sur une pointe acérée. Notre technique permet de toujours obtenir un plan menant à des intersections avec la surface, même dans des régions de discontinuité. Elle a néanmoins, certaines limites, particulièrement quand le point d'intérêt est nettement décentré par rapport à ses points adjacents. La figure 9a montre un exemple d'une empreinte invalide obtenue dans une telle situation. Ces empreintes, qui clairement, ne peuvent pas être obtenues par un capteur physique, doivent être rejetées suite à une inspection visuelle, avant de générer les modèles de classification.



Figure 9 : Exemple de : (a) empreinte invalide, causée par un axe de translation des plans désaxé par rapport à la normale à la surface; et (b) empreinte valide - l'axe de translation des plans est parallèle à la normale à la surface.

Les images de la figure 10 montre la série d'empreintes valides obtenues pour le modèle de la vache (« cow » dans la base de données [71]) en utilisant la méthode d'attention visuelle pour la détection des points d'intérêt.

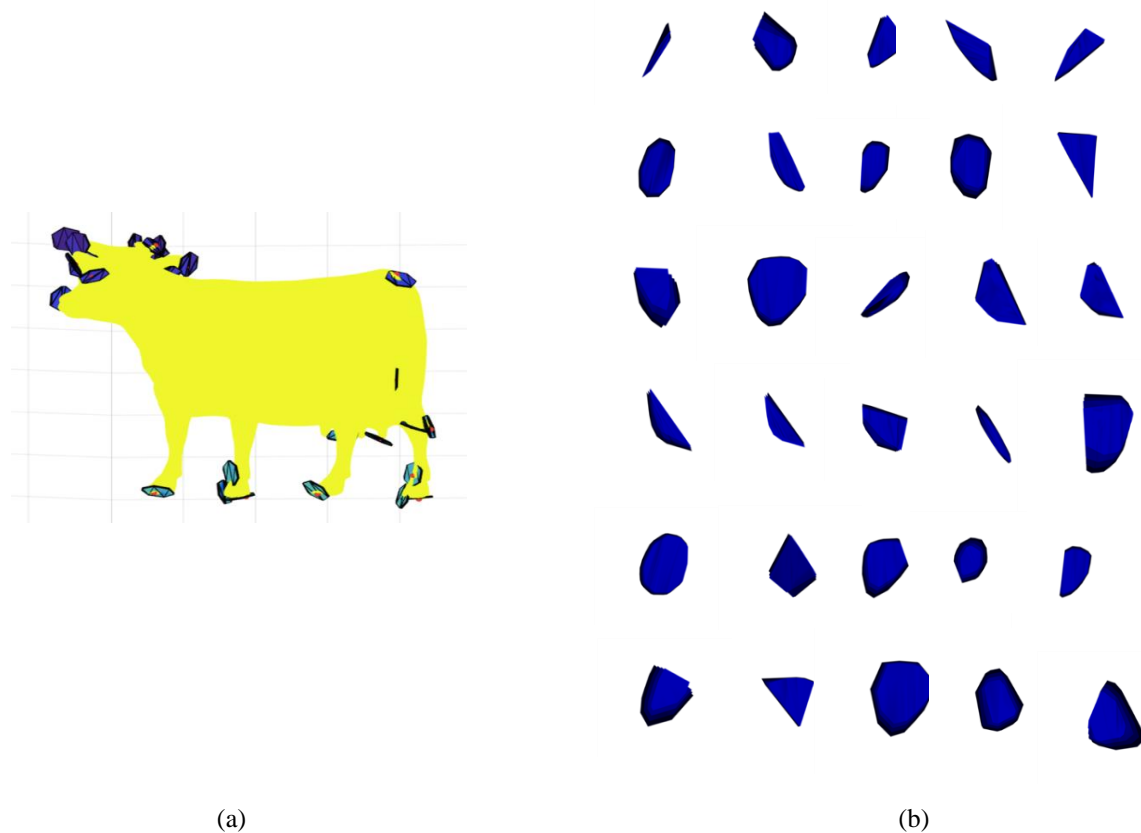


Figure 10 : (a) Illustration en jaune de la figurine de vache, ses points d'intérêt détectés par l'algorithme d'attention visuelle en rouge et les empreintes obtenues aux points de contact du capteur en bleu et (b) la série d'empreintes générées pour la figurine de la vache.

4.5. Extraction de caractéristiques à partir des données tactiles

Des caractéristiques extraites à partir des séries d'images semblables à celle dans la figure 10, ou encore, les images elles-mêmes, sous-échantillonnées et transformées dans des vecteurs, deviendront ensuite les produits consommés par *RapidMiner* pour classifier les objets selon leurs empreintes obtenues à l'aide des algorithmes d'identification de points d'intérêt.

4.5.1. *RapidMiner*

RapidMiner [79] est une plateforme logicielle développée par une compagnie allemande du même nom. Elle est utilisée, par exemple, pour l'apprentissage machine, la fouille de données et de textes, la prédiction de tendances, l'analyse de sentiments et l'intelligence d'affaire. *RapidMiner* a été développée en 2001 par Ralf Klinkenberg, Ingo Mierswa et Simon Fischer. Depuis ce temps, *RapidMiner* a évolué et se découplent maintenant en plusieurs produits (Studio, Server et Radoop).

4.5.1.1. Présentation de *RapidMiner*

RapidMiner Studio est une plateforme intégrant les techniques d'apprentissage les plus avancées dans un environnement visuel fort convivial (figure 11). De l'ingestion de données en passant par leur analyse jusqu'à la présentation des résultats, tout y est fait pour faciliter la vie de l'utilisateur.

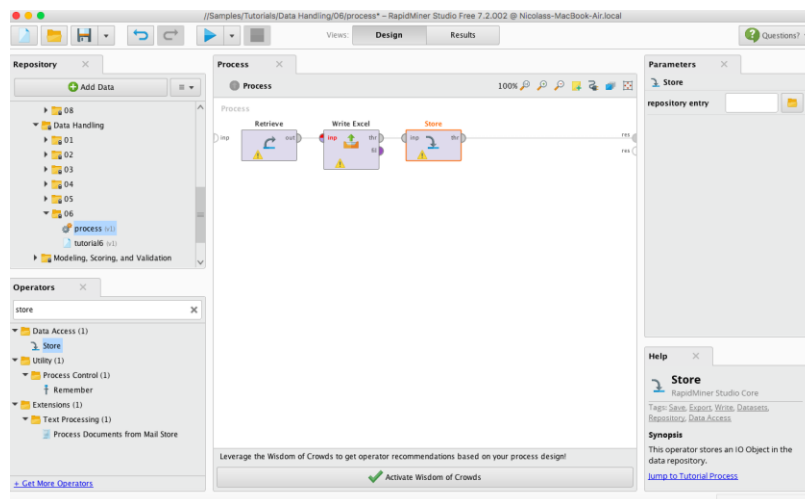


Figure 11: Interface graphique de *RapidMiner Studio*.

RapidMiner est conçu de manière à ce que les méthodes d'ingestion et de traitement de données soient sélectionnées, glissées et déposées (“*drag and drop*”) dans le plan de travail. De cette façon, l'utilisateur n'a pas à apprendre un langage de programmation et peut rapidement se mettre à l'analyse de données dans la mesure où il a déjà une base en techniques d'apprentissage.

4.5.1.2. Module d'analyse d'images pour *RapidMiner*

Puisque *RapidMiner* n'offre pas d'emblée des modules d'analyse d'images, nous avons dû nous tourner vers des tierces parties offrant les fonctionnalités désirées. Dans le cas présent, nous avons utilisé le module *Image Mining Extension IMMI* de la compagnie *Signal Processing Laboratory* [72]. Grâce à celui-ci, nous avons été en mesure d'ingérer les images et d'en extraire les caractéristiques nécessaires à la génération de modèles de classification.

4.5.2. Extraction des caractéristiques

Une fois que les profils de pression ont été acquis, nous en extrayons à partir des images résultantes (figure 10b), certaines caractéristiques telles que:

- les statistiques globales de l'image:
 - moyenne : correspond à la moyenne des valeurs de gris de tous les pixels de l'image;
 - médiane : calcule la valeur médiane des valeurs de gris de tous les pixels de l'image;
 - déviation standard : déviation obtenue en calculant la moyenne des valeurs de gris;
 - asymétrie : mesure de l'asymétrie observée autour de la courbe de distribution des valeurs de gris de l'image;
 - aplatissement : expression de l'aplatissement de la courbe de distribution des valeurs de gris de l'image;
 - sommets : incluent les couleurs les plus élevées de l'histogramme et leurs valeurs relatives;
 - valeur minimale de gris ;
 - valeur maximale de gris ;

- centre de masse normalisé : la valeur moyenne de brillance normalisée pour tous les points x et y de l'image;
- fraction de zone : le pourcentage de pixels non-nuls de l'image;
- un histogramme contenant de multiples valeurs relatives à l'image ;
- classification contour/intérieur (*Border/Interior Classification, BIC* à la figure 12) suivie de la distance logarithmique (dLog à la figure 12) des images entre les pixels intérieurs et les pixels de contour; et
- OBCF (minimum, maximum et médiane des valeurs de gris pour chacun des pixels de l'image ; ce qui requière préalablement une conversion de l'image en niveaux de gris).

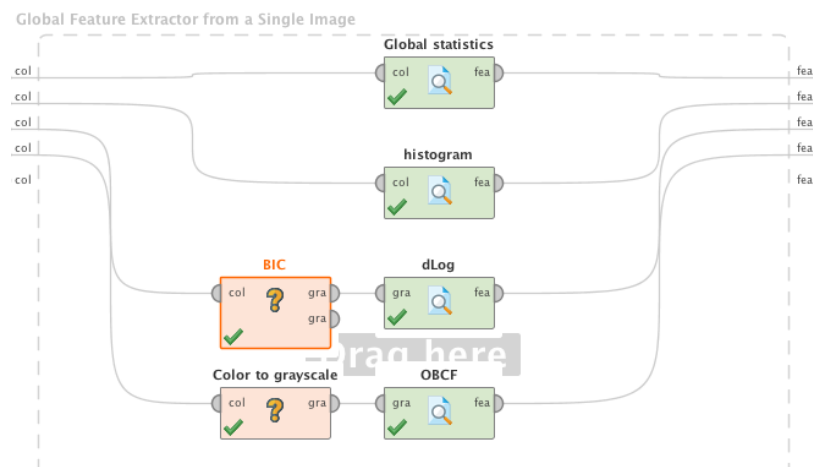


Figure 12 : Capture d'écran : caractéristiques extraites des images dans RapidMiner.

Celles-ci sont extraites directement par RapidMiner (figure 12) par bloc de chaque image en utilisant *IMMI*, résultant au total en une série de 720 caractéristiques pour chaque empreinte tactile. Une analyse de ces caractéristiques a mis en évidence le fait que certaines caractéristiques étaient associées avec des valeurs constantes. Puisque les données constantes nuisent à l'apprentissage, les caractéristiques ayant ce type de valeurs ont été éliminées. Les mêmes caractéristiques sont éliminées dans la série représentant chaque algorithme (“points saillants”, “attention visuelle”, “attention visuelle optimisée”, “sillance du maillage” et HKS) pour ne pas affecter la généralisation de la solution proposée et permettre une juste comparaison entre les différentes méthodes. Parmi les données nettoyées résultantes (158 caractéristiques), seulement 24 caractéristiques sont retenues et ensuite assemblées dans un vecteur représentant chaque empreinte. Les vecteurs sont utilisés pour la classification d'objets.

Nous avons également utilisé une autre manière d'extraire des caractéristiques sous forme vectorielle à partir des empreintes. Les empreintes ont été réduites à de matrices de 16x16 et de 32x32, respectivement. Cette technique de sous-échantillonnage (*downsampling*), a été effectuée à l'aide de Matlab. Ces empreintes sous-échantillonnées reflètent mieux les capacités réelles du capteur *FSR*. Ensuite chaque matrice a été convertie dans un vecteur de 256 et de 1024 éléments respectivement, contenant les valeurs de gris contenus dans chaque matrice. Ces vecteurs ont été utilisés aussi en RapidMiner pour effectuer des tests de classification.

4.6. Classification des objets

La classification en statistique se réfère aux processus supervisés qui permettent de déterminer qu'un item possède les caractéristiques particulières d'un groupe d'items, et qu'il peut donc être associé à ce groupe. Pour déterminer les caractéristiques des groupes permettant de catégoriser les items, il est impératif d'avoir accès au départ à un ensemble d'items hétérogènes, composé de sous-groupes ayant des caractéristiques communes, permettant d'étiqueter les items; cet ensemble devient alors le référent initial, appelé ensemble d'entraînement. Chacun des items de cet ensemble doit avoir une étiquette l'associant à un sous-groupe. Dans le cadre de ce projet, les items à classer sont des empreintes obtenues en apposant un capteur tactile sur des objets. Ces objets sont d'abord virtuels et ensuite de véritables objets physiques. Nous tentons de déterminer qu'une certaine empreinte appartient à un objet en particulier. Chacun de ces objets (figure 19 pour les objets virtuels) représentent donc un regroupement, ou une classe, dans laquelle nous tentons de placer nos empreintes.

À partir de l'ensemble d'entraînement, on extrait des caractéristiques (section 4.5.2) qui permettent de distinguer les sous-ensembles entre eux. Ces caractéristiques sont ensuite utilisées pour générer des modèles de classification. Il existe un grand nombre d'algorithmes permettant de générer des modèles de classification, qui doivent préalablement être validés avant de procéder à la classification d'items non-étiquetés. Une fois le modèle obtenu et validé, on procède à la classification.

Plusieurs techniques de classification sont testés pour la classification des objets 3D à partir de ces caractéristiques, à savoir l'apprentissage profond, les réseaux bayésiens, les arbres de décision, les forêts d'arbres, le *boosting* d'arbre de décision, les machines de vecteurs de support et les *K* plus proches voisins. Ceux-ci sont tous des algorithmes connus, qui ont été

utilisés avec succès sur des données tactiles, tel qu'on a pu le constater dans le chapitre 2. Ils sont brièvement présentés dans les sections suivantes.

4.6.1. Les réseaux de neurones et l'apprentissage profond

Les réseaux neuronaux artificiels, ou simplement les réseaux neuronaux, sont conçus de manière à reproduire la manière dont le cerveau humain traite l'information. Haykin [73] les définit de la manière suivante : « *Un réseau neuronal est un processeur massivement parallélisé, formé de processeurs simples, possédant une propension naturelle pour le stockage de connaissance basé sur l'expérience et en mesure d'en restituer le contenu pour usage ultérieur. Il est semblable au cerveau humain selon deux aspects :*

1. *La connaissance est acquise par le réseau à partir de son environnement par l'entremise d'un processus d'apprentissage;*
2. *La force des connections inter-neurones, connue sous le nom de poids synaptique, est utilisée pour stocker la connaissance acquise. »*

Un réseau neuronal simple est schématisé selon la figure 13.

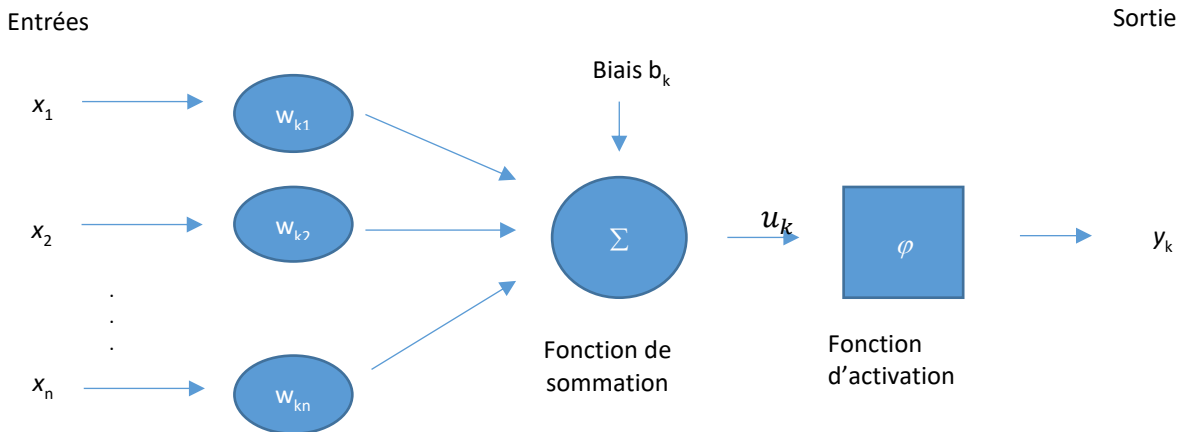


Figure 13 : Exemple d'un réseau neuronal simple.

Les équations s'appliquant à ce réseau sont les suivantes :

$$u_k = \sum_{j=1}^n w_{kj} x_j \quad (1)$$

$$y_k = \varphi(u_k + b_k) \quad (2)$$

En somme, un réseau neuronal reçoit des stimuli x_j de son environnement et ceux-ci sont multipliés par des poids w_{kj} associés à ces stimuli, le tout cumulé pour j stimuli. Ce résultat est ensuite traité par une fonction, dite d'activation. Cette fonction peut prendre diverses formes, mais elle est souvent sigmoïde, dont l'équation a la forme suivante:

$$y_k = \frac{1}{1 + e^{-u_k}} \quad (3)$$

Graphiquement, une fonction sigmoïde ressemble à celle illustrée à la figure 14.

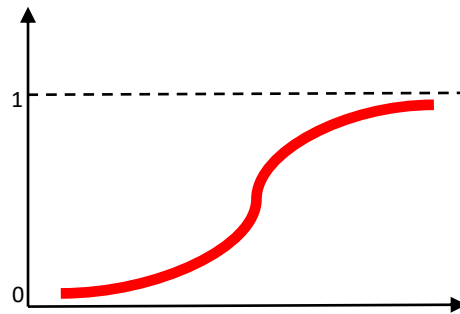


Figure 14 : Fonction d'activation sigmoïde.

Le résultat de la fonction d'activation est la valeur finale du réseau neuronal, et détermine si la condition finale désirée est atteinte ou non. Un des algorithmes le plus populaires pour l'apprentissage avec un réseau neuronal est la rétro-propagation, décrit dans la section suivante.

Dans le cadre de ce mémoire, un réseau neuronal sera utilisé en tant que classificateur. Afin d'entraîner le réseau, nous utilisons comme valeurs d'entrée les vecteurs des caractéristiques extraites à partir des empreintes des objets. À la sortie, nous obtiendrons une classe pour cet objet. Cette classe sera associée à un des objets de notre ensemble de départ.

En matière de réseau de neurones, le véritable défi réside dans l'assignation et la modification des poids nécessaires à l'obtention des résultats désirés. En outre, plus le réseau de neurones est profond ou possède de couches de niveaux, et plus le calcul des poids devient complexe. Pour effectuer ces calculs, la méthode dite de rétro-propagation a été développée dans les années 1960 et 1970, et mise en œuvre dans les réseaux de neurones à partir des années 1990 [74]. Par ailleurs, quand le réseau de neurones possède plusieurs couches et un grand nombre de neurones dans chacune, on parle alors d'apprentissage profond (*deep learning*).

La rétro-propagation utilise la technique des gradients descendants pour minimiser l'erreur pour l'apprentissage supervisée dans des réseaux discrets, différentiables et de profondeurs arbitraires [74]. Celle-ci cherche à minimiser les erreurs à tous les niveaux de neurones. La rétro-propagation effectue ces corrections dynamiquement, lors de l'apprentissage du modèle. Le processus est illustré à la figure 15.

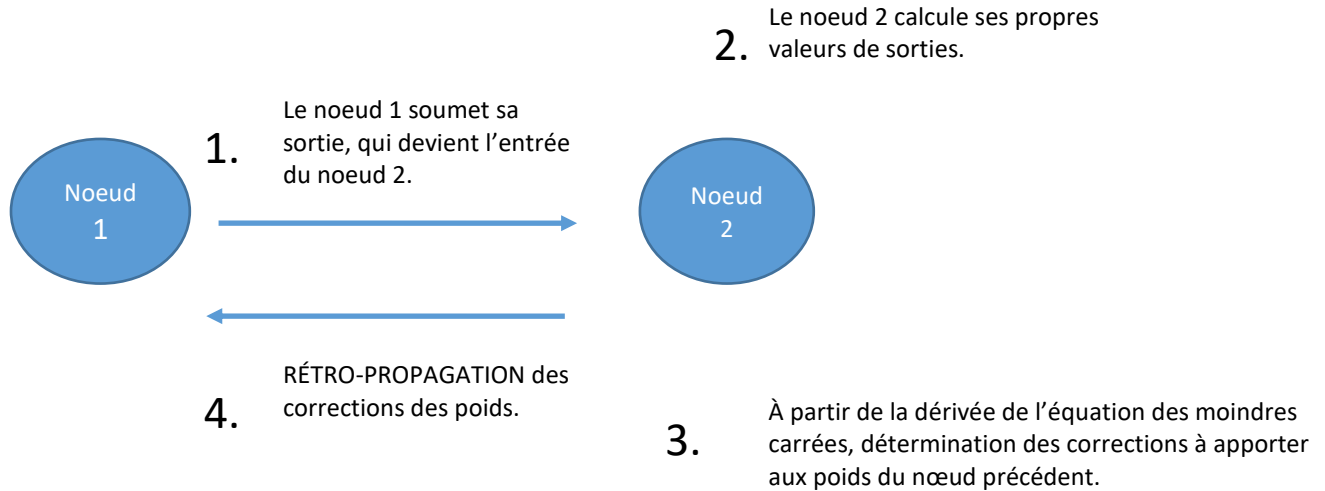


Figure 15 : Étapes nécessaires à la rétro-propagation.

Pour arriver à obtenir les corrections nécessaires pour les poids du nœud précédent, la somme des écarts entre la valeur de sortie estimée y_i et la valeur réelle t_i doit être minimisée.

$$E = \frac{1}{2} \sum_i (t_i - y_i)^2 \quad (4)$$

Pour ce faire, il est nécessaire de dériver l'équation précédente par rapport au poids w_k , en prenant soin d'inclure les dérivées partielles des variables ayant servi à déterminer la valeur de u_k .

$$\frac{dE}{dy} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial u_k} \cdot \frac{\partial u_k}{\partial w_k} \quad (5)$$

Au final, pour une équation de sortie de type sigmoïde, la variation des poids du nœud précédent est donnée par l'équation suivante :

$$\Delta w_k = \sum_i \varepsilon x_{ki} y_i (1 - y_i) (t_i - y_i) \quad (6)$$

ϵ étant un facteur dit « facteur d'apprentissage » (*learning rate*).

Dans le cadre de nos travaux nous avons utilisé l'apprentissage profond de la même manière que les réseaux de neurones pour classifier les empreintes que nous avons générées à partir de nos objets.

4.6.2. Les réseaux bayésiens

Les réseaux bayésiens sont basés sur le théorème de Bayes, qui permet de calculer la probabilité qu'un événement se produise en sachant qu'un autre événement lui étant relié se soit déjà produit. Étant donné un événement B, l'équation suivante permet de déterminer la probabilité qu'un événement A se produise :

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (7)$$

Dans un réseau bayésien, la probabilité qu'un événement A_i (i étant un nombre de 1 à n) se produise étant donnée une étiquette C, préalablement calculée. Ensuite, la probabilité qu'une étiquette C soit sélectionnée étant donnée des événements A_j , j étant un nombre de 1 à n , est calculée à l'aide de l'équation (7) [75].

La structure d'un réseau bayésien simple est montrée à la figure 16.

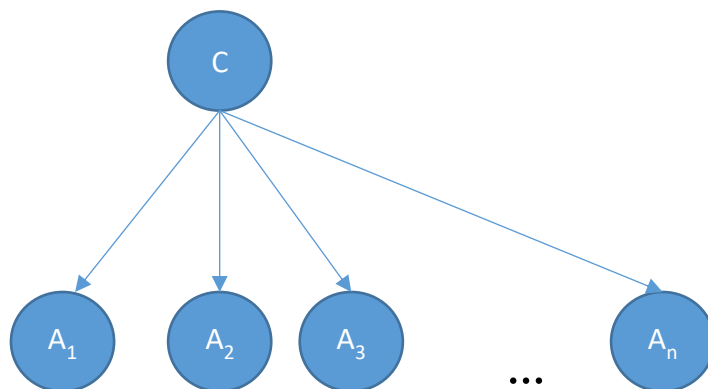


Figure 16 : Structure d'un réseau bayésien simple [75].

Dans le cadre de nos travaux, nous avons fait appel au module *Naïve Bayes* de RapidMiner pour classifier nos objets.

4.6.3. Les arbres de décision

Les arbres de décision sont des structures supervisées de classification possédant des branches et des feuilles. Les éléments d'un ensemble sont distribués dans un arbre de décision selon leurs caractéristiques. Plus une caractéristique est discriminante, plus elle est haute dans l'arbre de décision.

La figure 17 montre un arbre de décision pour la classification d'empreintes collectées selon l'algorithme HKS de détection de point d'intérêt. On peut y voir deux nœuds de décision, basés sur des valeurs extraites des empreintes et permettant de classifier les objets.

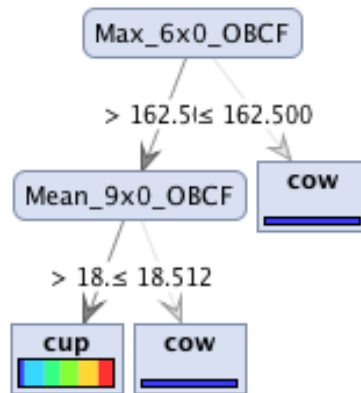


Figure 17 : Exemple d'arbre de décision pour classification d'empreintes.

Le facteur discriminant d'une caractéristique est formellement déterminé en calculant le gain d'information généré par celui-ci. Par exemple, pour l'algorithme d'arbre de classification ID3, le gain est déterminé par l'équation suivante [76] :

$$gain(A) = I(p, n) - E(A) \quad (8)$$

où

$$I(p, n) = -\log_2 \frac{p}{p+n} - -\log_2 \frac{n}{p+n} \quad (9)$$

et

$$E(A) = \sum_{i=1}^v \frac{p_i}{p_i + n_i} I(p_i, n_i) \quad (10)$$

où

$I \equiv$ Information

$E \equiv$ entropie

$p \equiv$ probabilité d'avoir un élément appartenant à une classe C

$p_i \equiv$ probabilité d'avoir un élément appartenant à une classe C_i

$n \equiv$ probabilité d'avoir un élément n'appartenant pas à une classe C

$n_i \equiv$ probabilité d'avoir un élément n'appartenant pas à une classe C_i

Le module d'arbre de décision de RapidMiner utilise l'algorithme C4.5, une version améliorée de l'algorithme ID3, décrit ci-haut.

4.6.4. Forêt d'arbres décisionnels (*Random Forest*)

L'algorithme de classification appelé forêt d'arbre décisionnels, mieux connu sous le nom *random forest* en anglais, est une technique qui crée une multitude d'arbres de classification pendant la période d'entraînement du modèle de classification [77]. Cette technique a comme avantage de pallier au surentraînement (*overfitting*) normalement associée aux arbres de décision.

Dans le cadre de présent travail, nous avons fait appel de cette technique par l'entremise du module *Random Forest* de RapidMiner.

4.6.5. Boosting d'arbres de décision (*Gradient Boosted Tree*)

Le *boosting* d'arbres de décision permet d'améliorer la performance de la technique des arbres en créant une succession d'arbres de décision [78]. Cette procédure flexible et non-linéaire applique une succession d'algorithmes faibles de classification sur des données qui changent petit à petit, ce qui crée une série d'arbres de décision et autant de modèles de classification faibles.

Dans le cadre de présent travail, nous avons fait appel à cette technique par l'entremise du module *Gradient Boosted Tree* de RapidMiner.

4.6.6. *K* plus proches voisins (*k-Nearest Neighbours, k-NN*)

La technique des *K* plus proches voisins est utilisée en apprentissage machine pour regrouper des éléments ayant des caractéristiques semblables. Celle-ci est utilisée en apprentissage non-supervisée, mais elle peut également être utile en apprentissage supervisée. Les regroupements sont faits en calculant les distances (par exemple, la distance euclidienne) autour d'un point central pour chaque regroupement. Ce point central est ensuite recalculé lorsqu'un regroupement est terminé; cette succession de calculs de distances et du point central à un groupe s'arrête lorsqu'il n'y a plus de changements significatifs de la position de ce dernier, ou après un nombre d'itérations donné [80].

Dans le cadre de nos travaux, nous avons utilisé le module KNN de RapidMiner pour générer nos modèles.

4.6.7. Machines à vecteurs de support (*SVM*)

Les machines à vecteurs de support sont une technique de classification supervisée qui consiste à établir des plans ou des hyperplans dans un espace à plusieurs dimensions afin de trouver une façon de séparer les éléments d'un ensemble de manière à ce que les regroupements expriment certaines caractéristiques communes à ses membres [81]. Mathématiquement, cette technique cherche à optimiser la distance entre les éléments des regroupements et les frontières, ou les vecteurs de support, qui divisent l'ensemble original en différents regroupements.

Dans le cadre de nos travaux, nous avons utilisés le module *Evolutionary SVM* de RapidMiner, qui utilise un algorithme évolutif pour résoudre le problème d'optimisation des machines à vecteurs de support, qui consiste à maximiser les distances entre les patrons d'entraînement et les frontières de décision.

4.6.8. L'évaluation de la classification

Afin d'évaluer la performance des différents classificateurs, nous utilisons la validation croisée. Celle-ci divise l'ensemble des données en deux groupes, soit le groupe d'entraînement, et le groupe de test. Le groupe d'entraînement est utilisé pour créer le modèle, alors que le groupe de tests l'est pour valider ce dernier. Typiquement, le groupe d'entraînement représente 90% des items de l'ensemble original, alors que le groupe de tests représente 10% de celui-ci. On

parle dans ce cas de la validation croisée à 10 plis. Ce processus de séparation des items en deux groupes, suivi de la génération d'un modèle et de tests sur celui-ci, est effectué à plusieurs reprises (10 fois au total dans le cas de 10 plis), et les résultats de chacune de ces itérations sont combinés (par moyennage) au final pour donner une matrice de confusion. Un exemple de matrice de confusion est fourni à la figure 18.

Sur la matrice de confusion, les éléments sur la diagonale plus foncée sont correctement classifiés. Les éléments d'une même colonne appartiennent réellement à la catégorie décrite par l'étiquette placée au haut de celle-ci ; sur cette même colonne, les éléments en dehors de la diagonale représentent des faux-négatifs par rapport à l'étiquette au haut de la colonne.

accuracy: 70.45% +/- 45.62% (mikro: 70.45%)

	true cow	true cup	true dromed...	true glasses	true hand	true plane	class precision
pred. cow	10	0	0	1	1	0	83.33%
pred. cup	1	2	0	0	0	0	66.67%
pred. drome...	0	0	9	0	0	1	90.00%
pred. glasses	0	0	0	3	1	1	60.00%
pred. hand	0	0	0	1	3	0	75.00%
pred. plane	1	0	2	2	1	4	40.00%
class recall	83.33%	100.00%	81.82%	42.86%	50.00%	66.67%	

Figure 18 : Exemple de matrice de confusion.

Les éléments sur une même ligne représentent les prédictions du modèle pour l'étiquette de celle-ci. Les éléments sur la diagonale sont bien classés, mais ceux en dehors de celle-ci ne le sont pas. On parle alors de faux-positifs par rapport à l'étiquette de la ligne.

Chapitre 5. Résultats expérimentaux

5.1. Données simulées de tests

Tel qu'indiqué précédemment, nous débutons par la simulation d'acquisition d'empreintes tactiles pour une série d'objets pour lesquels les modèles tridimensionnels et les points d'intérêt sont disponibles [71]. Dans le cadre de ce projet de mémoire, les résultats sont présentés d'abord sur six objets virtuels, montrés à la figure 19, et ensuite sur un sous-ensemble de quatre objets. Ceux-ci sont choisis puisqu'on possède des objets (jouets) réels dans le laboratoire qui sont semblables à ceux-ci. Nous tenterons d'obtenir des résultats à partir de ces objets physiques dans une étape ultérieure du présent travail, le but ultime de celui-ci étant de comparer nos résultats de simulation avec des résultats obtenus avec des objets réels.

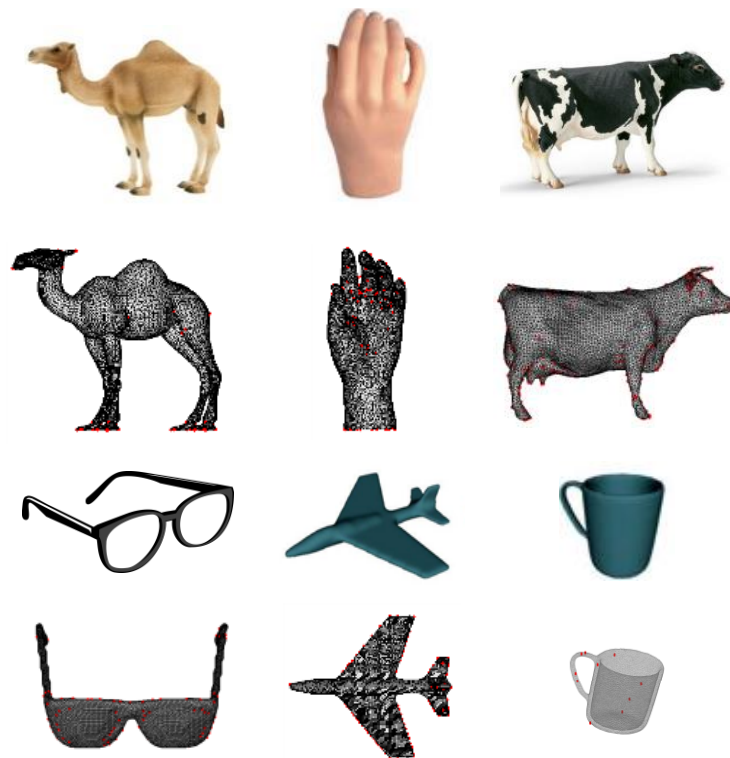


Figure 19 : Objets réels et modèles (maillages) correspondants, avec les points d'intérêt marqués en rouge.

Pour chacun des objets de la figure 19, nous générons des empreintes à l'aide de la technique présentée au chapitre 4. La figure 20 présente, par exemple, les empreintes obtenues à

partir de trois algorithmes de détection des points d'intérêt testés, pour l'objet tasse ("cup" dans la base de données [71]).

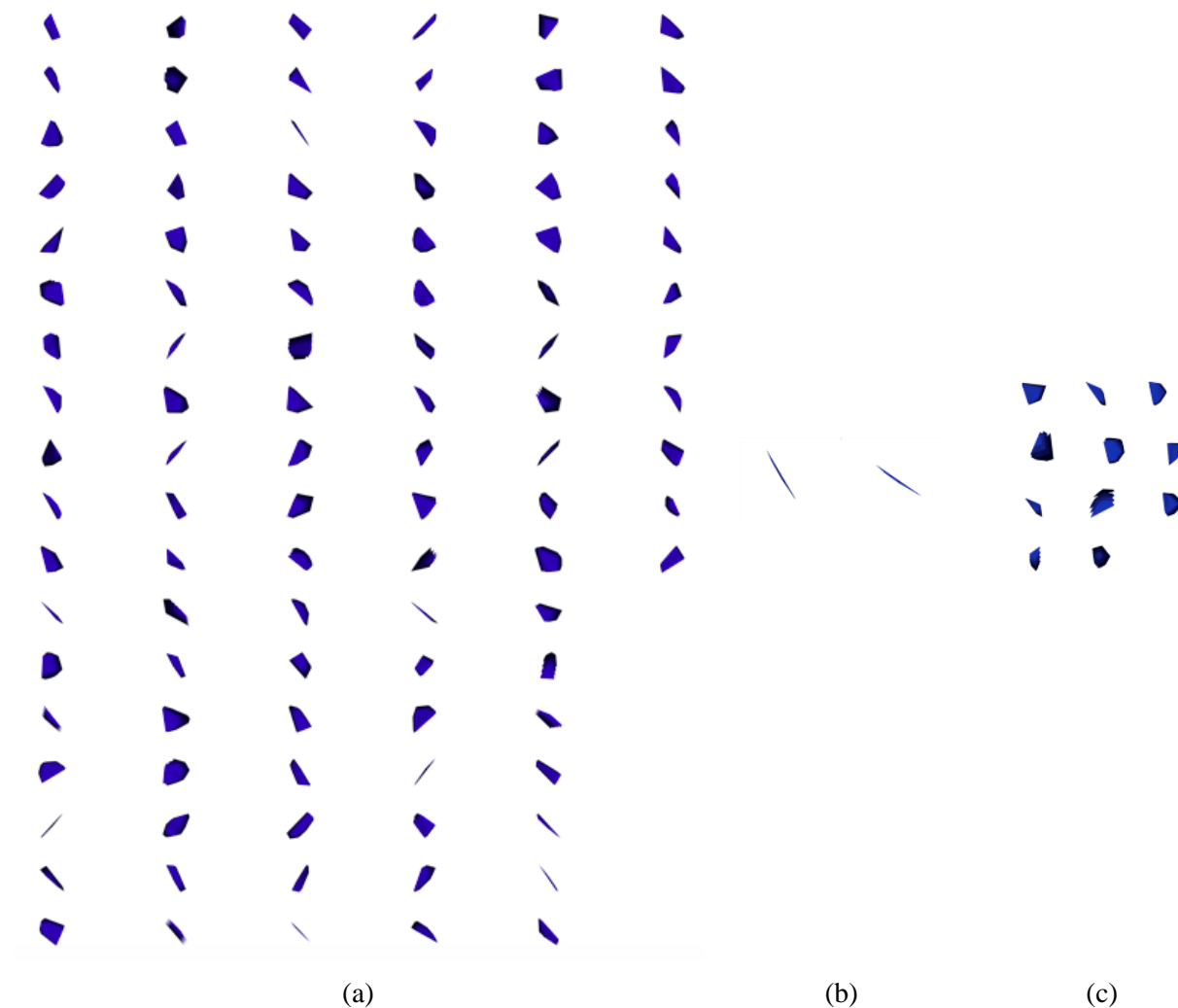


Figure 20 : Série des empreintes générées pour trois algorithmes d'identification de points d'intérêt pour l'objet tasse ("cup") : (a) "points saillants", (b) HKS et (c) "attention visuelle".

La figure 21 montre, pour chaque objet, le nombre d'empreintes générées par chaque algorithme. On peut constater que la méthode qui obtient le moins de points est l'attention visuelle optimisée, dont les nombres sont constants pour tous les objets; suivie de HKS, de l'algorithme d'attention visuelle et finalement par la méthode des points saillants.

Une fois ces empreintes obtenues, on procède avec l'identification des caractéristiques. Celles-ci doivent également être épurées, tel que mentionné à la section 4.5.2, afin d'éliminer les multiples colonnes de données constantes.

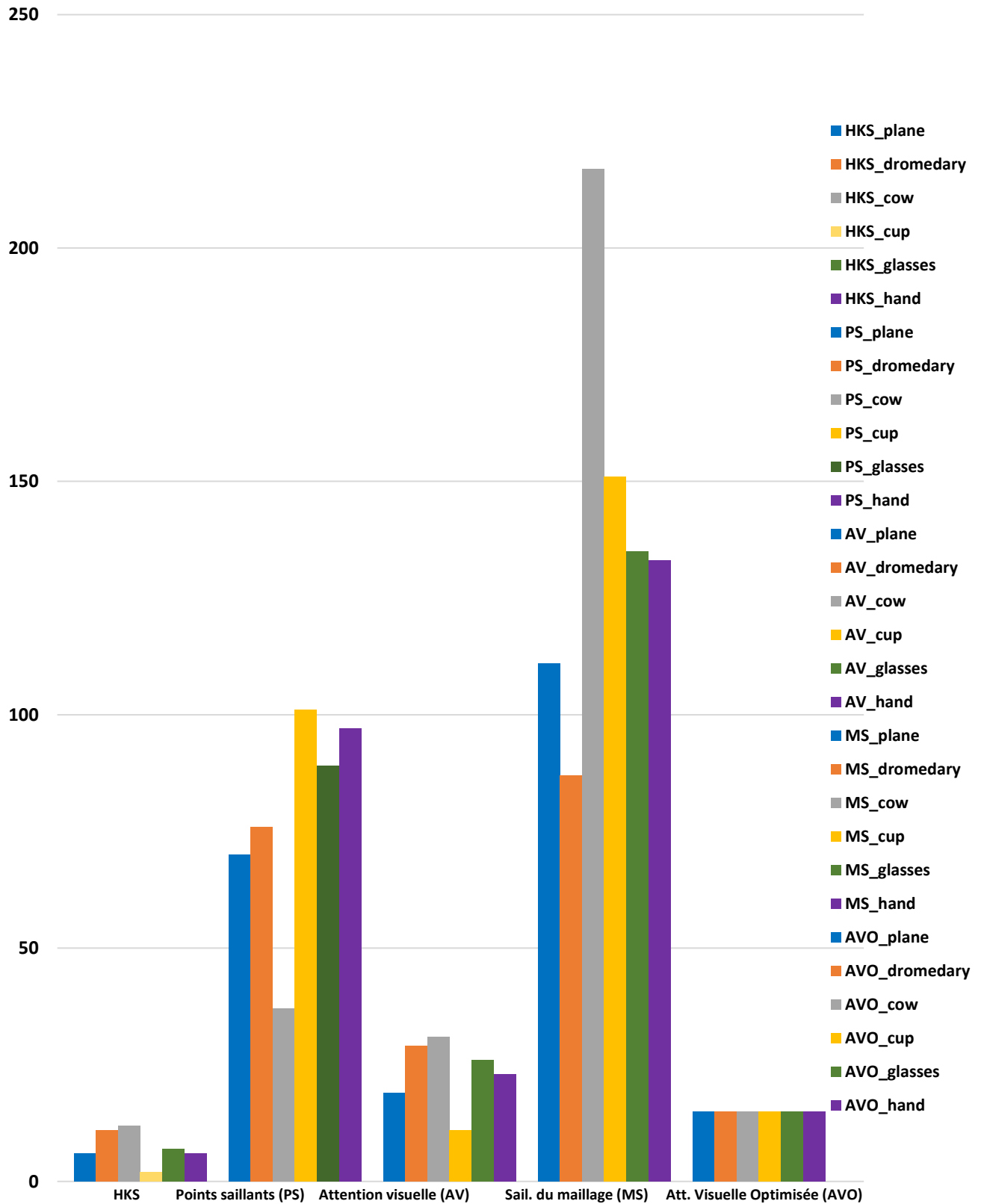


Figure 21 : Nombre d'empreintes par objet, pour chacun des algorithmes.

L'effet des colonnes de données constantes obtenues une fois les caractéristiques extraites des images avec le module IMMI est d'ailleurs fort important pour la classification. En l'absence de cette étape, aucun des algorithmes utilisés dans le cadre de nos travaux ne réussissaient à produire des résultats dignes de mention. Une fois les colonnes constantes éliminées, ces données deviennent finalement les entrées pour des classificateurs tels que l'apprentissage profond, un réseau de Bayes, un arbre de décision, le *boosting* d'arbres de décision, une forêt d'arbres de décision, une machine de vecteurs de support ou les K plus proches voisins.

5.2. Résultats expérimentaux pour la classification des données simulées en utilisant les caractéristiques IMMI pour les six objets de test

Cette section présente et compare les résultats de classification obtenus sur les six objets virtuels testés pour les cinq algorithmes de détection des points d'intérêt, à savoir HKS, "attention visuelle", "attention visuelle optimisée", "points saillants" et "sailance du maillage" pour les diverses techniques d'apprentissage mentionnées ci-dessus dans le cas où le module IMMI est utilisé pour l'extraction des caractéristiques à partir des empreintes.

5.2.1. Résultats expérimentaux utilisant l'apprentissage profond

Les modèles de classification de données pour l'apprentissage profond ont été générés avec *RapidMiner* à l'aide du module *Deep Learning* (figure 22). Celui-ci est basé sur un réseau neuronal multicouche entraîné avec un gradient stochastique descendant, utilisant la rétro-propagation. Chaque modèle a six sorties, une correspondant à chaque type d'objet. Chacune des empreintes a été étiquetée et traitée individuellement. L'architecture du module d'apprentissage profond utilise une fonction correctrice d'activation linéaire et est entraînée à l'aide de 10 itérations. Elle utilise un taux d'apprentissage adaptatif et les facteurs de régularisation $L_1=e^{-5}$ et $L_2=0$, respectivement. Les résultats sont montrés pour les divers algorithmes testés dans le tableau 1. Notez que les résultats complets sous forme de matrice de confusion sont disponibles en annexe.

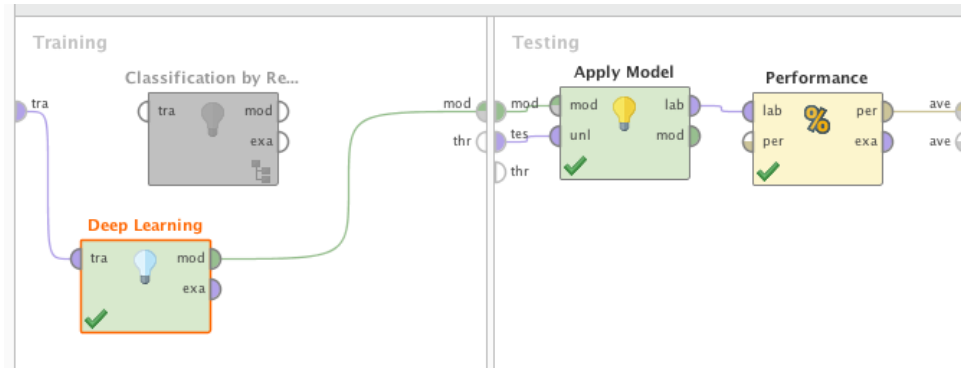


Figure 22 : Capture d'écran du sous-processus de classification utilisant l'algorithme Deep Learning.

En somme, la génération de modèles de classification avec l'apprentissage profond n'est pas adaptée au type de classification que nous tentons d'effectuer dans le présent travail, puisque la performance se situe en bas de 50%. Notons toutefois que les meilleurs résultats sont ceux dont les points d'intérêt pour les empreintes ont été obtenus avec l'algorithme d'attention visuelle.

Tableau 1 : Résultats du modèle de classification basé sur l'apprentissage profond pour six objets.

Apprentissage profond	Résultats IMMI
HKS	40,15%
Attention visuelle	46,72%
Attention visuelle optimisée	32,49%
Points saillants	35,26%
Saillance du maillage	14,89%

5.2.2. Résultats expérimentaux utilisant le module *Naïve Bayes*

Le tableau 2 montre les résultats pour *Naïve Bayes*. Cet algorithme de classification utilise la correction de Laplace.

Tableau 2 : Résultats du modèle de classification pour *Naïve Bayes* pour six objets.

<i>Naïve Bayes</i>	Résultats
HKS	40,91%
Attention visuelle	29.06%
Attention visuelle optimisée	21,1%
Points saillants	28,47%
Saillance du maillage	2,52%

À la lumière des résultats de classification obtenus avec Naïve Bayes, nous estimons que cette technique n'est pas adaptée au type de classification que nous tentons d'effectuer. Dans le cas présent, c'est l'algorithme HKS qui a retourné les meilleurs résultats, mais ceux-ci se trouvent encore en bas de 50%.

5.2.3 Résultats expérimentaux utilisant les arbres de décision

Nous avons également expérimenté avec les arbres de décision, pour chacun de nos algorithmes d'identification de points d'intérêt. Le ratio de gain est utilisé comme critère de décision, et l'élagage et le pré-élagage de l'arbre sont également utilisés. Le paramètre de confiance utilisé pour le calcul pessimiste de l'erreur est établi à 0,25. Les résultats sont montrés dans le tableau 3.

Tableau 3 : Résultats du modèle de classification pour les arbres de décision pour six objets.

Arbres de décision	Résultats
HKS	36,36%
Attention visuelle	21,37%
Attention visuelle optimisée	26,58%
Points saillants	24,82%
Saillance du maillage	25,57%

Les résultats de classification avec les arbres de décision se sont avérés peu convaincants, le meilleur étant 36,36%, pour l'algorithme HKS.

5.2.4 Résultats de classification obtenus avec la forêt d'arbres décisionnels

La forêt d'arbres décisionnels utilisé contient dix arbres avec les mêmes paramètres que ceux utilisés par l'algorithme des arbres de décision et a une profondeur maximale de 20. Le tableau 4 présente les résultats de classification obtenus pour l'algorithme de forêt d'arbres décisionnels. Dans le cas présent, les meilleurs résultats ont été obtenus avec l'attention visuelle optimisée, avec un taux de 34,6%. Toutefois, les résultats de classification pour la forêt d'arbres décisionnels ne sont pas à la hauteur des attentes.

Tableau 4 : Résultats du modèle de classification pour les forêts d'arbres décisionnels pour six objets.

Forêt d'arbres	Résultats
HKS	34,09%
Attention visuelle	21,37%
Points saillants	26,28%
Attention visuelle optimisée	34,6%
Saillance du maillage	26,29%

5.2.5 Résultats de classification avec le *boosting* d'arbres de décision

La méthode de *boosting* d'arbres de décision utilise 20 arbres d'une profondeur maximum de cinq, ainsi qu'un taux d'apprentissage de 0,1. Le tableau 5 montre les résultats de classification effectuée avec le module *Gradient Boosted Tree* de RapidMiner.

Tableau 5 : Résultats du modèle de classification pour le *boosting* d'arbres de décision pour six objets.

<i>Boosting</i> d'arbres	Résultats
HKS	43,18%
Attention visuelle	36,75%
Attention visuelle optimisée	16,46%
Points saillants	36,50%
Saillance du maillage	2,40%

En observant les résultats dans les trois cas, on constate que le meilleur parmi ceux-ci est obtenu avec HKS, soit une précision de 43,18%.

5.2.6 Résultats de classification avec les machines de vecteurs de support

L'implémentation des machines de vecteurs de support (SVM) utilise un algorithme évolutif pour résoudre le problème d'optimisation de paramètre d'une SVM; elle utilise le noyau radial avec les paramètres $\gamma=1$ et $\varepsilon=0.1$; pour la partie évolutive, une population aléatoire de départ est utilisée, où seulement les meilleurs individus survivent, avec un maximum de 10 000 générations et une fraction de tournoi de 0,75. Les résultats obtenus avec le module *Evolutionary SVM* de *RapidMiner* sont montrés au tableau 6, le meilleur résultat ayant été obtenu avec l'attention visuelle optimisée.

Tableau 6 : Résultats du modèle de classification pour le module *Evolutionary SVM* pour six objets.

<i>Evolutionary SVM</i>	Résultats
HKS	27,27%
Attention visuelle	7,91%
Attention visuelle optimisée	35,02%
Points saillants	30,66%
Saillance du maillage	25,09%

5.2.7 Résultats de classification avec les *K* plus proches voisins

Pour les *K* plus proches voisins, chaque échantillon est simplement assigné à la classe la plus près, en utilisant la distance Euclidienne comme mesure de proximité. Les résultats obtenus avec le module *KNN* de *RapidMiner* sont montrés au tableau 7 pour les quatre algorithmes de détection des points d'intérêt.

Tableau 7 : Résultats du modèle de classification pour le module pour *K* plus proches voisins pour six objets.

<i>K</i> plus proches voisins	Résultats
HKS	70,45%
Attention visuelle	43,59%
Attention visuelle optimisée	35,02%
Points saillants	56,20%
Saillance du maillage	24,25%

Dans l'ensemble, la classification faite avec les *K* plus proches voisins s'est avérée la plus performante, notamment avec un résultat de 70,45% pour l'algorithme HKS, et de 56,20% pour les "points saillants".

La figure 23 résume les résultats obtenus pour tous les classificateurs en utilisant les caractéristiques récupérées par le module *IMMI* sur les données simulées. Une observation sommaire des résultats présentés à la figure 23 permet de rapidement noter qu'à l'exception de la classification avec *KNN*, il y a peu de résultats significatifs. Or, ceux-ci peuvent s'expliquer par le fait que les empreintes collectées sur certains de nos objets ont des formes très similaires.

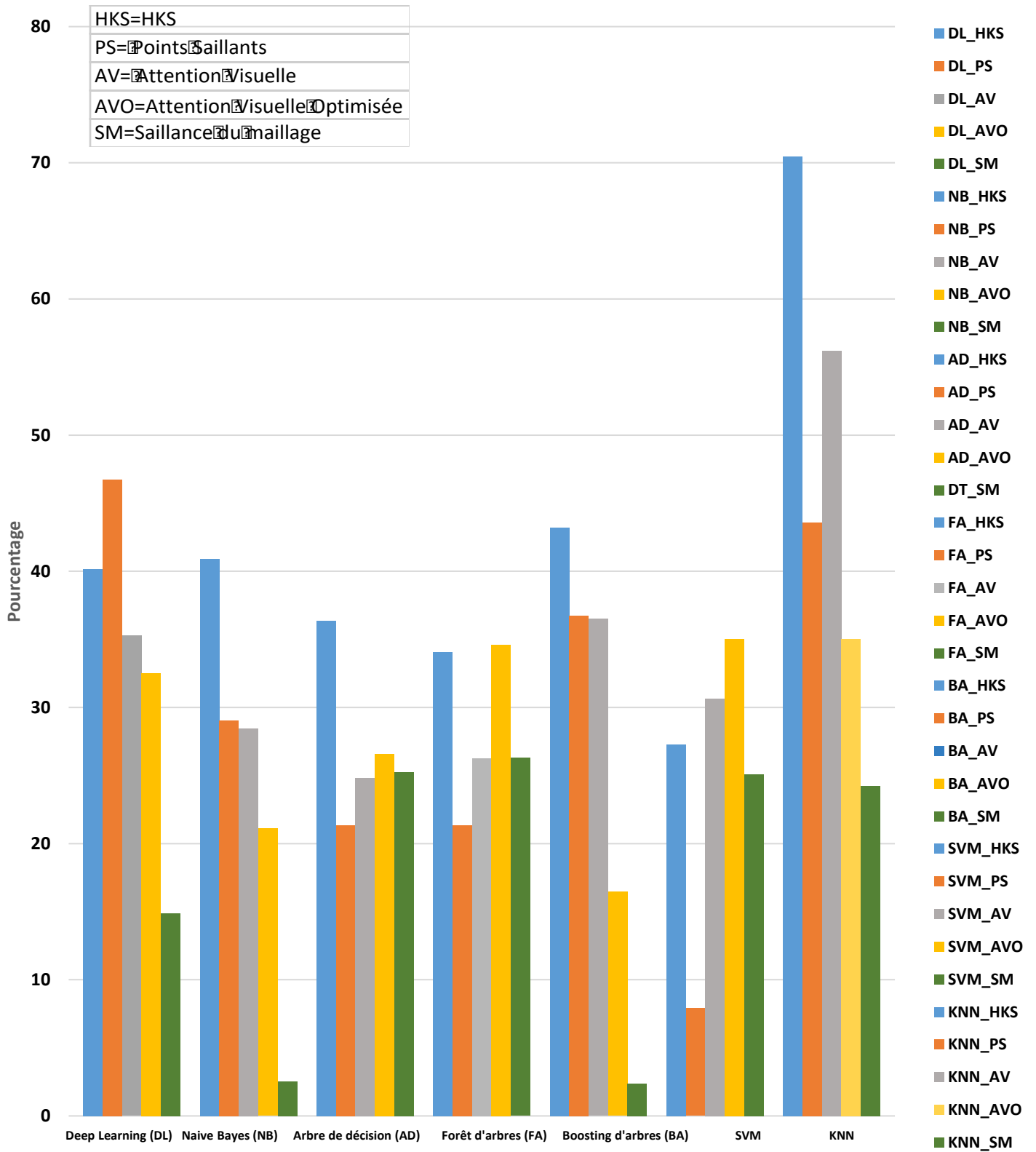


Figure 23 : Résultats obtenus pour les données simulées sur les six objets avec tous les classificateurs en utilisant le module IMMI.

Une analyse de la similarité basée sur la corrélation croisée normalisée, faite à l'aide de Matlab, l'a d'ailleurs démontré. La figure 24 montre, par exemple, la corrélation croisée normalisée pour deux objets, soit "cow" et "dromedary". Dans cette figure, plus les marqueurs sont près de la ligne supérieure, plus la ressemblance est grande. Dans ce cas-ci, la vaste majorité des items se retrouvent au-delà de 0,50 ce qui démontre une très grande ressemblance entre les empreintes des deux objets.

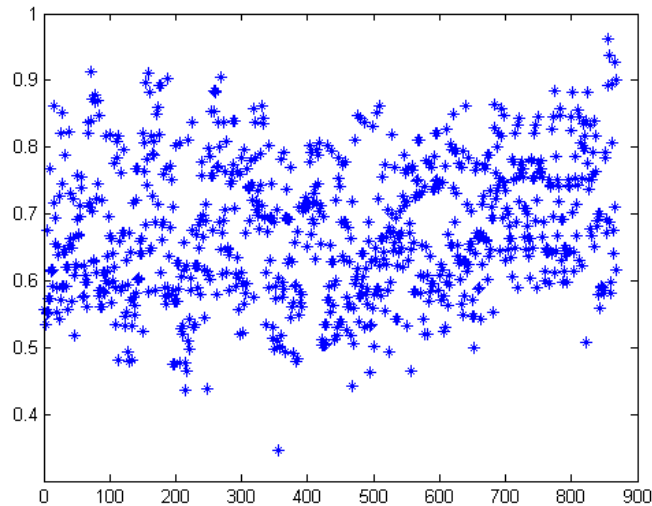


Figure 24 : Exemple de similarité (calculée à l'aide de la corrélation croisée normalisée) entre les empreintes de "cow" et "dromedary".

Pour valider qu'il s'agit d'un problème de similarité, nous avons, en premier lieu, procéder avec des tests sur quatre objets, en s'assurant que les objets présentant la plus grande similarité ne sont pas utilisés en même temps. Un échantillon des résultats pour les objets : "cow", "cup", "hand", et "glasses" sont présentés dans la section suivante.

5.3. Résultats expérimentaux pour la classification des données simulées en utilisant les caractéristiques IMMI pour quatre objets de test

5.3.1. Résultats expérimentaux utilisant l'apprentissage profond

Les modèles de classification de données pour l'apprentissage profond ont été générés avec *RapidMiner*, tel que mentionné à la section 5.2.1 à l'aide du module *Deep Learning*. Le réseau multicouche à six sorties utilisé a les mêmes paramètres que celui de la section 5.2.1. Les

résultats sont présentés pour les divers algorithmes de détection des points d'intérêt testés dans le tableau 8.

Tableau 8 : Résultats du modèle de classification basé sur l'apprentissage profond pour quatre objets.

Apprentissage profond	Résultats IMMI
HKS	25,93%
Attention visuelle	35,42%
Attention visuelle optimisée	81,60%
Points saillants	44,03%
Saillance du maillage	19,21%

L'attention visuelle optimisée est l'algorithme ayant retourné le meilleur taux de classification, avec un pourcentage de 81,60%.

5.3.2. Résultats expérimentaux utilisant le module *Naïve Bayes*

Le tableau 9 montre les résultats pour *Naïve Bayes*. Dans ce cas, le meilleur résultat est aussi obtenu à l'aide de l'attention visuelle optimisée, avec un pourcentage de précision de 78,90%.

Tableau 9 : Résultats du modèle de classification pour *Naïve Bayes* pour quatre objets.

<i>Naïve Bayes</i>	Résultats
HKS	33,33%
Attention visuelle	23,75%
Attention visuelle optimisée	78,90%
Points saillants	35,81%
Saillance du maillage	5,20%

5.3.3 Résultats expérimentaux utilisant les arbres de décision

Pour le modèle utilisant un arbre de décision avec les mêmes paramètres qu'à la section 5.2.3, les résultats sont montrés dans le tableau 10.

Tableau 10 : Résultats du modèle de classification pour les arbres de décision pour quatre objets.

Arbres de décision	Résultats
HKS	14,81%
Attention visuelle	35,42%
Attention visuelle optimisée	45,87%
Points saillants	32,89%
Saillance du maillage	34,02%

Les résultats de classification avec les arbres de décision se sont avérés peu convaincants, le meilleur étant 45,87%, avec l'attention visuelle optimisée.

5.3.4 Résultats de classification obtenus avec la forêt d'arbres décisionnels

Le tableau 11 présente les résultats de classification obtenus pour l'algorithme de forêt d'arbres décisionnels. Encore une fois, les meilleurs résultats ont été obtenus avec l'attention visuelle optimisée, avec un taux de 50,46%.

Tableau 11 : Résultats du modèle de classification pour les forêts d'arbres décisionnels pour quatre objets.

Forêt d'arbres	Résultats
HKS	37,04%
Attention visuelle	34,58%
Attention visuelle optimisée	50,46%
Points saillants	31,43%
Saillance du maillage	34,02%

5.3.5 Résultats de classification avec le *boosting* d'arbres de décision

L'attention visuelle optimisée a généré les meilleurs résultats, avec une précision de 88,99%, pour la méthode de *boosting* d'arbres de décision, tel que montré au tableau 12.

Tableau 12: Résultats du modèle de classification pour le *boosting* d'arbres de décision pour quatre objets.

<i>Boosting</i> d'arbres	Résultats
HKS	29,63%
Attention visuelle	30,42%
Attention visuelle optimisée	88,99%
Points saillants	45,62%
Saillance du maillage	2,68%

5.3.6 Résultats de classification avec les machines de vecteurs de support

Les résultats obtenus avec le module *Evolutionary SVM* de *RapidMiner* sont montrés au tableau 13. Le meilleur résultat est à nouveau obtenu avec l'attention visuelle optimisée, avec une précision de 92,66%.

Tableau 13 : Résultats du modèle de classification avec le module *Evolutionary SVM*, pour quatre objets.

SVM évolutif	Résultats
HKS	22,22%
Attention visuelle	35,98%
Attention visuelle optimisée	92,66%
Points saillants	29,05%
Saillance du maillage	27,72%

5.3.7 Résultats de classification avec les K plus proches voisins

Les résultats obtenus avec le module *KNN* de *RapidMiner* sont montrés au tableau 14 pour les cinq algorithmes de détection de points d'intérêt. L'attention visuelle optimisée donne encore une fois les meilleurs résultats, avec une précision de 95,41%. Celui-ci représente d'ailleurs le meilleur résultat de classification obtenu par le présent effort de recherche. On constate également un net recul de la performance de HKS par rapport à la classification faite pour six objets; en effet, on obtient ici un résultat de 22,22% alors qu'il était de 70,45% pour six objets.

Tableau 14 : Résultats du modèle de classification pour le module pour K plus proches voisins pour quatre objets.

K plus proches voisins	Résultats
HKS	22,22%
Attention visuelle	37,08%
Attention visuelle optimisée	95,41%
Points saillants	29,97%
Saillance du maillage	27,72%

Les résultats de classification pour quatre objets par rapport à six se sont avérés nettement meilleurs, particulièrement pour l'attention visuelle optimisée. Un seul résultat s'est avéré nettement meilleur pour six objets, soit celui obtenu avec les K plus proches voisins pour l'algorithme de détection de points d'intérêt HKS. Ce résultat est d'ailleurs tout à fait inconsistant avec les autres résultats de six à quatre objets, qui sont soit meilleurs ou, à quelques occasions, légèrement inférieurs. Au final, c'est l'attention visuelle optimisée qui a donné les meilleurs résultats de classification pour quatre objets, avec des résultats de 92,66% et de 95,41%, pour SVM évolutive et les K plus proches voisins respectivement.

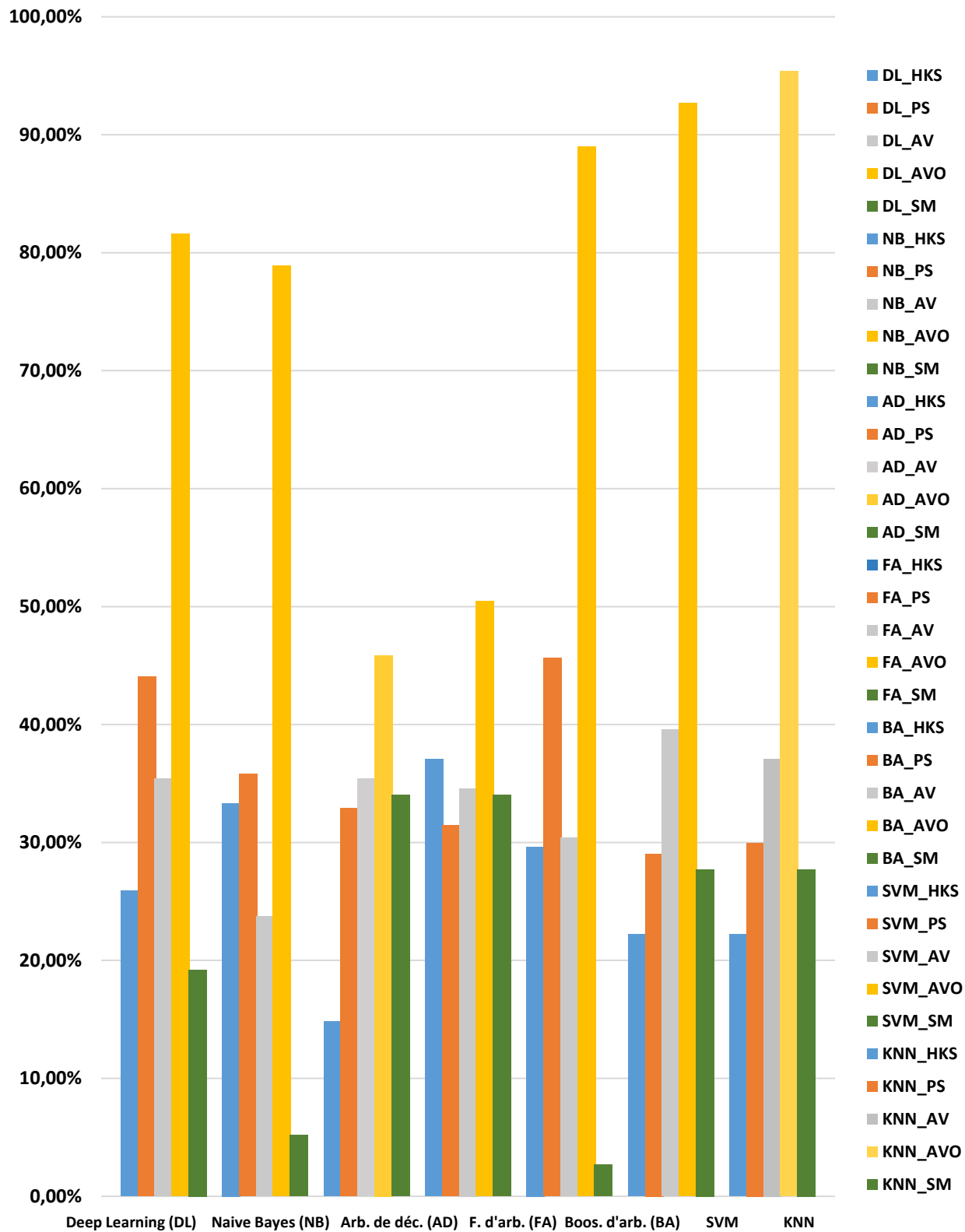


Figure 25 : Résultats obtenus pour les données simulées sur les quatre objets avec tous les classificateurs en utilisant le module IMMI.

5.4. Résultats expérimentaux pour la classification des données simulées en utilisant le sous-échantillonnage

En plus de classer les données à partir de caractéristiques obtenues à l'aide du module IMMI, nous les avons également classées à partir des vecteurs obtenus par le sous-échantillonnage des empreintes. Les empreintes ont d'abord été transformées en matrice de 32x32 et de 16x16 respectivement. Nous avons ensuite transformé ces matrices dans des vecteurs (par concaténation des lignes) qui ont été utilisés pour la classification. Les résultats sont montrés à la figure 24 pour le cas de l'attention visuelle optimisée. La figure compare le taux de reconnaissance pour les données simulées quand le module IMMI et le sous-échantillonnage sont utilisés. Les classifications effectuées avec des données sous-échantillonnées (16x16 et 32x32) ont toutes deux obtenu des performances semblables avec une différence d'au plus 2% entre les deux. Encore une fois, ce sont les méthodes des K plus proches voisins et SVM évolutif qui ont donné les meilleurs résultats de classification, soit 63,74% pour chacune d'elles dans le cas 32x32 (les dernières deux barres orange à la figure 26). Néanmoins, la performance de la méthode utilisant le module IMMI pour l'extraction de caractéristiques leur demeure nettement supérieure.

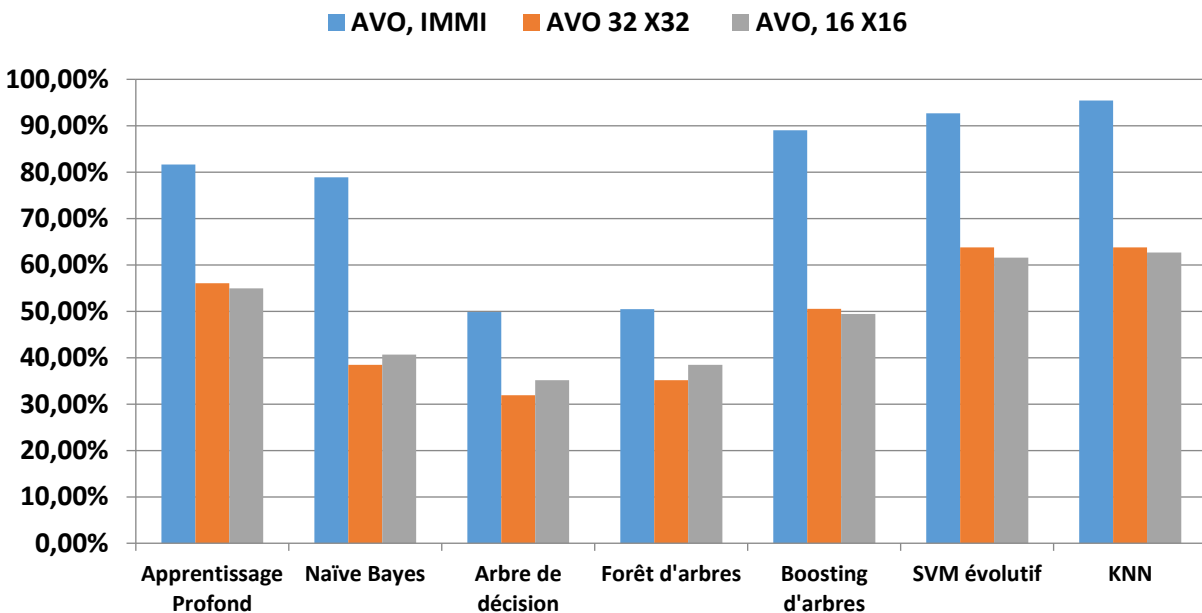


Figure 26 : Résultats obtenus avec tous les classificateurs, pour quatre objets virtuels, avec la méthode de l'attention visuelle optimisée.

5.5 Utilisation de la corrélation croisée normalisée pour améliorer les résultats de la classification

Étant donné les constats sur la similarité des empreintes à la section 5.2, pour améliorer la performance de classificateurs, nous avons décidé, pour chaque objet et en se basant sur la similitude calculée, (1) de ne conserver que les empreintes ayant un niveau de similarité inférieur à 0,50 ou (2) de ne retenir que les empreintes avec une similarité inférieure à 0,85 afin d'entraîner les classificateurs. Pour ce faire, nous avons déterminé la corrélation croisée normalisée entre toutes les empreintes, et nous avons retenu dans un cas celles dont le résultat de similarité était inférieur à 0,50 et dans l'autre, celles avec une similarité inférieure à 0,85. Au final, les empreintes utilisées pour effectuer la classification sont celles qui sont différentes l'une de l'autre. Les résultats sont illustrés au tableau 15 pour les quatre objets et au tableau 16 pour six objets. La figure 27 compare ensuite les résultats pour quatre objets virtuels.

Tableau 15 : Résultats de classification pour les quatre objets virtuels, empreintes les plus dissimilaires, attention visuelle optimisée.

Classificateurs	Rétention des empreintes avec la similarité < 0,50	Rétention des empreintes avec la similarité < 0,85
Apprentissage profond	56,99%	79,95%
Naïve Bayes	59,35%	71,25%
Arbre de décision	55,25%	60,25%
Forêt d'arbres	50,12%	55,26%
<i>Boosting</i> d'arbres	59,23%	58,00%
SVM évolutif	68,93%	81,82%
K plus proches voisins	75,00%	87,89%

Tableau 16 : Résultats de classification pour les six objets virtuels, empreintes les plus dissimilaires, attention visuelle optimisée.

Classificateurs	Rétention des empreintes avec la similarité < 0,50	Rétention des empreintes avec la similarité < 0,85
Apprentissage profond	44,99%	56.13%
Naïve Bayes	63,35%	75.18%
Arbre de décision	48,25%	56.00%
Forêt d'arbres	50,12%	54.25%
<i>Boosting</i> d'arbres	44,23%	55.17%
SVM évolutif	63,93%	73.69%
K plus proches voisins	68,00%	75.82%

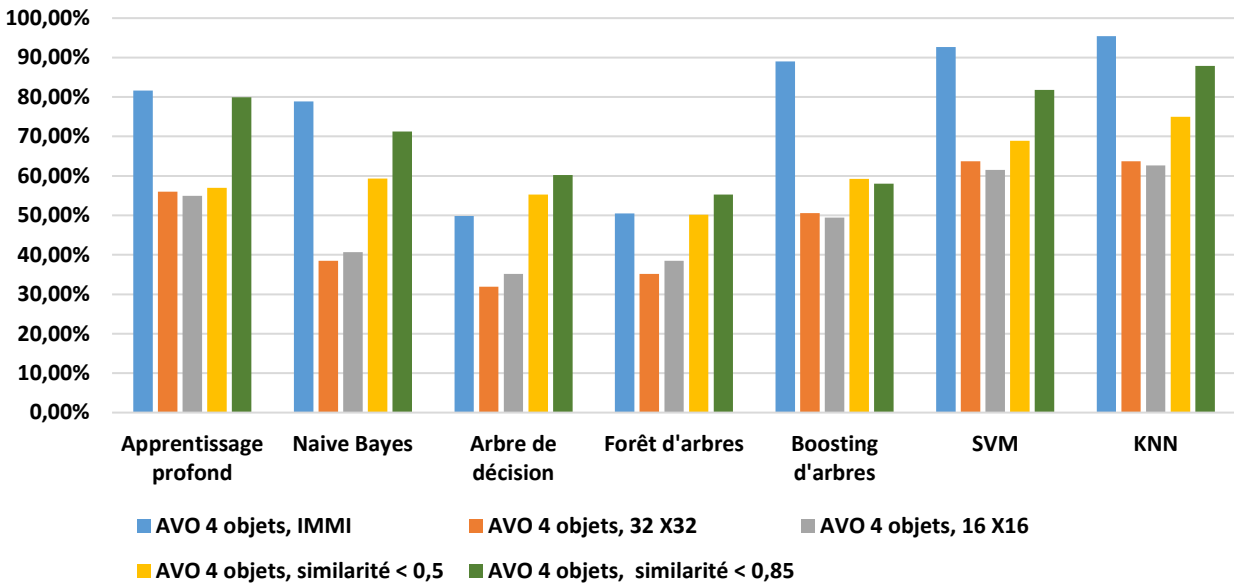


Figure 27 : Graphique illustrant les résultats obtenus avec tous les classificateurs, pour les objets virtuels, avec la méthode de l'attention visuelle optimisée et les empreintes les plus dissimilaires, comparativement à l'attention visuelle et toutes les empreintes.

Les résultats obtenus avec l'attention visuelle optimisée et IMMI, suivis des résultats générés à partir des empreintes les plus différentes (similarité < 0,85) ont produit les meilleurs résultats de classification. Pour l'attention visuelle optimisée seule et en utilisant les caractéristiques extraites par IMMI, nous avons obtenu la plus haute performance avec une précision de 95,41% pour quatre objets virtuels avec le regroupement des K plus proches voisins, et de 92,66% avec la méthode SVM évolutive (en bleu à la figure 27).

En comparant la performance quand le sous-échantillonnage est utilisé (en orange pour 32×32 et en gris pour 16×16, figure 27) avec le cas lorsque l'on utilise la similarité (les colonnes en jaune et vert à la figure 27), on constate que l'utilisation de la corrélation croisée afin d'éliminer les empreintes les plus similaires a également permis d'améliorer sensiblement les résultats obtenus par sous-échantillonnage. Pour les quatre objets, en éliminant les empreintes les plus semblables (similarité < 0.85, en vert à la figure 27), la meilleure performance que nous avons réussi à obtenir est de 87,89% (tableau 15), également avec les K plus proches voisins.

Des tests supplémentaires effectués ont démontré que l'utilisation de la corrélation croisée normalisée pour la sélection des empreintes a également permis d'améliorer les résultats de classification pour six objets virtuels, tel que présenté à la figure 28. Le meilleur résultat de classification (75.82% au tableau 16) a été obtenu avec les K plus proches voisins, avec les

empreintes ayant des similarités inférieures à 0,85. On y constate que pour six objets, l'élimination des empreintes les plus semblables pour ne conserver que les empreintes les plus différentes les unes des autres (ensemble des empreintes dont la similarité est inférieure à 0,85) est la technique ayant donnée les meilleurs résultats de classification. Si on se rapporte à la figure 24, on constate que de conserver l'ensemble des images ayant une corrélation croisée normalisée inférieure à 0,85 permet en fait de conserver la majorité des images, mais en éliminant toutefois les images présentant le plus de ressemblance.

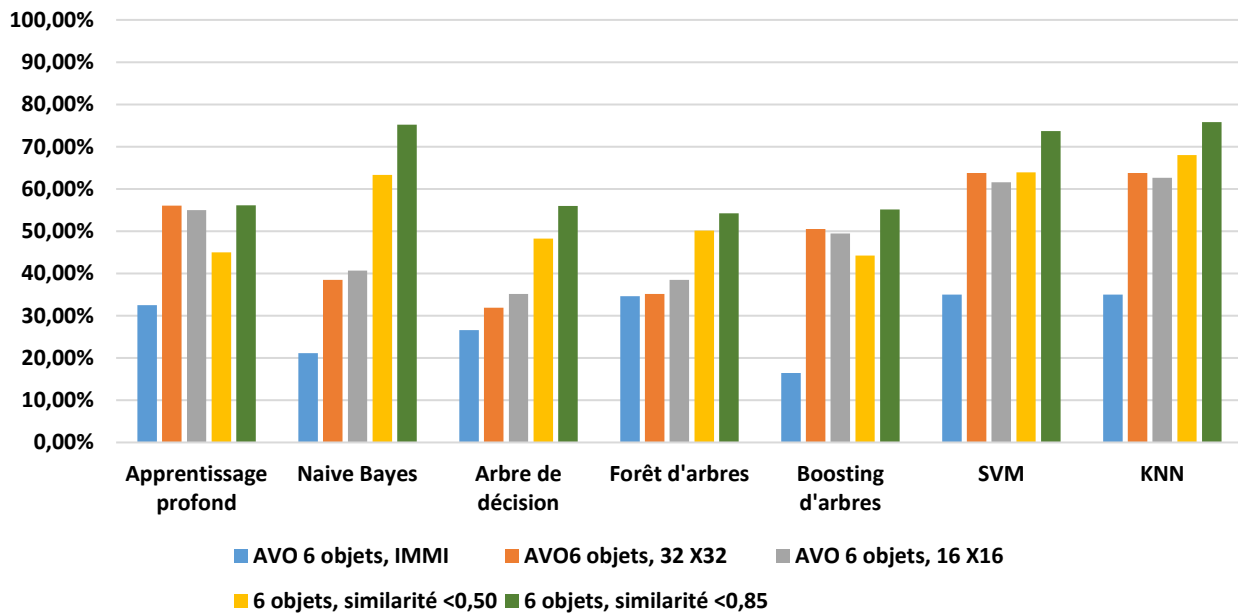


Figure 28 : Graphique illustrant les résultats obtenus avec tous les classificateurs, pour six objets virtuels avec les méthodes IMMI, pour les vecteurs 16x16 et pour les vecteurs 32 x32, ainsi qu'avec la corrélation croisée normalisée.

Les mêmes constats restent valables pour la figure 28 qui montre les résultats pour six objets que pour les quatre objets de la figure 27, principalement que les performances en tenant compte de la similarité sont meilleures que celles obtenues avec le sous-échantillonnage; par contre, dans ce cas-ci, les performances obtenues avec les caractéristiques extraites par IMMI y sont moins bonnes.

5.6 Résultats expérimentaux obtenus sur les objets réels

Nous avons aussi effectué des tests avec des empreintes obtenues à partir d'objets réels. Celle-ci ont dû subir un certain prétraitement, illustré à la figure 29. Celui-ci a permis d'éliminer

le quadrillage des images originales, susceptible de générer des distorsions pendant la classification ; le prétraitement a aussi permis de lisser l'image et de créer des gradients dans les zones de transition de ses contours de manière à la rendre plus en phase avec la réalité. Les empreintes du type (b) ont d'abord été transformées en matrice de 32x32 et de 16x16 ; nous en avons ensuite extrait des caractéristique IMMI en plus de transformer les matrices dans des vecteurs qui ont été utilisés pour la classification. Le tableau 17 et la figure 30 présentent les résultats lorsque l'extraction est basée sur IMMI et aussi lorsque des vecteurs construits à partir des matrices 16x16 et 32x32 sont extraits des images afin de les classifier.

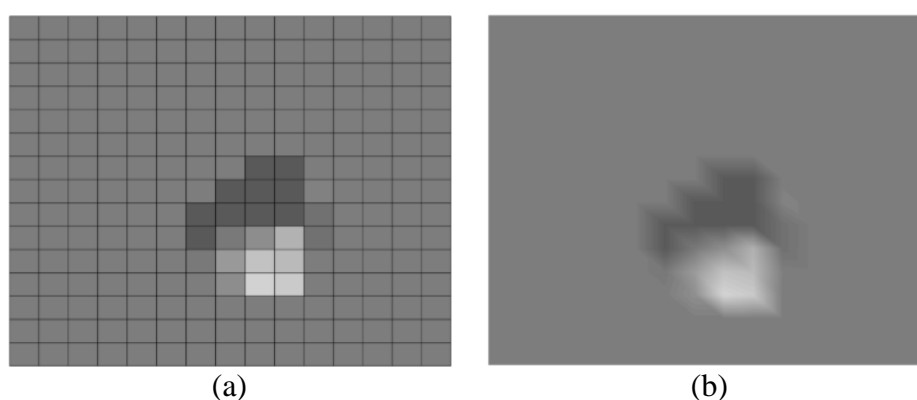


Figure 29 : Empreinte originale (a) et empreinte finale (b). On y constate l'adoucissement des contours de (a) vers (b).

À partir des empreintes des objets réels, nous avons aussi appliqué la méthode d'élimination des empreintes les plus semblables, d'abord en ne conservant que les empreintes ayant une similarité inférieure à 0,50, et ensuite celles ayant une similarité inférieure à 0,85.

Tableau 17 : Résultats de classification pour les objets réels.

Classificateurs	Résultats IMMI	Résultats 32×32	Résultats 16×16
Apprentissage profond	52.23%	56.25%	47.10%
Naïve Bayes	29.30%	68.75%	27.54%
Arbre de décision	33.76%	62.50%	22.46%
Forêt d'arbres	36.94%	50.00%	25.36%
Boosting d'arbre de décision	22.29%	56.25%	36.96%
SVM évolutif	40.76%	56.50%	57.25%
K plus proches voisins	40.13%	56.50%	57.97%

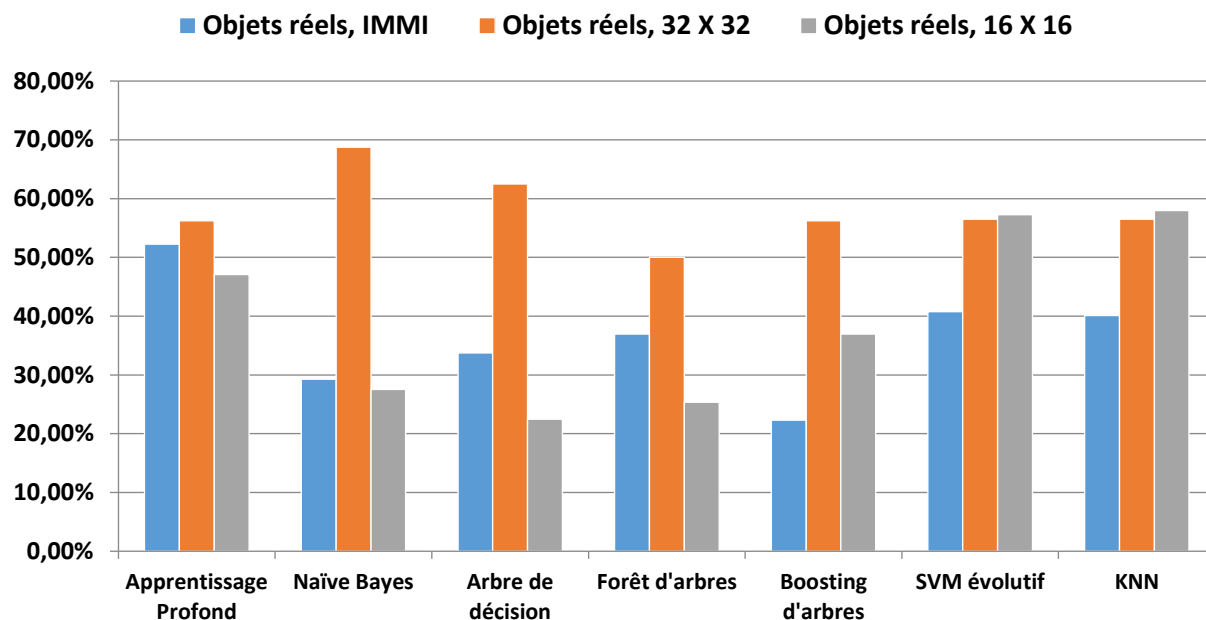


Figure 30 : Résultats obtenus avec tous les classificateurs, pour les quatre objets réels pour les caractéristiques extraites par le module IMMI et obtenues par sous-échantillonnage.

On peut observer que la meilleure performance est obtenue dans ce cas par l’algorithme Naïve Bayes et elle est de 68.75% pour une matrice de 32x32. Nous avons comparé les résultats de classification de ces ensembles avec les meilleurs résultats obtenus avec les ensembles de caractéristiques IMMI et sous-échantillonnées avec des vecteurs de 16x16 et de 32x32 de la figure 17. Cette comparaison est présentée dans le tableau 18, et à la figure 31.

Tableau 18 : Résultats de classification pour les quatre objets réels.

Classificateurs	Meilleurs résultats (parmi IMMI, 16×16, 32×32)	Rétention des empreintes avec la similarité < 0,50	Rétention des empreintes avec la similarité < 0,85
Apprentissage profond	56.25%	50.00%	59.14%
Naïve Bayes	68.75%	41.67%	58.10%
Arbre de décision	62.50%	55.56%	60.14%
Forêt d’arbres	50.00%	38.33%	54.05%
Boosting d’arbre de décision	56.25%	44.33%	47.62%
SVM évolutif	57.25%	67.23%	72.14%
K plus proches voisins	57.97%	68.91%	72.25%

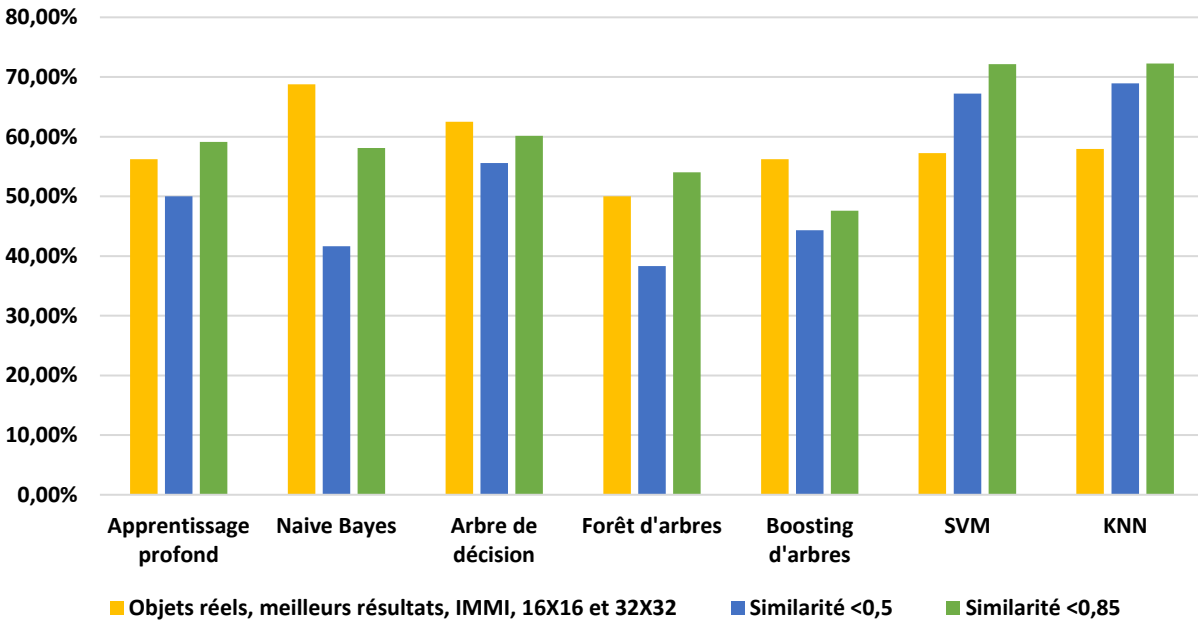


Figure 31 : Résultats obtenus avec tous les classificateurs, pour les quatre objets réels, en utilisant la similarité.

À la lumière de ces résultats, il apparaît évident que pour obtenir de bons taux de classification, il est impératif d'éliminer les empreintes présentant le plus de ressemblance possible (similarité plus grande que 0,85). À ce titre, l'utilisation de la corrélation croisée normalisée s'est avérée cruciale pour identifier les empreintes présentant de trop grandes similarités, et les éliminer. Le tableau 17 démontre qu'il est préférable de n'éliminer que les empreintes les plus semblables (avec une similarité $> 0,85$), que de ne conserver que celles les plus dissemblables (similarité $< 0,50$). Cette méthode permet également de conserver un plus grand nombre d'empreintes. Les résultats pour les empreintes dont la corrélation est inférieure à 0,50 sont moins bons que pour ceux dont la corrélation est inférieure à 0,85. Ces constats sont similaires à ceux fait pour les empreintes simulées à la section 5.5. Un taux maximal de 72,25% est obtenu pour l'algorithme des K plus proches voisins pour les quatre objets réels. Les résultats basés sur la similarité sont meilleurs que ceux quand toutes les empreintes sont utilisées.

Finalement, le tableau 19 et la figure 32 et présente une comparaison des meilleurs résultats obtenus pour les données simulées et pour les données réelles, pour quatre ainsi que six objets. On y note que pour Naïve Bayes, la forêt d'arbres, le *boosting* d'arbres, SVM et les K plus proches voisins, les résultats pour la similarité inférieure à 0,85 pour six objets virtuels sont légèrement meilleurs que pour quatre objets virtuels.

Tableau 19 : Résumé des meilleurs résultats obtenus.

Classificateurs	Attention visuelle optimisée, quatre objets virtuels, IMMI	Quatre objets réels, similarité < 0,85	Attention visuelle optimisée, six objets virtuels, similarité < 0,85	Six objets réels, similarité < 0,85
Apprentissage profond	81.65%	59.14%	56.13%	55.13%
Naïve Bayes	78.90%	58.10%	75.18%	55.67%
Arbre de décision	49.83%	60.14%	56.00%	58.16%
Forêt d'arbres	50.46%	54.05%	54.25%	50.25%
<i>Boosting</i> d'arbres	88.99%	47.62%	55.17%	45.50%
SVM	92.66%	72.14%	73.69%	68.25%
KNN	95.41%	72.25%	75.82%	70.47%

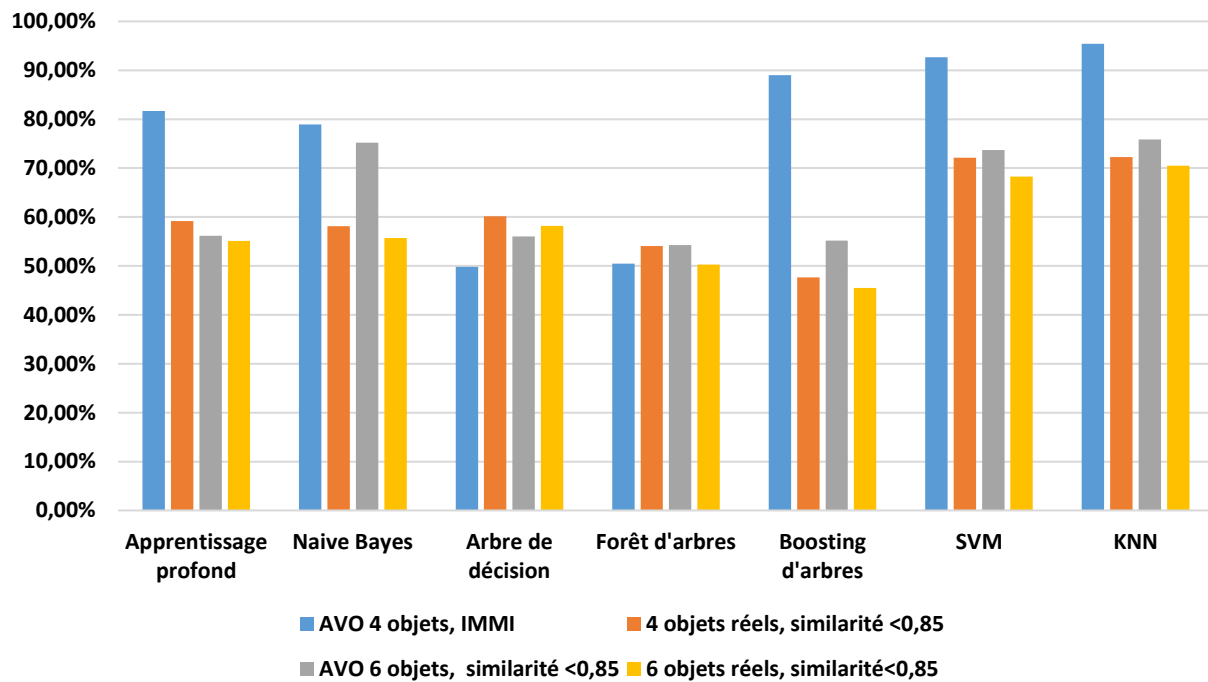


Figure 32 : Meilleurs résultats sur les données simulées et réelles.

La performance générale de la classification pour les objets réels s'avère moins bonne; ce résultat était prévisible puisque les images obtenues par le capteur tactile ont une très basse résolution, en plus d'être moins nettes. Tel qu'illustré au tableau 19, l'algorithme de *K* plus proches voisins est celui qui obtient la plus haute performance dans tous les cas. Le tableau 19,

ainsi que la figure 32, présentent également les résultats de classification pour six objets réels, qui se sont avérés un peu moins bons que pour quatre objets réels (4,01% inférieur dans le pire des cas), mais tout de même valable avec une précision de 70,47% pour les K plus proches voisins par rapport à la précision maximale de 72,25% obtenu pour les 4 objets en utilisant le même algorithme. Toutes les autres méthodes de classification performant sensiblement de la même manière pour six objets que pour quatre objets.

Chapitre 6. Contributions et conclusions

Dans le cadre de ce travail nous avons proposé une nouvelle technique pour obtenir des empreintes sur des objets virtuels. Afin de simuler cette prise d’empreinte sur une surface virtuelle plane, nous avons modélisé le corps du capteur virtuel comme un volume rectangulaire simulant la partie déformable du capteur résistif sensible à la force. L’intersection avec le capteur a ensuite été modélisée comme une série des courbes de différentes profondeurs. Celles-ci ont été projetées sur un plan unique afin de former des images en deux dimensions, en vue de leur analyse et de la reconnaissance de l’objet auquel ces empreintes appartiennent.

Nous avons aussi effectué de nombreux tests de classification avec sept classificateurs différents, soit l’apprentissage profond, les réseaux bayésiens, les arbres de décision, les forêts d’arbres, le *boosting* d’arbres de décision, les machines de vecteurs de support et les K plus proches voisins sur six ainsi que quatre objets virtuels. Avec une version optimisée de la technique d’attention visuelle pour identifier les points d’intérêts, nous avons obtenu d’excellents résultats de classification pour cinq des sept techniques décrites ci-haut, le meilleur résultat étant de 95,41% avec l’algorithme des K plus proches voisins pour quatre objets virtuels et de 72,25% pour six objets virtuels en utilisant le même algorithme. Nous avons montré des comparaisons de cette version optimisée avec d’autres méthodes de détection des points d’intérêt.

Nous avons également effectué des tests de classification sur des objets réels, avec le module *IMMI* de *RapidMiner*. Les résultats se sont avérés moins bons que pour les objets virtuels, avec une précision maximale de 68,75% pour la technique *Naïve Bayes*.

Nous avons démontré que l’élimination des empreintes les plus semblables à l’aide de la corrélation croisée normalisée augmentait sensiblement la précision de la classification. Grâce à cette technique, nous avons obtenu une précision de 72,25% pour la classification de quatre objets réels, avec l’algorithme des K plus proches voisins, et de 70,47% pour 6 objets réels avec ce même classificateur. Cette même technique appliquée à six objets virtuels nous a également permis d’obtenir un résultat de 75,82% (extraction *IMMI*).

En résumé, le présent projet a apporté les contributions suivantes, conformes aux objectifs décrits dans le chapitre 3 :

- Une étude de l'impact de l'utilisation des points d'intérêt obtenus par des algorithmes de détection inspirés de l'attention visuelle sur la collecte des données tactiles et la reconnaissance des objets à partir des données acquises ;
- La conception et la mise en œuvre d'une technique pour la simulation du processus d'acquisition de données tactiles sous forme d'empreintes (profils de pression) aux endroits identifiés par les points d'intérêt sur des objets 3D ;
- L'étude d'une série de techniques d'apprentissage pour la reconnaissance d'objets à partir des empreintes obtenues, incluant une technique originale de sélection des données basée sur la similarité des empreintes;
- Une validation expérimentale sur une base de données virtuelle ;
- Une validation expérimentale sur des données réelles collectées avec un capteur résistif sensible à la force, *FSR*;
- Une comparaison des résultats obtenus pour diverses méthodes de détection des points d'intérêt, des méthodes d'extraction des caractéristiques et des techniques d'apprentissage pour la reconnaissance d'objets.

Les résultats de ce travail sont le sujet des deux articles de recherche:

- N. Pedneault, A.-M. Cretu, "3D Object Recognition from Tactile Array Data Acquired at Salient Points", *IEEE Int. Symp. Robotics and Intelligent Sensors*, Ottawa, Canada, 2017 – accepté pour la publication.
- G. Rouhafzay, N. Pedneault, A.-M. Cretu, "A 3D Visual Attention Model to Guide Tactile Data Acquisition for Object Recognition", *4th MDPI Int. Electron. Conf. Sensors and Applications*, 15-30. Nov. 2017, Sciforum El. Conf. Series, 2017 - soumis.

En ce qui a trait aux travaux futurs, nous considérons que l'approche "sac de caractéristiques" devraient être explorée afin d'améliorer la précision de la classification des objets. Il serait aussi intéressant d'étudier l'impact de l'ordre de présentation des empreintes lors de l'apprentissage sur la performance de classificateurs. Dans le même ordre d'idée, le problème abordé dans ce mémoire pourrait bénéficier des algorithmes d'apprentissage en ligne. Il serait également souhaitable de déterminer avec précision le degré de similarité entre les empreintes au-delà duquel les empreintes doivent être rejetées avant de procéder à la classification des objets. De possibles améliorations sont aussi envisageables en utilisant un algorithme votant qui capitalise sur la décision de plusieurs classificateurs.

Bibliographie

- [1] P.K. Allen, "Integrating vision and touch for object recognition tasks." *The International Journal of Robotics Research* 7.6 (1988): 15-33.
- [2] O. Kroemer, C. H. Lampert, and J. Peters, "Learning Dynamic Tactile Sensing With Robust Vision-Based Training," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 55–557, Jun. 2011.
- [3] J. Sinapov, V. Sukhoy, R. Sahai, and A. Stoytchev, "Vibrotactile Recognition and Categorization of Surfaces by Humanoid Robot", *IEEE Trans. Robotics*, vol.27, no.3, pp. 488-497, 2011.
- [4] T.E.A. de Oliveira et al. "Touch sensing for humanoid robots." *IEEE Instrumentation & Measurement Magazine* 18.5 (2015): 13-19.
- [5] B. Hannaford and A.M. Okamura, "Haptics." *Springer Handbook of Robotics*. Springer Berlin Heidelberg, 2008. 719-739.
- [6] S.J. Lederman and R. L. Klatzky. "Hand movements: A window into haptic object recognition." *Cognitive psychology* 19.3 (1987): 342-368.
- [7] J.A. Fishel, and G. E. Loeb. "Bayesian exploration for intelligent identification of textures." (2012).
- [8] M. Schöpfer et al. "Identifying relevant tactile features for object identification." *Towards Service Robots for Everyday Environments*. Springer Berlin Heidelberg, 2012: 417-430.
- [9] R.L. Klatzky, and S. J. Lederman, "Identifying objects from a haptic glance." *Perception & Psychophysics* 57.8 (1995): 1111-1123.
- [10] A.M. Okamura, M. L. Turner, and M. R. Cutkosky. "Haptic exploration of objects with rolling and sliding." *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*. Vol. 3. IEEE, 1997.
- [11] T. Winiarski, and C. Zieliński, "Specification of multi-robot controllers on an example of a haptic device." *Robot Motion and Control 2009*. Springer London, 2009. 227-242.
- [12] C. Antonya, "Force Feedback in String based Haptic Systems." *Procedia Computer Science* 25 (2013): 90-97.
- [13] P. Giguère, and G. Dudek. "A simple tactile probe for surface identification by mobile robots." *IEEE Transactions on Robotics* 27.3 (2011): 534-55.

- [14] S. Youssefian, N. Rahbar, and E. Torres-Jara, "Contact Behavior of Soft Spherical Tactile Sensors," *IEEE Sens. J.*, vol. 14, no. 5, pp. 1435–1452, May 2014.
- [15] S. Ratnasingam and T. M. McGinnity, "Object recognition based on tactile form perception," *IEEE SSCI 2011 Symp. Ser. Comput. Intell. - RIIS 2011 2011 IEEE Work. Robot. Intell. Informationally Struct. Sp.*, pp. 26–31, 2011.
- [16] Y. Tenzer, L. P. Jentoft, and R. D. Howe, "The Feel of MEMS Barometers: Inexpensive and Easily Customized Tactile Array Sensors," *IEEE Robot. Autom. Mag.*, 2014.
- [17] M. Zillich and W. Feiten, "A versatile tactile sensor system for covering large and curved surface areas," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 20–24, 2012.
- [18] C. Pasca, et al. "Intelligent haptic sensor system for robotic manipulation." *IEEE Instrumentation and measurement technology conference proceedings*. Vol. 1. IEEE; 2004.
- [19] A. M. Cretu, T. E. A. De Oliveira, V. Prado Da Fonseca, B. Tawbe, E. M. Petriu, and V. Z. Groza, "Computational intelligence and mechatronics solutions for robotic tactile object recognition," *WISP 2015 - IEEE Int. Symp. Intell. Signal Process. Proc.*, no. 2, 2015.
- [20] A. Drimus, G. Kootstra, A. Bilberg, and D. Kragic, "Design of a flexible tactile sensor for classification of rigid and deformable objects," *Rob. Auton. Syst.*, vol. 62, no. 1, pp. 3–15, 2014.
- [21] H. Liu, D. Guo, and F. Sun, "Object Recognition Using Tactile Measurements: Kernel Sparse Coding Methods," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 3, pp. 656–665, Mar. 2016.
- [22] Z. Pezzementi et al. "Tactile-object recognition from appearance Information." *IEEE Transactions on Robotics* 27.3 (2011): 473-487.
- [23] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object identification with tactile sensors using bag-of-features," 2009 *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 243–248, Oct. 2009.
- [24] T. Bhattacharjee, J. M. Rehg, and C. C. Kemp. "Haptic classification and recognition of objects using a tactile sensing forearm." *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012.
- [25] C. M. Oddo, L. Beccai, M. Felder, F. Giovacchini, and M. C. Carrozza, "Artificial roughness encoding with a bio-inspired MEMS- Based tactile sensor array," *Sensors*, vol. 9, no. 5, pp. 3161–3183, 2009.
- [26] Y.-L. Park, R. J. Black, B. Moslehi, and M. R. Cutkosky, "Fingertip force control with embedded fiber Bragg grating sensors," 2008 *IEEE Int. Conf. Robot. Autom.*, pp. 3431–3436, May 2008.

- [27] Y. L. Park, K. Chau, R. J. Black, and M. R. Cutkosky, "Force sensing robot fingers using embedded fiber Bragg grating sensors and 74 deposition manufacturing," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2007, vol. 24, no. 7, pp. 1510–1516.
- [28] G. De Maria, C. Natale, and S. Pirozzi, "Force/tactile sensor for robotic applications," *Sensors Actuators, A Phys.*, vol. 175, pp. 60–72, 2012.
- [29] G. De Maria, C. Natale, and S. Pirozzi, "Tactile data modeling and interpretation for stable grasping and manipulation," *Rob. Auton. Syst.*, vol. 61, no. 9, pp. 1008–1020, Sep. 2013.
- [30] G. De Maria, C. Natale, and S. Pirozzi, "Tactile sensor for human-like manipulation," 2012 4th *IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics*, pp. 1686–1691, Jun. 2012.
- [31] C. Chorley, C. Melhuish, T. Pipe, and J. Rossiter, "Development of a tactile sensor based on biologically inspired edge encoding," 2009 *Int. Conf. Adv. Robot.*, 2009.
- [32] T. Assaf, C. Roke, J. Rossiter, T. Pipe, and C. Melhuish, "Seeing by touch: Evaluation of a soft biologically-inspired artificial fingertip in real-time active touch," *Sensors (Switzerland)*, vol. 14, no. 2, pp. 2561–2577, 2014.
- [33] P. Dallaire, Patrick, "Autonomous tactile perception: a combined improved sensing and Bayesian nonparametric approach." *Robotics and Autonomous Systems* 62.4 (2014): 422-435.
- [34] J. Sinapov, V. Sukhoy, R. Sahai, and A. Stoytchev, "Vibrotactile Recognition and Categorization of Surfaces by a Humanoid Robot," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 488–497, Jun. 2011.
- [35] D.R. Faria, R. Martins, and J. Dias, "Grasp Exploration for 3D Object 74 Representation using Probabilistic Map." *Doctoral Conference on Computing, Electrical and Industrial Systems*. Springer Berlin Heidelberg, 2010.
- [36] P. Payeur et al. "Intelligent haptic sensor system for robotic manipulation." *IEEE transactions on instrumentation and measurement* 54.4 (2005): 1583-1592.
- [37] C. Pasca, *Smart tactile sensor*. Diss. University of Ottawa (Canada), 2004.
- [38] S. Datta, et al. "Performance analysis of object 74 classification and matching from tactile images using wavelet energy features." *Procedia Technology* 10 (2013): 805-812.

- [39] T. Corradi, P. Hall, and P. Iravani, "Tactile Features: Recognizing Touch Sensations with a Novel and Inexpensive Tactile Sensor." *Conference Towards Autonomous Robotic Systems*. Springer International Publishing, 2014.
- [40] A. Aggarwal, and F. Kirchner, "Object recognition and localization: The role of tactile sensors." *Sensors* 14.2 (2014): 3227-3266.
- [41] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing-from humans to humanoids," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 1–20, Feb. 2010.
- [42] D. S. Chaturanga, Z. Wang, V. A. Ho, A. Mitani, and S. Hirai, "A biomimetic soft fingertip applicable to haptic feedback systems for texture identification," in 2013 *IEEE International Symposium on Haptic Audio Visual Environments and Games (HAVE)*, 2013, pp. 29–33.
- [43] T.E.A. de Oliveira, et al. "Data-driven analysis of kinaesthetic and tactile information for classification." *Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2015 IEEE International Conference on*. IEEE, 2015.
- [44] M. Bordegoni et al. "A force and touch sensitive self-deformable haptic strip for exploration and deformation of digital surfaces." *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*. Springer Berlin Heidelberg, 2010.
- [45] R. Bajcsy, "Active perception." *Proceedings of the IEEE* 76.8 (1988): 966-1005.
- [46] A. Aggarwal, and P. Kampmann, "Tactile sensors based object recognition and 6d pose estimation." *International Conference on Intelligent Robotics and Applications*. Springer Berlin Heidelberg, 2012
- [47] A.M. Okamura, "Uniting haptic exploration", *Robotics Research*. Springer Berlin Heidelberg, 2003. 225-238.
- [48] D. Xu, G. E. Loeb, and J. A. Fishel. "Tactile identification of objects using Bayesian exploration." *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013.
- [49] N.F. Lepora, U. Martinez-Hernandez, and T. J. Prescott. "A SOLID case for active bayesian perception in robot touch." *Conference on Biomimetic and Biohybrid Systems*. Springer Berlin Heidelberg, 2013.
- [50] S. Luo et al. "Tactile Object Recognition with Semi-Supervised Learning." *International Conference on Intelligent Robotics and Applications*. Springer International Publishing, 2015.
- [51] S.K. Yeung et al. "High sampling resolution tactile sensor for object recognition." *IEEE Transactions on Instrumentation and Measurement* 43.2 (1994): 277-282.

- [52] J. Sturm, "Object Recognition Using Tactile Sensors." *Approaches to Probabilistic Model Learning for Mobile Manipulation Robots*. Springer Berlin Heidelberg, 2013, pp. 125-139.
- [53] T.C. Phung et al. "Edge identification of a small object through a low-resolution tactile sensor array." *International Journal of Precision Engineering and Manufacturing* 11.2 (2010): 247-254.
- [54] O. P. Odiase, and R. C. Richardson, "Object Characterisation using a Haptic Probe." *Climbing and Walking Robots*. Springer Berlin Heidelberg, 2006. 825-832.
- [55] M. R. Cutkosky, R. D. Howe, and W. R. Provancher, "Force and tactile sensors." *Springer Handbook of Robotics*. Springer Berlin Heidelberg, 2008. 455-476.
- [56] J. Sturm, "Basics." *Approaches to Probabilistic Model Learning for Mobile Manipulation Robots*. Springer Berlin Heidelberg, 2013. 13-33.
- [57] R. Ma et al. "Linear dynamic system method for tactile object classification." *Science China Information Sciences* 57.12 (2014): 1-11.
- [58] L. Carotenuto et al. "A fuzzy classifier for tactile sensing." *Journal of Intelligent and Robotic systems* 20.1 (1997): 71-86.
- [59] A. Khasnobish et al. "Object-74 recognition and 3D reconstruction from tactile sensor images." *Medical & biological engineering & computing* 52.4 (2014): 353-362.
- [60] D. Babu et al. "Machine Learning Based 74 Classification Using Tactile Sensor Array." *Advanced Computing, Networking and Informatics-Volume 1*. Springer International Publishing, 2014. 47-55.
- [61] Y. Chebotar, O. Kroemer and J. Peters. "Learning robot tactile sensing for object manipulation." *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014.
- [62] E. Castillo-Muñiz, J. Rivera-Rovelo and E. Bayro-Corrochano. "Using haptic and neural networks for surface and mechanical properties 3D reconstruction." *International Conference on Ubiquitous Computing and Ambient Intelligence*. Springer Berlin Heidelberg, 2012.
- [63] D. Keren et al. "Recognizing 3D objects using tactile sensing and curve invariants." *Journal of Mathematical Imaging and Vision* 12.1 (2000): 5-23.
- [64] Y.B. Kim et al. "Exploration and reconstruction of unknown object by active touch of robot hand." *Intelligent Service Robotics* 8.3 (2015): 141-149.

- [65] S.-H. Kim et al. "Texture classification using a polymer-based MEMS tactile sensor." *Journal of micromechanics and microengineering* 15.5 (2005): 912.
- [66] M. Chagnon-Forget, et al. "Enhanced Visual-Attention Model for Perceptually Improved 3D Object Modeling in Virtual Environments." *3D Research* 7.4 (2016): 30.
- [67] H. Dutagaci, C. P. Cheung, and A. Godil, "Evaluation of 3D interest point detection techniques via human-generated ground truth," *The Visual Computer*, vol. 28, pp. 901- 917, 2012.
- [68] U. Castellani, M. Cristani, S. Fantoni, and V. Murino, "Sparse points matching by combining 3D mesh saliency", *Eurographics*, vol. 27, pp. 643-652, 2008.
- [69] J. Sun, M. Ovsjanikov and L. Guibas. "A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion." *Computer graphics forum*. Vol. 28. No. 5. Blackwell Publishing Ltd, 2009
- [70] C. H. Lee, A. Varshney, and D. W. Jacobs, "Mesh Saliency", *ACM SIGGRAPH*, vol. 174, pp. 659-666, 2005.
- [71] *Shape Analysis Research Project*, A Benchmark for 3D Interest Point Detection Algorithms, <http://www.itl.nist.gov/iad/vug/sharp/benchmark/3DInterestPoint/>, NIST, consulté en janvier 2017.
- [72] Signal Processing Laboratory, <http://77ab.cz/en/research/data-mining/articles>.
- [73] S.S. Haykin et al. *Neural networks and learning machines*. Vol. 3. Upper Saddle River, NJ, USA:: Pearson, 2009, chapitre 1.
- [74] J. Schmidhuber, "Deep learning in neural networks: An overview." *Neural networks* 61 (2015): 85-117
- [75] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian Network Classifiers." *Machine Learning* 2.29 (1997): 131-163.
- [76] J.R. Quinlan, "Induction of decision trees." *Machine learning* 1.1 (1986): 81-106.
- [77] T.K. Ho, "Random decision forests." *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*. vol. 1. IEEE, 1995.
- [78] J.H. Friedman, "Greedy function approximation: a gradient boosting machine." *Annals of statistics* (2001): 1189-1232.
- [79] rapidminer.com

[80] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic press, 2013, chapitre 6.

[81] B.E. Boser, I. M. Guyon, and V. N. Vapnik. "A training algorithm for optimal margin classifiers." *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992.

[82] G. Rouhafzay and A.-M. Cretu, "Selectively-Densified Mesh Construction for Virtual Reality Applications using Visual Attention Derived Salient Points", *IEEE Int. Conf. Computational Intelligence and Virtual Environments for Measurements Systems and Applications*, pp. 99-104, Annecy, France, 2017.

Annexes

Annexe A – code pour générer les intersections

```
%%% Projet de maîtrise - RECONNAISSANCE D'OBJETS
%%% TRIDIMENSIONNELS
%%% À PARTIR DE DONNÉES TACTILES
%%% Université du Québec en Outaouais - 2017
%%% Created by: Nicolas Pedneault, from a script developed
%%% originally
%%% by Francis Gorman.
%%% Intersection between a 3D SURFACE and a PLANE, at a
%%% precise point.

% -----
% Part 1
% Drawing Object

% Setting up object environment
method_name = 'Salient_points';
ALGORITHM_IPs_MAIN_FOLDER
='/Users/Nick/Documents/MATLAB/ALGORITHMS_INTEREST_POINTS_2/';
ALGORITHM_IPs_DIR = [ALGORITHM_IPs_MAIN_FOLDER method_name '/'];

% Ex.: Object #1
strb=sprintf ('%s%d%s','object', 12, '.mat');
load([ALGORITHM_IPs_DIR strb]);
load(['/Users/Nick/Documents/MATLAB/MODEL_DATASET_2/' strb]);

% For visual attention algorithm, use these variables:
%strb=sprintf ('%s%s','hand_3_new','.mat');
%strb2=sprintf ('%s%d%s','object', 24, '.mat');
%load(['/Users/Nick/Documents/MATLAB/MODEL_DATASET_2/' strb2]);
%load(['/Users/Nick/Documents/MATLAB/VisAtt/' strb]);%load interesting points from a method

% -----
% Part 2
% Drawing points of interest

IP_subject = V(IP_vertex_indices,:);
myPlot=trisurf(F,V(:,1),V(:,2),V(:,3),ones(length(F),1),'edgealpha',0);
hold on;
myPlot2=plot3(IP_subject(:,1),IP_subject(:,2),IP_subject(:,3),'r','MarkerSize',15);
%direction = [0 1 0];
%rotate(myPlot,direction,180);
%rotate(myPlot2,direction,180);
%view(0,90);
VOXEL_NUMBER = 5 ;           % Amount of voxels to analyze

%Code added by Nick Pedneault; identifying connected points.

pointsLength=length(IP_vertex_indices);
```



```

for i=1:pointsLength
    theSalientPoint=IP_vertex_indices(i);
    fprintf('The salient Point is: %d\n',theSalientPoint);
    [row,col]=find(F==theSalientPoint);%find all occurrences of the salient point in the matrix of triangles (F).
    %row is an array of indexes where the salient point was found in the F
    %matrix. Each index corresponds to a triangle.
    %fprintf('%d\t %d\t %d\t',V(myVal(1)));
    %fprintf('%d\t %d\t %d\t',L(1),L(2), L(3));
    x=[];%array of all points touching the salient point.
    pointsLimit=size(row);
    for j=1:pointsLimit %iterating over the triangles that include the salient point.
        %fprintf('This triangle includes the salient point. The triangle # is: %d\t',row(j));
        %fprintf('\n');
        %fprintf('%d\t',F(row(j),:));
        %fprintf('\n');
        %fprintf('Touching points and coordinates: ');
        %x=[];

        for k=1:3

            if F(row(j),k)==theSalientPoint
                continue
            else
                %x=add(x,F(row(j),k));
                %fprintf(' %d\t %d\t',F(row(j),k), V(F(row(j),k),:));
                %fprintf('\n Touching Point and coordinates: ');
                y=F(row(j),k);
                %fprintf('%d\n',y);
                x=[x y];
                x=unique(x); %x contains all the points that touch the salient point.
                    %For each point of x, I now want to find all triangles that
                    %touch these points, and build an a
            end
            %fprintf('Array of touching points \n %d\n',x);

        end

    end

end
fprintf('Array of touching points is: \n');

fprintf('%d ',x(1,:));
fprintf('\n');

p0=x(1,1); %Identifying indexes of points on plan.
p1=x(1,2);
p2=x(1,3);
planP0=V(p0,:); %Obtaining coordinates of these points.
planP1=V(p1,:);
planP2=V(p2,:);

orthogonalVector=cross(planP0-planP1,planP0-planP2); %Obtaining normal vector.
normalVector=orthogonalVector./norm(orthogonalVector);
%normalVector=normalVector;
fprintf('normal vector is: \t');
fprintf('%d',normalVector);
fprintf('\n');

```

```

fprintf('p0 = %d \t',p0);
fprintf(' coordinates %d %d %d',V(p0,:));
fprintf('\n');

%%%%%%%%%%

EXT = '.mat';
DIR = '/USERS/Nick/Documents/MATLAB/INPRINTS_NEW/Salient_points/object12_cow/'; %needs to
change for object, and then for method.
strb3="";
strb4="";

strb3=strcat('point_new_',num2str(i));
strb4=strcat(strb3, '/');
fprintf('\n%s\n', strb4);
mkdir(DIR,strb4);
DIR1=strcat(DIR,strb4);
fprintf('\n%s\n', DIR1);

fv.faces=F;
fv.vertices=[V(:,1),V(:,2),V(:,3)];

DEPTH = 5;           % Capture occurrences
k = 0.001;           % Plane position
voxel = cell(DEPTH,1); % Intersection container

for l = 1:DEPTH
    fprintf('%d',planP0);
    try
        planP0=k*(normalVector) + planP0;
        %normalVector=normalVector;
        % circlePlane3D was obtained from the MatLab portal.
        fvc=circlePlane3D(planP0,normalVector,0.03, 1, 0, 'b', '-');
        fvc2.faces=get(fvc,'Faces');
        fvc2.vertices=get(fvc,'Vertices');

        [intersect12, Surf12]=SurfaceIntersection(fv,fvc2);

        temp=Surf12.vertices;
        [m n]=size(temp);
        voxel(l,1) = mat2cell(temp,m);

        catch
            fprintf('intersect problem\n');

    end;
end;

for m = 1:VOXEL_NUMBER
    OBJ = ['voxel_' num2str(m)];

    FILE = [DIR1 OBJ EXT];

```

```
    save(FILE, 'voxel', '-v7.3');  
end;  
  
clear voxel;  
  
end;  
%%%%%%%%%%  
  
%End of code added by Nick Pedneault
```

Annexe B – code pour générer les empreintes

```
% % % Projet synthèse - RECONNAISSANCE D'OBJETS
% % %TRIDIMENSIONNELS
% % % À PARTIR DE DONNÉES TACTILES
% % % Université du Québec en Outaouais - 2017
% % % Created by: Nicolas Pedneault, from a script originally
% % %developed
% % % by Francis Gorman.
% % % Creation of imprints from intersects.

% -----
% Part 1
% Loading voxel

%DIR = '/Users/Nick/Documents/MATLAB/INPRINTS/Salient_points/object12_hand3/';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Image='object24_hand3';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
strb=";

EXT = '.mat';
POINTS = 5;
f=Figure();
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:21
    DIR = '/Users/Nick/Documents/MATLAB/INPRINTS_NEW/VisAtt/object24_hand3/';
    %strb=strcat('point_new_positive_',num2str(j));
    strb=strcat('point_new_negative_',num2str(j));
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    strb2=strcat(Image,'_');
    strb2=strcat(strb2,strb);
    strb5=strcat(strb2,'_b');
    strb2=strcat(strb2,'.png');
    strb5=strcat(strb5,'.png');

    strb=strcat(strb,'/');
    DIR=strcat(DIR,strb);

for i = 1:POINTS
    OBJ = ['voxel_' num2str(i)];
    FILE = [DIR OBJ EXT];
    capt_data = matfile(FILE, 'Writable', true);
    voxel = capt_data.voxel;
end

% -----
% Part 2
% Encapsulating data

OCC = 5;
h = zeros(OCC,1);
```

```

data = []; % stretches to contain all the data associated to a 3D object point
data1 = []; %at the end, data1 will contain all the projected points making the inprint.
data2 = [];
data3 = [];
data4 = [];
data5 = [];
%finalUsableData = [];
%color=[r b y r b y r b];
projData = [];

for i = 1:OCC
    temp = voxel(6-i,1);
    temp = temp{1,1};

    %calculate normal vector with bottom intersect
    if i==1
        %define points required to calculate normal vector.
        try
            x0=temp(1,1);
            y0=temp(2,1);
            z0=temp(3,1);
            x1=temp(1,2);
            y1=temp(2,2);
            z1=temp(3,2);
            x2=temp(1,3);
            y2=temp(2,3);
            z2=temp(3,3);
            p0=[x0 y0 z0];
            p1=[x1 y1 z1];
            p2=[x2 y2 z2];
        catch
            fprintf('empty voxel\n');
            continue;
        end;

        p=p0; %p in the equation
        %q in the projection equation will be every other points from other
        %voxel.

        orthogonalVector=cross(p0-p1,p0-p2); %Obtaining normal vector.
        normalVector=orthogonalVector./norm(orthogonalVector); %n in the equation

        [theta,rho,z] = cart2pol(normalVector(1),normalVector(2),normalVector(3)); %trying to obtain
        cylindrical coordinates of normal vector.

        %add points to graph
        data = horzcat(data, temp);
        %fill3(data(1,:), data(2,:), data(3,:),[.1 .1 .1]);
        %hold on;
        data1= data;
    else
        %project all other points onto bottom plane
        lenTemp=length(temp);
        data = [];
    end
end

```

```

dataTemp = [[];[];[]];
for j=1:lenTemp
    qX=temp(1,j);
    qY=temp(2,j);
    qZ=temp(3,j);
    q=[qX qY qZ];

    %projection function here

    proj=q-dot(q-p,normalVector)*normalVector;
    proj = proj .';
    dataTemp = horzcat(dataTemp, proj);
    data=[data proj];

    %data=projData;

end;
%finalUsableData=[finalUsableData data1];

if i==2
    data2=data;
end;
if i==3
    data3=data;
end;
if i==4
    data4=data;
end
if i==5
    data5=data;
end;

end

end
try

tri = delaunay(data4(1,:), data4(2,:));

[r,c] = size(tri);
disp(r);

grey4 = [0 0 .1];

h = trisurf(tri, data4(1,:), data4(2,:), data4(3,:), 'Facecolor',grey4,'EdgeColor','none');
axis off;
hold on;

tri = delaunay(data3(1,:), data3(2,:));

[r,c] = size(tri);
disp(r);

grey3 = [0 0 .3];

```

```

h = trisurf(tri, data3(1,:), data3(2,:), data3(3,:), 'Facecolor', grey3, 'EdgeColor', 'none');
axis vis3d;
axis off;

tri = delaunay(data2(1,:), data2(2,:));

[r,c] = size(tri);
disp(r);

grey2 = [0 0 .5];

h = trisurf(tri, data2(1,:), data2(2,:), data2(3,:), 'Facecolor', grey2, 'EdgeColor', 'none');
axis vis3d;
axis off;

hold on;

tri = delaunay(data1(1,:), data1(2,:));

%%
% How many triangles are there?

[r,c] = size(tri);
disp(r);

%% Plot it with TRISURF

grey1 =[0 0 .7];

h = trisurf(tri, data1(1,:), data1(2,:), data1(3,:), 'Facecolor', grey1, 'EdgeColor', 'none');
axis vis3d;

%% Clean it up

axis off;

hold on;

%tri = delaunay(data5(1,:), data5(2,:));

% How many triangles are there?

%[r,c] = size(tri2);
%disp(r);

%% Plot it with TRISURF

%grey5 = [0 0 .9];

%h = trisurf(tri, data5(1,:), data5(2,:), data5(3,:), 'Facecolor', grey5, 'EdgeColor', 'none');
%axis vis3d;
%axis off;

```

```

view(theta,rho);%angles obtained from normalVector
saveas(f, strb2);
hold off;
catch
fprintf('cannot draw inprint\n');
end;

%%reverse plans

try

tri = delaunay(data1(1,:), data1(2,:));

[r,c] = size(tri);
disp(r);

grey4 = [0 0 .1];

h = trisurf(tri, data1(1,:), data1(2,:), data1(3,:), 'Facecolor',grey4, 'EdgeColor', 'none');
axis off;
hold on;

tri = delaunay(data2(1,:), data2(2,:));

[r,c] = size(tri);
disp(r);

grey3 = [0 0 .3];

h = trisurf(tri, data2(1,:), data2(2,:), data2(3,:), 'Facecolor',grey3, 'EdgeColor', 'none');
axis vis3d;
axis off;

tri = delaunay(data3(1,:), data3(2,:));

[r,c] = size(tri);
disp(r);

grey2 = [0 0 .5];

h = trisurf(tri, data3(1,:), data3(2,:), data3(3,:), 'Facecolor',grey2, 'EdgeColor', 'none');
axis vis3d;
axis off;

hold on;

tri = delaunay(data4(1,:), data4(2,:));

%%
% How many triangles are there?

[r,c] = size(tri);
disp(r);

```



```

%% Plot it with TRISURF

grey1 = [0 0 .7];

h = trisurf(tri, data4(1,:), data4(2,:), data4(3,:), 'Facecolor',grey1, 'EdgeColor', 'none');
axis vis3d;

%% Clean it up

axis off;

%%%%%%%%%%

hold on;

% tri = delaunay(data5(1,:), data5(2,:));

% How many triangles are there?

%[r,c] = size(tri2);
%disp(r);

%% Plot it with TRISURF

%grey5 = [0 0 .9];
%h = trisurf(tri, data5(1,:), data5(2,:), data5(3,:), 'Facecolor',grey5, 'EdgeColor', 'none');
%axis vis3d;
%axis off;

view(theta,rho);%angles obtained from normalVector
saveas(f, strb5);
hold off;
catch
fprintf('cannot draw inprint 2\n');
end;
clear voxel;
end;

```

Annexe C – matrices de confusion

Apprentissage profond, six objets virtuels

Tableau C1 : Résultats du modèle de classification basé sur l'apprentissage profond pour l'algorithme HKS.

accuracy: 40.15% +/- 49.02% (mikro: 40.15%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	14	2	4	4	3	5	43.75%
pred. cup	2	2	1	1	1	0	28.57%
pred. drome...	6	2	14	4	4	1	45.16%
pred. glasses	6	2	3	11	5	7	32.35%
pred. hand	2	2	5	3	9	1	40.91%
pred. plane	1	0	2	2	1	5	45.45%
class recall	45.16%	20.00%	48.28%	44.00%	39.13%	26.32%	

Tableau C2 : Résultats du modèle de classification basé sur l'apprentissage profond pour l'algorithme d'attention visuelle.

accuracy: 46.72% +/- 49.89% (mikro: 46.72%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	13	2	1	3	5	7	41.94%
pred. cup	1	3	1	0	1	0	50.00%
pred. drome...	4	2	18	5	2	1	56.25%
pred. glasses	8	0	3	13	4	1	44.83%
pred. hand	3	2	4	2	10	3	41.67%
pred. plane	2	1	2	2	1	7	46.67%
class recall	41.94%	30.00%	62.07%	52.00%	43.48%	36.84%	

Tableau C3 : Résultats du modèle de classification basé sur l'apprentissage profond pour l'algorithme des points saillants.

accuracy: 35.26% +/- 47.78% (mikro: 35.26%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	8	6	3	4	1	4	30.77%
pred. cup	4	40	23	11	23	17	33.90%
pred. drome...	11	20	22	14	8	8	26.51%
pred. glasses	5	12	12	40	17	5	43.96%
pred. hand	6	12	7	14	35	16	38.89%
pred. plane	3	10	9	6	12	20	33.33%
class recall	21.62%	40.00%	28.95%	44.94%	36.46%	28.57%	

Tableau C4 : Résultats du modèle de classification basé sur l'apprentissage profond pour l'algorithme de saillance de treillis.

accuracy: 14.89% +/- 35.59% (mikro: 14.89%)

	true cow	true cup	true glasses	true hand	true dromedary	true plane	class precision
pred. cow	34	37	50	38	21	22	16.83%
pred. cup	47	14	28	24	22	14	9.40%
pred. glasses	47	26	12	25	10	9	9.30%
pred. hand	49	40	23	19	18	15	11.59%
pred. drome...	22	14	11	18	6	12	7.23%
pred. plane	18	19	11	9	10	39	36.79%
class recall	15.67%	9.33%	8.89%	14.29%	6.90%	35.14%	

Naive Bayes, six objets virtuels

Tableau C5 : Résultats du modèle de classification basé sur Naïve Bayes pour l'algorithme HKS.

accuracy: 40.91% +/- 49.17% (mikro: 40.91%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	9	2	4	6	2	1	37.50%
pred. cup	0	0	0	0	0	0	0.00%
pred. dromedary	2	0	6	1	1	1	54.55%
pred. glasses	0	0	0	0	2	2	0.00%
pred. hand	0	0	0	0	1	0	100.00%
pred. plane	1	0	1	0	0	2	50.00%
class recall	75.00%	0.00%	54.55%	0.00%	16.67%	33.33%	

Tableau C6 : Résultats du modèle de classification basé sur Naïve Bayes pour l'algorithme des points saillants.

accuracy: 29.06% +/- 45.40% (mikro: 29.06%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	1	0	3	2	2	1	11.11%
pred. cup	10	51	23	17	25	19	35.17%
pred. dromedary	5	5	4	2	4	4	16.67%
pred. glasses	17	32	38	63	55	28	27.04%
pred. hand	1	1	1	0	0	1	0.00%
pred. plane	3	11	7	5	10	17	32.08%
class recall	2.70%	51.00%	5.26%	70.79%	0.00%	24.29%	

Tableau C7 : Résultats du modèle de classification basé sur Naïve Bayes pour l’algorithme d’attention visuelle.

accuracy: 28.47% +/- 45.13% (mikro: 28.47%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	2	0	1	3	2	1	22.22%
pred. cup	4	3	5	1	2	0	20.00%
pred. dromedary	9	2	7	5	2	1	26.92%
pred. glasses	5	2	6	4	4	1	18.18%
pred. hand	5	2	4	6	11	4	34.38%
pred. plane	6	1	6	6	2	12	36.36%
class recall	6.45%	30.00%	24.14%	16.00%	47.83%	63.16%	

Tableau C8 : Résultats du modèle de classification basé sur Naïve Bayes pour l’algorithme de saillance de treillis.

accuracy: 2.52% +/- 15.68% (mikro: 2.52%)

	true cow	true cup	true glasses	true hand	true dromedary	true plane	class precision
pred. cow	0	18	3	10	2	10	0.00%
pred. cup	39	0	11	81	4	40	0.00%
pred. glasses	0	0	2	0	0	0	100.00%
pred. hand	123	88	97	8	51	22	2.06%
pred. drome...	4	3	2	12	0	28	0.00%
pred. plane	51	41	20	22	30	11	6.29%
class recall	0.00%	0.00%	1.48%	6.02%	0.00%	9.91%	

Arbres de décision, six objets virtuels

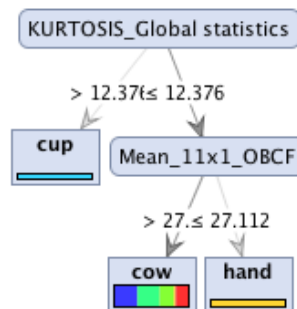


Figure C1 : Arbre de décision pour l’algorithme HKS.

Tableau C9 : Résultats du modèle de classification basé sur un arbre de décision pour l’algorithme HKS.

accuracy: 36.36% +/- 48.10% (mikro: 36.36%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	12	2	11	7	1	4	32.43%
pred. cup	0	0	0	0	0	1	0.00%
pred. dromedary	0	0	0	0	0	0	0.00%
pred. glasses	0	0	0	0	1	0	0.00%
pred. hand	0	0	0	0	4	1	80.00%
pred. plane	0	0	0	0	0	0	0.00%
class recall	100.00%	0.00%	0.00%	0.00%	66.67%	0.00%	

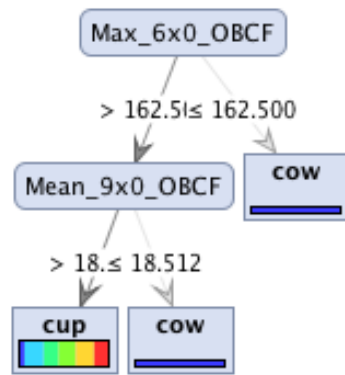


Figure C2 : Arbre de décision pour l’algorithme des points saillants.

Tableau C10 : Résultats du modèle de classification basé sur un arbre de décision pour l’algorithme des points saillants.

accuracy: 21.37% +/- 40.99% (mikro: 21.37%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	0	0	0	0	2	0	0.00%
pred. cup	37	100	76	89	94	70	21.46%
pred. dromedary	0	0	0	0	0	0	0.00%
pred. glasses	0	0	0	0	0	0	0.00%
pred. hand	0	0	0	0	0	0	0.00%
pred. plane	0	0	0	0	0	0	0.00%
class recall	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%	

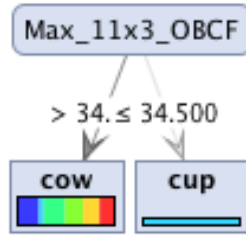


Figure C3 : Arbre de décision pour l’algorithme d’attention visuelle.

Tableau C11 : Résultats du modèle de classification basé sur un arbre de décision pour l’algorithme d’attention visuelle.

accuracy: 24.82% +/- 43.20% (mikro: 24.82%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	31	7	29	24	22	19	23.48%
pred. cup	0	3	0	0	1	0	75.00%
pred. dromedary	0	0	0	0	0	0	0.00%
pred. glasses	0	0	0	0	0	0	0.00%
pred. hand	0	0	0	0	0	0	0.00%
pred. plane	0	0	0	1	0	0	0.00%
class recall	100.00%	30.00%	0.00%	0.00%	0.00%	0.00%	

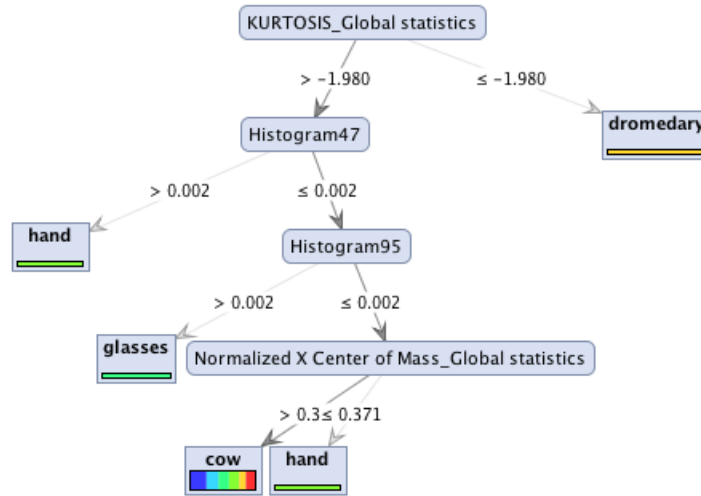


Figure C4 : Arbre de décision pour l’algorithme de saillance du maillage.

Tableau C11 : Résultats du modèle de classification basé sur un arbre de décision pour l’algorithme de saillance du maillage.

accuracy: 25.57% +/- 43.63% (mikro: 25.57%)

	true cow	true cup	true glasses	true hand	true dromedary	true plane	class precision
pred. cow	212	148	134	130	87	111	25.79%
pred. cup	0	0	0	0	0	0	0.00%
pred. glasses	1	0	0	2	0	0	0.00%
pred. hand	2	1	1	1	0	0	20.00%
pred. drome...	1	0	0	0	0	0	0.00%
pred. plane	1	1	0	0	0	0	0.00% <input type="text" value="0.00%"/>
class recall	97.70%	0.00%	0.00%	0.75%	0.00%	0.00%	

Forêt d'arbres, six objets virtuels

Tableau C12 : Résultats du modèle de classification basé sur le module *Random Forest* pour l'algorithme HKS.

accuracy: 34.09% +/- 47.40% (mikro: 34.09%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	3	2	1	2	2	2	25.00%
pred. cup	1	0	0	0	0	0	0.00%
pred. dromedary	4	0	7	3	1	1	43.75%
pred. glasses	2	0	0	0	0	1	0.00%
pred. hand	2	0	1	1	3	0	42.86%
pred. plane	0	0	2	1	0	2	40.00%
class recall	25.00%	0.00%	63.64%	0.00%	50.00%	33.33%	

Tableau C13 : Résultats du modèle de classification basé sur le module *Random Forest* pour l'algorithme des points saillants.

accuracy: 21.37% +/- 40.99% (mikro: 21.37%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	0	0	0	0	0	0	0.00%
pred. cup	30	77	62	69	66	60	21.15%
pred. dromedary	1	2	1	2	6	2	7.14%
pred. glasses	4	5	4	3	6	1	13.04%
pred. hand	2	14	9	12	17	5	28.81%
pred. plane	0	2	0	3	1	2	25.00%
class recall	0.00%	77.00%	1.32%	3.37%	17.71%	2.86%	

Tableau C14 : Résultats du modèle de classification basé sur le module *Random Forest* pour l'algorithme attention visuelle.

accuracy: 26.28% +/- 44.01% (mikro: 26.28%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	8	4	11	3	13	8	17.02%
pred. cup	0	0	0	1	0	0	0.00%
pred. dromedary	11	4	11	6	4	2	28.95%
pred. glasses	2	0	4	10	2	5	43.48%
pred. hand	6	2	2	2	3	0	20.00%
pred. plane	4	0	1	3	1	4	30.77%
class recall	25.81%	0.00%	37.93%	40.00%	13.04%	21.05%	

Tableau C14 : Résultats du modèle de classification basé sur le module *Random Forest* pour l'algorithme de saillance du maillage.

accuracy: 26.29% +/- 44.02% (mikro: 26.29%)

	true cow	true cup	true glasses	true hand	true dromedary	true plane	class precision
pred. cow	205	144	122	127	82	98	26.35%
pred. cup	2	0	2	1	0	0	0.00%
pred. glasses	2	3	5	1	1	3	33.33%
pred. hand	2	2	4	4	3	4	21.05%
pred. drome...	1	1	0	0	0	1	0.00%
pred. plane	5	0	2	0	1	5	38.46%
class recall	94.47%	0.00%	3.70%	3.01%	0.00%	4.50%	

Boosting d'arbres de décision, six objets virtuels

Tableau C15 : Résultats du modèle de classification basé sur le module *Gradient Boosted Tree* pour l'algorithme HKS.

accuracy: 43.18% +/- 49.53% (mikro: 43.18%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	4	1	0	5	2	0	33.33%
pred. cup	1	0	0	0	0	0	0.00%
pred. dromedary	3	1	8	0	1	1	57.14%
pred. glasses	3	0	2	1	0	1	14.29%
pred. hand	1	0	0	0	3	1	60.00%
pred. plane	0	0	1	1	0	3	60.00%
class recall	33.33%	0.00%	72.73%	14.29%	50.00%	50.00%	

Tableau C16 : Résultats du modèle de classification basé sur le module *Gradient Boosted Tree* pour l'algorithme des points saillants.

accuracy: 36.75% +/- 48.21% (mikro: 36.75%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	2	4	3	2	3	0	14.29%
pred. cup	11	46	16	12	19	15	38.66%
pred. dromedary	2	19	23	5	13	5	34.33%
pred. glasses	4	4	9	43	18	12	47.78%
pred. hand	10	18	17	17	31	11	29.81%
pred. plane	8	9	8	10	12	27	36.49%
class recall	5.41%	46.00%	30.26%	48.31%	32.29%	38.57%	

Tableau C17 : Résultats du modèle de classification basé sur le module *Gradient Boosted Tree* pour l'algorithme d'attention visuelle.

accuracy: 36.50% +/- 48.14% (mikro: 36.50%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	13	2	5	3	6	8	35.14%
pred. cup	1	2	1	0	2	0	33.33%
pred. dromed...	6	2	12	6	4	1	38.71%
pred. glasses	2	0	6	10	5	3	38.46%
pred. hand	4	4	1	5	6	0	30.00%
pred. plane	5	0	4	1	0	7	41.18%
class recall	41.94%	20.00%	41.38%	40.00%	26.09%	36.84%	

Tableau C18 : Résultats du modèle de classification basé sur le module *Gradient Boosted Tree* pour l'algorithme de saillance du maillage.

accuracy: 2.40% +/- 15.31% (mikro: 2.40%)

	true cow	true cup	true glasses	true hand	true dromedary	true plane	class precision
pred. cow	15	54	42	28	19	32	7.89%
pred. cup	59	0	40	29	15	26	0.00%
pred. glasses	79	35	0	48	25	18	0.00%
pred. hand	23	29	30	0	14	14	0.00%
pred. drome...	20	13	14	12	0	16	0.00%
pred. plane	21	19	9	16	14	5	5.95%
class recall	6.91%	0.00%	0.00%	0.00%	0.00%	4.50%	

Evolutionary SVM, six objets virtuels

Tableau C19 : Résultats du modèle de classification basé sur le module *Evolutionary SVM* pour l'algorithme HKS.

accuracy: 27.27% +/- 44.54% (mikro: 27.27%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	12	2	11	7	6	6	27.27%
pred. cup	0	0	0	0	0	0	0.00%
pred. dromedary	0	0	0	0	0	0	0.00%
pred. glasses	0	0	0	0	0	0	0.00%
pred. hand	0	0	0	0	0	0	0.00%
pred. plane	0	0	0	0	0	0	0.00%
class recall	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	

Tableau C20 : Résultats du modèle de classification basé sur le module *Evolutionary SVM* pour l'algorithme des points saillants.

accuracy: 7.91% +/- 26.98% (mikro: 7.91%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	37	100	76	89	96	70	7.91%
pred. cup	0	0	0	0	0	0	0.00%
pred. dromedary	0	0	0	0	0	0	0.00%
pred. glasses	0	0	0	0	0	0	0.00%
pred. hand	0	0	0	0	0	0	0.00%
pred. plane	0	0	0	0	0	0	0.00%
class recall	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	

Tableau C21 : Résultats du modèle de classification basé sur le module *Evolutionary SVM* pour l'algorithme d'attention visuelle.

accuracy: 30.66% +/- 46.11% (mikro: 30.66%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	31	10	23	20	23	19	24.60%
pred. cup	0	0	0	0	0	0	0.00%
pred. dromed...	0	0	6	0	0	0	100.00%
pred. glasses	0	0	0	5	0	0	100.00%
pred. hand	0	0	0	0	0	0	0.00%
pred. plane	0	0	0	0	0	0	0.00%
class recall	100.00%	0.00%	20.69%	20.00%	0.00%	0.00%	

Tableau C22 : Résultats du modèle de classification basé sur le module *Evolutionary SVM* pour l'algorithme de saillance du maillage.

accuracy: 25.09% +/- 43.35% (mikro: 25.09%)

	true cow	true cup	true glasses	true hand	true dromedary	true plane	class precision
pred. cow	54	43	40	34	23	20	25.23%
pred. cup	49	29	29	24	12	11	18.83%
pred. glasses	32	29	25	22	14	10	18.94%
pred. hand	28	23	23	30	15	8	23.62%
pred. drome...	23	13	9	12	16	7	20.00%
pred. plane	31	13	9	11	7	55	43.65%
class recall	24.88%	19.33%	18.52%	22.56%	18.39%	49.55%	

K plus proches voisins, six objets virtuels

Tableau C23 : Résultats du modèle de classification basé sur le module *KNN* pour l'algorithme HKS.

accuracy: 70.45% +/- 45.62% (mikro: 70.45%)

	true cow	true cup	true dromed...	true glasses	true hand	true plane	class precision
pred. cow	10	0	0	1	1	0	83.33%
pred. cup	1	2	0	0	0	0	66.67%
pred. drome...	0	0	9	0	0	1	90.00%
pred. glasses	0	0	0	3	1	1	60.00%
pred. hand	0	0	0	1	3	0	75.00%
pred. plane	1	0	2	2	1	4	40.00%
class recall	83.33%	100.00%	81.82%	42.86%	50.00%	66.67%	

Tableau C24 : Résultats du modèle de classification basé sur le module *KNN* pour l'algorithme des points saillants.

accuracy: 43.59% +/- 49.59% (mikro: 43.59%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	15	4	5	6	6	9	33.33%
pred. cup	2	41	14	9	16	9	45.05%
pred. drome...	4	17	27	6	11	7	37.50%
pred. glasses	6	15	8	48	8	9	51.06%
pred. hand	5	13	9	7	48	11	51.61%
pred. plane	5	10	13	13	7	25	34.25%
class recall	40.54%	41.00%	35.53%	53.93%	50.00%	35.71%	

Tableau C25 : Résultats du modèle de classification basé sur le module *KNN* pour l’algorithme d’attention visuelle.

accuracy: 56.20% +/- 49.61% (mikro: 56.20%)

	true cow	true cup	true dromedary	true glasses	true hand	true plane	class precision
pred. cow	21	0	3	3	2	6	60.00%
pred. cup	0	6	0	2	0	1	66.67%
pred. drome...	2	2	17	5	2	1	58.62%
pred. glasses	2	2	5	12	2	1	50.00%
pred. hand	4	0	3	2	13	2	54.17%
pred. plane	2	0	1	1	4	8	50.00%
class recall	67.74%	60.00%	58.62%	48.00%	56.52%	42.11%	

Tableau C26 : Résultats du modèle de classification basé sur le module *KNN* pour l’algorithme de saillance du maillage.

accuracy: 24.25% +/- 42.86% (mikro: 24.25%)

	true cow	true cup	true glasses	true hand	true dromedary	true plane	class precision
pred. cow	58	41	32	31	24	18	28.43%
pred. cup	48	25	31	22	13	11	16.67%
pred. glasses	29	30	26	27	15	15	18.31%
pred. hand	29	28	24	30	14	8	22.56%
pred. drome...	24	15	11	11	14	10	16.47%
pred. plane	29	11	11	12	7	49	41.18%
class recall	26.73%	16.67%	19.26%	22.56%	16.09%	44.14%	

Apprentissage profond, quatre objets virtuels

Tableau C27 : Résultats du modèle de classification basé sur l’apprentissage profond pour l’algorithme HKS.

accuracy: 25.93% +/- 43.82% (mikro: 25.93%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	5	1	5	4	33.33%
pred. cup	1	0	0	0	0.00%
pred. glasses	4	1	0	0	0.00%
pred. hand	2	0	2	2	33.33%
class recall	41.67%	0.00%	0.00%	33.33%	

Tableau C28 : Résultats du modèle de classification basé sur l'apprentissage profond pour l'algorithme d'attention visuelle.

accuracy: 35.42% +/- 47.83% (mikro: 35.42%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	37	8	29	32	34.91%
pred. cup	8	10	9	2	34.48%
pred. glasses	20	6	23	9	39.66%
pred. hand	19	6	7	15	31.91%
class recall	44.05%	33.33%	33.82%	25.86%	

Tableau C29 : Résultats du modèle de classification basé sur l'apprentissage profond pour l'algorithme d'attention visuelle optimisée.

accuracy: 81.65% +/- 38.71% (mikro: 81.65%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	36	0	2	4	85.71%
pred. cup	0	27	0	2	93.10%
pred. glasses	6	0	23	5	67.65%
pred. hand	0	0	1	3	75.00%
class recall	85.71%	100.00%	88.46%	21.43%	

Tableau C30 : Résultats du modèle de classification basé sur l'apprentissage profond pour l'algorithme des points saillants.

accuracy: 44.03% +/- 49.64% (mikro: 44.03%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	4	0	2	1	57.14%
pred. cup	46	188	78	101	45.52%
pred. glasses	23	29	49	33	36.57%
pred. hand	17	23	69	91	45.50%
class recall	4.44%	78.33%	24.75%	40.27%	

Tableau C31 : Résultats du modèle de classification basé sur l'apprentissage profond pour l'algorithme de saillance du maillage.

accuracy: 19.21% +/- 39.40% (mikro: 19.21%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	64	45	54	44	30.92%
pred. cup	60	21	35	42	13.29%
pred. glasses	51	43	18	28	12.86%
pred. hand	42	41	28	19	14.62%
class recall	29.49%	14.00%	13.33%	14.29%	

Naïve Bayes, quatre objets virtuels

Tableau C32 : Résultats du modèle de classification basé sur Naïve Bayes pour l'algorithme HKS.

accuracy: 33.33% +/- 47.14% (mikro: 33.33%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	3	2	1	0	50.00%
pred. cup	1	0	0	0	0.00%
pred. glasses	5	0	2	2	22.22%
pred. hand	3	0	4	4	36.36%
class recall	25.00%	0.00%	28.57%	66.67%	

Tableau C32 : Résultats du modèle de classification basé sur Naïve Bayes pour l'algorithme des points saillants.

accuracy: 35.81% +/- 47.94% (mikro: 35.81%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	3	2	1	2	37.50%
pred. cup	31	113	55	73	41.54%
pred. glasses	13	11	14	11	28.57%
pred. hand	43	114	128	140	32.94%
class recall	3.33%	47.08%	7.07%	61.95%	

Tableau C33 : Résultats du modèle de classification basé sur Naïve Bayes pour l'algorithme d'attention visuelle.

accuracy: 23.75% +/- 42.56% (mikro: 23.75%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	1	3	4	4	8.33%
pred. cup	2	0	2	0	0.00%
pred. glasses	9	1	4	2	25.00%
pred. hand	72	26	58	52	25.00%
class recall	1.19%	0.00%	5.88%	89.66%	

Tableau C34 : Résultats du modèle de classification basé sur Naïve Bayes pour l’algorithme d’attention visuelle optimisée.

accuracy: 78.90% +/- 40.80% (mikro: 78.90%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	39	0	4	2	86.67%
pred. cup	0	24	0	0	100.00%
pred. glasses	3	3	22	11	56.41%
pred. hand	0	0	0	1	100.00%
class recall	92.86%	88.89%	84.62%	7.14%	

Tableau C35 : Résultats du modèle de classification basé sur Naïve Bayes pour l’algorithme de saillance de maillage.

accuracy: 5.20% +/- 22.20% (mikro: 5.20%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	21	55	13	22	18.92%
pred. cup	72	0	23	101	0.00%
pred. glasses	0	0	2	0	100.00%
pred. hand	124	95	97	10	3.07%
class recall	9.68%	0.00%	1.48%	7.52%	

Arbres de décision, quatre objets virtuels

Tableau C36 : Résultats du modèle de classification basé sur un arbre de décision pour l’algorithme HKS.

accuracy: 14.81% +/- 35.52% (mikro: 14.81%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	3	2	4	1	30.00%
pred. cup	0	0	0	0	0.00%
pred. glasses	7	0	1	5	7.69%
pred. hand	2	0	2	0	0.00%
class recall	25.00%	0.00%	14.29%	0.00%	

Tableau C37 : Résultats du modèle de classification basé sur un arbre de décision pour l’algorithme des points saillants.

accuracy: 32.89% +/- 46.98% (mikro: 32.89%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	2	2	1	3	25.00%
pred. cup	83	233	193	212	32.32%
pred. glasses	3	2	3	1	33.33%
pred. hand	2	3	1	10	62.50%
class recall	2.22%	97.08%	1.52%	4.42%	

Tableau C38 : Résultats du modèle de classification basé sur un arbre de décision pour l’algorithme d’attention visuelle.

accuracy: 35.42% +/- 47.83% (mikro: 35.42%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	83	28	67	58	35.17%
pred. cup	1	2	1	0	50.00%
pred. glasses	0	0	0	0	0.00%
pred. hand	0	0	0	0	0.00%
class recall	98.81%	6.67%	0.00%	0.00%	

Tableau C39 : Résultats du modèle de classification basé sur un arbre de décision pour l’algorithme d’attention visuelle optimisée.

accuracy: 45.87% +/- 49.83% (mikro: 45.87%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	42	27	20	11	42.00%
pred. cup	0	0	0	0	0.00%
pred. glasses	0	0	6	1	85.71%
pred. hand	0	0	0	2	100.00%
class recall	100.00%	0.00%	23.08%	14.29%	

Tableau C40 : Résultats du modèle de classification basé sur un arbre de décision pour l’algorithme de saillance du maillage.

accuracy: 34.02% +/- 47.38% (mikro: 34.02%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	213	150	132	128	34.19%
pred. cup	0	0	0	0	0.00%
pred. glasses	1	0	1	3	20.00%
pred. hand	3	0	2	2	28.57%
class recall	98.16%	0.00%	0.74%	1.50%	

Forêt d’arbres, quatre objets virtuels

Tableau C41 : Résultats du modèle de classification basé sur le module *Random Forest* pour l’algorithme HKS.

accuracy: 37.04% +/- 48.29% (mikro: 37.04%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	7	1	4	6	38.89%
pred. cup	0	1	0	0	100.00%
pred. glasses	3	0	2	0	40.00%
pred. hand	2	0	1	0	0.00%
class recall	58.33%	50.00%	28.57%	0.00%	

Tableau C42 : Résultats du modèle de classification basé sur le module *Random Forest* pour l’algorithme des points saillants.

accuracy: 31.43% +/- 46.42% (mikro: 31.43%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	0	0	0	0	0.00%
pred. cup	83	231	192	220	31.82%
pred. glasses	3	0	2	2	28.57%
pred. hand	4	9	4	4	19.05%
class recall	0.00%	96.25%	1.01%	1.77%	

Tableau C43 : Résultats du modèle de classification basé sur le module *Random Forest* pour l'algorithme attention visuelle.

accuracy: 34.58% +/- 47.56% (mikro: 34.58%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	8	29	65	57	34.91%
pred. cup	0	0	1	0	0.00%
pred. glasses	3	1	2	1	28.57%
pred. hand	0	0	0	0	0.00%
class recall	96.43%	0.00%	2.94%	0.00%	

Tableau C44 : Résultats du modèle de classification basé sur le module *Random Forest* pour l'algorithme attention visuelle optimisée.

accuracy: 50.46% +/- 50.00% (mikro: 50.46%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	42	19	19	11	46.15%
pred. cup	0	7	1	2	70.00%
pred. glasses	0	1	6	1	75.00%
pred. hand	0	0	0	0	0.00%
class recall	100.00%	25.93%	23.08%	0.00%	

Tableau C45 : Résultats du modèle de classification basé sur le module *Random Forest* pour l'algorithme de saillance du maillage.

accuracy: 34.02% +/- 47.38% (mikro: 34.02%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	210	149	128	126	34.26%
pred. cup	3	0	2	2	0.00%
pred. glasses	2	0	3	2	42.86%
pred. hand	2	1	2	3	37.50%
class recall	96.77%	0.00%	2.22%	2.26%	

Boosting d'arbres de décision, quatre objets virtuels

Tableau C46 : Résultats du modèle de classification basé sur le module *Gradient Boosted Tree* pour l'algorithme HKS.

accuracy: 29.63% +/- 45.66% (mikro: 29.63%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	1	0	0	0	100.00%
pred. cup	0	0	0	1	0.00%
pred. glasses	11	2	7	5	28.00%
pred. hand	0	0	0	0	0.00%
class recall	8.33%	0.00%	100.00%	0.00%	

Tableau C47 : Résultats du modèle de classification basé sur le module *Gradient Boosted Tree* pour l'algorithme des points saillants.

accuracy: 45.62% +/- 49.81% (mikro: 45.62%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	8	6	9	11	23.53%
pred. cup	29	151	54	83	47.63%
pred. glasses	29	33	98	45	47.80%
pred. hand	24	50	37	87	43.94%
class recall	8.89%	62.92%	49.49%	38.50%	

Tableau C48 : Résultats du modèle de classification basé sur le module *Gradient Boosted Tree* pour l'algorithme d'attention visuelle.

accuracy: 30.42% +/- 46.01% (mikro: 30.42%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	31	11	21	13	40.79%
pred. cup	11	4	9	9	12.12%
pred. glasses	27	11	21	19	26.92%
pred. hand	15	4	17	17	32.08%
class recall	36.90%	13.33%	30.88%	29.31%	

Tableau C49 : Résultats du modèle de classification basé sur le module *Gradient Boosted Tree* pour l'algorithme d'attention visuelle optimisée.

accuracy: 88.99% +/- 31.30% (mikro: 88.99%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	42	0	0	4	91.30%
pred. cup	0	27	0	2	93.10%
pred. glasses	0	0	26	6	81.25%
pred. hand	0	0	0	2	100.00%
class recall	100.00%	100.00%	100.00%	14.29%	

Tableau C50 : Résultats du modèle de classification basé sur le module *Gradient Boosted Tree* pour l'algorithme de saillance du maillage.

accuracy: 2.68% +/- 16.14% (mikro: 2.68%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	17	68	49	36	10.00%
pred. cup	70	0	43	43	0.00%
pred. glasses	71	36	0	54	0.00%
pred. hand	59	46	43	0	0.00%
class recall	7.83%	0.00%	0.00%	0.00%	

Evolutionary SVM, quatre objets virtuels

Tableau C51 : Résultats du modèle de classification basé sur le module *Evolutionary SVM* pour l'algorithme HKS.

accuracy: 22.22% +/- 41.57% (mikro: 22.22%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	4	2	5	2	30.77%
pred. cup	0	0	0	0	0.00%
pred. glasses	3	0	0	2	0.00%
pred. hand	5	0	2	2	22.22%
class recall	33.33%	0.00%	0.00%	33.33%	

Tableau C52 : Résultats du modèle de classification basé sur le module *Evolutionary SVM* pour l'algorithme des points saillants.

accuracy: 29.05% +/- 45.40% (mikro: 29.05%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	15	23	31	23	16.30%
pred. cup	25	94	61	70	37.60%
pred. glasses	24	51	42	65	23.08%
pred. hand	26	72	64	68	29.57%
class recall	16.67%	39.17%	21.21%	30.09%	

Tableau C53 : Résultats du modèle de classification basé sur le module *Evolutionary SVM* pour l'algorithme d'attention visuelle.

accuracy: 39.58% +/- 48.90% (mikro: 39.58%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	46	10	28	26	41.82%
pred. cup	5	1	10	3	5.26%
pred. glasses	21	10	24	5	40.00%
pred. hand	12	9	6	24	47.06%
class recall	54.76%	3.33%	35.29%	41.38%	

Tableau C54 : Résultats du modèle de classification basé sur le module *Evolutionary SVM* pour l'algorithme d'attention visuelle optimisée.

accuracy: 92.66% +/- 26.08% (mikro: 92.66%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	42	0	0	4	91.30%
pred. cup	0	27	0	2	93.10%
pred. glasses	0	0	26	2	92.86%
pred. hand	0	0	0	6	100.00%
class recall	100.00%	100.00%	100.00%	42.86%	

Tableau C55 : Résultats du modèle de classification basé sur le module *Evolutionary SVM* pour l'algorithme de saillance du maillage.

accuracy: 27.72% +/- 44.76% (mikro: 27.72%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	77	53	46	40	35.65%
pred. cup	64	37	34	31	22.29%
pred. glasses	37	30	31	31	24.03%
pred. hand	39	30	24	31	25.00%
class recall	35.48%	24.67%	22.96%	23.31%	

K plus proches voisins, quatre objets virtuels

Tableau C56 : Résultats du modèle de classification basé sur le module *KNN* pour l'algorithme HKS.

accuracy: 22.22% +/- 41.57% (mikro: 22.22%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	3	1	5	2	27.27%
pred. cup	1	1	0	0	50.00%
pred. glasses	3	0	0	2	0.00%
pred. hand	5	0	2	2	22.22%
class recall	25.00%	50.00%	0.00%	33.33%	

Tableau C57 : Résultats du modèle de classification basé sur le module *KNN* pour l'algorithme des points saillants.

accuracy: 29.97% +/- 45.81% (mikro: 29.97%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	7	25	27	20	8.86%
pred. cup	26	94	53	73	38.21%
pred. glasses	36	49	53	61	26.63%
pred. hand	21	72	65	72	31.30%
class recall	7.78%	39.17%	26.77%	31.86%	

Tableau C58 : Résultats du modèle de classification basé sur le module *KNN* pour l'algorithme d'attention visuelle.

accuracy: 37.08% +/- 48.30% (mikro: 37.08%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	40	9	21	18	45.45%
pred. cup	5	2	12	9	7.14%
pred. glasses	5	9	25	9	36.76%
pred. hand	14	10	10	22	39.29%
class recall	47.62%	6.67%	36.76%	37.93%	

Tableau C59 : Résultats du modèle de classification basé sur le module *KNN* pour l'algorithme d'attention visuelle optimisée.

accuracy: 95.41% +/- 20.92% (mikro: 95.41%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	42	0	0	3	93.33%
pred. cup	0	27	0	0	100.00%
pred. glasses	0	0	26	2	92.86%
pred. hand	0	0	0	9	100.00%
class recall	100.00%	100.00%	100.00%	64.29%	

Tableau C60 : Résultats du modèle de classification basé sur le module *KNN* pour l'algorithme de saillance du maillage.

accuracy: 27.72% +/- 44.76% (mikro: 27.72%)

	true cow	true cup	true glasses	true hand	class precision
pred. cow	80	51	42	37	38.10%
pred. cup	66	33	34	30	20.25%
pred. glasses	35	33	31	34	23.31%
pred. hand	36	33	28	32	24.81%
class recall	36.87%	22.00%	22.96%	24.06%	

Objets réels

Tableau C61 : Résultats du modèle de classification basé sur le module *KNN* pour les objets réels.

accuracy: 56.25%

	true hand	true cup	true cow	true glass	class precision
pred. hand	2	0	0	2	50.00%
pred. cup	0	6	2	1	66.67%
pred. cow	1	1	1	0	33.33%
pred. glass	0	0	0	0	0.00%
class recall	66.67%	85.71%	33.33%	0.00%	

Tableau C62 : Résultats du modèle de classification basé sur le module Deep Learning pour les objets réels.

accuracy: 56.25%

	true hand	true cup	true cow	true glass	class precision
pred. hand	2	1	0	1	50.00%
pred. cup	0	5	1	2	62.50%
pred. cow	1	0	2	0	66.67%
pred. glass	0	1	0	0	0.00%
class recall	66.67%	71.43%	66.67%	0.00%	

Tableau C63 : Résultats du modèle de classification basé sur le module *Naive Bayes* pour les objets réels.

accuracy: 68.75%

	true hand	true cup	true cow	true glass	class precision
pred. hand	2	1	0	1	50.00%
pred. cup	0	6	0	2	75.00%
pred. cow	1	0	3	0	75.00%
pred. glass	0	0	0	0	0.00%
class recall	66.67%	85.71%	100.00%	0.00%	

Tableau C64 : Résultats du modèle de classification basé sur le module *Decision Tree* pour les objets réels.

accuracy: 62.50%

	true hand	true cup	true cow	true glass	class precision
pred. hand	0	0	0	0	0.00%
pred. cup	3	7	3	0	53.85%
pred. cow	0	0	0	0	0.00%
pred. glass	0	0	0	3	100.00%
class recall	0.00%	100.00%	0.00%	100.00%	

Tableau C65 : Résultats du modèle de classification basé sur le module *Random Forest* pour les objets réels.

accuracy: 50.00%

	true hand	true cup	true cow	true glass	class precision
pred. hand	0	0	0	0	0.00%
pred. cup	3	7	3	2	46.67%
pred. cow	0	0	0	0	0.00%
pred. glass	0	0	0	1	100.00%
class recall	0.00%	100.00%	0.00%	33.33%	

Tableau C66 : Résultats du modèle de classification basé sur le module *Gradient Boosted Tree* pour les objets réels.

accuracy: 56.25%

	true hand	true cup	true cow	true glass	class precision
pred. hand	1	1	0	1	33.33%
pred. cup	1	6	1	2	60.00%
pred. cow	0	0	2	0	100.00%
pred. glass	1	0	0	0	0.00%
class recall	33.33%	85.71%	66.67%	0.00%	

Tableau C67 : Résultats du modèle de classification basé sur le module *SVM* pour les objets réels.

accuracy: 56.25%

	true hand	true cup	true cow	true glass	class precision
pred. hand	1	1	0	1	33.33%
pred. cup	0	5	0	1	83.33%
pred. cow	1	0	3	1	60.00%
pred. glass	1	1	0	0	0.00%
class recall	33.33%	71.43%	100.00%	0.00%	