RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE

ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université 20 août 1955 Skikda
Faculté des sciences
Département d'informatique
Spécialité intelligence artificielle

جامعة 20 أوت 1955 سكيكدة
كلية العلوم
قسم الإعلام الآلي
تخصص ذكاء إصطناعي

Mémoire de fin d'études

En vue de l'obtention du diplôme Master2

THÈME:

Système de reconnaissance des gestes de la main pour le contrôle d'actions informatiques

Présenté par :
BOUSMAT Aya
ZAIR Mohamed Walid

Soutenu le 23/06/2026, devant le jury suivant :

N°	Nom et prénom	Établissement d'affiliation	Qualité
1	Dr.BOULNEMOUR Imene	Université 20 Août 1955-Skikda	Encadrante
2	Dr.LAROUM Toufik	Université 20 Août 1955-Skikda	Président
3	Dr.RAMDANE Chafika	Université 20 Août 1955-Skikda	Examinatrice

Année universitaire : 2025-2026

Remerciement

Avant tout, nous remercions Allah, le Tout-Puissant, qui nous a accordé la force, la patience et la persévérance nécessaires pour mener à bien ce travail.

Nous adressons nos plus sincères remerciements à notre encadreure BOULNEMOUR Imene, pour son accompagnement précieux, sa disponibilité constante, ses conseils avisés et son soutien tout au long de la réalisation de ce mémoire. Son expertise et ses orientations ont été d'une aide inestimable dans l'aboutissement de ce projet.

Nous exprimons également notre profonde gratitude aux membres du jury qui ont accepté d'évaluer ce travail et de lui consacrer leur temps. Leurs remarques et suggestions contribueront sans aucun doute à l'amélioration de nos connaissances et de nos compétences.

Nos remerciements s'étendent à l'ensemble des enseignants du département, qui nous ont transmis leur savoir, leurs expériences et les valeurs académiques qui ont marqué notre parcours universitaire.

Nous adressons également nos sincères remerciements à nos familles pour leur soutien moral, leur patience et leurs encouragements tout au long de notre parcours universitaire.

Enfin, nous adressons nos remerciements à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce mémoire et nous ont apporté leur aide, leurs conseils ou leur soutien.

À toutes et à tous, nous exprimons notre profonde reconnaissance et notre gratitude.

Dédicace

Nous dédions ce modeste travail,

Avant tout, à nos très chères mères, sources inépuisables d'amour, de tendresse et de dévouement. À celles qui ont veillé sur nous avec patience, qui ont partagé nos joies comme nos difficultés, et qui ne nous ont jamais oubliés dans leurs prières.

Aucun mot ne saurait exprimer toute notre reconnaissance envers elles, ni traduire l'immensité de leurs sacrifices et de leur soutien tout au long de notre parcours. Que Dieu les protège, leur accorde une longue vie pleine de santé, de bonheur et de sérénité.

À nos pères, pour leurs efforts, leurs conseils, leur soutien et les valeurs qu'ils nous ont transmises. Nous leur témoignons notre profond respect et notre sincère gratitude.

À nos frères et sœurs pour leur affection, leurs encouragements et leur présence à nos côtés.

À nos neveux et nièces qui illuminent nos vies par leur innocence et leur joie.

À nos amis les plus proches pour leur amitié sincère, leur soutien constant et tous les moments inoubliables partagés ensemble.

À nous-mêmes,

Parce que personne ne connaît mieux que nous les obstacles que nous avons traversés, les nuits de travail, les moments de fatigue, de doute et d'incertitude.

Nous nous dédions ce travail en reconnaissance de notre courage, de notre persévérance et de notre capacité à poursuivre notre chemin malgré les difficultés. Que cette réalisation nous rappelle toujours que les efforts sincères finissent par porter leurs fruits.

À toutes les personnes qui nous ont accompagnés, soutenus et encouragés, de près ou de loin, tout au long de notre parcours universitaire.

Nous dédions ce travail avec respect, gratitude et affection.

Résumé

L'interaction gestuelle constitue une alternative prometteuse aux périphériques traditionnels tels que le clavier et la souris. Dans ce contexte, ce mémoire présente la conception et la réalisation d'un système intelligent de reconnaissance des gestes de la main destiné au contrôle d'actions informatiques en temps réel.

Le système proposé repose sur l'utilisation de la vision artificielle et de l'intelligence artificielle. Les points clés de la main sont extraits à l'aide de MediaPipe, puis exploités par deux modèles complémentaires : un classificateur SVM (Support Vector Machine) pour la reconnaissance des gestes statiques et un réseau de neurones BiLSTM (Bidirectional Long Short-Term Memory) pour la reconnaissance des gestes dynamiques. Une interface graphique a également été développée afin de permettre une interaction intuitive avec l'utilisateur.

Les expérimentations réalisées sur un jeu de données personnalisé de plus de 2200 échantillons ont montré des performances élevées, avec une précision de 97,3 % pour le modèle SVM et de 99,4 % pour le modèle BiLSTM. Les résultats obtenus démontrent la robustesse et l'efficacité de l'approche proposée pour le contrôle d'actions informatiques par gestes de la main.

Mots-clés : Reconnaissance gestuelle, Vision par ordinateur, Intelligence artificielle, MediaPipe, SVM, BiLSTM, Interaction homme-machine, Contrôle gestuel.

Abstract

Gesture-based interaction represents a promising alternative to traditional input devices such as the keyboard and mouse. In this context, this thesis presents the design and implementation of an intelligent hand gesture recognition system for real-time computer control.

The proposed system relies on computer vision and artificial intelligence techniques. Hand landmarks are extracted using MediaPipe and processed by two complementary models: an SVM classifier for static gesture recognition and a BiLSTM neural network for dynamic gesture recognition. A graphical user interface was also developed to provide intuitive interaction with the user.

Experiments conducted on a custom dataset containing more than 2200 samples demonstrated high performance, achieving an accuracy of 97.3% for the SVM model and 99.4% for the BiLSTM model. The obtained results confirm the robustness and effectiveness of the proposed approach for controlling computer actions through hand gestures.

Keywords: Gesture Recognition, Computer Vision, Artificial Intelligence, MediaPipe, SVM, BiLSTM, Human-Computer Interaction, Gesture-Based Control.

الملخص

يمثل التفاعل القائم على الإيماءات بديلاً واعدًا لوسائل الإدخال التقليدية مثل لوحة المفاتيح والفأرة. في هذا السياق تقدم هذه المذكرة تصميم وإنجاز نظام ذكي للتعرف على إيماءات اليد بهدف التحكم في الأوامر الحاسوبية في الـ زمن الحقيقي.

يعتمد النظام المقترح على تقنيات الرؤية الحاسوبية والذكاء الاصطناعي، حيث يتم استخراج النقاط المرجعية لليد باستعمال MediaPipe ، ثم معالجتها بواسطة نموذجين متكاملين : مصنف SVM للتعرف على الإيماءات السّاكنة، وشبكة عصبية من نوع BiLSTM للتعرف على الإيماءات الديناميكيّة. كما تمّ تطوير واجهة رسوميّة تفا عليّة لتسهيل استخدام النّظام.

أظهرت التّجارب المنجزة على مجموعة بيانات خاصة تضمّ 2200 عيّنة نتائج متميزة، حيث بلغت الدّقة 97.3% بالنسبة لنموذج SVM و 99.4% بالنسبة لنموذج BiLSTM .

وتؤكد هذه النّتائج فعاليّة ومتانة المنهجيّة المقترحة في التّحكم بالأوامر الحاسوبيّة اعتمادا على إيماءات اليد.

الكلمات المفتاحية : التّعرف على الإيماءات، الرؤية الحاسوبيّة، الذكاء الإصطناعي، MediaPipe ، SVM ، BiLSTM ، التّفاعل بين الإنسان والحاسوب، التّحكم بالإيماءات.

Table des matières

Introduction générale 1
1. Contexte du mémoire 1
2. Problématique 2
3. Objectif du mémoire 3
4. Organisation du mémoire 4
Chapitre1: État de l'art sur la reconnaissance gestuelle 6
1.1. Introduction 6
1.2. Définition des gestes(statiques/ dynamiques) 6
1.2.1. Gestes statiques 6
1.2.2. Gestes dynamiques 6
1.3. Domaines d'application 7
1.3.1. Interface Homme-Machine (IHM) 7
1.3.2. Réalité Virtuelle et Augmentée (RV / RA) 7
1.3.3. Jeux vidéo 8
1.3.4. Robotique 8
1.3.5. Santé et Médecine 9
1.3.6. Présentation et Éducation 9
1.4. Conclusion 10
Chapitre2: Techniques et méthodes de reconnaissance gestuelle 11
2.1. Introduction 11
2.2. Pipeline d'un système de reconnaissance gestuelle 11
2.2.1. Acquisition de données (Sensing) 12
2.2.2. Prétraitement et Segmentation (Preprocessing) 12
2.2.3. Extraction des caractéristiques (Feature Extraction) 12
2.2.4. Classification et Reconnaissance 13
2.2.5. Post-traitement et Exécution de la commande 13
2.3. Comparaison entre les méthodes de classification 14
2.3.1. Machines à Vecteurs de Support (SVM) 14
2.3.2. Bidirectional Long Short-Term Memory (BiLSTM) 14
2.3.3. Random Forest (RF) 15
2.3.4. k-plus proches voisins (k-NN) 15
2.3.5. Réseaux de Neurones Convolutionnels (CNN) 15
2.3.6. Gated Recurrent Unit (GRU) 16
2.3.7. Synthèse comparative 17
2.4. Conclusion 17
Chapitre3: Interaction gestuelle pour la manipulation des ordinateurs 19
3.1. Introduction 19
3.2. Interfaces homme-machine basées sur les gestes 19
3.3. Gestes utilisées pour la manipulation 21
3.4. Ergonomie et expérience utilisateur 22
3.4.1. Le confort physique et la fatigue (Gorilla Arm Syndrome) 22

3.4.2. L'affordance et la métaphore naturelle 23
3.4.3. La boucle de rétroaction (Feedback) 23
3.4.4. Loi de Fitts appliquée au pointage gestuel 23
3.5. Contraintes et défis 24
3.5.1. Contraintes 24
3.5.2. Défies 24
3.6. Conclusion 25
Chapitre4: Conception et réalisation du système proposé 26
4.1. Introduction 26
4.2. Analyse des besoins 26
4.2.1. Définir l'objectif 26
4.2.2. Spécification des besoins fonctionnels 27
4.2.3. Spécification des besoins non-fonctionnels 27
4.2.4. Synthèse des besoins techniques 27
4.3. Architecture du système 28
4.3.1. Acquisition des données 28
4.3.2. Prétraitement et segmentation du signal 28
4.3.3. Extraction des caractéristiques 29
4.3.4. Classification et reconnaissance 30
4.3.5. Application & commande 30
4.4. Description et gestes reconnues 31
4.4.1. Gestes statiques 31
4.4.1.1. Déplacement du curseur (Suivi de l'index) 31
4.4.1.2. Clic Simple (Single Click = Main ouverte) 31
4.4.1.3. Clic Double (Double Click = Index et majeur levés) 31
4.4.1.4. Clic Droit (Right Click = Index et auriculaire levés) 31
4.4.1.5. Ouverture du Clavier Virtuel (Index, majeur et annulaire levés) . 31
4.4.1.6. Lecture/Pause Vidéo (Poing Fermé) 32
4.4.2. Gestes Dynamiques 32
4.4.2.1. Défilement Vers le Haut (Scroll Up) 32
4.4.2.2. Défilement Vers le Bas (Scroll Down) 32
4.4.2.3. Glissement Vers la Droite (Slide Right) 33
4.4.2.4. Glissement Vers la Gauche (Slide Left) 33
4.4.2.5. Zoom avant (Zoom In) 33
4.4.2.6. Zoom arrière (Zoom Out) 33
4.5. Implémentation du système 34
4.5.1. Fonctionnement global du système 34
4.5.1.1.Démarrage et initialisation 35
4.5.1.2. Initialisation de la caméra HD(High Definition) 35
4.5.1.3. Détection et visualisation de la main 35
4.5.1.4. Extraction des caractéristiques 35
4.5.1.5. Classification des gestes 35
4.5.1.6. Validation et filtrage temporel 35
4.5.1.7. Mappage et exécution des actions 36

4.5.1.8. Fin du traitement 36
4.5.2. Branche des gestes statiques 37
4.5.2.1. Collecte de données 37
4.5.2.2. Entraînement du modèle de classification statique 39
4.5.2.3. Validation du modèle sur des données réelles en temps réel 42
4.5.3. Branche des gestes dynamiques 44
4.5.3.1. Collecte de données 44
4.5.3.2. Entraînement du modèle de la classification dynamique 46
4.5.3.3. Validation du modèle sur des données réelles en temps réel 50
4.6. Conclusion 51
Chapitre 5: Expérimentation et résultats 53
5.1. Introduction 53
5.2. Protocole expérimental 53
5.3. Jeux de données utilisées 54
5.3.1. Jeu de données des gestes statiques 54
5.3.2. Jeu de données des gestes dynamiques 55
5.3.3. Organisation du jeu de données 55
5.3.4. Taille et diversité des données 55
5.4. Métriques d'évaluation 56
5.4.1. Taux de reconnaissance (Accuracy) 56
5.4.2. Précision (Precision) 56
5.4.3. Rappel (Recall) 57
5.4.4. F1-Score 57
5.5. Analyse des résultats 57
5.5.1. Résultats du modèle SVM 57
5.5.1.1. Performance globale 57
5.5.1.2. Matrice de confusion 58
5.5.1.3. Analyse détaillée par classe 59
5.5.2. Résultats du modèle BiLSTM 60
5.5.2.1. Performance globale 60
5.5.2.2. Courbes d'apprentissage (Accuracy et Loss) 61
5.5.2.3. Matrice de confusion 62
5.5.2.4. Analyse détaillée par classe 63
5.5.3. Comparaison entre les deux modèles 64
5.6. Comparaison avec d'autres méthodes 64
5.7. Discussion 66
5.7.1. Points forts du système 66
5.7.2. Limites du système 67
5.7.3. Perspectives d'amélioration 67
5.8. Interface et démonstration du système 68
5.8.1. Exemples d'utilisation du système 69
5.9. Conclusion 71
Conclusion générale 73
Références 75

Liste des figures

Figure1.1: figure montrant la différence entre les gestes Statique et dynamique 7
Figure1.2: figure montrant l' Interaction gestuelle en Réalité Virtuelle et Augmentée. 8
Figure1.3: Jouer à des jeux à l'aide des gestes de la main 8
Figure1.4: Applications gestuelles en robotique industrielle. 9
Figure1.5: Applications gestuelles en santé et médecine. 9
Figure1.6 : Applications gestuelles en présentation et éducation. 10

Figure2.1 :Architecture modulaire du pipeline du reconnaissance gestuelle 13

Figure3.1: Taxonomie multidimensionnelle des gestes en Interaction Homme-Machine .. 20
Figure3.2: Les étapes clés d'un système de reconnaissance des gestes 21
Figure3.3: Délimitation des zones d'interaction ergonomiques et prévention du syndrome du « bras de gorille » 22

Figure4.1: Figure illustrant une architecture modulaire du pipeline de reconnaissance gestuelle 28
Figure4.2: Figure présente l'étape de prétraitement et segmentation du signal 29
Figure4.3: Illustration de l'étape d'extraction des caractéristiques 29
Figure4.4: Illustration de l'étape de classification et reconnaissance 30
Figure4.5: Illustration de l'étape d'application & commande 30
Figure4.6: Figure représentant des gestes statiques pour le contrôle de l'ordinateur 32
Figure4.7: représentant les gestes dynamiques pour le contrôle de l'ordinateur. 33
Figure4.8: Schéma global du système proposé pour la reconnaissance de geste de la main. 34
Figure4.9: Schéma de collecte des données statiques. 37
Figure 4.10: Schéma d' entraînement du modèle statique. 39
Figure 4.11: Figure représentant la classification par SVM 41
Figure 4.12: Figure illustrant l'évaluation du modèle statique en temps réel. 42
Figure 4.13 : Figure illustrant le processus de collecte des données dynamiques. 44
Figure 4.14: Figure illustrant le processus de collecte des données dynamiques. 46
Figure 4.15: Figure illustrant le principe de fonctionnement du modèle LSTM 49
Figure 4.16: Figure illustrant l'évaluation du modèle dynamique en temps réel. 50

Figure 5.1: Matrice de confusion du modèle SVM 58
Figure 5.2: Courbes d'apprentissage du modèle BiLSTM (Accuracy et Loss) 61
Figure 5.3: Matrice de confusion du modèle BiLSTM 62
Figure 5.4: Interface principale du système de reconnaissance des gestes. 68
Figure 5.5 : Module de visualisation des gestes reconnus par le système. 69
Figure 5.6: Reconnaissance du geste statique « Pause » et exécution de la commande correspondante. 70
Figure 5.7: État initial avant l'exécution du geste de zoom. 70
Figure 5.8: Résultat obtenu après l'exécution du geste dynamique de zoom. 71

Liste des tableaux

Tableau4.1: Tableau récapitulatif les besoins prioritaires pour la réalisation du système .27

Tableau 5.1: Tableau des performances du modèle SVM (Accuracy, Précision, Rappel et F1-Score) 57
Tableau 5.2: Résultats détaillées des métriques d'évaluation par classe 59
Tableau 5.3: Tableau des performances du modèle BiLSTM (Accuracy, Précision, Rappel et F1-Score) 60
Tableau 5.5: Comparaison des performances des modèles SVM et BiLSTM 64
Tableau 5.6: Comparaison de notre approche avec les travaux existants 65

Liste des abréviations

SVM : Support Vector Machine

LSTM : Long Short-Term Memory

IHM: Interface Homme Machine

NUI: Natural User Interface

RV : Réalité Virtuelle

RA : Réalité Augmentée

K-NN: K-Nearest Neighbors

CNN: Convolutional Neural Network

GRU: Gated Recurrent Unit

FPS: Frame Per Second

ROI: Region Of Interest

RNN: Recurrent Neural Network

RF: Randoom Forest

GPU: Graphics Processing Unit

UX: User Experience

UI: User Interface

RGB: Red, Green, Blue

API: Application Programming Interface

Caméra HD: High Definition

BiLSTM: Bidirectional Long Short-Term

Noyau RBF: Radial Basis Functin kernel

ReLU: Rectified Linear Unit

Introduction générale

Introduction générale

1. Contexte du mémoire

Au début de l'informatique, nous utilisions des interfaces purement textuelles, alors que ces dernières années nous évoluons dans un environnement numérique omniprésent et multidimensionnel. Cela est dû à l'intégration croissante des outils informatiques dans tous les secteurs de la vie quotidienne (santé, industrie, éducation). Les technologies de l'Information et de la Communication ont apporté des avantages dont les utilisateurs ne peuvent plus se passer, notamment la rapidité de traitement et l'accès instantané aux données. Mais cela impose à l'utilisateur une dépendance vis-à-vis de périphériques physiques traditionnels (clavier, souris) qui créent une barrière matérielle et souvent rigide.

En outre, ces systèmes d'interaction classiques se montrent parfois limités, voire inadaptés, pour répondre aux besoins de mobilité, d'immersion ou d'ergonomie exigés par les nouvelles applications de réalité virtuelle ou les environnements de travail sans contact.

Pour pallier ce problème, il faut avoir la possibilité d'interagir avec la machine de manière plus naturelle et intuitive. Les mécanismes d'interface doivent disposer de données biométriques et comportementales sur l'utilisateur, telles que ses mouvements, la position de ses mains ou ses intentions gestuelles. De ce fait, le système de reconnaissance gestuelle, capable de traduire le mouvement humain en commandes numériques, devient central dans tout système basé sur les Interfaces Homme-Machine (IHM) de nouvelle génération.

Cependant, l'utilisation d'une simple reconnaissance de formes statiques n'est pas suffisante pour les systèmes de manipulation complexe d'un ordinateur, en particulier à cause de la difficulté d'interpréter la dynamique du mouvement dans des conditions variables (éclairage, occlusion, vitesse). Par conséquent, l'utilisation d'un système de reconnaissance robuste, tant par sa précision que par sa capacité de traitement en temps réel, est incontournable afin de pouvoir interpréter et exploiter correctement les gestes dans le but de satisfaire les attentes des utilisateurs en termes de fluidité et de contrôle.

Dans ce mémoire, nous nous focalisons sur une approche par vision artificielle visant à concevoir une architecture de reconnaissance gestuelle, optimisée à l'aide de techniques d'intelligence artificielle. Le système repose sur deux modèles complémentaires : un SVM pour les gestes statiques et un BiLSTM pour les gestes dynamiques, afin de permettre l'exécution de tâches de navigation et de sélection de manière précise et ergonomique.

2. Problématique

Depuis l'apparition des interfaces naturelles (NUI), les mécanismes de reconnaissance gestuelle dédiés à la manipulation des ordinateurs n'arrivent pas encore à offrir une alternative totalement fiable et fluide aux périphériques conventionnels (clavier, souris). Bien que de nombreux systèmes aient opté pour l'utilisation de capteurs de vision et de caméras lors de leurs mises en œuvre afin de remédier à la rigidité des interfaces classiques, ils se heurtent à des obstacles majeurs en situation réelle. Malgré ces avancées, une mauvaise distinction entre un mouvement de déplacement (navigation) et une action de validation (sélection), ou encore une sensibilité excessive aux bruits environnementaux (variations d'éclairage, arrière-plans complexes), engendre l'exécution de commandes erronées. Or, les performances de ces mécanismes de contrôle sont étroitement dépendantes de la qualité de l'extraction des caractéristiques et de la robustesse des modèles de classification. En effet, un système de reconnaissance conçu par vision artificielle, capable d'analyser précisément la morphologie de la main, permettrait aux interfaces homme-machine de disposer d'un moyen efficace pour interpréter correctement les intentions de l'utilisateur, dans la perspective d'aboutir à une manipulation précise, ergonomique et sans contact.

En outre, le geste humain est généralement effectué de manière dynamique et varie significativement d'un individu à l'autre. De par la nature temporelle de ce processus, l'intention de l'utilisateur n'est jamais figée à un instant précis, ce qui rend les systèmes de manipulation gestuelle classiques peu performants. Ces derniers peinent souvent à maintenir une interprétation stable de la trajectoire, provoquant des ambiguïtés au point de ne pas pouvoir recommander l'action adéquate (clic, glissement, défilement) au moment voulu. Ce manque de précision peut mener au phénomène du « Midas Touch », où chaque

mouvement involontaire est interprété comme une commande, nuisant ainsi gravement à l'expérience utilisateur.

De ce fait, l'optimisation de l'architecture de reconnaissance, à travers l'intégration de modèles d'apprentissage profond et l'utilisation de points clés (Landmarks), s'avère nécessaire afin de résoudre ce problème. Cette approche permet d'obtenir une meilleure séparation sémantique entre les différentes classes des gestes à partir des flux vidéo capturés, en particulier en prédisant avec exactitude la transition entre la phase de navigation et celle de sélection. Ceci permet d'aboutir à un système suffisamment robuste et prêt à être exploité par un système d'exploitation, offrant ainsi à l'utilisateur une alternative de contrôle intuitive et performante, en adéquation avec ses attentes en matière d'interaction moderne.

3. Objectif du mémoire

L'objectif principal de ce mémoire est de concevoir et de réaliser un système de reconnaissance gestuelle robuste et performant, basé sur l'Intelligence Artificielle, permettant la manipulation fluide d'un ordinateur en temps réel. Pour atteindre cet objectif global, plusieurs sous-objectifs spécifiques sont définis :

Étude et analyse de la taxonomie des gestes : Définir un ensemble des gestes intuitifs et ergonomiques permettant de couvrir les fonctionnalités essentielles de navigation (déplacement du curseur) et de sélection (clics et manipulations d'objets) afin de minimiser la fatigue de l'utilisateur.

Développement d'un pipeline de traitement d'image : Mettre en place une architecture logicielle capable d'acquérir le flux vidéo, de prétraiter les données et d'extraire les points clés de la main (Landmarks) de manière efficace, même dans des environnements visuels variables.

Conception d'un modèle de classification performant : Exploiter les algorithmes d'apprentissage profond pour entraîner un modèle capable de distinguer avec précision les différentes classes des gestes, en assurant une séparation nette entre les intentions de navigation et les actions de sélection.

Implémentation d'une interface de contrôle en temps réel : Développer un module de communication entre le système de reconnaissance et le système d'exploitation, permettant de traduire les gestes détectés en commandes exécutables (mouvements de souris, événements de clic, défilement) sans latence perceptible.

Évaluation et validation des performances : Tester le système proposé à travers une série d'expérimentations afin de mesurer son taux de précision, sa robustesse face aux contraintes environnementales et son acceptabilité ergonomique par les utilisateurs.

À travers ces objectifs, ce travail vise à démontrer qu'il est possible de substituer, ou du moins de compléter, les périphériques d'entrée traditionnels par une interface naturelle plus immersive et adaptée aux nouveaux besoins de l'interaction homme-machine moderne.

4. Organisation du mémoire

Afin de répondre à la problématique posée et d'atteindre les objectifs fixés, ce mémoire est structuré en cinq chapitres complémentaires.

Le premier chapitre présente un état de l'art sur la reconnaissance gestuelle. Nous y introduisons les concepts fondamentaux liés aux interfaces homme-machine, aux gestes de la main ainsi qu'aux principales applications de la reconnaissance gestuelle. Une analyse des travaux existants est également réalisée afin de mettre en évidence les approches les plus utilisées et les défis encore ouverts dans ce domaine.

Le deuxième chapitre est consacré aux techniques et méthodes de reconnaissance gestuelle. Nous y détaillons les différentes approches basées sur la vision artificielle, les méthodes d'extraction de caractéristiques ainsi que les algorithmes d'apprentissage automatique et d'apprentissage profond utilisés pour la classification des gestes.

Le troisième chapitre traite de l'interaction gestuelle pour la manipulation des ordinateurs. Il présente les mécanismes permettant de traduire les gestes reconnus en commandes informatiques, ainsi que les contraintes ergonomiques et techniques liées au contrôle gestuel des systèmes informatiques.

Le quatrième chapitre décrit la conception et la réalisation du système proposé. Nous y présentons l'architecture globale de la solution développée, les différentes étapes du traitement des données, la constitution des jeux de données, l'extraction des points clés de

la main à l'aide de MediaPipe, ainsi que le développement des modèles SVM et BiLSTM et de l'interface graphique du système.

Enfin, le cinquième chapitre est consacré à l'expérimentation et à l'analyse des résultats. Nous y présentons le protocole expérimental adopté, les métriques d'évaluation utilisées, les performances obtenues par les modèles développés, une comparaison avec des travaux de référence ainsi qu'une discussion des avantages, des limites et des perspectives d'amélioration du système proposé.

Chapitre 1

État de l’art sur la reconnaissance gestuelle

Chapitre1: État de l'art sur la reconnaissance gestuelle

1.1. Introduction

La reconnaissance des gestes de la main constitue un domaine clé de l'intelligence artificielle et du traitement d'images. Elle vise à permettre aux systèmes informatiques d'interpréter les mouvements de la main humaine comme une forme d'interaction naturelle. Elle repose sur des techniques de vision par ordinateur combinées à des méthodes d'apprentissage automatique et d'apprentissage profond pour analyser des gestes statiques et dynamiques. Cette approche contribue au développement d'interfaces homme-machine plus intuitives et efficaces.[1]

1.2. Définition des gestes(statiques/ dynamiques)

Les gestes de la main sont ces mouvements ou positions naturelles de la main qu'on utilise pour donner des ordres ou contrôler un ordinateur sans le toucher, surtout dans les interfaces homme-machine. On les repère grâce à la vision par ordinateur, ce qui rend l'interaction fluide et intuitive, comme si on parlait avec la machine.

1.2.1. Gestes statiques

Les gestes statiques, c'est une seule pose figée de la main, sans mouvement dans le temps – par exemple, le pouce levé pour dire « OK » ou le poing fermé pour arrêter. Ils sont parfaits pour des commandes simples comme sélectionner ou activer quelque chose, car ils sont hyper faciles à détecter sans complication.[1]

1.2.2. Gestes dynamiques

Les gestes dynamiques, eux, évoluent avec le temps : un glissement (swipe) pour faire défiler, ou dessiner dans l'air pour écrire virtuellement. Ça demande plus de calcul pour suivre la trajectoire, mais c'est idéal pour du contrôle avancé, genre en réalité virtuelle, où l'interaction devient riche et immersive.[2]



[2]

Figure1.1: figure montrant la différence entre les gestes Statique et dynamique

1.3. Domaines d'application

Les gestes de la main sans contact élargissent les possibilités d'interaction avec les machines bien au-delà des clavier et souris traditionnels. Ils s'intègrent dans une variété de secteurs où l'intuitivité et la précision gestuelle apportent un vrai plus, en rendant les systèmes plus naturels et accessibles. Voici les principaux domaines, avec des exemples concrets pour mieux illustrer leur impact.

1.3.1. Interface Homme-Machine (IHM)

Dans les IHM, les gestes remplacent les commandes physiques pour contrôler des écrans tactiles virtuels ou des bureaux multimédias. Par exemple, un balayage de la main peut faire défiler une page web, ou une pose ouverte peut zoomer sur une carte, comme dans les smart TV modernes ou les tableaux interactifs en entreprise. Cela rend l'expérience plus fluide, surtout pour les réunions collaboratives ou les postes de travail sans les mains (hands-free).

1.3.2. Réalité Virtuelle et Augmentée (RV / RA)

La RV et RA exploitent pleinement les gestes pour une immersion totale : imaginez manipuler des objets 3D en les « saisissant » virtuellement dans un casque Oculus, ou superposer des infos augmentées sur le monde réel via HoloLens (casque de réalité mixte combinant réel et virtuel). Dans l'industrie, les ingénieurs conçoivent des prototypes en tournant la main pour pivoter un modèle, ou dans l'éducation, les étudiants dissèquent un organe virtuel sans bistouri. C'est révolutionnaire pour la formation et la simulation.[3]



Figure1.2: figure montrant l' Interaction gestuelle en Réalité Virtuelle et Augmentée.[3]

1.3.3. Jeux vidéo

Les jeux adoptent les gestes pour une jouabilité immersive, comme dans « Tom Clancy's Ghost Recon» où un poing levé ordonne une pause tactique, ou dans les jeux de danse comme Just Dance qui suivent les mouvements complets. Sur consoles comme la Nintendo Switch ou PlayStation avec PS Move (manette de mouvement), cela augmente l'engagement physique, rendant les parties plus dynamiques et sociales, surtout en multijoue.



Figure1.3: Jouer à des jeux à l'aide des gestes de la main

1.3.4. Robotique

En robotique, les gestes dirigent les robots à distance : un ouvrier en usine peut pointer pour qu'un bras robotisé saisisse une pièce, ou un opérateur guide un drone par des glissements aériens. Dans les drones de livraison ou les robots chirurgicaux, cela évite les ralentissements des joysticks et améliore la précision en environnements dangereux, comme les sites nucléaires ou les zones de catastrophe.[4]



Figure1.4: Applications gestuelles en robotique industrielle.

1.3.5. Santé et Médecine

Le secteur médical utilise les gestes pour l'asepsie : un chirurgien contrôle un scanner 3D sans toucher l'écran, évitant toute contamination pendant une opération. Dans la rééducation, des patients revalident leur motricité fine via des exosquelettes guidés par gestes, ou des applications suivent les tremblements pour diagnostiquer Parkinson précocement. C'est un gain énorme en hygiène et en suivi personnalisé.



Figure1.5: Applications gestuelles en santé et médecine.

1.3.6. Présentation et Éducation

Pour les présentations, les gestes animent les diapositives: un mouvement de la main avance une diapositive qui met en évidence un graphique, comme avec des outils comme Leap Motion (capteur de mouvements de la main) intégrés à PowerPoint. En classe, les profs contrôlent des tableaux blancs interactifs, et les élèves manipulent des molécules virtuelles en chimie. Cela dynamise l'enseignement et rend les cours plus interactifs pour tous les âges.



Figure1.6 : Applications gestuelles en présentation et éducation.

1.4. Conclusion

Dans ce chapitre, nous avons présenté les notions fondamentales de la reconnaissance gestuelle ainsi que ses principaux domaines d'application. Cette étude a mis en évidence l'importance croissante de cette technologie dans le développement d'interfaces homme-machine plus naturelles et intuitives. Le chapitre suivant sera consacré aux techniques et méthodes utilisées pour la détection et la reconnaissance des gestes.

Chapitre 2

Techniques et méthodes de reconnaissance gestuelle

Chapitre2: Techniques et méthodes de reconnaissance gestuelle

2.1. Introduction

Après avoir présenté les concepts fondamentaux de la reconnaissance gestuelle et ses domaines d'application dans le chapitre précédent, il est nécessaire d'étudier les techniques et méthodes qui permettent de concevoir un système de reconnaissance performant. En effet, la reconnaissance des gestes repose sur une chaîne de traitement comprenant plusieurs étapes, allant de l'acquisition des données jusqu'à l'interprétation et l'exécution de la commande correspondante.

Dans ce chapitre, nous présentons les principales étapes du pipeline de reconnaissance gestuelle, notamment l'acquisition des données, le prétraitement, l'extraction des caractéristiques et la classification. Nous examinons également les méthodes de classification les plus utilisées, telles que le SVM, le Random Forest, le k-NN (K-Nearest Neighbors), les réseaux de neurones convolutifs (CNN) ainsi que les architectures récurrentes comme le GRU (Gated Recurrent Unit) . Enfin, une comparaison de ces approches permettra d'identifier les solutions les plus adaptées à la reconnaissance des gestes statiques et dynamiques en temps réel.

2.2. Pipeline d'un système de reconnaissance gestuelle

L'évolution de la robotique collaborative et des interfaces utilisateur naturelles a placé la reconnaissance gestuelle au cœur des systèmes d'Interaction Homme-Machine (HRI). L'objectif est de permettre à un utilisateur de commander un système (un robot, un logiciel de présentation, ou une interface domotique) de manière intuitive, sans contact physique. Cependant, passer d'un flux de données brutes à une commande logicielle fiable nécessite une architecture robuste capable de traiter les informations en temps réel tout en minimisant les erreurs de détection.[14]

2.2.1. Acquisition de données (Sensing)

L'étape initiale consiste à capturer le flux vidéo de l'utilisateur. Pour assurer une accessibilité maximale et un coût matériel réduit, nous utilisons une caméra RGB standard. Contrainte temps réel : Le système doit traiter un nombre d'images par seconde (FPS) suffisant pour ne pas manquer les transitions rapides des gestes dynamiques.[22]

Variabilité : Cette étape doit tenir compte des conditions d'éclairage changeantes et de la résolution de la caméra intégrée.

2.2.2. Prétraitement et Segmentation (Preprocessing)

Une fois l'image capturée, elle subit des transformations pour isoler l'information utile :

- **Réduction du bruit :** Application de filtres (ex: flou gaussien) pour lisser l'image.
- **Segmentation de la main :** Il s'agit d'extraire la main de l'arrière-plan (souvent complexe dans une salle de classe ou de conférence). Le pipeline doit définir une région d'intérêt (ROI) pour concentrer les calculs uniquement sur la zone où se trouve la main.
- **Normalisation :** Ajustement de l'échelle et de l'orientation pour que le système reconnaisse le geste, que l'utilisateur soit proche ou loin de la caméra.

2.2.3. Extraction des caractéristiques (Feature Extraction)

C'est le cœur technique du pipeline. Au lieu de traiter l'image entière, le système extrait des vecteurs mathématiques représentatifs:

- **Points caractéristiques (Landmarks) :** Identification des articulations des doigts et des coordonnées de la paume.
- **Descripteurs spatiaux :** Pour les gestes statiques, on analyse la configuration relative des doigts (ex: index levé vs poing fermé).
- **Descripteurs temporels :** Pour les gestes dynamiques, on suit l'évolution des coordonnées dans le temps pour identifier une trajectoire (ex: un balayage vers la droite).[15]

2.2.4. Classification et Reconnaissance

Les caractéristiques extraites sont envoyées à un classifieur. Le choix de l'algorithme est crucial pour l'efficacité sur des ordinateurs à ressources limitées:

- **Méthodes classiques** : Des algorithmes comme les SVM ou k-NN peuvent suffire pour des gestes statiques simples.[23]
- **Apprentissage profond léger :** L'utilisation de modèles optimisés permet de reconnaître des gestes complexes avec une grande précision sans saturer le processeur de l'ordinateur.[25]

2.2.5. Post-traitement et Exécution de la commande

La dernière étape du pipeline consiste à traduire la classification en action concrète dans le logiciel de présentation:

- **Filtrage temporel :** Pour éviter les déclenchements accidentels (faux positifs), le système peut exiger que le geste soit maintenu pendant quelques millisecondes ou qu'il soit validé par une étape de « lissage ».
- **Interface de commande :** Le geste validé est traduit en un événement clavier virtuel (ex: touche « Flèche Droite » pour la diapositive suivante.[4]

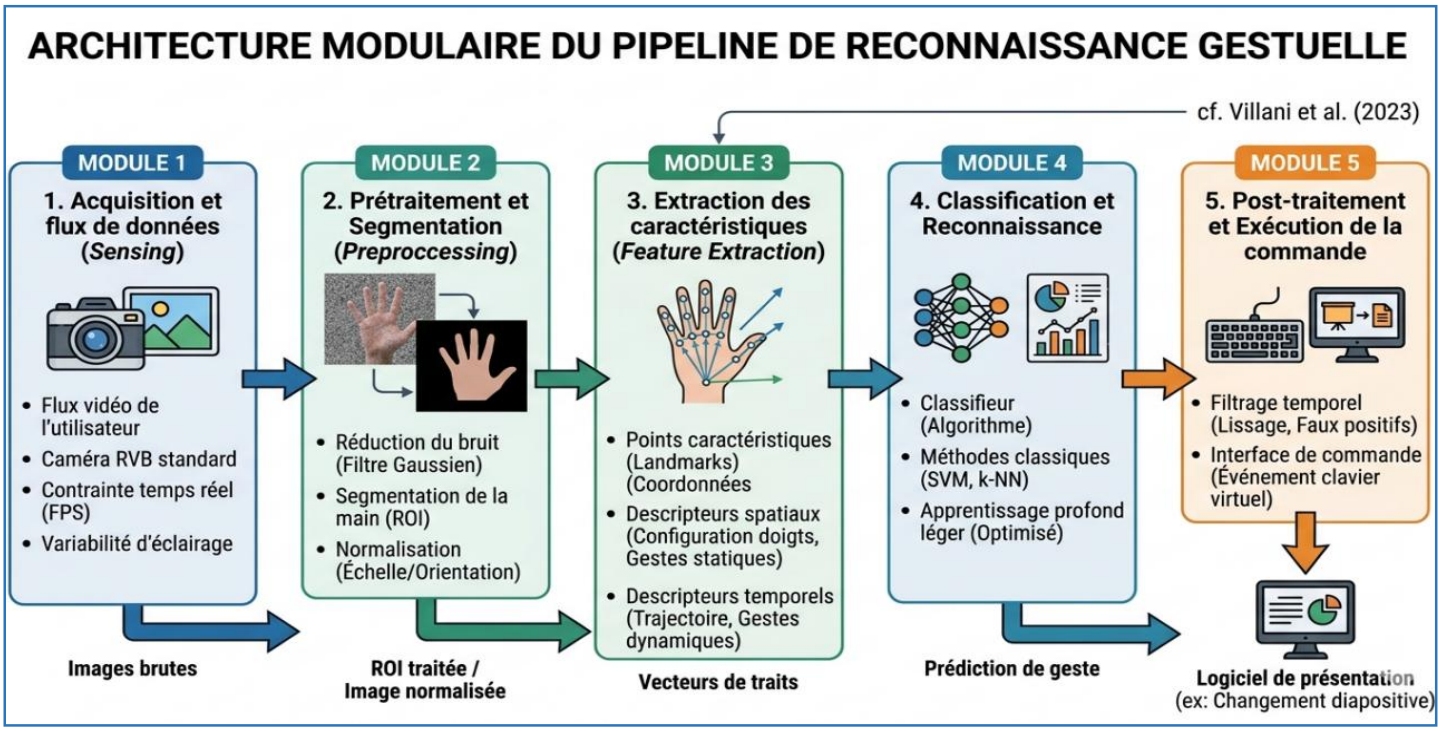


Figure2.1 : Architecture modulaire du pipeline du reconnaissance gestuelle

2.3. Comparaison entre les méthodes de classification

Le choix d'un algorithme de classification est une étape critique qui détermine la précision et la réactivité du système de reconnaissance gestuelle

Dans cette section, nous aborderons certaines de ces méthodes:

2.3.1. Machines à Vecteurs de Support (SVM)

Le SVM est un classifieur supervisé qui cherche à définir l'hyperplan optimal pour séparer les données en différentes classes des gestes.

- **Performance sur les gestes statiques :** Il excelle dans la reconnaissance de poses fixes où la position relative des doigts est le seul facteur discriminant.
- **Efficacité computationnelle :** Une fois entraîné, le modèle SVM est extrêmement léger, ce qui permet une exécution fluide sur des ordinateurs à ressources limitées.
- **Dépendance aux caractéristiques :** Son efficacité dépend fortement de la qualité de l'extraction manuelle des descripteurs (forme, contours), car il ne traite pas nativement la séquence temporelle. [24]

2.3.2. Bidirectional Long Short-Term Memory (BiLSTM)

Le BiLSTM est une variante des réseaux de neurones récurrents (RNN) conçue pour mémoriser des informations sur de longues périodes dans les deux directions temporelles, ce qui le rend particulièrement adapté aux données séquentielles.

- **Maîtrise de la dynamique :** Contrairement au SVM, le BiLSTM capture naturellement l'évolution temporelle dans les deux sens, ce qui est indispensable pour identifier des gestes dynamiques comme les balayages (swipes) ou les rotations.
- **Apprentissage de bout en bout** : Il peut apprendre directement des représentations complexes à partir des flux de données, réduisant ainsi le besoin d'une extraction de caractéristiques artisanale.
- **Coût matériel :** Cette méthode nécessite une puissance de calcul plus élevée et un volume de données d'entraînement beaucoup plus important pour éviter le sur-apprentissage.[29]

2.3.3. Random Forest (RF)

Le Random Forest est un algorithme d'apprentissage ensembliste qui combine plusieurs arbres de décision pour améliorer la robustesse et la précision de la classification.

- Robustesse aux données bruitées : En moyennant les résultats de nombreux arbres, il réduit considérablement le risque de sur-apprentissage lié aux variations de luminosité ou aux imprécisions de capture des points clés de la main.
- Évaluation de l'importance des variables : Cette méthode permet d'identifier quels descripteurs (angles des doigts, distance entre les articulations) sont les plus discriminants pour chaque geste.
- Rapidité d'exécution : Bien que l'entraînement puisse être plus long qu'un SVM, la phase de prédiction est extrêmement rapide, répondant ainsi aux exigences du temps réel.

2.3.4. k-plus proches voisins (k-NN)

L'algorithme k-NN est une méthode non-paramétrique qui classifie un geste en fonction de sa proximité avec les exemples mémorisés lors de l'apprentissage dans l'espace des caractéristiques.

- **Simplicité et adaptabilité** : Il ne nécessite pas de phase d'entraînement complexe ; l'ajout de nouveaux gestes se fait simplement en ajoutant des exemples à la base de données.
- **Sensibilité à l'échelle** : Son efficacité repose sur une normalisation rigoureuse des données (coordonnées des landmarks), car les distances calculées sont sensibles aux variations d'échelle des mains.
- **Coût mémoire et latence** : La recherche des voisins les plus proches peut devenir coûteuse en ressources de calcul si la base de données est volumineuse, ce qui peut impacter la fluidité du système.

2.3.5. Réseaux de Neurones Convolutionnels (CNN)

Initialement conçus pour l'analyse d'images, les CNN sont utilisés ici pour extraire automatiquement des hiérarchies de caractéristiques spatiales à partir des cartes de profondeur ou des représentations matricielles des mains.

- **Extraction automatique de traits** : Contrairement au SVM, le CNN élimine l'étape d'extraction manuelle des descripteurs, car il apprend lui-même à reconnaître les formes et contours pertinents de la main.
- **Performance spatiale** : Il offre une précision inégalée pour la reconnaissance des gestes statiques complexes et de poses de doigts entrelacées.
- **Exigences matérielles** : Le déploiement d'un CNN nécessite souvent une accélération matérielle GPU (Graphics Processing Unit) pour maintenir une cadence de traitement fluide (FPS) lors de l'analyse vidéo.

2.3.6. Gated Recurrent Unit (GRU)

Le GRU est une variante simplifiée du LSTM qui utilise des mécanismes de porte (mise à jour et réinitialisation) pour modéliser les dépendances temporelles dans les séquences des gestes.

- **Optimisation des paramètres** : Possédant une structure plus légère que le LSTM, le GRU s'entraîne plus rapidement et nécessite moins de données pour converger, tout en offrant des performances similaires pour la reconnaissance dynamique.
- **Efficacité pour le temps réel** : Sa complexité algorithmique réduite permet une latence plus faible, ce qui est crucial pour la réactivité lors de la manipulation d'interfaces (navigation, défilement).
- **Gestion des séquences courtes** : Il s'avère particulièrement performant pour les gestes dynamiques brefs, comme le clic rapide ou le changement de page.

2.3.7. Synthèse comparative

Le tableau suivant résume les différences fondamentales entre ces deux approches dans le contexte de l'interaction homme-machine :

Critère	SVM	KNN	Random Forest	BiLSTM	CNN	GRU
Cible principale	Gestes statiques(poses)	Poses simples	Gestes statiques robustes	Gestes dynamique(mouvements)	Poses complexes (images)	Gestes dynamiques
Complexité algorithmique	Faible à modérée	Très faible	Modérée	Élevée	Élevée	Élevée
Gestion du temps	Nécessite des descripteurs temporels externes	Nulle	Nulle (traitement instantané)	Temps de calcul plus élevé (inférence plus lente que SVM)	Faible(analyse spatiale uniquement)	Native (optimisée pour le temps réel)
Volume de données requis	Faible (quelques dizaines d'exemples)	Faible	Modéré (centaines d'exemples)	Élevé (milliers de séquences)	Élevée(milliers d'images)	Elevé (milliers de séquences)
Usage recommandé	Systèmes temps réel sur matériel modeste	Prototypage et systèmes de base	Classification de Landmarks / Points clés	Interfaces immersives et complexes (RV/RA)	Reconnaissance visuelle brute / Vision artificielle	Pilotage de présentations numériques en temps réel

Tableau 2.1: Étude comparative entre les méthodes de classification[5]

2.4. Conclusion

Ce chapitre a permis de décomposer l'architecture complexe d'un système de reconnaissance gestuelle en un pipeline modulaire et structuré. Nous avons vu que la fiabilité du système ne repose pas uniquement sur l'intelligence de l'algorithme final, mais sur la qualité de chaque étape : de l'acquisition par capteur standard à l'extraction de caractéristiques (landmarks), qui transforme une image pixelisée en un vecteur mathématique exploitable.

L'étude comparative des différentes méthodes de classification met en évidence un compromis nécessaire entre puissance de calcul et nature du geste :

Les approches classiques comme le SVM ou le Random Forest s'avèrent idéales pour la reconnaissance de postures statiques sur des machines aux ressources limitées, grâce à leur légèreté et leur rapidité.

À l'inverse, pour capturer la fluidité des gestes dynamiques, les architectures récurrentes comme le BiLSTM ou le GRU sont indispensables, bien qu'elles exigent un volume de données et une puissance de calcul supérieurs.

En définitive, le choix d'une architecture optimale pour une interface de commande (comme un logiciel de présentation) semble s'orienter vers une solution hybride : capable de traiter des points clés de la main (plus légers que l'image brute) via un classifieur optimisé pour le temps réel.

C'est sur cette base méthodologique que nous allons maintenant appuyer la conception de notre propre solution technique .

Chapitre 3

Interaction gestuelle pour la manipulation des ordinateurs

Chapitre3: Interaction gestuelle pour la manipulation des ordinateurs

3.1. Introduction

La reconnaissance gestuelle ne constitue pas une finalité en soi, mais un moyen de permettre une interaction plus naturelle entre l'utilisateur et l'ordinateur. Dans ce contexte, les interfaces homme-machine basées sur les gestes représentent une évolution importante des modes de contrôle traditionnels, en offrant la possibilité d'exécuter des commandes sans contact physique avec les périphériques classiques.

Cependant, la conception d'un système de manipulation gestuelle efficace ne dépend pas uniquement des performances des algorithmes de reconnaissance. Elle doit également prendre en compte les aspects ergonomiques, les contraintes d'utilisation ainsi que les attentes des utilisateurs en matière de confort, de précision et de fluidité d'interaction. Le choix des gestes, la qualité du retour visuel et la réduction de la fatigue physique jouent un rôle essentiel dans l'acceptation et l'efficacité de ces interfaces.

Dans ce chapitre, nous présentons les principes des interfaces homme-machine basées sur les gestes, les principaux gestes utilisés pour la manipulation des ordinateurs ainsi que les notions d'ergonomie et d'expérience utilisateur. Nous abordons également les contraintes et les défis liés à l'interaction gestuelle avant d'identifier les exigences nécessaires à la conception d'un système de contrôle gestuel performant et adapté à une utilisation en temps réel.

3.2. Interfaces homme-machine basées sur les gestes

Les interfaces homme-machine basées sur les gestes visent à offrir une interaction plus naturelle que les périphériques traditionnels. Elles reposent sur l'utilisation des mouvements de la main pour transmettre des commandes à l'ordinateur.

Les gestes peuvent être classés selon plusieurs critères. Selon leur effet, on distingue les gestes discrets, qui déclenchent une action unique, et les gestes continus, qui permettent un contrôle en temps réel d'un paramètre. Selon leur dimension temporelle, on distingue les gestes statiques, correspondant à une posture fixe de la main, et les gestes dynamiques,

définis par une succession de mouvements dans le temps. La reconnaissance des gestes dynamiques est généralement plus complexe car elle nécessite l'analyse de l'évolution du mouvement ainsi que l'identification du début et de la fin de l'action.[6]

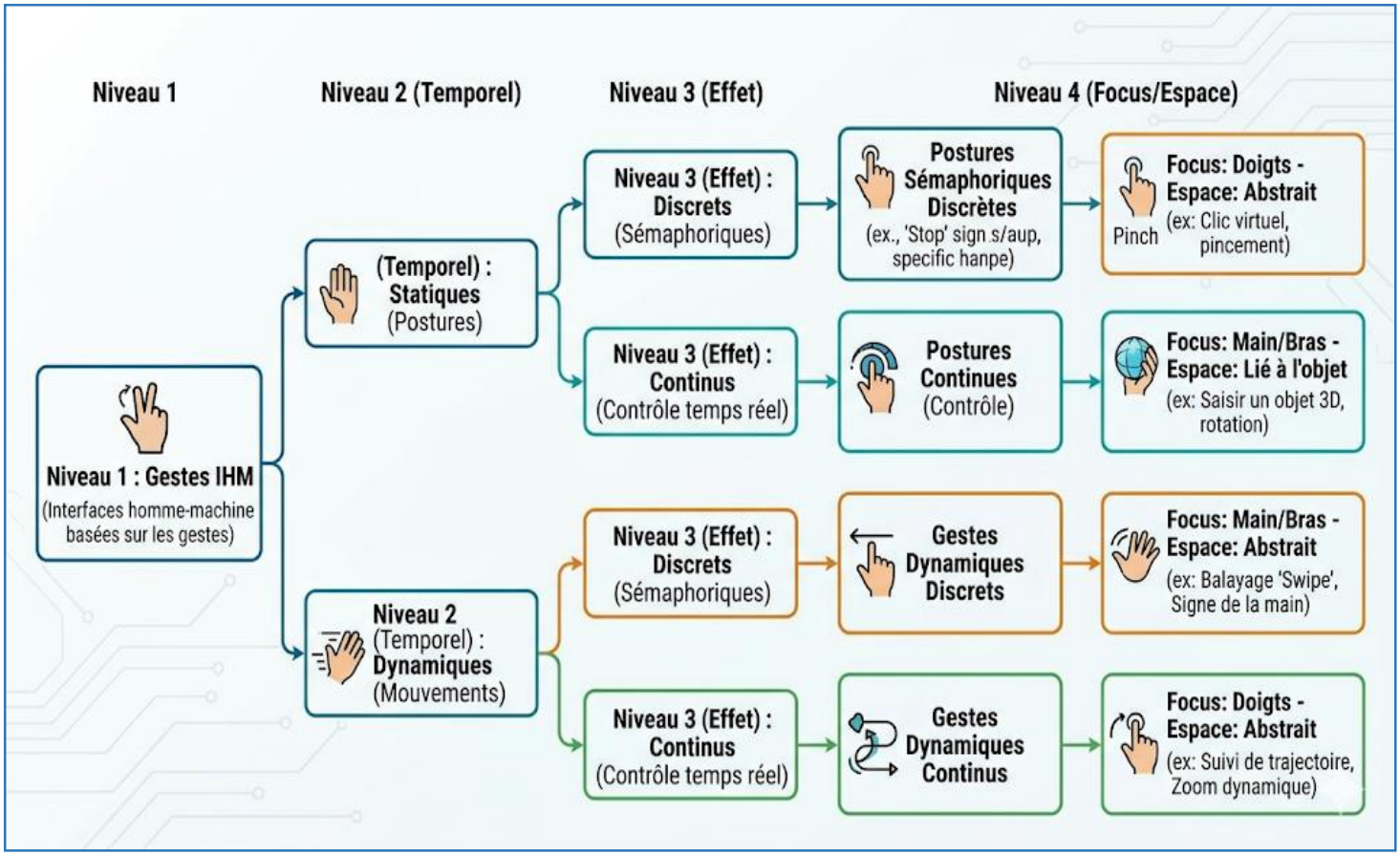


Figure3.1: Taxonomie multidimensionnelle des gestes en Interaction Homme-Machine

Les interfaces gestuelles utilisent soit des caméras et capteurs de vision, soit des capteurs portables pour capturer les mouvements de la main. Le processus de reconnaissance comprend l'acquisition des données, leur prétraitement, l'extraction des caractéristiques pertinentes et la classification du geste. Selon le type de geste, différents algorithmes peuvent être utilisés, notamment les réseaux de neurones récurrents pour les gestes dynamiques et des classificateurs plus simples pour les gestes statiques.[6]

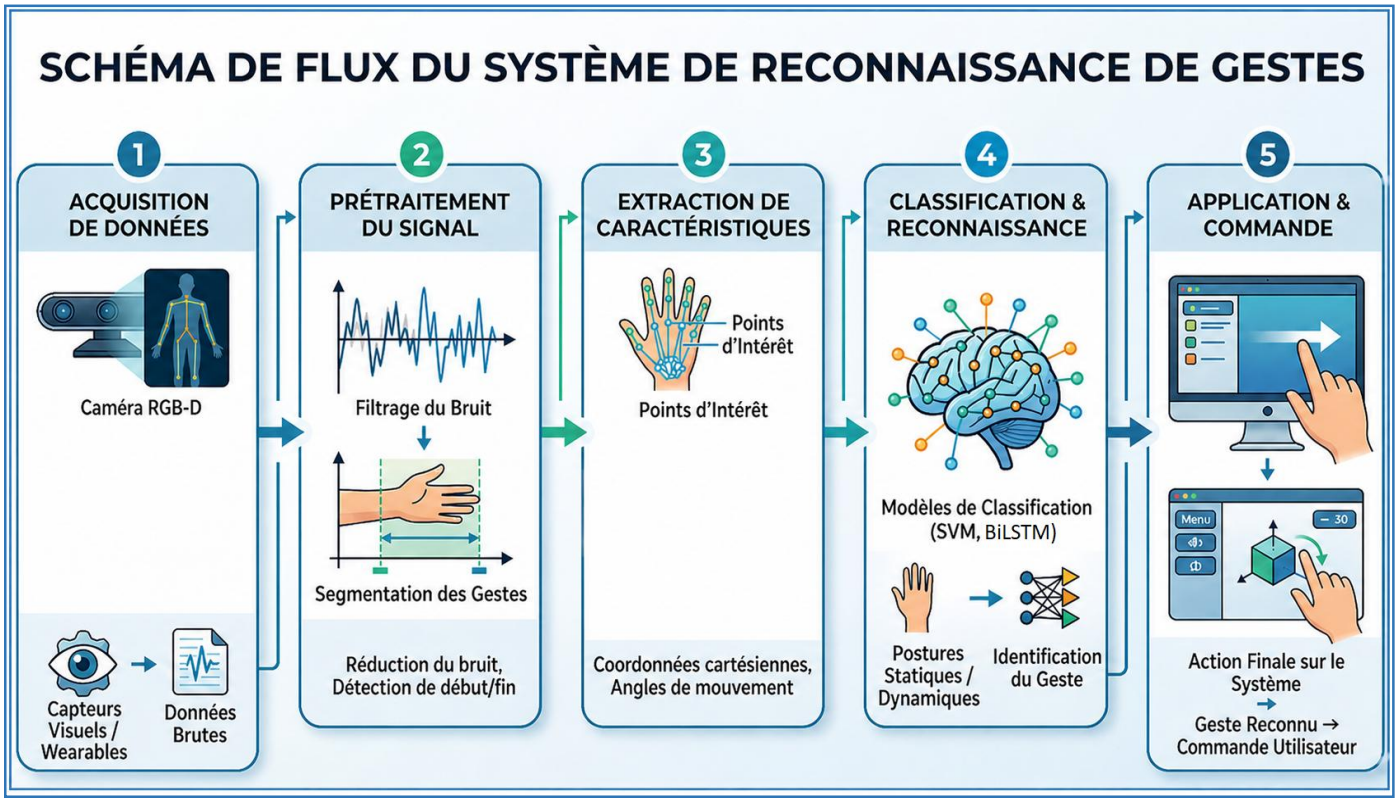


Figure3.2: Les étapes clés d'un système de reconnaissance des gestes

3.3. Gestes utilisées pour la manipulation

La manipulation d'objets virtuels repose sur une taxonomie des gestes permettant de transformer les mouvements de la main en commandes numériques, réparties principalement en deux catégories : la navigation et la sélection.

La navigation regroupe les gestes liés au déplacement et à l'exploration de l'interface, tels que le déplacement (pan), le zoom par rapprochement ou écartement des doigts, la rotation d'un objet ainsi que le défilement des contenus. Ces gestes assurent une interaction fluide et continue avec l'environnement.

La sélection correspond aux actions discrètes permettant d'interagir avec un élément précis. Elle inclut le pointage, la validation (air tap ou maintien du curseur), le geste de pincement pour saisir un objet, ainsi que la sélection par lasso pour regrouper plusieurs éléments. Cette catégorie exige une grande précision afin d'éviter les erreurs d'interprétation.[7]

3.4. Ergonomie et expérience utilisateur

L'efficacité d'un système de reconnaissance gestuelle ne se mesure pas uniquement à sa précision algorithmique, mais aussi à sa capacité à offrir une interaction confortable et intuitive. Dans le cadre d'un système de contrôle de présentation, l'ergonomie (UX: User Experience /UI: User Interface) doit répondre à des défis physiologiques et cognitifs spécifiques.

3.4.1. Le confort physique et la fatigue (Gorilla Arm Syndrome)

Le confort physique est un enjeu majeur dans les interfaces gestuelles, notamment en raison du syndrome du « bras de gorille », qui apparaît lorsque l'utilisateur maintient ses bras levés trop longtemps, entraînant fatigue et inconfort. Pour limiter ce problème, les gestes doivent être réalisés dans une zone de confort proche du corps, et les interactions doivent rester rapides et peu contraignantes, notamment pour les commandes de navigation.

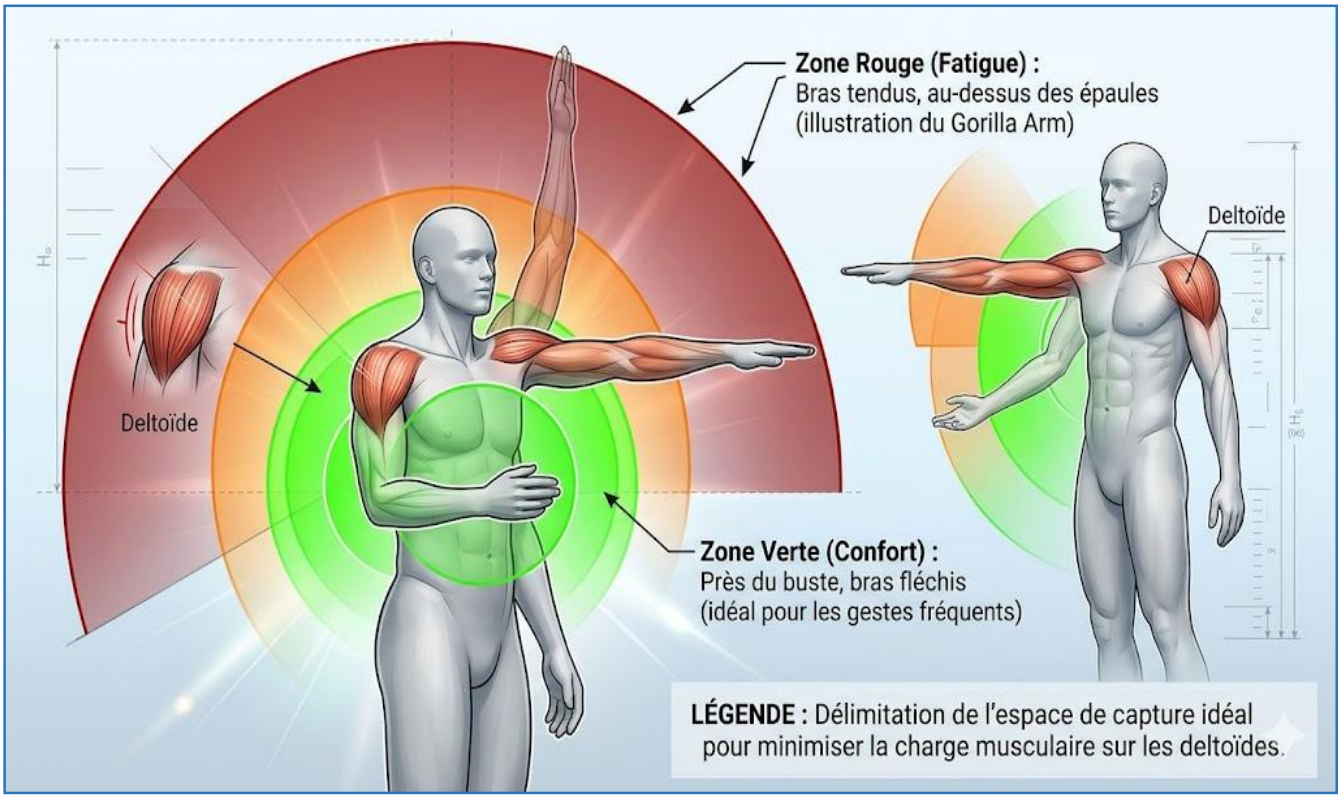


Figure3.3: Délimitation des zones d'interaction ergonomiques et prévention du syndrome du « bras de gorille »

3.4.2. L'affordance et la métaphore naturelle

Pour qu'une interface soit intuitive, elle doit exploiter des métaphores naturelles que l'utilisateur possède déjà. L'objectif est de réduire la charge cognitive : l'utilisateur ne doit pas « réfléchir » à la commande, mais l'exécuter par réflexe.

Mapping Naturel : Un mouvement de la main vers la droite pour passer à la diapositive suivante est une extension logique de la lecture linéaire occidentale.

Simplicité : Un geste trop complexe (ex: une rotation spécifique des doigts) augmente le risque d'erreur de la part de l'utilisateur et du système de classification.

3.4.3. La boucle de rétroaction (Feedback)

L'interaction gestuelle nécessite un mécanisme de retour d'information (feedback), car elle ne dispose pas de sensation physique comme le clic d'une souris. Le système doit donc fournir un feedback visuel ou sonore immédiat pour confirmer la bonne détection du geste, par exemple via un changement d'icône ou de couleur. De plus, la latence doit être très faible (idéalement inférieure à 100–200 ms), afin d'éviter les répétitions des gestes et les erreurs de type faux positifs.[8]

3.4.4. Loi de Fitts appliquée au pointage gestuel

Pour les fonctions de sélection ou de pointage (par exemple, utiliser la main comme un pointeur laser), les principes de la loi de Fitts s'appliquent. Elle stipule que le temps nécessaire pour atteindre une cible dépend de la distance à la cible et de sa taille.

$$T = a + b \log_2 \left(1 + \frac{W}{D} \right)$$

T : Temps de mouvement.

D : Distance vers le centre de la cible.

W : Largeur (taille) de la cible.

Dans une interface gestuelle, les cibles (boutons de navigation) doivent être plus larges que sur une interface classique pour compenser l'instabilité naturelle de la main humaine en l'air (le « jitter »). [8]

3.5. Contraintes et défis

3.5.1. Contraintes

La reconnaissance gestuelle par vision artificielle est confrontée à plusieurs contraintes qui rendent sa mise en œuvre complexe. Celles-ci incluent d'abord les conditions environnementales, telles que les variations de luminosité et les arrière-plans complexes, qui perturbent la détection de la main. Ensuite, la nature même de la main humaine pose des difficultés liées à sa non-rigidité, aux auto-occlusions et à la variabilité entre utilisateurs.

D'autres contraintes concernent l'acquisition des données, notamment le flou de mouvement, la distance par rapport à la caméra et la fréquence d'images insuffisante, pouvant entraîner une perte d'informations. Enfin, des défis spatio-temporels et algorithmiques apparaissent, comme la reconnaissance indépendante de l'échelle et de l'orientation, la segmentation précise des gestes dans le temps, ainsi que la nécessité de respecter les contraintes du temps réel tout en maintenant de bonnes performances. [9]

3.5.2. Défies

Les défis liés à la reconnaissance gestuelle en conditions réelles sont nombreux et concernent plusieurs aspects fondamentaux du système. L'un des principaux problèmes est la segmentation temporelle, qui consiste à détecter avec précision le début et la fin d'un geste dans un flux vidéo continu. À cela s'ajoute le phénomène du Midas Touch, où il devient difficile de distinguer les gestes intentionnels des mouvements involontaires.

Un autre défi important concerne la variabilité des utilisateurs, aussi bien entre différents individus qu'au sein d'un même utilisateur, ce qui complique la généralisation des modèles. La reconnaissance doit également gérer la dynamique temporelle des gestes, notamment leur vitesse d'exécution et l'ordre des mouvements, qui sont essentiels pour éviter les confusions entre gestes similaires.

Enfin, l'augmentation du nombre des gestes à reconnaître pose un problème de scalabilité, avec un risque accru de confusion entre classes proches et une charge computationnelle plus élevée. Les systèmes basés uniquement sur la vision RGB (Red,

Green, Blue) présentent également des limites, notamment l'absence d'information de profondeur et leur sensibilité aux conditions environnementales.[9]

3.6. Conclusion

Ce chapitre a présenté une vue d'ensemble des interfaces homme-machine basées sur la reconnaissance gestuelle, en mettant en évidence la complexité de ce type d'interaction.

Il a montré que la conception d'un système efficace repose sur trois éléments essentiels : la définition claire des types des gestes, la robustesse des méthodes techniques face aux contraintes de l'environnement, et la prise en compte de l'ergonomie pour garantir une interaction naturelle et confortable.

Enfin, les différentes limites et défis identifiés soulignent la nécessité d'une approche adaptée et optimisée pour les applications en temps réel. Ces éléments constituent la base du chapitre suivant, consacré à la conception et à la réalisation du système proposé.

Chapitre 4

Conception et réalisation du système proposé

Chapitre4: Conception et réalisation du système proposé

4.1. Introduction

Après avoir présenté les concepts théoriques liés à la reconnaissance gestuelle et aux interfaces homme-machine dans les chapitres précédents, ce chapitre est consacré à la conception et à la réalisation du système proposé de contrôle de l'ordinateur par les mouvements de la main.

Nous commençons par l'analyse des besoins fonctionnels et non fonctionnels ainsi que la définition des objectifs du système. Ensuite, nous décrivons l'architecture générale adoptée, depuis l'acquisition des données jusqu'à l'exécution des commandes. Nous présentons également les gestes statiques et dynamiques retenus ainsi que les actions associées. Enfin, nous détaillons l'implémentation du système, les méthodes de classification utilisées pour la reconnaissance des gestes et la validation du fonctionnement en temps réel.

4.2. Analyse des besoins

L'analyse des besoins constitue le socle de la phase de conception. Elle vise à identifier précisément les objectifs fonctionnels et les contraintes techniques nécessaires à la mise en place d'un système de reconnaissance gestuelle efficace. Cette étape est cruciale pour définir comment le système doit interpréter les mouvements de l'utilisateur afin de transformer un flux vidéo brut en commandes de navigation fluides et intuitives, tout en respectant les limites ergonomiques liées à l'utilisation prolongée.

4.2.1. Définir l'objectif

L'objectif principal de ce système est de développer une interface de contrôle sans contact permettant de manipuler une présentation numérique à l'aide des mouvements naturels de la main. Le système assure l'acquisition et le traitement vidéo en temps réel, l'extraction des points caractéristiques de la main, la reconnaissance des gestes statiques et dynamiques, ainsi qu'une interaction fluide grâce à une faible latence d'exécution.

4.2.2. Spécification des besoins fonctionnels

Le système doit assurer la détection et le suivi de la main en temps réel, reconnaître les différents gestes de commande, distinguer les mouvements intentionnels des gestes naturels afin d'éviter les activations involontaires, et convertir les gestes reconnus en actions exploitables par les applications de présentation.

4.2.3. Spécification des besoins non-fonctionnels

Le système doit offrir une reconnaissance précise et robuste des gestes, garantir un fonctionnement en temps réel avec une faible latence, assurer une utilisation confortable grâce à des gestes ergonomiques, et fonctionner avec une webcam standard sans nécessiter de matériel spécialisé.

4.2.4. Synthèse des besoins techniques

Le tableau suivant récapitule les besoins prioritaires pour la réalisation du système:

Besoin	description
Capture	Acquisition du flux vidéo via une webcam standard (RGB).
Traitement	Extraction des 21 landmarks de la main en temps réel.
Ergonomie	Définition d'une taxonomie des gestes intuitifs et peu fatigants.
Feedback	Retour visuel immédiat pour confirmer la prise en compte du geste

Tableau4.1: Tableau récapitulatif les besoins prioritaires pour la réalisation du système

4.3. Architecture du système

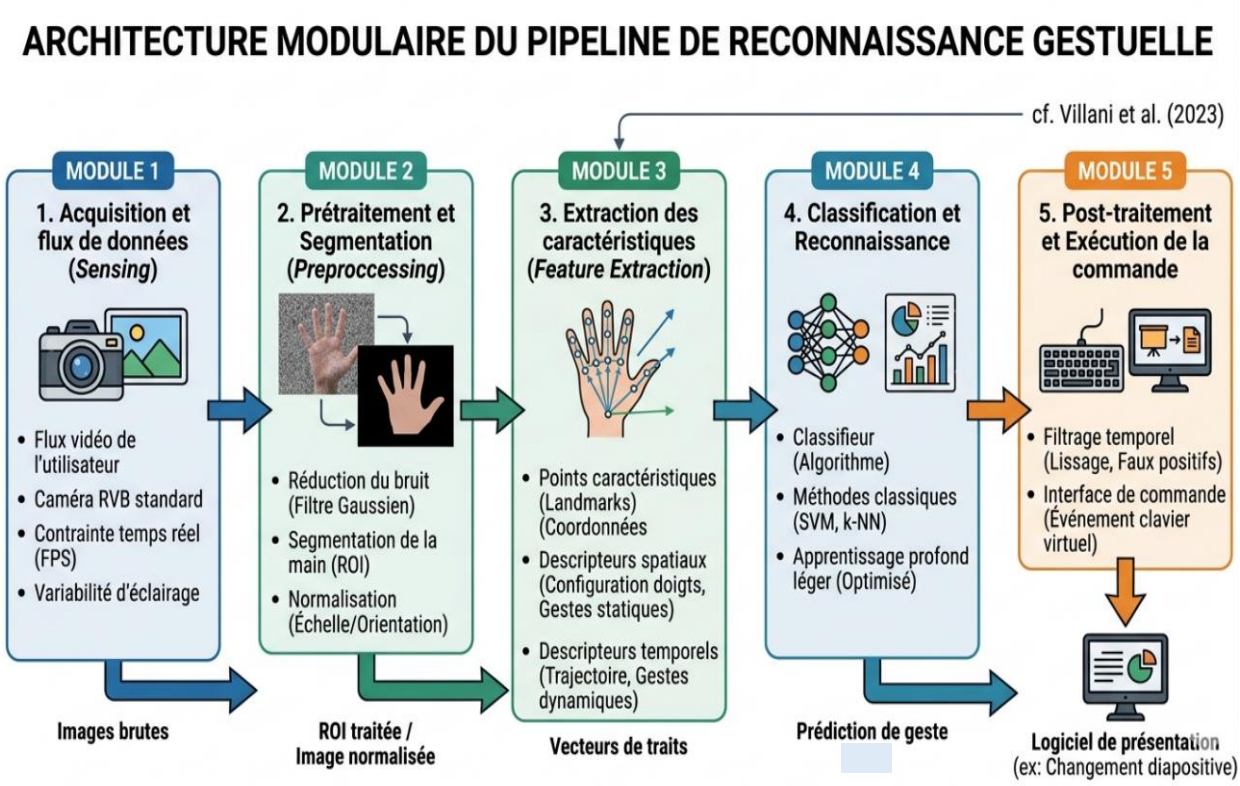


Figure4.1: Figure illustrant une architecture modulaire du pipeline de reconnaissance gestuelle

4.3.1. Acquisition des données

L'acquisition des données constitue la première étape du système. Elle repose sur une webcam RGB standard qui capture les images en temps réel et les convertit en données numériques exploitables. Le flux vidéo obtenu sert de base à l'analyse des formes et des mouvements de la main pour la reconnaissance gestuelle.

4.3.2. Prétraitement et segmentation du signal

Le prétraitement prépare les données vidéo pour l'analyse en éliminant le bruit, en isolant la main de l'arrière-plan et en détectant les 21 points clés (landmarks). Les coordonnées sont ensuite normalisées afin de garantir une reconnaissance fiable des gestes indépendamment de la position ou de la distance de l'utilisateur.[16]

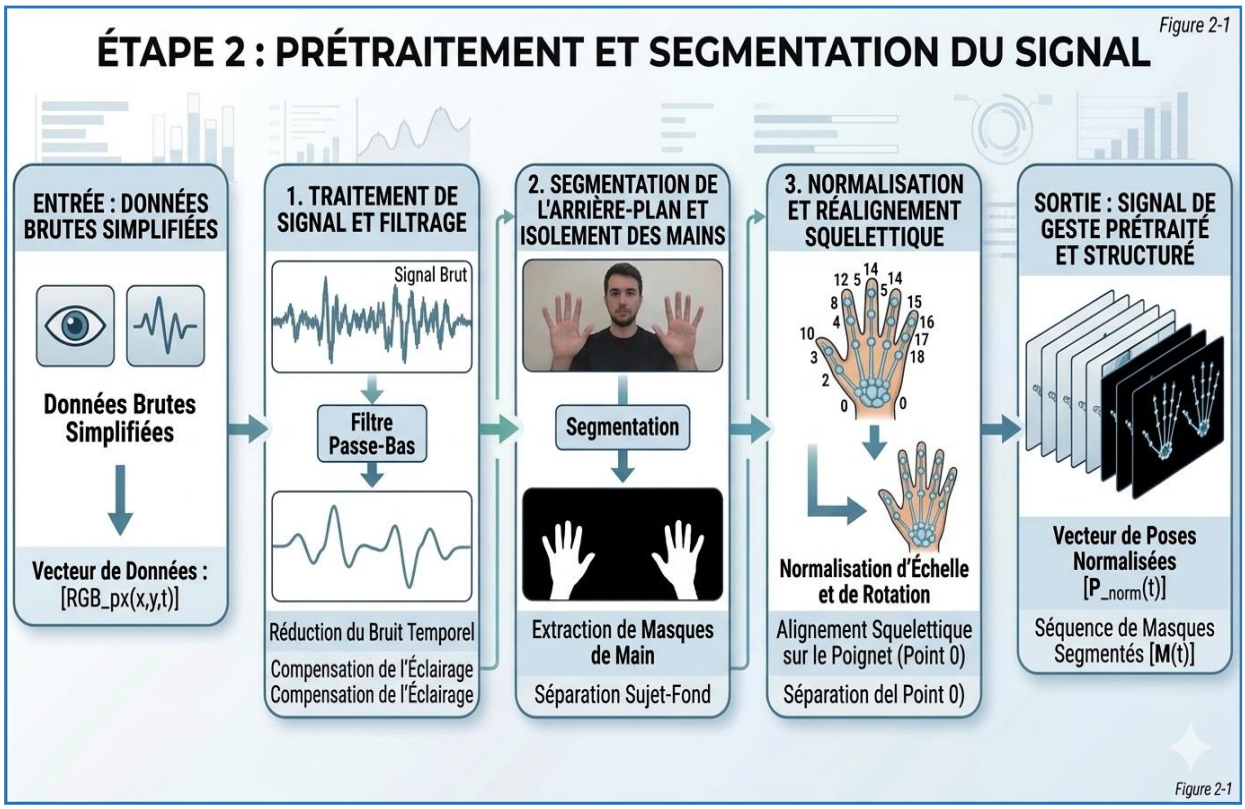


Figure4.2: Figure présente l'étape de prétraitement et segmentation du signal

4.3.3. Extraction des caractéristiques

L'extraction des caractéristiques consiste à transformer les données brutes de la main en informations exploitables par les modèles d'apprentissage. Pour les gestes statiques, le système extrait quatre distances géométriques entre les doigts à partir d'une seule image (1 frame). Pour les gestes dynamiques, il utilise l'évolution temporelle des 21 landmarks sur une séquence de 30 frames afin de capturer le mouvement. Cette étape permet de conserver uniquement les caractéristiques pertinentes, réduisant ainsi la complexité des données tout en améliorant l'efficacité et la rapidité de la reconnaissance des gestes.[15]

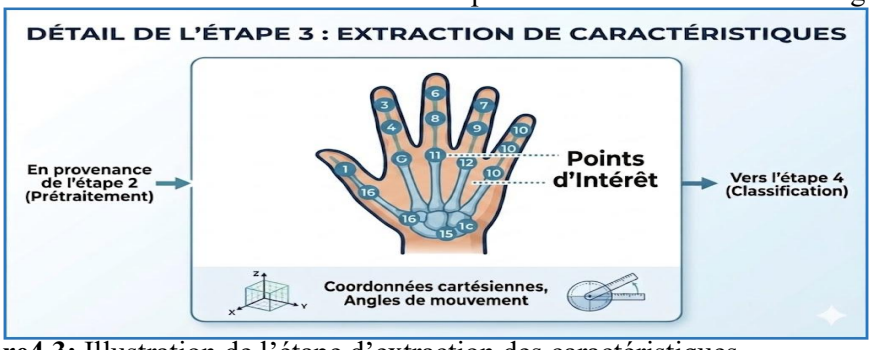


Figure4.3: Illustration de l'étape d'extraction des caractéristiques

4.3.4. Classification et reconnaissance

La classification et la reconnaissance consistent à interpréter les caractéristiques extraites afin d'identifier le geste effectué. Le système s'appuie sur un modèle d'apprentissage automatique pré entraîné pour reconnaître les postures statiques et les mouvements dynamiques. En comparant les données aux modèles appris, il attribue une étiquette au geste avec un niveau de confiance, permettant ainsi de transformer les données numériques en commande interprétable.

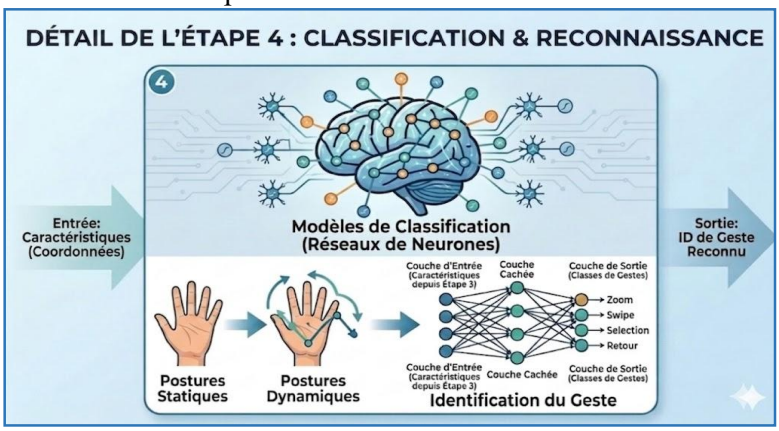


Figure4.4: Illustration de l'étape de classification et reconnaissance

4.3.5. Application & commande

L'étape d'application et de commande consiste à transformer le geste reconnu en une action concrète dans l'interface cible. Elle repose sur un mapping sémantique qui associe chaque ID de geste à une commande spécifique, une logique de contrôle qui valide l'exécution selon le contexte, ainsi que l'envoi de la commande vers l'application via une API (Application Programming Interface). Un retour visuel est ensuite fourni à l'utilisateur pour confirmer l'exécution et assurer une interaction fluide.



Figure4.5: Illustration de l'étape d'application & commande

4.4. Description et gestes reconnues

Dans ce système de contrôle de l'ordinateur par la reconnaissance des gestes de la main, les interactions sont classées en deux grandes catégories : les gestes statiques et les gestes dynamiques.

4.4.1. Gestes statiques

Les gestes statiques correspondent à des positions fixes de la main détectées par le système sans mouvement significatif.

4.4.1.1. Déplacement du curseur (Suivi de l'index)

Ce geste utilise la pointe de l'index levée (seule ou avec le pouce) pour contrôler le curseur de la souris. Le système traduit sa position devant la caméra en mouvements fluides sur l'écran, offrant une navigation précise et intuitive sans entraînement préalable, comme une souris sans contact.

4.4.1.2. Clic Simple (Single Click = Main ouverte)

Le clic simple consiste à ouvrir complètement la main devant la caméra. Cette position est reconnue instantanément pour simuler un clic gauche, offrant une interaction rapide sans mouvement.

4.4.1.3. Clic Double (Double Click = Index et majeur levés)

Le double clic est effectué en levant l'index et le majeur tandis que les autres doigts restent repliés. Cette pose est reconnue pour exécuter deux clics rapides, facilitant l'ouverture de fichiers ou d'applications.

4.4.1.4. Clic Droit (Right Click = Index et auriculaire levés)

Le clic droit consiste à lever l'index et l'auriculaire en formant un « V », les autres doigts repliés. Cette pose est détectée pour afficher le menu contextuel, facilitant l'accès rapide aux options avancées.

4.4.1.5. Ouverture du Clavier Virtuel (Index, majeur et annulaire levés)

Le clavier virtuel est activé en levant l'index, le majeur et l'annulaire. Ce geste est reconnu pour afficher le clavier à l'écran et faciliter la saisie de texte.

4.4.1.6. Lecture/Pause Vidéo (Poing Fermé)

La lecture ou la pause d'une vidéo est contrôlée en formant un poing fermé. Ce geste est reconnu pour basculer entre lecture et pause, offrant un contrôle simple du contenu multimédia.



Figure4.6: Figure représentant des gestes statiques pour le contrôle de l'ordinateur

4.4.2. Gestes Dynamiques

Les gestes dynamiques reposent sur le mouvement de la main dans l'espace devant la caméra. Ils permettent d'exécuter des actions continues ou directionnelles.

4.4.2.1. Défilement Vers le Haut (Scroll Up)

Un mouvement de la main de haut en bas est reconnu comme un défilement vers le haut. Ce geste permet de remonter facilement dans une page ou un document.

4.4.2.2. Défilement Vers le Bas (Scroll Down)

Le défilement vers le bas est déclenché par un mouvement de la main de bas en haut.

4.4.2.3. Glissement Vers la Droite (Slide Right)

Le glissement vers la droite est réalisé par un mouvement de la main de droite à gauche. Ce geste est interprété comme un glissement horizontal pour naviguer entre des diapositives.

4.4.2.4. Glissement Vers la Gauche (Slide Left)

Le glissement vers la gauche est effectué par un mouvement de la main de gauche à droite. Ce geste permettant de revenir en arrière entre les vues.

4.4.2.5. Zoom avant (Zoom In)

Le zoom avant est effectué lorsque le pouce et l'index, en contact, s'éloignent progressivement. Cette augmentation de distance est interprétée pour agrandir l'élément sélectionné de manière progressive.

4.4.2.6. Zoom arrière (Zoom Out)

Le zoom arrière est réalisé lorsque le pouce et l'index, d'abord éloignés, se rapprochent jusqu'à se toucher. Ce geste permet de réduire progressivement la taille du contenu affiché.

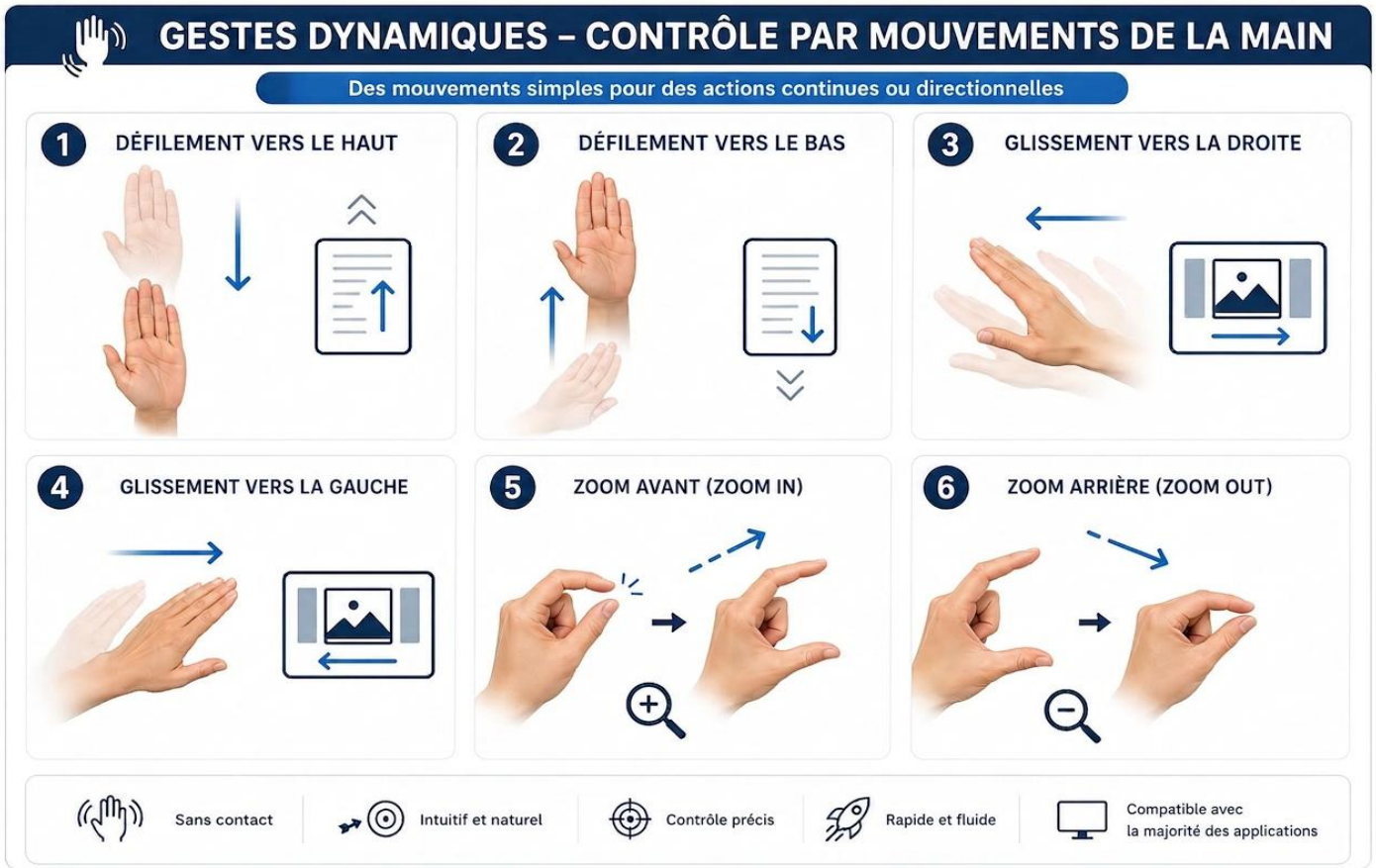


Figure4.7: représentant les gestes dynamiques pour le contrôle de l'ordinateur.

4.5. Implémentation du système

4.5.1. Fonctionnement global du système

Dans un premier temps, la figure ci-dessous illustre le schéma global du système qui montre les grandes étapes de son fonctionnement.

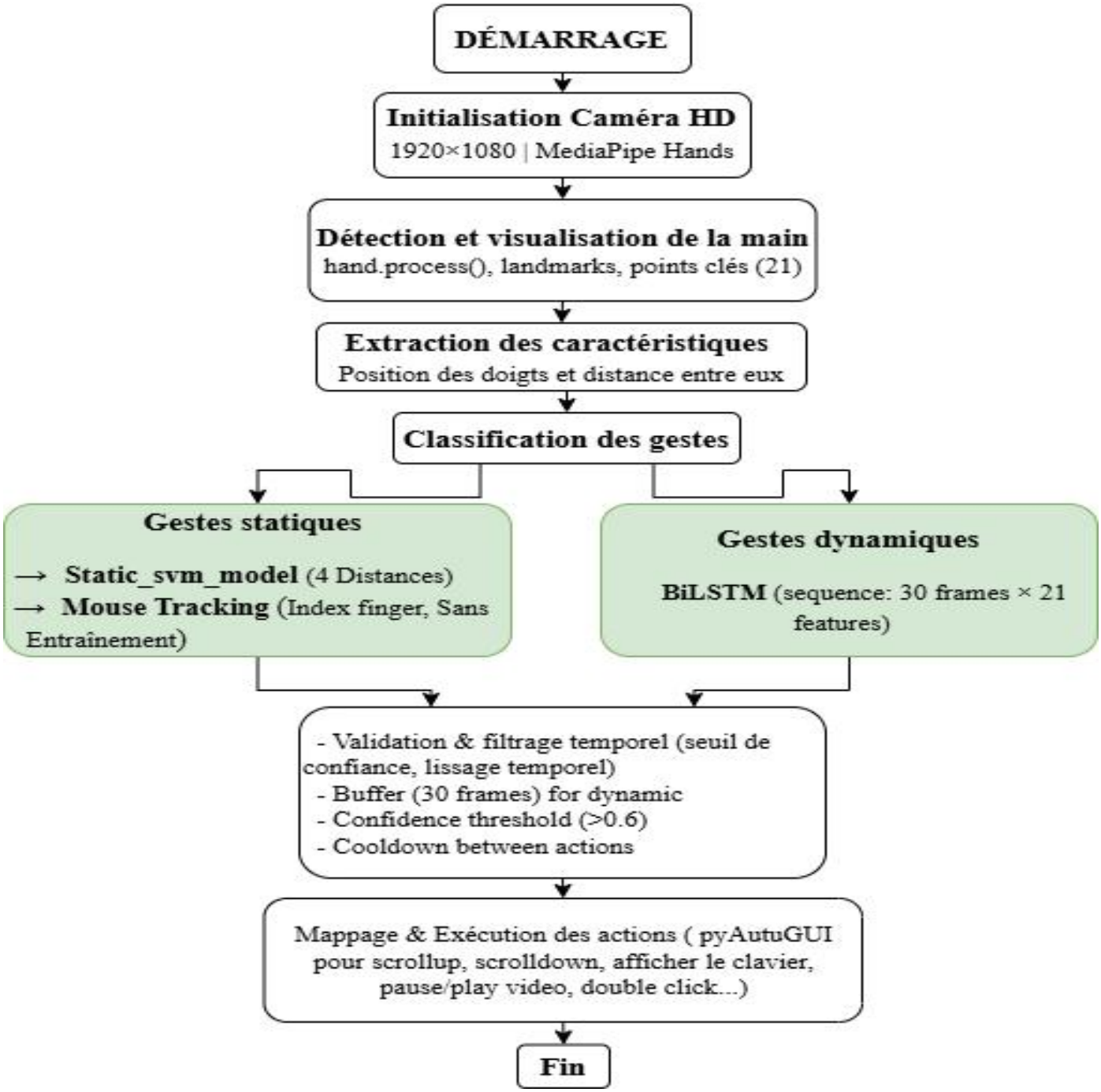


Figure4.8: Schéma global du système proposé pour la reconnaissance de geste de la main.

4.5.1.1. Démarrage et initialisation

Au démarrage, le système lance le flux de traitement en initialisant les modules nécessaires à la reconnaissance des gestes. Cette étape permet de préparer l'environnement d'exécution, de vérifier l'accès à la caméra et de configurer les paramètres de base avant l'analyse des images.

4.5.1.2. Initialisation de la caméra HD(High Definition)

Cette étape consiste à activer la caméra avec une résolution élevée (1920×1080) pour obtenir des images détaillées. Une bonne qualité d'image améliore la détection de la main et réduit les erreurs lors de la reconnaissance des gestes.

4.5.1.3. Détection et visualisation de la main

Dans cette phase, le système détecte la main dans l'image grâce à MediaPipe Hands et affiche les points de repère en temps réel. Ce modèle permet de localiser les articulations de la main et de suivre ses mouvements pour analyser les gestes.[10]

4.5.1.4. Extraction des caractéristiques

Après la détection, le système extrait des caractéristiques comme la position des doigts et les distances entre les points de la main. Ces informations forment un vecteur plus compact et facilitent la reconnaissance des gestes par le classifieur.[1]

4.5.1.5. Classification des gestes

Les caractéristiques extraites sont transmises au module de classification afin d'identifier le geste réalisé. Les gestes statiques sont reconnus à l'aide d'un modèle SVM basé sur les caractéristiques géométriques de la main, tandis que les gestes dynamiques sont analysés sur une séquence temporelle de trames à l'aide d'un modèle BiLSTM (Bidirectional Long Short-Term).

Le déplacement du curseur, sont exécutées directement par un module de suivi de la souris sans nécessiter de phase de prédiction.[11]

4.5.1.6. Validation et filtrage temporel

Après la prédiction, le système applique une validation temporelle afin de limiter les fausses détections. Cette étape inclure un seuil de confiance, un lissage sur plusieurs trames (frales), une mémoire tampon (buffer) temporel pour les gestes dynamiques, ainsi qu'un temps de pause entre deux actions pour éviter les déclenchements répétés.[12]

4.5.1.7. Mappage et exécution des actions

Une fois le geste validé, le système le transforme en action concrète via un module de commande pyAutoGUI. Cela permet d'associer chaque geste à une opération précise, par exemple le défilement, l'affichage de la souris, la touche espace, la lecture/pause d'une vidéo ou un double clic.

4.5.1.8. Fin du traitement

La phase finale consiste à interpréter le geste, exécuter l'action associée, puis revenir automatiquement à l'état d'attente pour traiter les trames suivantes en temps réel. Lorsque l'utilisateur appuie sur ESC, le système se ferme, la boucle s'arrête, la caméra est libérée et les fenêtres OpenCV sont fermées.

Le système repose sur plusieurs étapes générales, et les gestes statiques et dynamiques sont traités séparément selon un même processus basé sur la collecte des données, l'entraînement du modèle et son utilisation pour la prédiction.

4.5.2. Branche des gestes statiques

4.5.2.1. Collecte de données

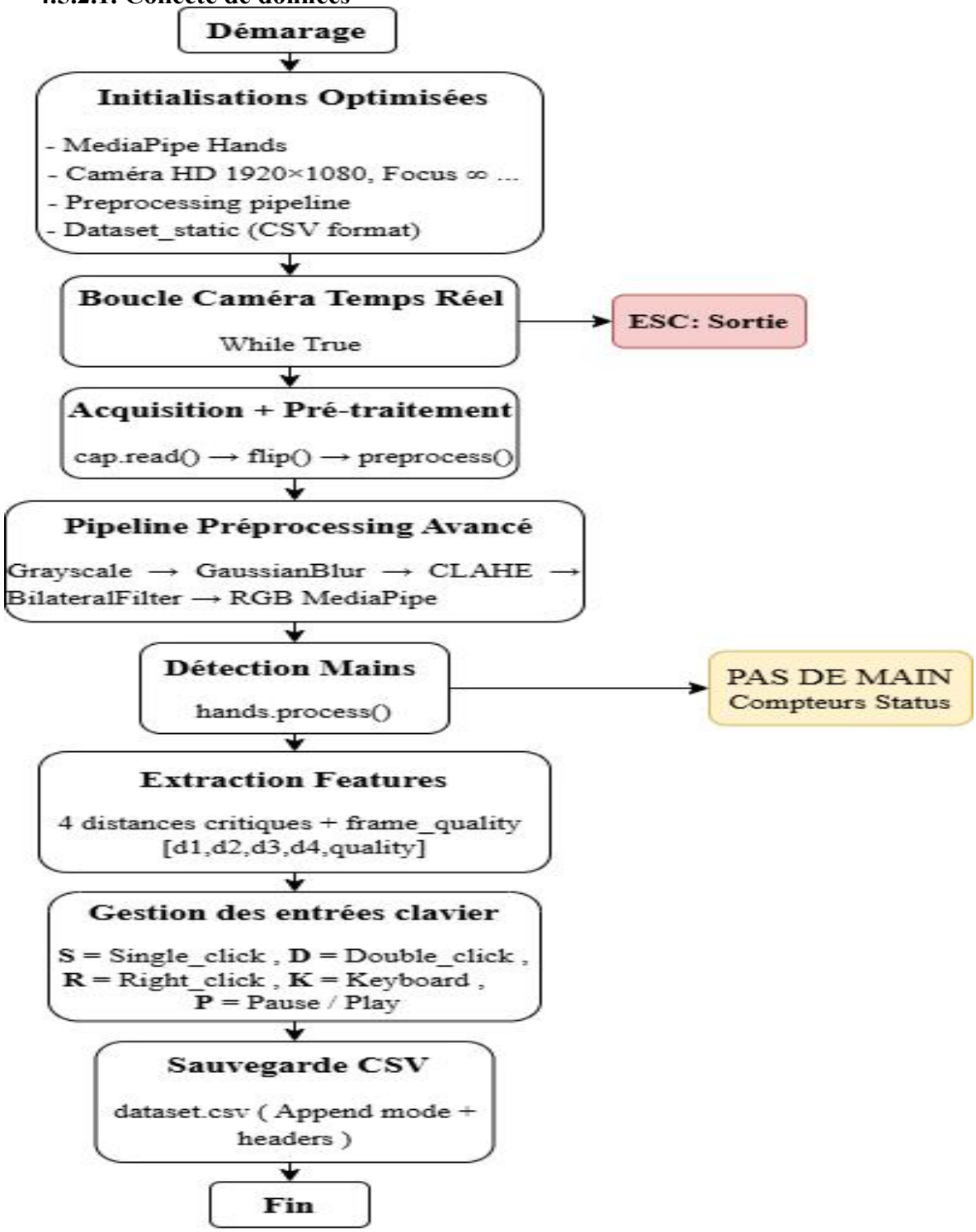


Figure4.9: Schéma de collecte des données statiques.

La collecte des données des gestes statiques débute par l'initialisation de l'environnement logiciel, de la caméra HD et du module MediaPipe Hands. Les images sont capturées en temps réel puis soumises à un prétraitement afin d'améliorer leur qualité avant la détection des landmarks de la main.

À partir des points détectés, le système extrait plusieurs caractéristiques géométriques représentatives de la posture de la main. Chaque échantillon est ensuite associé à une étiquette (label) correspondant au geste réalisé. L'étiquetage est effectué à l'aide du clavier : la touche p correspond au geste Pause, s à Single Click, d à Double Click, r à Right Click et k à Keyboard. Lorsqu'une touche est pressée alors qu'une main est détectée, les caractéristiques extraites sont enregistrées avec leur classe associée.

Les données annotées sont ensuite sauvegardées dans un fichier CSV afin de constituer une base de données supervisée destinée à l'entraînement du modèle SVM. À la fin de la collecte, un bilan statistique du nombre d'échantillons enregistrés pour chaque classe est généré afin de vérifier l'équilibre du jeu de données.

4.5.2.2. Entraînement du modèle de classification statique

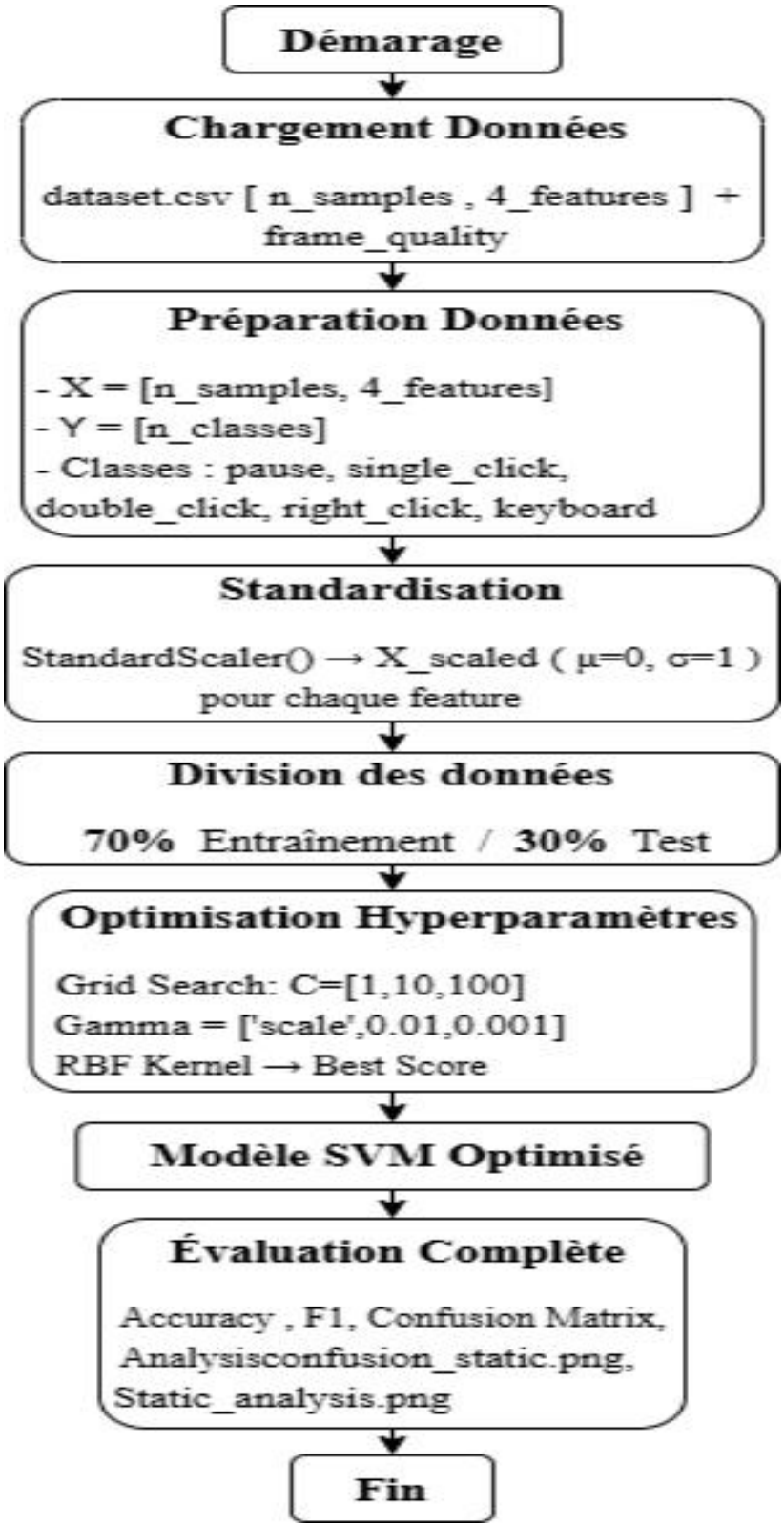


Figure 4.10: Schéma d' entraînement du modèle statique.

L'entraînement du modèle statique commence par le chargement des données collectées et stockées dans des fichiers CSV. Chaque échantillon est représenté par quatre caractéristiques géométriques extraites à partir des landmarks de la main, auxquelles s'ajoute un indicateur de qualité de l'image. Les étiquettes associées aux différents gestes constituent les classes cibles du modèle.

Avant l'apprentissage, les données sont standardisées à l'aide de StandardScaler afin d'harmoniser l'échelle des variables. Le jeu de données est ensuite divisé en deux ensembles : 70 % pour l'entraînement et 30 % pour le test, avec une stratification des classes afin de préserver leur distribution.

La classification est réalisée à l'aide d'un SVM à noyau RBF (Radial Basis Functin kernel),[17] dont les hyperparamètres principaux C et γ (gamma) sont optimisés par une recherche par grille (Grid Search) afin d'obtenir la meilleure configuration.[18] Le modèle est ensuite évalué à l'aide de plusieurs métriques, notamment l'accuracy, la précision, le rappel et le score F1, ainsi qu'une matrice de confusion permettant d'analyser les performances de classification. Enfin, le modèle entraîné et le scaler sont sauvegardés pour être utilisés lors de la reconnaissance des gestes en temps réel.

✧ **Modèle SVM optimisé**

Ce choix est justifié par le fait que les données ne sont pas linéairement séparables, comme dans de nombreux systèmes de reconnaissance des gestes. Le noyau RBF permet de projeter les données dans un espace de dimension supérieure afin de construire des frontières de décision non linéaires, améliorant ainsi la séparation entre les classes et la marge. Les SVM restent également une méthode robuste pour ce type de problème, offrant un bon équilibre entre précision, généralisation et complexité.[19]

❖ **Formulation mathématique du modèle SVM**

1. Extraction → [4 distances doigts] ∈ $\mathbb{R}^4$
- $\mathbf{x \in R4}$ représente le vecteur de caractéristiques extrait à partir des distances entre les doigts.

2. Standardisation → X_scaled ~ N(0,1)

- · La standardisation permet de normaliser les données pour éviter qu'une caractéristique domine les autres.
  3. Noyau RBF → $\phi:\mathbb{R}^4 \rightarrow \mathbb{R}\infty$ (non-linéaire)
- · La fonction φ correspond à la transformation implicite effectuée par le noyau RBF afin de rendre les données séparables.
  4. Optimisation → $\max(1/||w||) \mid y_i(w\cdot\phi(x_i)+b) \geq 1$
- · L'expression d'optimisation vise à maximiser la marge entre les classes tout en respectant les contraintes de classification.
  5. Vecteurs Supports → $\alpha_i > 0$ définissent frontière
- · Les coefficients α_i permettent d'identifier les vecteurs de support, qui sont les points critiques définissant la frontière de décision.
  6. Prédiction → $f(x)=\text{sign}(\sum\alpha_i y_i K(x_i,x)+b)$
- Enfin, la fonction f(x) est utilisée pour prédire la classe d'un nouveau geste.

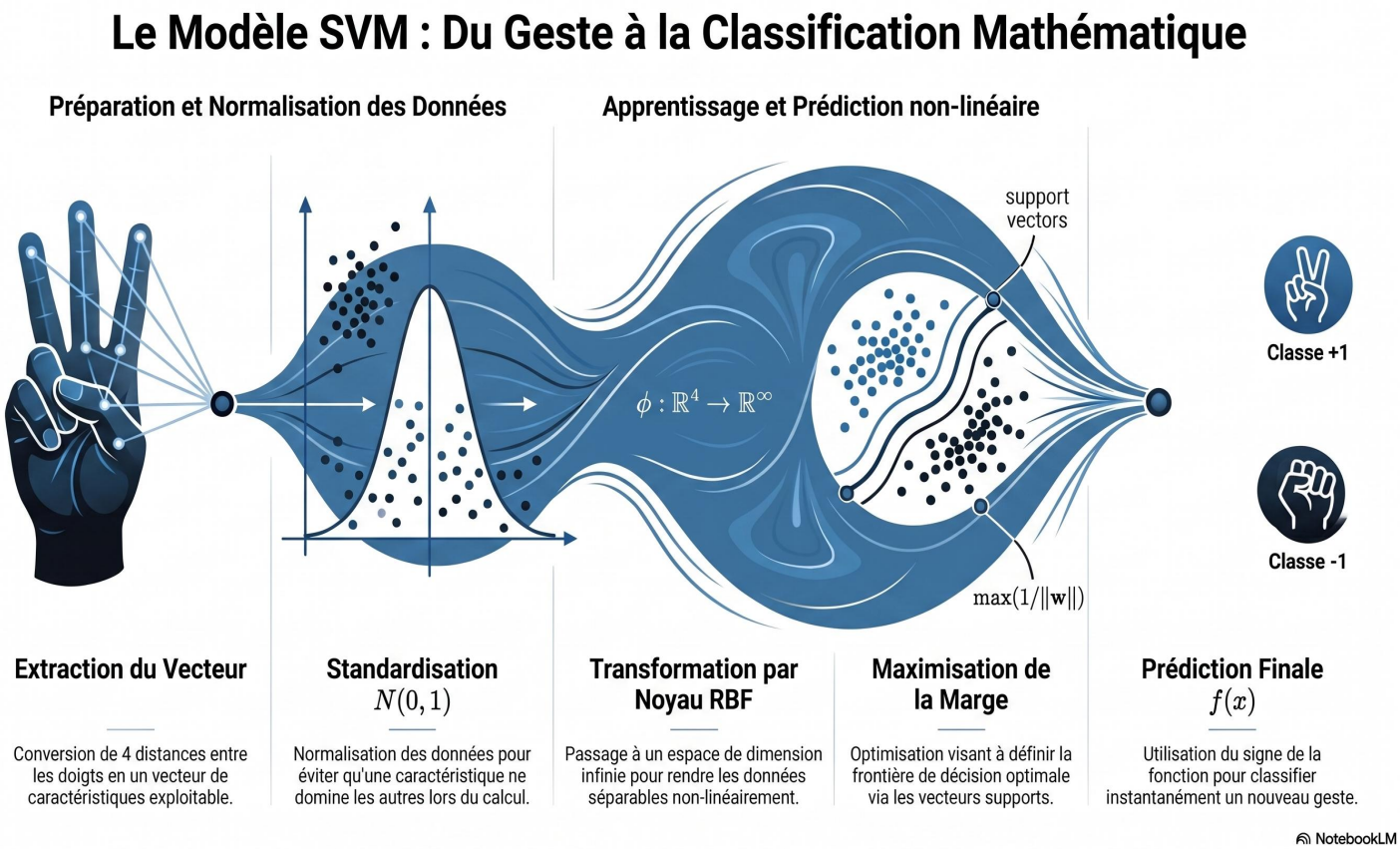


Figure 4.11: Figure représentant la classification par SVM

Avantages SVM pour Gestes Statiques:

- ✓ Interprétable (hyperplan clair),
- ✓ Pas de séquence temporelle requise,
- ✓ Efficace sur 4D → temps réel,
- ✓ Robustesse aux variations éclairage. [20]

4.5.2.3. Validation du modèle sur des données réelles en temps réel

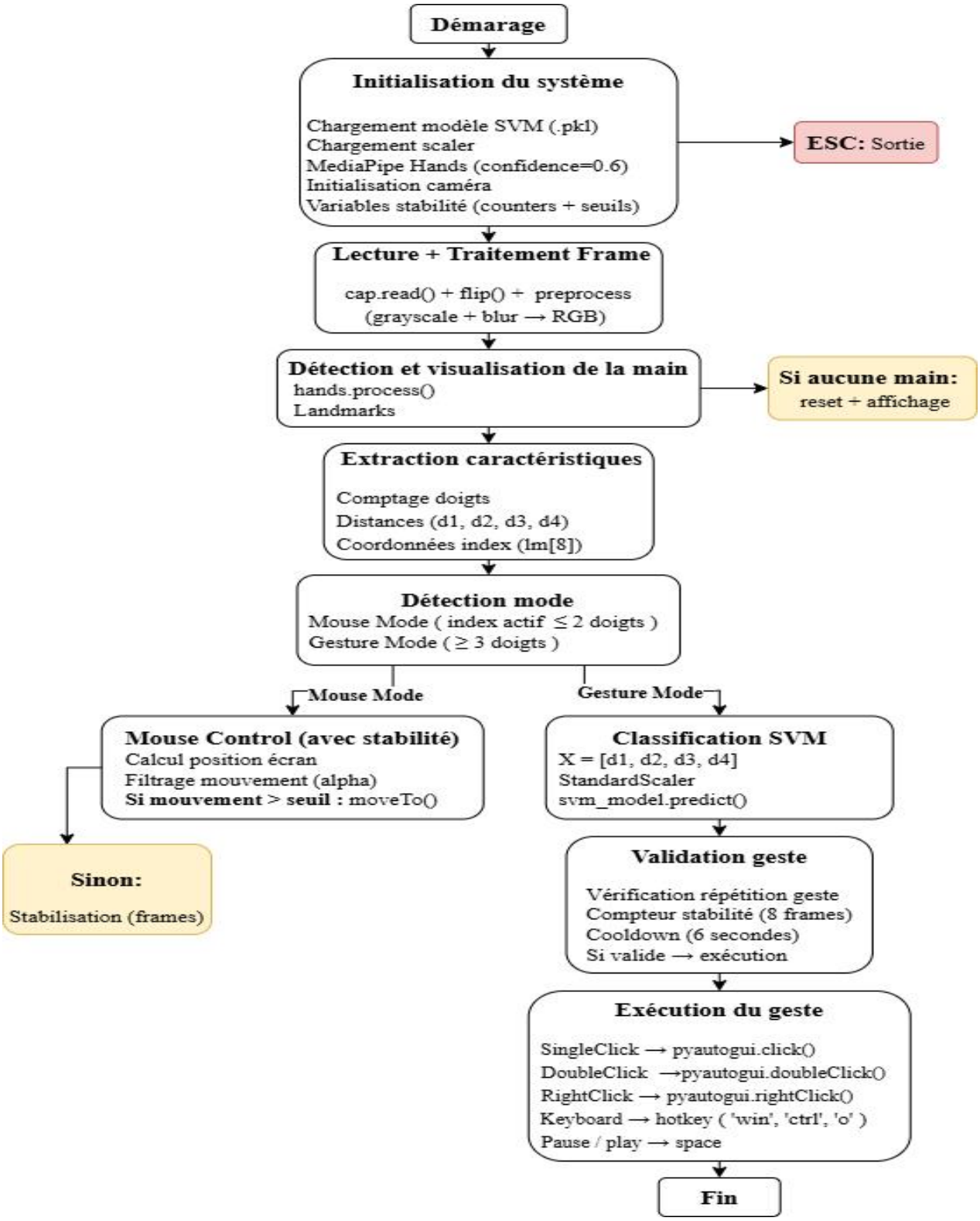


Figure 4.12: Figure illustrant l'évaluation du modèle statique en temps réel.

Après l'entraînement, le modèle SVM est intégré dans une application de reconnaissance gestuelle fonctionnant en temps réel. Le système charge le modèle entraîné ainsi que le scaler, puis capture les images provenant de la caméra. Les landmarks de la main sont extraits à l'aide de MediaPipe Hands et transformés en caractéristiques géométriques identiques à celles utilisées lors de l'apprentissage.

Selon le nombre de doigts détectés, le système bascule entre le mode de contrôle du curseur et le mode de reconnaissance gestuelle. Dans ce dernier cas, les caractéristiques sont normalisées puis transmises au classifieur SVM afin d'identifier le geste réalisé. Une étape de validation temporelle basée sur la stabilité de la prédiction et un délai de temporisation (cooldown) permet de réduire les déclenchements involontaires. Une fois le geste validé, l'action correspondante est exécutée à l'aide de PyAutoGUI, assurant ainsi une interaction fluide et fiable dans des conditions réelles d'utilisation.

4.5.3. Branche des gestes dynamiques

4.5.3.1. Collecte de données

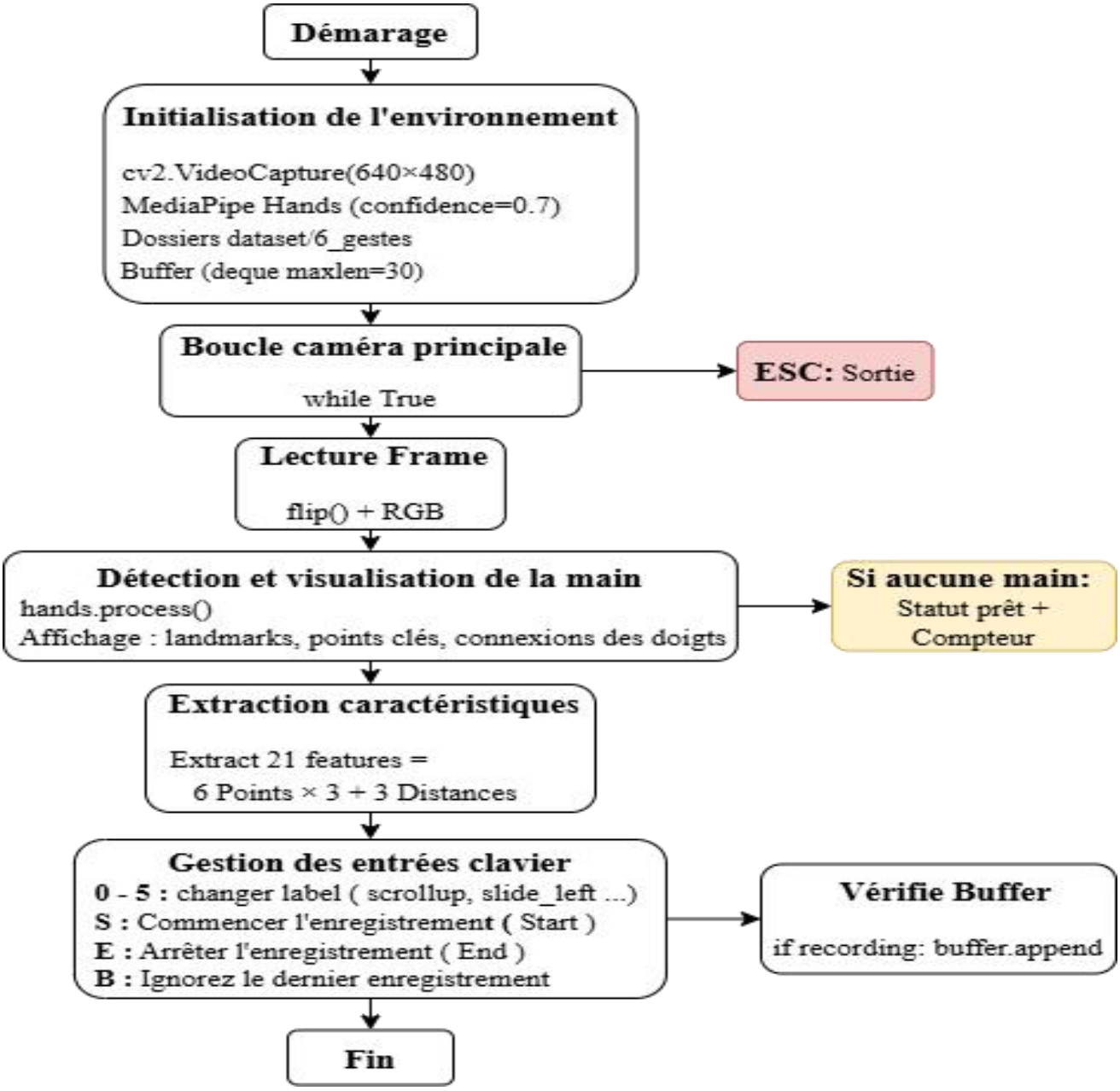


Figure 4.13 : Figure illustrant le processus de collecte des données dynamiques.

La collecte des gestes dynamiques débute par l'initialisation de l'environnement de travail, de la caméra et du module MediaPipe Hands. Les images sont capturées en temps réel puis analysées afin d'extraire les landmarks de la main. Pour chaque trame, le système sélectionne six points clés (le poignet et l'extrémité des cinq doigts) et récupère leurs coordonnées (x, y, z). Trois distances inter-doigts sont également calculées afin de mieux représenter la configuration de la main. L'ensemble forme un vecteur de 21 caractéristiques, décrivant à la fois la position spatiale et la posture de la main.

Contrairement aux gestes statiques, les gestes dynamiques sont représentés par des séquences temporelles. Le système utilise donc un buffer de 30 frames consécutives permettant de capturer l'évolution complète du mouvement. L'utilisateur contrôle la collecte à l'aide du clavier : les touches 0 à 5 permettent de sélectionner la classe du geste à enregistrer (**0** = Défilement, **1** = Défilement vers le bas, **2** = Glissement vers le haut, **3** = Glissement vers le bas, **4** = Zoom avant, **5** = Zoom arrière), **S** démarre l'enregistrement, **E** arrête et sauvegarde la séquence, tandis que **B** supprime la dernière séquence en cas d'erreur.

Une fois la séquence validée, les caractéristiques sont lissées puis enregistrées sous forme de fichiers .npy afin de constituer une base de données supervisée destinée à l'entraînement du modèle BiLSTM. À la fin de la collecte, les ressources sont libérées et un bilan du nombre de séquences enregistrées est affiché.

4.5.3.2. Entraînement du modèle de la classification dynamique

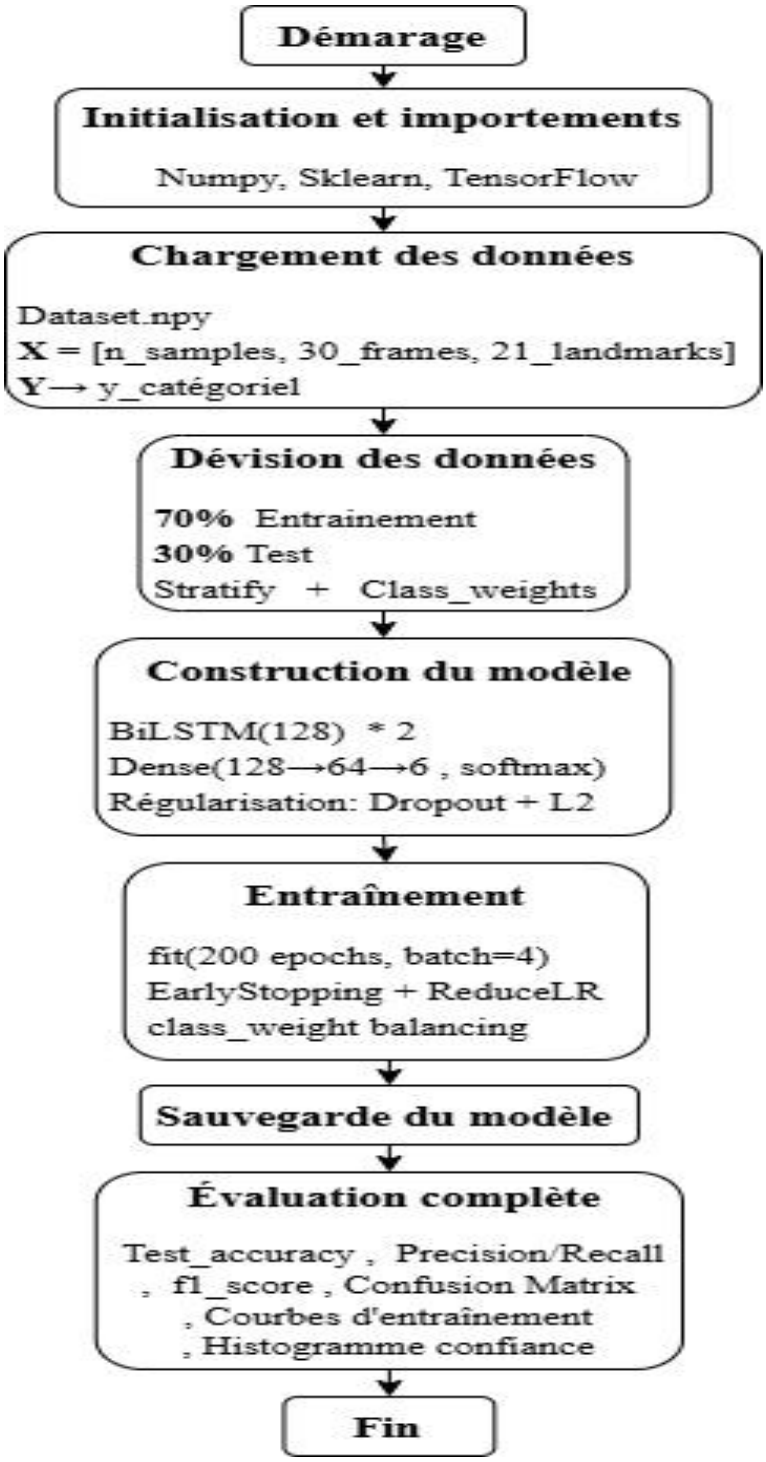


Figure 4.14: Figure illustrant le processus de collecte des données dynamiques.

L'entraînement du modèle dynamique repose sur une architecture BiLSTM destinée à l'apprentissage des dépendances temporelles des gestes. Les données utilisées sont chargées à partir de fichiers .npy, où chaque échantillon est représenté par une séquence de 30 frames et 21 landmarks de la main, avec des étiquettes encodées en one-hot pour la classification multi-classes.

Le jeu de données est divisé en 70 % pour l'entraînement et 30 % pour le test, avec une stratification des classes afin de préserver leur distribution.[26] Des poids de classes (class weights) sont également appliqués pour compenser le déséquilibre éventuel des données.

Le modèle est construit avec deux couches BiLSTM, suivies de couches Dropout pour réduire le surapprentissage et de couches Dense (128 → 64 → 6) pour la classification finale. Une régularisation L2 est utilisée pour limiter la complexité du modèle, et la couche Softmax permet de générer des probabilités pour chaque classe.[27]

L'entraînement est réalisé sur 200 epochs avec un batch size de 4, en utilisant les techniques EarlyStopping et ReduceLROnPlateau afin d'améliorer la convergence. Enfin, le modèle est évalué à l'aide de l'accuracy, la précision, le rappel et le F1-score, ainsi que d'une matrice de confusion et des courbes d'apprentissage. Le modèle entraîné est sauvegardé au format .keras pour une utilisation en temps réel.[28]

❖ **Architecture du modèle BiLSTM**

Entrée : [30 pas de temps × 21 caractères] ↓

Représente une séquence temporelle de 30 trames, où chaque trame contient 21 caractéristiques (points de repère de la main"landmarks"). Cette structure permet de capturer l'évolution du geste dans le temps.

BiLSTM (128) → [30 × 256] ↓

Une couche Bidirectional LSTM avec 128 unités dans chaque direction (avant et arrière). Elle produit 256 caractéristiques (128 × 2) pour chaque timestep, ce qui permet de capturer les dépendances temporelles dans les deux sens.

Dropout (0.5) ↓

Technique de régularisation qui désactive aléatoirement 50 % des neurones pendant l'entraînement afin de réduire le surapprentissage (overfitting).

BiLSTM (128) → [256] ↓

Deuxième couche BiLSTM (128 unités dans chaque direction, soit 256 caractéristiques en sortie) qui résume toute la séquence temporelle en un vecteur unique représentant les informations les plus pertinentes du geste.

Dropout (0.5) ↓

Permet de renforcer la généralisation du modèle en évitant la dépendance excessive à certaines neurones.

Dense (128, ReLU + L2) → [128] ↓

Couche entièrement connectée qui transforme les caractéristiques extraites.

- ReLU (Rectified Linear Unit) : introduit de la non-linéarité.
- Régularisation L2 : pénalise les poids élevés pour éviter le surapprentissage.

Dropout (0.4) ↓

Réduction supplémentaire du surapprentissage avec un taux de 40 %.

Dense (64, ReLU) → [64] ↓

Réduction progressive de la dimension des données afin de concentrer l'information utile pour la classification.

Dense (6, Softmax) → [6 classes] ↓

Couche de sortie qui produit une distribution de probabilité sur les 6 gestes. La fonction Softmax permet de choisir la classe la plus probable.

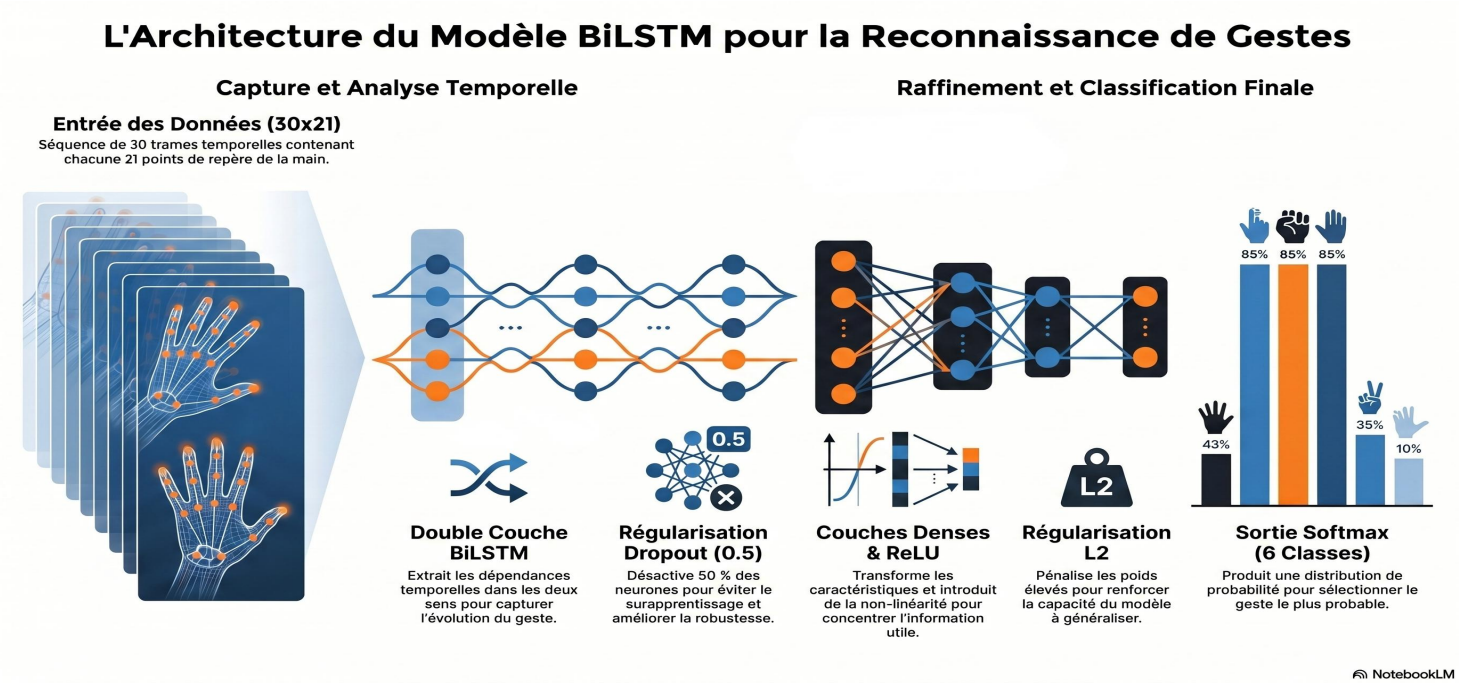


Figure 4.15: Figure illustrant le principe de fonctionnement du modèle LSTM

- **Avantages de l'architecture BiLSTM**

  - ✓ Capture bidirectionnelle des dépendances temporelles,
  - ✓ Robustesse aux variations de vitesse gestuelle,
  - ✓ Traitement efficace des séquences de longueur fixe,
  - ✓ Régularisation multiple contre l'overfitting.[30]

4.5.3.3. Validation du modèle sur des données réelles en temps réel

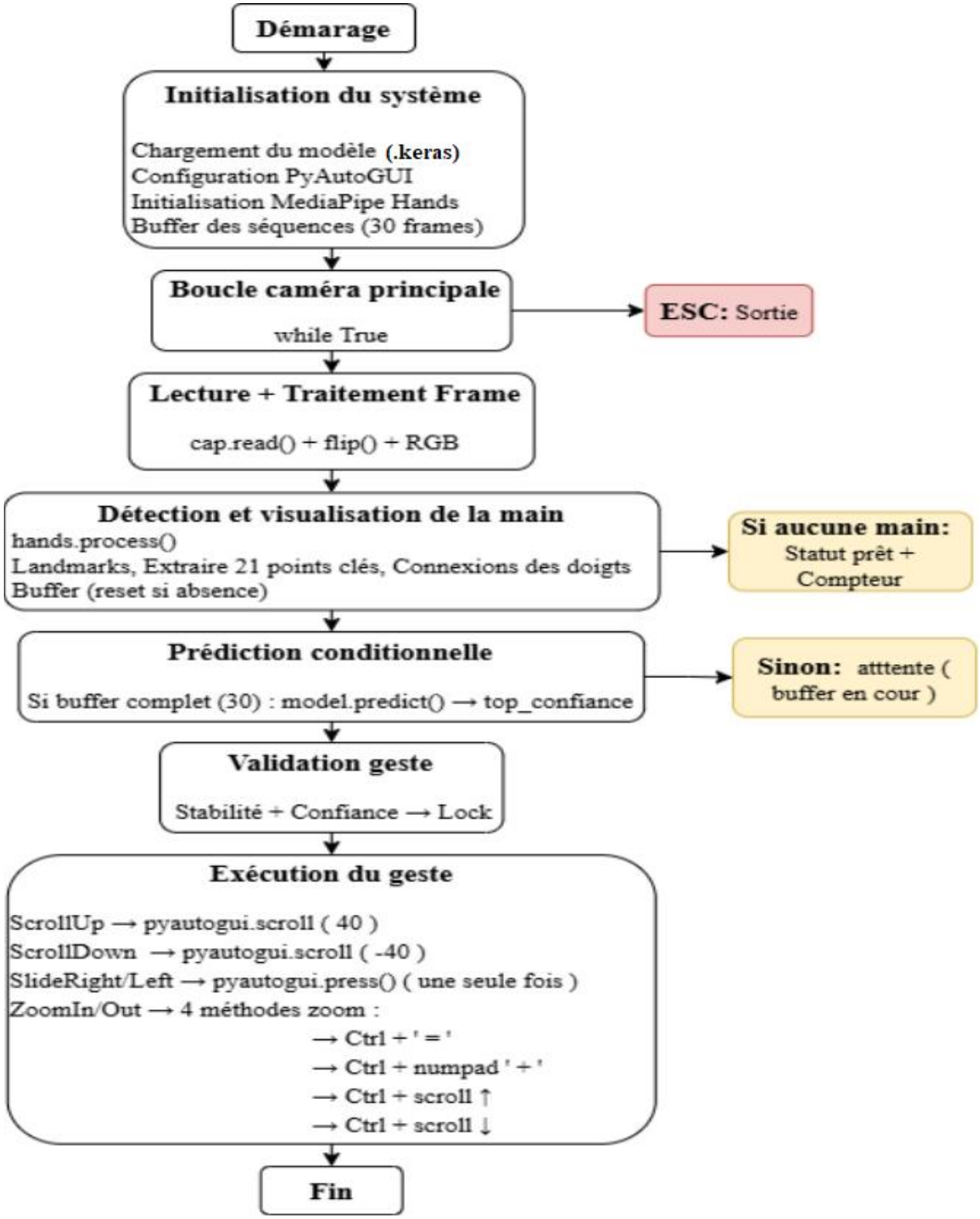


Figure 4.16: Figure illustrant l'évaluation du modèle dynamique en temps réel.

La validation du modèle dynamique s'effectue en temps réel à travers une boucle continue de capture vidéo.[21] Le modèle BiLSTM préalablement entraîné est chargé en mémoire, tandis que MediaPipe Hands est utilisé pour extraire les 21 landmarks de la main. Les caractéristiques sont calculées pour chaque frame et stockées dans un buffer temporel de 30 frames afin de former une séquence complète représentant un geste dynamique.

Une fois la séquence complète disponible, elle est envoyée au modèle qui retourne une distribution de probabilités. Le geste est validé uniquement si la confiance dépasse un seuil défini (60 %) et si la stabilité de la prédiction est assurée à travers un mécanisme de verrouillage temporel afin d'éviter les oscillations ou les fausses détections.

Après validation, le geste est converti en action système à l'aide de PyAutoGUI, permettant l'exécution de commandes telles que le défilement, la navigation ou le zoom. Un mécanisme de contrôle (cooldown et compteur interne) est utilisé pour éviter les répétitions excessives et garantir une interaction fluide en temps réel.

4.6. Conclusion

Ce chapitre a présenté la conception et la réalisation du système de contrôle d'un ordinateur par gestes de la main. Il a permis de détailler l'ensemble de la chaîne de traitement, depuis l'acquisition des données vidéo jusqu'à l'exécution des commandes en temps réel.

Dans un premier temps, l'architecture globale du système a été décrite en mettant en évidence les différentes étapes du pipeline, notamment la capture des images, la détection de la main, l'extraction des caractéristiques, ainsi que la classification des gestes statiques et dynamiques. Ensuite, les besoins fonctionnels et non fonctionnels ont été définis afin de garantir un système performant, fluide et adapté à une utilisation en temps réel.

Par ailleurs, les deux approches de reconnaissance ont été présentées : les gestes statiques basés sur un modèle SVM et les gestes dynamiques basés sur un réseau BiLSTM. Chaque branche a été accompagnée des phases de collecte de données, d'entraînement et de validation en conditions réelles. Enfin, l'intégration du système avec des outils de contrôle tels que PyAutoGUI a permis de transformer les gestes reconnus en actions concrètes sur l'ordinateur.

Ainsi, ce chapitre constitue la mise en œuvre pratique du système proposé, en validant l'efficacité de l'approche hybride adoptée pour la reconnaissance gestuelle en temps réel.

Chapitre 5

Expérimentation et résultats

Chapitre 5: Expérimentation et résultats

5.1. Introduction

Après la conception et l'implémentation du système de reconnaissance gestuelle, il est essentiel d'évaluer ses performances afin de vérifier sa capacité à reconnaître correctement les gestes statiques et dynamiques en temps réel. Dans ce chapitre, nous présentons le protocole expérimental ainsi que les jeux de données utilisés pour l'apprentissage et l'évaluation des modèles. Nous décrivons également les métriques de performance retenues pour mesurer la qualité de la reconnaissance. Les résultats obtenus par les modèles SVM et BiLSTM sont ensuite analysés à l'aide de plusieurs indicateurs, notamment l'accuracy, la précision, le rappel, le F1-score et les matrices de confusion. Une comparaison entre les deux approches est réalisée afin de mettre en évidence leurs performances respectives. Enfin, nous discutons les avantages et les limites du système proposé avant de présenter son interface graphique et quelques exemples d'utilisation en conditions réelles.

5.2. Protocole expérimental

Dans cette phase expérimentale, nous avons évalué les performances du système intelligent de reconnaissance des gestes développé dans le cadre de ce projet.

Le système proposé repose sur deux modèles de classification complémentaires entraînés sur des données collectées manuellement :

un modèle SVM dédié à la reconnaissance des gestes statiques à partir d'images ,

un modèle BiLSTM utilisé pour la reconnaissance des gestes dynamiques à partir de séquences vidéo composées de 30 trames.

L'implémentation du système a été réalisée en Python en utilisant plusieurs **bibliothèques spécialisées** telles que :

- OpenCV pour le traitement vidéo,[31]
- MediaPipe pour la détection et l'extraction des points clés de la main , [32]
- TensorFlow/Keras pour le modèle BiLSTM ,

- Scikit-learn pour le modèle SVM ,[33]
- Math (math.hypot) pour le calcul des distances euclidiennes entre les points clés de la main,[34]
- OS pour la gestion des chemins du dataset et l'organisation des données,[35]
- Collections (deque) pour la gestion des files de mémoire temporelle (buffer des séquences),[36]
- PyAutoGUI pour l'exécution des commandes associées aux gestes reconnus.[36]

Le protocole expérimental suit les étapes suivantes :

- Acquisition des images et des vidéos en temps réel à partir de la caméra.
- Détection de la main et extraction des landmarks grâce à MediaPipe.
- Prétraitement des données extraites avant leur transmission au modèle approprié.
- Utilisation du modèle SVM pour reconnaître les gestes statiques.
- Utilisation du modèle BiLSTM pour reconnaître les gestes dynamiques basés sur une séquence temporelle de frames.
- Validation de la prédiction afin d'améliorer la stabilité et de réduire les fausses détections.
- Exécution automatique de l'action correspondante après reconnaissance du geste.

Les expériences ont été réalisées dans différentes conditions d'éclairage et avec plusieurs utilisateurs afin d'évaluer la robustesse, la précision et la stabilité du système en temps réel.

5.3. Jeux de données utilisées

Pour entraîner et évaluer le système proposé, nous avons construit notre propre jeu de données dédié à la reconnaissance des gestes statiques et dynamiques.

La collecte des données a été réalisée à l'aide d'un programme développé en Python permettant d'enregistrer automatiquement les caractéristiques de la main détectée par la caméra.

5.3.1. Jeu de données des gestes statiques

Pour les gestes statiques, le système capture une image de la main puis extrait les coordonnées et les distances entre les 21 points clés (landmarks) détectés par MediaPipe.

Les caractéristiques extraites sont ensuite sauvegardées sous forme de fichiers CSV.

Les gestes statiques utilisés sont les suivants :Single Click, Double Click, Right Click, Keyboard, Pause (lire).

5.3.2. Jeu de données des gestes dynamiques

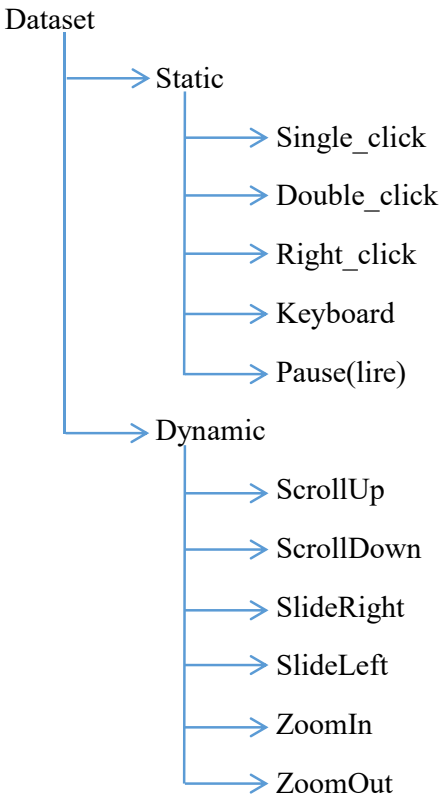
Pour les gestes dynamiques, le système enregistre des séquences vidéo composées de 30 frames consécutives.

Pour chaque frame, les caractéristiques de la main sont extraites puis stockées sous forme de tableaux NumPy (.npy).

Les gestes dynamiques utilisés sont : Scroll Up, Scroll Down, Slide Right, Slide Left, Zoom In, Zoom Out.

5.3.3. Organisation du jeu de données

Le dataset a été organisé selon la structure suivante :



5.3.4. Taille et diversité des données

Afin d'améliorer la robustesse du système, nous avons collecté environ 200 échantillons pour chaque geste dans différentes conditions :

- variation de distance par rapport à la caméra ,
- différentes tailles de main ,
- conditions d'éclairage variées ,
- variations naturelles des mouvements.

Le dataset contient au total environ 2200 échantillons représentant les différentes catégories des gestes.

Cette diversité permet d'améliorer les capacités de généralisation du système et de réduire les erreurs de reconnaissance dans des situations réelles d'utilisation.

5.4. Métriques d'évaluation

Afin d'évaluer les performances du système proposé, plusieurs métriques d'évaluation ont été utilisées pour mesurer la qualité de la reconnaissance des gestes statiques et dynamiques.

Ces métriques permettent d'analyser la précision du système ainsi que sa capacité à reconnaître correctement les différents gestes réalisés par l'utilisateur.

5.4.1. Taux de reconnaissance (Accuracy)

Le taux de reconnaissance représente le pourcentage des gestes correctement reconnus par le système par rapport au nombre total de tests effectués.[38]

Il est calculé à l'aide de la formule suivante :

$$Accuracy = \frac{Nombre\ de\ prédictions\ corrects}{Nombre\ total\ de\ prédictions} * 100$$

Cette métrique permet d'évaluer les performances globales du système.

5.4.2. Précision (Precision)

La précision mesure la capacité du modèle à produire des prédictions correctes pour une classe donnée.[38]

Elle est définie par la formule suivante :

$$Précision = \frac{Vrais\ Positifs}{Vrais\ Positifs + Faux\ Positifs}$$

Une précision élevée signifie que le système génère peu de fausses détections.

5.4.3. Rappel (Recall)

Le rappel mesure la capacité du système à détecter correctement les gestes réellement présents.[38]

Il est calculé comme suit :

$$Recall = \frac{Vrais\ Positifs}{Vrais\ Positifs + Faux\ Négatifs}$$

Un rappel élevé indique que le système reconnaît correctement la majorité des gestes effectués.

5.4.4. F1-Score

Est une mesure de performance qui combine la précision (Précision) et le rappel (Recall) en une seule métrique. Il représente la moyenne harmonique de ces deux mesures et permet d'évaluer l'équilibre entre la capacité du modèle à éviter les fausses prédictions positives et à détecter correctement les instances positives. Une valeur de F1-score proche de 1 indique une excellente performance du modèle.[39]

$$F1 - Score = 2 * \frac{Précision * Recall}{Précision + Recall}$$

5.5. Analyse des résultats

Afin d'analyser les performances du système proposé, plusieurs expériences ont été réalisées sur les modèles SVM et BiLSTM entraînés sur notre jeu de données.

Les résultats obtenus montrent une bonne capacité de reconnaissance des gestes statiques et dynamiques avec un taux de précision élevé et une stabilité satisfaisante en temps réel.

5.5.1. Résultats du modèle SVM

5.5.1.1. Performance globale

Accuracy	Précision	Recall	F1-Score
97.3	97.5	97.3	97.4

Tableau 5.1: Tableau des performances du modèle SVM (Accuracy, Précision, Rappel et F1-Score)

Les résultats globaux obtenus par le modèle SVM montrent des performances très satisfaisantes. En effet, le modèle a atteint une accuracy de 97,3 %, ce qui signifie qu'il a

correctement classifié la majorité des échantillons de test. Les valeurs élevées de précision (97,5 %), de rappel (97,3 %) et de F1-score (97,4 %) confirment la capacité du modèle à reconnaître efficacement les différents gestes statiques tout en limitant les erreurs de classification. Ces résultats démontrent la pertinence des caractéristiques extraites ainsi que l'efficacité du classifieur SVM pour l'application de contrôle d'ordinateur basée sur les gestes de la main.

5.5.1.2. Matrice de confusion

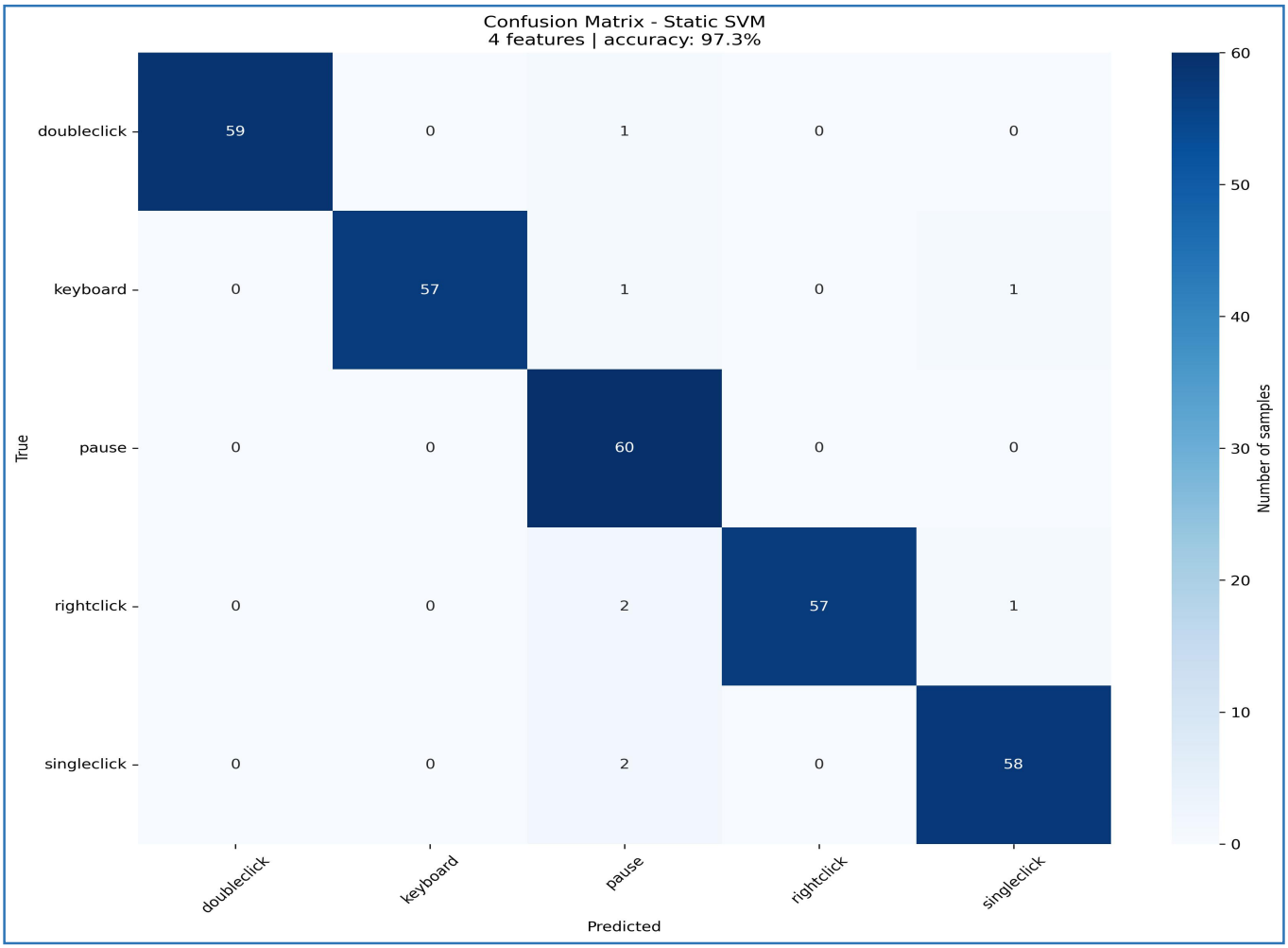


Figure 5.1: Matrice de confusion du modèle SVM

La matrice de confusion montre que le modèle SVM obtient une excellente performance globale avec une exactitude (Accuracy) de 97,3 %. La majorité des échantillons sont correctement classifiés, comme l'indique la forte concentration des valeurs sur la diagonale principale.

La classe « Pause » présente les meilleurs résultats avec 60 échantillons correctement reconnus sur 60, ce qui correspond à un taux de reconnaissance de 100 %. Les gestes « Double Click », « Keyboard », « Right Click » et « Single Click » affichent également des performances élevées avec respectivement 59, 57, 57 et 58 prédictions correctes.

Les erreurs observées restent limitées. Un échantillon de la classe « Double Click » a été confondu avec « Pause ». De même, quelques échantillons des classes « Keyboard », « Right Click » et « Single Click » ont été classés comme « Pause ». Cette confusion peut être expliquée par certaines similitudes géométriques entre les positions des doigts utilisées pour ces gestes.

Malgré ces quelques erreurs, le faible nombre de classifications incorrectes confirme la capacité du modèle SVM à distinguer efficacement les différents gestes statiques. Les résultats obtenus démontrent ainsi la robustesse du modèle et son adéquation pour une utilisation dans un système de contrôle d'ordinateur basé sur les gestes de la main.

5.5.1.3. Analyse détaillée par classe

Geste	Précision	Recall	F1-Score
Clic_gauche	0.975	0.967	0.967
Double_Clic	1	0.983	0.992
Clic_droit	1	0.950	0.974
Clavier	1	0.966	0.983
Pause/Lecture	0.909	1	0.952

Tableau 5.2: Résultats détaillées des métriques d'évaluation par classe

L'analyse des résultats du modèle SVM montre des performances globalement élevées pour la reconnaissance des gestes statiques. Les valeurs de précision, de rappel et de F1-score dépassent 95 % pour la majorité des classes, ce qui indique une excellente capacité de classification.

Le geste « Double_clic » a obtenu les meilleurs résultats avec une précision de 100 % et un F1-score de 99,2 %, démontrant une séparation claire de cette classe par rapport aux autres gestes. Les gestes « Clic_droit » et « Clavier» présentent également des performances très satisfaisantes avec des F1-scores respectifs de 97,4 % et 98,3 %.

Le geste « Clic_gauche » a enregistré un F1-score de 96,7 %, ce qui confirme une bonne reconnaissance malgré quelques erreurs de classification mineures. Quant au geste « Pause/Lecture », il possède un rappel de 100 %, indiquant que toutes les occurrences réelles ont été correctement détectées. Cependant, sa précision est légèrement plus faible (90,9 %), ce qui suggère l'existence de quelques faux positifs.

Dans l'ensemble, ces résultats confirment l'efficacité du modèle SVM pour la reconnaissance des gestes statiques destinés au contrôle de l'ordinateur, avec une excellente robustesse et une capacité de généralisation satisfaisante.

5.5.2. Résultats du modèle BiLSTM

5.5.2.1. Performance globale

Accuracy	Précision	Recall	F1-Score
99.44%	99.5%	99.4%	99.4%

Tableau 5.3: Tableau des performances du modèle BiLSTM (Accuracy, Précision, Rappel et F1-Score)

Les résultats globaux du modèle BiLSTM montrent d'excellentes performances dans la reconnaissance des gestes dynamiques. Le modèle a atteint une accuracy de 99,44 %, indiquant qu'il a correctement classifié la quasi-totalité des séquences de test. Les valeurs élevées de précision (99,5 %), de rappel (99,4 %) et de F1-score (99,4 %) confirment la capacité du modèle à distinguer efficacement les différentes classes tout en minimisant les erreurs de classification.

Ces résultats démontrent que l'architecture BiLSTM est particulièrement adaptée à l'analyse des séquences temporelles de mouvements de la main. Grâce à sa capacité à exploiter les dépendances temporelles entre les images successives, le modèle assure une reconnaissance fiable et robuste des gestes dynamiques utilisés pour le contrôle de l'ordinateur.

5.5.2.2. Courbes d'apprentissage (Accuracy et Loss)

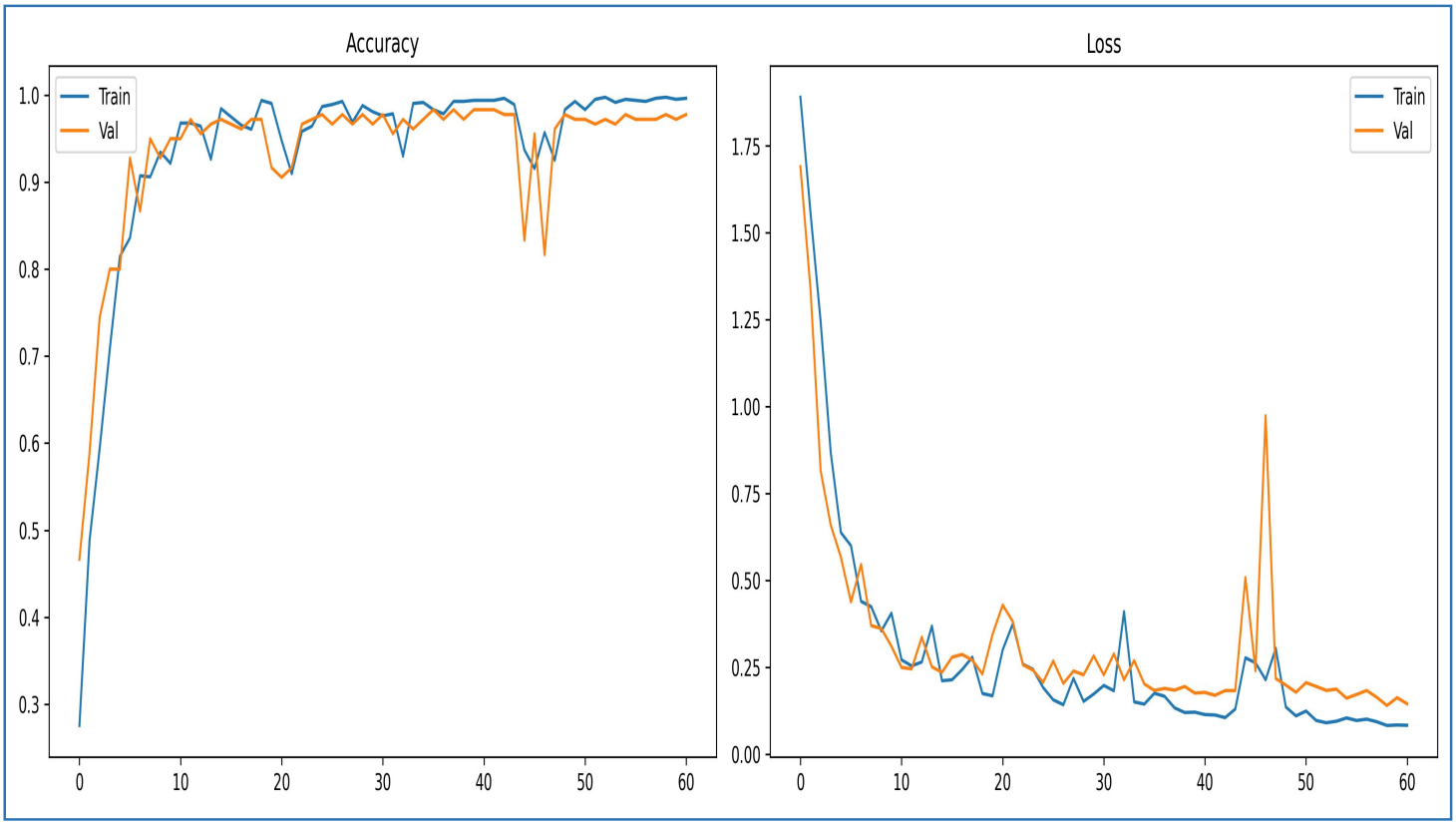


Figure 5.2: Courbes d'apprentissage du modèle BiLSTM (Accuracy et Loss)

Les courbes d'apprentissage montrent une amélioration progressive des performances du modèle BiLSTM au cours de l'entraînement. La précision (Accuracy) augmente rapidement durant les premières époques, passant d'une valeur initiale relativement faible à plus de 95 %, puis se stabilise progressivement autour de 99 % pour les ensembles d'entraînement et de validation.

La proximité entre les courbes de précision d'entraînement et de validation indique une bonne capacité de généralisation du modèle et l'absence d'un surapprentissage significatif. Bien que quelques fluctuations soient observées vers les dernières époques, les performances restent globalement stables.

Concernant la fonction de perte (Loss), les courbes d'entraînement et de validation présentent une diminution rapide au début de l'apprentissage, traduisant une amélioration continue de la qualité des prédictions. Les faibles valeurs finales de la perte confirment la convergence du modèle. Malgré l'apparition de quelques pics isolés sur la courbe de

validation, la tendance générale reste décroissante, ce qui témoigne de la stabilité du processus d'apprentissage.

Ces résultats démontrent que le modèle BiLSTM a appris efficacement les caractéristiques temporelles des gestes dynamiques et qu'il est capable de fournir des performances élevées sur des données non vues.

5.5.2.3. Matrice de confusion

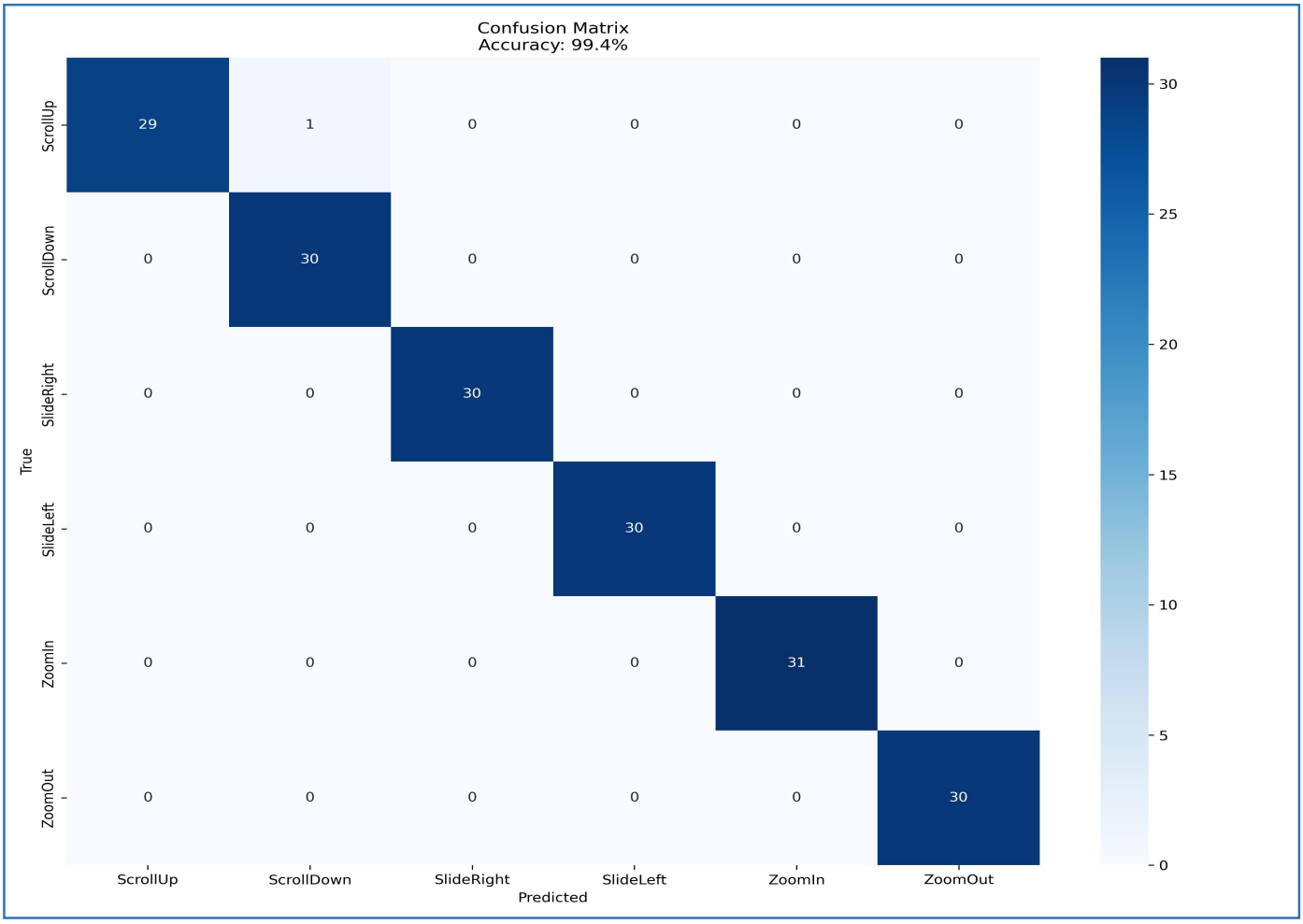


Figure 5.3: Matrice de confusion du modèle BiLSTM

On observe que la matrice est presque entièrement diagonale, ce qui signifie que la majorité des exemples ont été correctement classés par le modèle. Les classes ScrollDown, SlideRight, SlideLeft, ZoomIn et ZoomOut sont toutes reconnues parfaitement, avec respectivement 30, 30, 30, 31 et 30 bonnes prédictions. Cela montre que le modèle a appris de manière efficace les caractéristiques temporelles associées à ces gestes.

Une seule erreur de classification est visible : un exemple de la classe ScrollUp a été prédit à tort comme ScrollDown. Malgré cette confusion isolée, les performances globales restent excellentes, avec une accuracy de 99,4% indiquée sur la figure. Ce résultat traduit une très bonne capacité de généralisation du modèle BiLSTM sur les données de test.

En conclusion, la matrice de confusion confirme que le modèle proposé assure une classification très fiable des gestes étudiés.

5.5.2.4. Analyse détaillée par classe

Geste	Précision	Recall	F1-Score
Scroll_UP	1	0.97	0.98
Scroll_Down	0.97	1	0.98
Slide_Right	1	1	1
Slide_Left	1	1	1
Zoom_In	1	1	1
Zoom_Out	1	1	1

Tableau 5.4: Résultats détaillées des métriques d'évaluation par classe

L'analyse des indicateurs de performance par classe montre que le modèle BiLSTM obtient des résultats globalement très satisfaisants pour l'ensemble des gestes reconnus. Les valeurs de précision, de rappel et de F1-score sont quasi parfaites pour la majorité des classes, ce qui confirme la robustesse du modèle dans la classification des séquences temporelles.

La classe Scroll_UP présente une précision de 1, un rappel de 0,97 et un F1-score de 0,98. Cela signifie que toutes les prédictions faites pour cette classe sont correctes, mais qu'un petit nombre d'exemples réels n'a pas été détecté. La classe Scroll_Down affiche quant à elle une précision de 0,97, un rappel de 1 et un F1-score de 0,98, ce qui indique que tous les exemples réels ont été retrouvés, malgré la présence de quelques prédictions erronées.

Les classes Slide_Right, Slide_Left, Zoom_In et Zoom_Out obtiennent toutes des scores parfaits, avec une précision, un rappel et un F1-score égaux à 1. Ces résultats montrent que

le modèle reconnaît ces gestes sans aucune confusion ni omission, ce qui traduit une excellente capacité de discrimination entre les différentes classes.

5.5.3. Comparaison entre les deux modèles

Modèle	Accuracy	Précision	Recall	F1-Score
SVM	97.3 %	97.5 %	97.3 %	97.4 %
BiLSTM	99.4 %	99.5 %	99.4 %	99.4 %

Tableau 5.5: Comparaison des performances des modèles SVM et BiLSTM

Les résultats obtenus montrent que les deux approches sont performantes, mais chacune convient davantage à un type de geste. Le SVM a été utilisé pour les gestes statiques et a donné de bons résultats avec une accuracy de 97,3%. De son côté, le BiLSTM, appliqué aux gestes dynamiques, a obtenu de meilleures performances globales, avec une accuracy de 99,4%, une précision de 99,5%, un recall de 99,4% et un F1-score de 99,4%.

Cette différence s'explique par la nature des données traitées. Le SVM est bien adapté à des classes où les caractéristiques restent relativement fixes, tandis que le BiLSTM exploite efficacement la dimension temporelle des séquences, ce qui le rend plus approprié pour la reconnaissance des mouvements dynamiques. Ainsi, les deux modèles sont complémentaires, mais le BiLSTM se révèle plus robuste pour les gestes évolutifs dans le temps.

5.6. Comparaison avec d'autres méthodes

Travail	Auteur	Année	Titre	Méthode	Dataset	Accuracy (%)
T1: Article	Kenneth Lai et Svetlana N. Yanushkevicich	22 Jul 2020	CNN+RNN Depth and Skeleton based Dynamic Hand Gesture Recognition	CNN + RNN	Dynamic Hand Gesture -14/28	85.46
T2: Article	Koustav Mullick et Anoop M. Namboodiri	29 Dec 2017	LEARNING DEEP AND COMPACT MODELS FOR GESTURE RECOGNITIOgnition	3D-CNN + LSTM	ChaLearn (18.37 millions)	93.2

T3: Article	Xianyuan Wang, Zhenjiang Miao, Ruyi Zhang et Shanshan Hao	2019	I3D-LSTM: A New Model for Human Action Recognition	3D-CNN + LSTM	UCF-101	95.1
T4: Master's Degree	Abdallah ABDALLAH		Study of hand gesture recognition algorithms using the EMGs signals based on NINA PRO dataset	CNN	NINA PRO	95.98
T5: Article	Samat Mukhanov, Raissa Uskenbayeva et all	2023	GESTURE RECOGNITION OF MACHINE LEARNING AND CONVOLUTIONAL NEURAL NETWORK METHODS FOR KAZAKH SIGN LANGUAGE	SVM, CNN, LSTM		92
T6: Notre système	BOUSMAT Aya et ZAIR Mohamed Walid	2026	Système de reconnaissance des geste de la main pour le contrôle d'actions informatiques	SVM + BiLSTM	Jeu de données personnalisé (2200 échantillon)	97.3 (SVM) 99.4 (BiLSTM)

Tableau 5.6: Comparaison de notre approche avec les travaux existants

L'analyse comparative présentée dans le Tableau 5.X montre que les performances obtenues par notre système sont très compétitives par rapport aux travaux de référence de la littérature. Les approches basées sur les réseaux de neurones profonds, telles que CNN+RNN (Kenneth Lai et Yanushkevicich, 2020), 3D-CNN+LSTM (Mullick et Namboodiri, 2017) et I3D-LSTM (Wang et al., 2019), ont atteint des taux de reconnaissance compris entre 85,46 % et 95,1 %. De même, les méthodes basées sur CNN appliquées au jeu de données NINA PRO ont obtenu une précision de 95,98 %, tandis que les approches combinant SVM, CNN et LSTM pour la langue des signes kazakhe ont rapporté une précision de 92 %.

Notre système, fondé sur une architecture hybride combinant un modèle SVM pour les gestes statiques et un modèle BiLSTM pour les gestes dynamiques, a obtenu respectivement une précision de 97,3 % et 99,4 %. Ces résultats surpassent la majorité des travaux étudiés, malgré l'utilisation d'un jeu de données personnalisé de taille relativement modeste (2200 échantillons) comparativement à certains ensembles de données de grande envergure tels que ChaLearn ou UCF-101. Cette performance élevée peut être attribuée à la qualité du prétraitement des données, à l'extraction pertinente des caractéristiques de la main à l'aide de MediaPipe, ainsi qu'à la spécialisation de chaque modèle dans une tâche précise (SVM pour les gestes statiques et BiLSTM pour les séquences temporelles).

Ainsi, les résultats obtenus démontrent l'efficacité de l'approche proposée et confirment qu'une combinaison judicieuse de méthodes d'apprentissage automatique et d'apprentissage profond peut offrir des performances supérieures tout en conservant une complexité raisonnable et une exécution adaptée aux applications de contrôle en temps réel.

5.7. Discussion

Les résultats obtenus démontrent l'efficacité du système proposé pour la reconnaissance des gestes de la main et le contrôle d'actions informatiques en temps réel. Les performances élevées obtenues par les modèles SVM et BiLSTM montrent la pertinence de l'approche adoptée.

5.7.1. Points forts du système

Les principaux avantages du système sont les suivants :

- Excellente précision de reconnaissance (97,3 % pour le SVM et 99,4 % pour le BiLSTM).
- Temps de réponse rapide permettant une utilisation en temps réel.
- Bonne robustesse face aux variations normales d'éclairage.
- Capacité de généralisation à différents utilisateurs, y compris ceux non présents dans la phase d'apprentissage.
- Utilisation d'une simple caméra RGB sans capteurs spécialisés.
- Faible coût matériel et facilité de déploiement.

- Exploitation des points clés extraits par MediaPipe, ce qui réduit la quantité de données à traiter.
- Possibilité d'exécution sur des ordinateurs standards et des plateformes à ressources limitées.
- Architecture hybride combinant les avantages du SVM pour les gestes statiques et du BiLSTM pour les gestes dynamiques.

5.7.2. Limites du système

Malgré ses bonnes performances, certaines limitations ont été observées :

- Diminution de la qualité de détection dans des conditions de très faible luminosité.
- Nombre limité des gestes reconnus (12 gestes).
- Dépendance à la qualité de détection des points clés fournie par MediaPipe.
- Risque de confusion entre certains gestes présentant des configurations très similaires.
- Évaluation réalisée sur un jeu de données personnalisé, ce qui limite la comparaison directe avec certains benchmarks publics.

5.7.3. Perspectives d'amélioration

Pour améliorer davantage le système, plusieurs pistes peuvent être envisagées :

- Augmenter le nombre des gestes reconnus.
- Enrichir le jeu de données avec davantage d'utilisateurs et de scénarios.
- Améliorer la robustesse dans les environnements à faible éclairage.
- Tester le système sur des plateformes embarquées ou mobiles.
- Étendre les fonctionnalités pour contrôler un plus grand nombre d'applications informatiques.
- Intégrer des modèles plus avancés pour la reconnaissance des gestes complexes.

5.8. Interface et démonstration du système

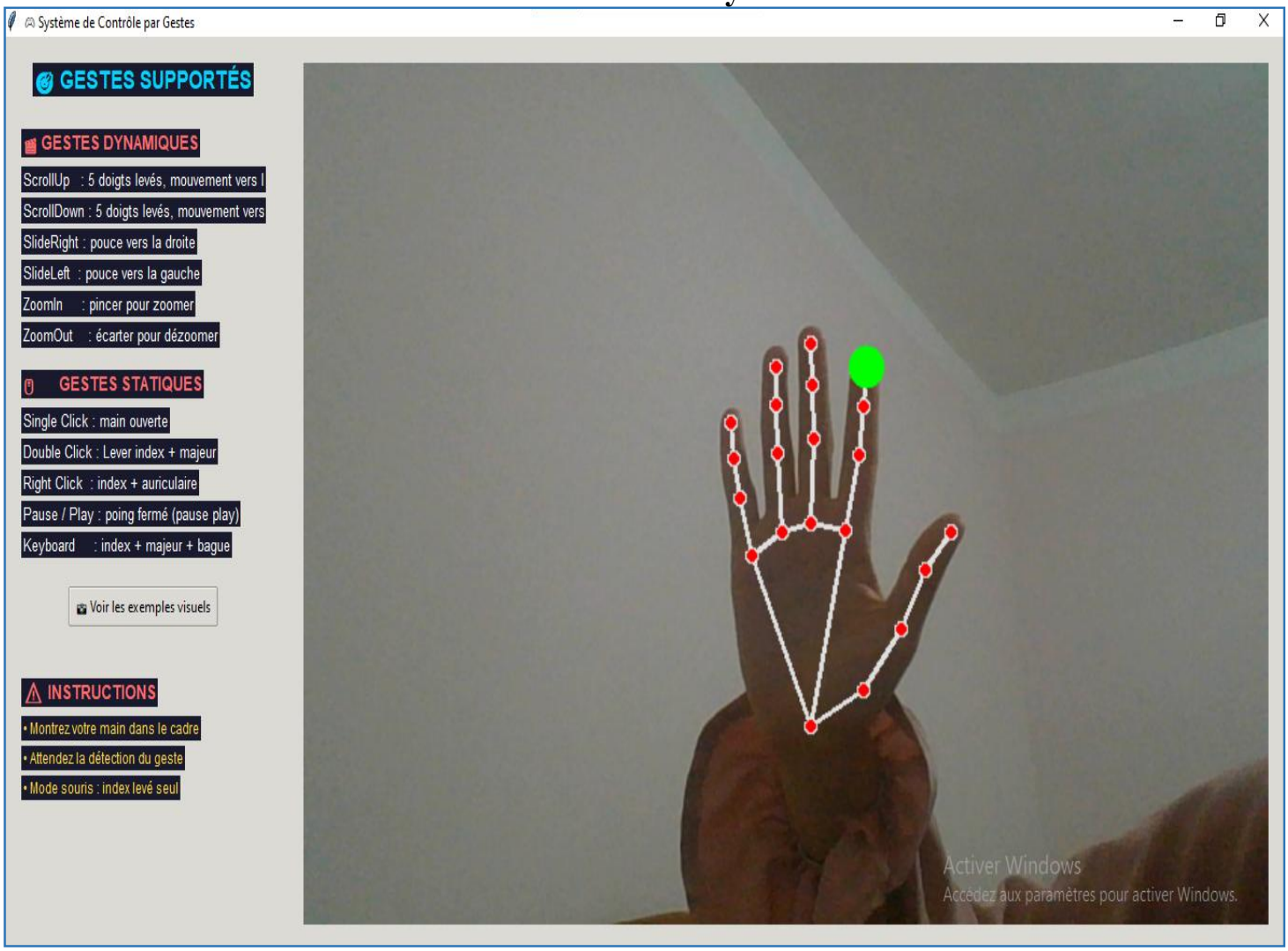


Figure 5.4: Interface principale du système de reconnaissance des gestes.

Afin d'améliorer l'expérience utilisateur, une interface graphique a été développée. Elle permet de visualiser en temps réel la détection et le suivi de la main, de consulter les gestes disponibles ainsi que les consignes d'utilisation. Un module de visualisation a également été intégré pour illustrer l'exécution correcte des gestes reconnus par le système.



Figure 5.5 : Module de visualisation des gestes reconnus par le système.

Cette interface présente une représentation graphique des gestes statiques et dynamiques reconnus par le système afin d'aider l'utilisateur à les exécuter correctement.

5.8.1. Exemples d'utilisation du système

Afin d ' illustrer le fonctionnement du système dans des situations réelles, deux exemples sont présentés : un geste statique associé à la commande Pause et un geste dynamique permettant d ' effectuer un zoom. Ces exemples montrent la capacité du système à reconnaître différents types des gestes et à exécuter les actions correspondantes en temps réel.

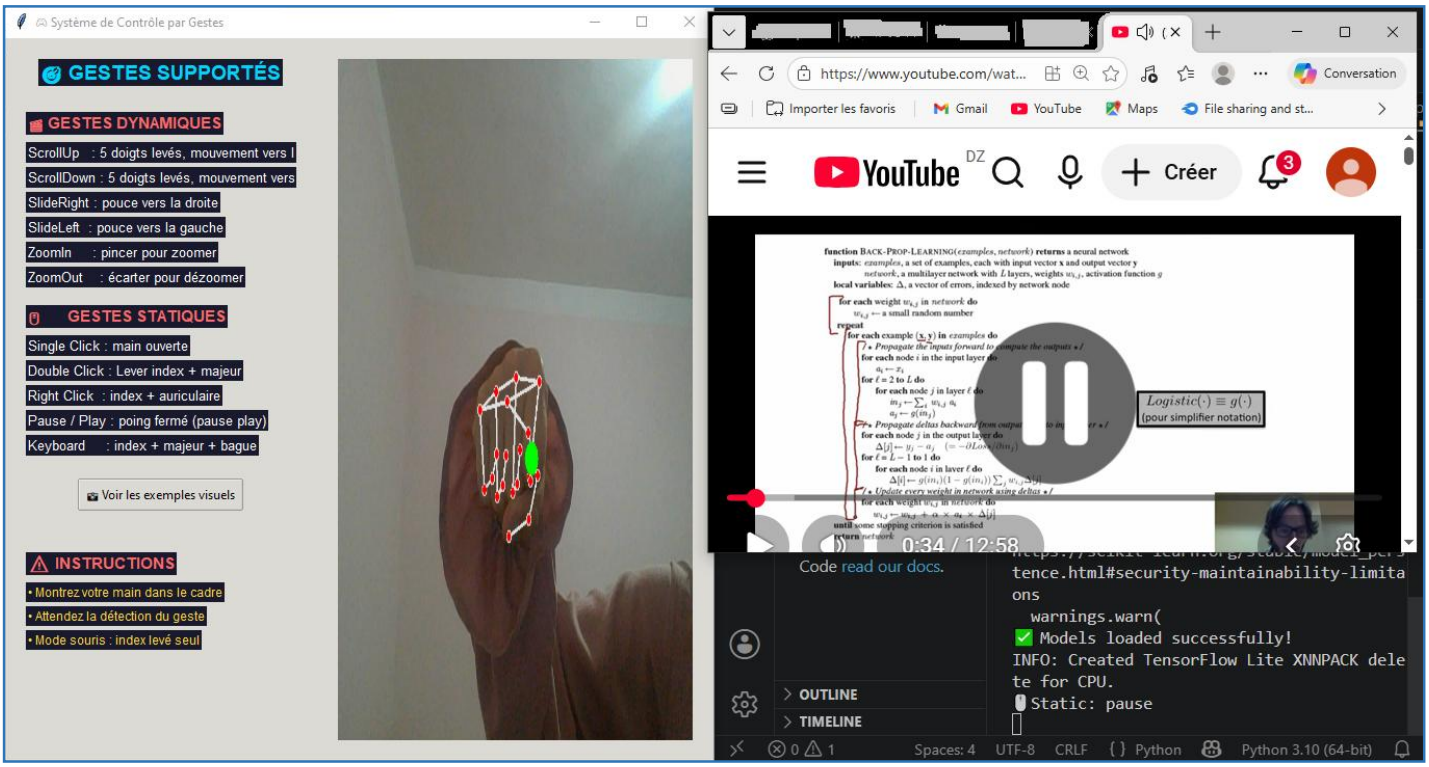


Figure 5.6: Reconnaissance du geste statique « Pause » et exécution de la commande correspondante.

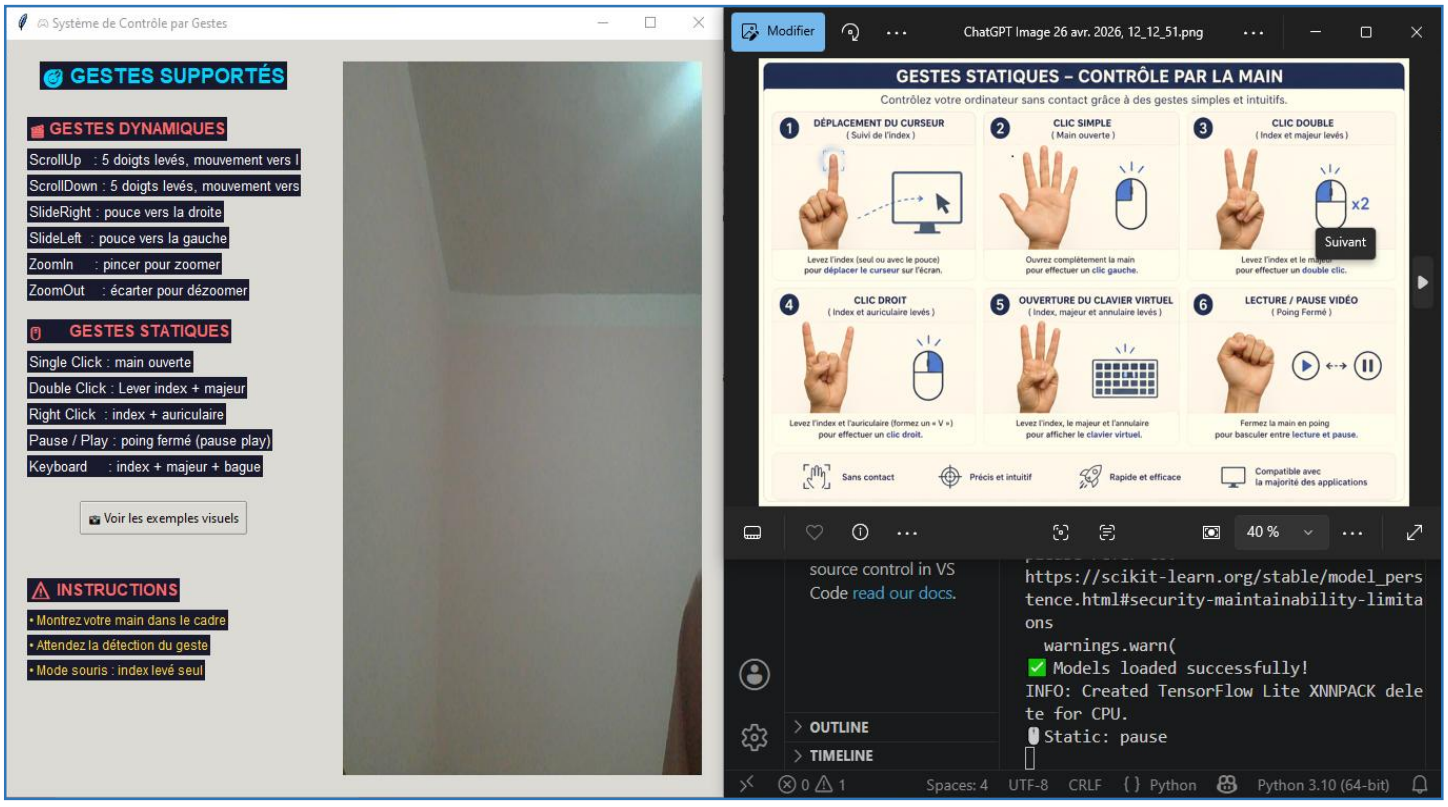


Figure 5.7: État initial avant l'exécution du geste de zoom.

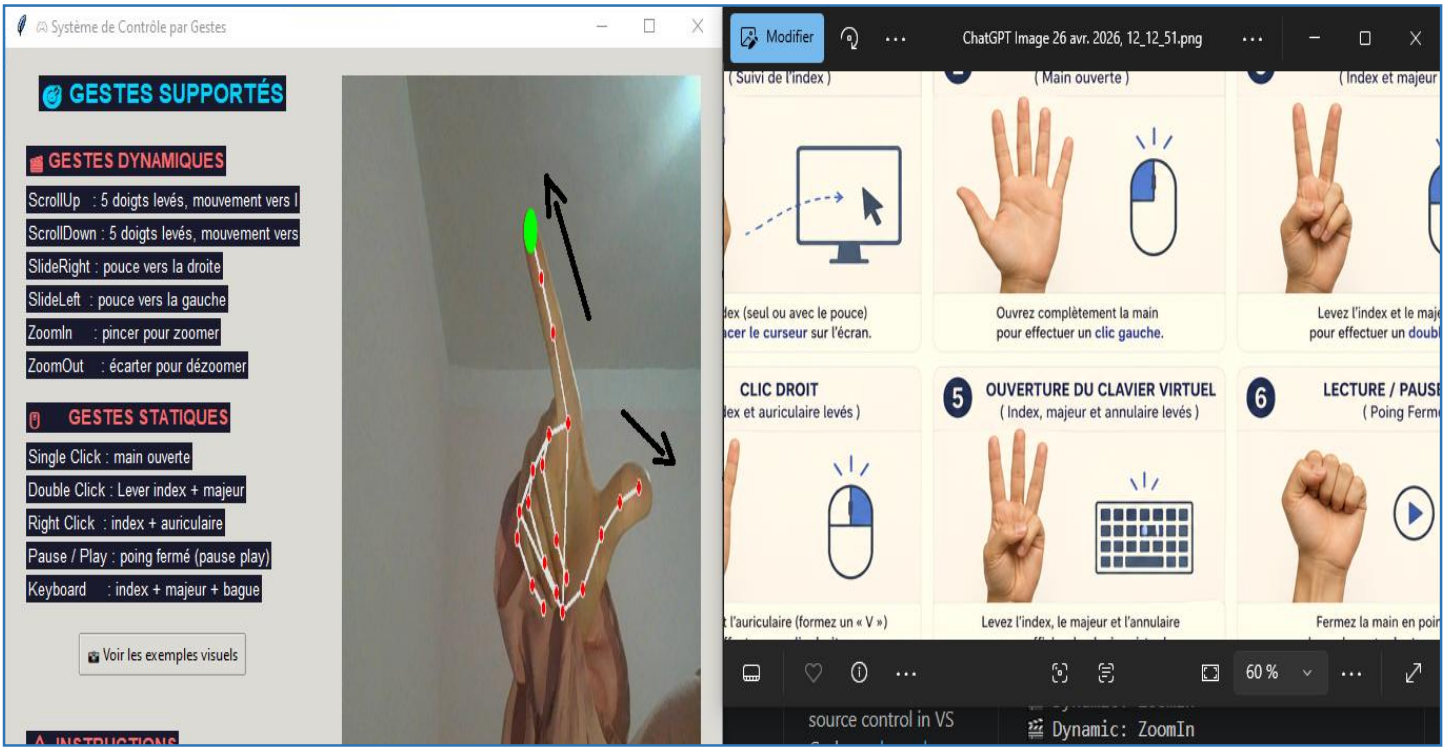


Figure 5.8: Résultat obtenu après l'exécution du geste dynamique de zoom.

5.9. Conclusion

Dans ce chapitre, nous avons présenté le protocole expérimental mis en œuvre pour évaluer le système de reconnaissance des gestes développé ainsi que les jeux de données utilisés pour l'apprentissage et les tests. L'analyse des résultats a montré que les deux modèles retenus offrent d'excellentes performances, avec une précision de 97,3 % pour le modèle SVM dédié aux gestes statiques et de 99,4 % pour le modèle BiLSTM destiné aux gestes dynamiques.

Les différentes métriques d'évaluation, les matrices de confusion et les analyses détaillées par classe ont confirmé la robustesse et la fiabilité du système. La comparaison avec des travaux de référence a également mis en évidence la compétitivité de l'approche proposée. Par ailleurs, la présentation de l'interface graphique et des exemples d'utilisation a démontré la capacité du système à assurer un contrôle d'actions informatiques en temps réel à partir des gestes de la main.

Ces résultats valident ainsi les choix techniques effectués tout au long de ce projet et confirment la pertinence de l'architecture hybride basée sur SVM et BiLSTM pour la reconnaissance des gestes statiques et dynamiques.

Conclusion générale

Conclusion générale

L'interaction homme-machine constitue aujourd'hui un domaine de recherche en pleine évolution, visant à rendre l'utilisation des systèmes informatiques plus naturelle, intuitive et accessible. Dans ce contexte, la reconnaissance gestuelle représente une solution prometteuse permettant de remplacer ou de compléter les dispositifs d'interaction traditionnels tels que la souris et le clavier.

L'objectif principal de ce travail était la conception et la réalisation d'un système de reconnaissance des gestes de la main destiné au contrôle d'actions informatiques en temps réel à l'aide d'une simple caméra. Pour atteindre cet objectif, nous avons d'abord étudié l'état de l'art de la reconnaissance gestuelle ainsi que les principales techniques utilisées dans ce domaine. Nous avons ensuite analysé les méthodes de vision par ordinateur et d'apprentissage automatique les plus adaptées à la reconnaissance des gestes statiques et dynamiques.

Sur la base de cette étude, nous avons développé un système hybride combinant MediaPipe pour la détection et le suivi de la main, un modèle SVM pour la reconnaissance des gestes statiques et un modèle BiLSTM pour la reconnaissance des gestes dynamiques. Une interface graphique conviviale a également été conçue afin de faciliter l'utilisation du système et d'améliorer l'expérience utilisateur.

Les expérimentations réalisées ont démontré l'efficacité de la solution proposée. Les résultats obtenus ont atteint une précision de 97,3 % pour le modèle SVM et de 99,4 % pour le modèle BiLSTM, confirmant ainsi la robustesse et la fiabilité du système dans des conditions d'utilisation réelles. De plus, le système présente plusieurs avantages, notamment son faible coût matériel, sa rapidité d'exécution, sa capacité de généralisation à différents utilisateurs et sa compatibilité avec des équipements informatiques standards.

Malgré ces résultats encourageants, certaines limitations subsistent, notamment la sensibilité aux conditions d'éclairage très faibles et le nombre limité des gestes actuellement pris en charge. Ces contraintes ouvrent plusieurs perspectives d'amélioration.

Comme travaux futurs, il serait intéressant d'augmenter le nombre des gestes reconnus, d'enrichir la base de données avec davantage d'utilisateurs et de scénarios, d'améliorer la

robustesse du système face aux conditions environnementales difficiles et d'étendre son déploiement vers des plateformes embarquées ou mobiles. L'intégration de modèles d'intelligence artificielle plus avancés pourrait également permettre la reconnaissance des gestes plus complexes et l'élargissement des possibilités d'interaction homme-machine.

En conclusion, ce travail a permis de concevoir et de valider un système performant de reconnaissance gestuelle pour le contrôle d'actions informatiques. Les résultats obtenus démontrent le potentiel de cette approche et constituent une base solide pour le développement de futures applications interactives basées sur les gestes de la main.

Références

[1] A.Cheloufi, " Système de Reconnaissance des gestes de la main ", Mémoire de master, Telemcen , 2018.

[2] Ch.Berrached, " Système de Reconnaissance des gestes de la main ", Mémoire de master, Telemcen, 2014.

[3] https://www.dynamique-mag.com/article/la-realite-augmentee-et-la-realite-virtuelle-comment-ces-technologies-transforment-les-industries-traditionnelles

[4] VILLANI, Valeria, SECCHI, Cristian, LIPPI, Marco et SABATTINI, Lorenzo. A general pipeline for online gesture recognition in human–robot interaction. IEEE Transactions on Human-Machine Systems, 2023, vol. 53, n° 2, p. 315-324.

[5] JAIN, Deepak Kumar, MAHANTI, Aniket, SHAMSOLMOALI, Pourya et MANIKANDAN, Ramachandran. Deep neural learning techniques with long short-term memory for gesture recognition. Neural Computing and Applications, 2020, vol. 32, n° 20, p. 16073-16089.

[6] CARFÌ, Alessandro et MASTROGIOVANNI, Fulvio. Gesture-based Human-Machine Interaction: Taxonomy, Problem Definition, and Analysis. IEEE Transactions on Cybernetics. 2022, vol. 52, n° 12, pp. 12759-12771. DOI 10.1109/TCYB.2021.31343

[7] DIAS, P., PINTO, J., ELISEU, S. et SANTOS, B. S. Gesture interactions for virtual immersive environments: Navigation, selection and manipulation. In : International Conference on Virtual, Augmented and Mixed Reality. Cham : Springer International Publishing, 2016, pp. 211-221.

[8] AROMAA, Susanna, FRANGAKIS, Nikos, TEDONE, Domenico, VIITANIEMI, Juhani et AALTONEN, Iina. Digital human models in human factors and ergonomics evaluation of gesture interfaces. In : Proceedings of the ACM on Human-Computer Interaction. Juin 2018. Vol. 2, n° EICS, Art. 06, pp. 1-14. DOI 10.1145/3229088.

[9] ESCALERA, Sergio, ATHITSOS, Vassilis et GUYON, Isabelle. Challenges in multi-modal gesture recognition. In : Gesture Recognition. Cham : Springer, 2017, pp. 1-60. DOI 10.1007/978-3-319-58457-7_1.

[10] KHETTAF Yasmine. Mémoire de master en informatique: « Etude de méthode de reconnaissance de geste ». Faculté des Sciences, Université de Skikda . 2022.

[11] T.S. Varshini and P. Rukmani. Article:" MP-GestLSTM: real time gesture detection using MediaPipe and LSTM ". School of Computer Science and Engineering, Vellore Institute of Technology Chennai Campus, Chennai, India. 2025.

[12] Théo Voillemin. THÈSE pour obtenir le grade de Docteur: « Reconnaissance de gestes et d'actions de la main, combinant vision par ordinateur et technologies de réalité augmentée » .Université de Lille. Ecole Doctorale SPI-MADIS.2021.

[13] LI, H., et al. sEMG-based Gesture-Free Hand Intention Recognition: System, Dataset, Toolbox, and Benchmark Results. arXiv, 2024.

[14] VALCHEVA, I., GANDHI, V. et CHINELLATO, E. Surface EMG Driven Gesture Recognition Using Machine Learning for Robotic Applications. Discover Robotics, 2026.

[15] YUAN, Y. sEMG Signal Processing and Feature Extraction for Gesture Recognition. PLOS ONE, 2024.

[16] SHIN, J., et al. Hand Gesture Recognition Using sEMG Signals with Feature Enhancement. Scientific Reports, 2024.

[17] YUAN, Y. Application of a sEMG Hand Motion Recognition Method in Rehabilitation Medicine. PLOS ONE, 2024.

[18] HAN, G. Improving sEMG-Based Hand Gesture Recognition Through Optimizing Parameters and Classifiers. Electronics, 2024.

[19] TU, P., LI, J. et WANG, H. Motion Recognition with Improved SVM Based on sEMG. Sensors, 2024.

[20] Hand Gesture Classification Using Multi-Class SVM with Feature Extraction. Expert Systems with Applications, 2024.

[21] Camillo Lugaresi et al. MediaPipe: A Framework for Building Perception Pipelines. arXiv, 2020.

[22] CHEN, X. et al. Real-Time Hand Gesture Interaction Systems for Human-Computer Interaction. IEEE Access, 2023.

[23] TU, P. et al. Improved SVM for Motion Recognition Using sEMG Signals. Sensors, 2024.

[24] HAN, G. Optimization of SVM Parameters for Classification Tasks. Electronics (MDPI), 2024.

[25] CHEN, W. et al. Hand Gesture Recognition Using Deep Learning: A Review. IEEE Access, 2021.

[26] UNIVERSITÉ DE LYON. Application des réseaux LSTM pour la reconnaissance de mouvements humains, 2023.

[27] UNIVERSITY OF OXFORD. Travaux sur sequence modeling et BiLSTM appliqués à la vision, 2020–2022.

[28] UNIVERSITÉ DE MONTRÉAL. Deep Learning for Human Action and Gesture Recognition, 2022.

[29] HOCHREITER, S. et SCHMIDHUBER, J. Revisited Works on LSTM Applied in Modern Deep Learning (TensorFlow/Keras Documentation), 2022–2024.

[30] HOCHREITER, S. et SCHMIDHUBER, J. Travaux sur les LSTM (Long Short-Term Memory).

[31] MDN WEB DOCS. Python. Mozilla, 2023. [en ligne]. Consulté le 11 avril 2026.

[32] WEBTECH. OpenCV. In : Glossaire Webtech [en ligne]. Consulté le 23 mai 2024. Disponible à l'adresse : https://webtech.fr/glossaire/opencv/

[33] INNOVATIANA. MediaPipe : la boîte à outils incontournable pour le Computer Vision. In : Blog Innovatiana [en ligne]. 21 août 2024. [Consulté le 16 avril 2026]. Disponible à l'adresse : https://www.innovatiana.com/fr/post/mediapipe-101

[34] SCIKIT-LEARN DEVELOPERS. sklearn.model_selection.train_test_split. In : scikit-learn v1.4.2 documentation [en ligne]. 2024. [Consulté le 16 avril 2026]. Disponible à l'adresse : https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[35] GOOGLE. MediaPipe Hands. In : MediaPipe Legacy Solutions Documentation [en ligne]. 2023. [Consulté le 16 avril 2026]. Disponible à l'adresse : https://mediapipe.readthedocs.io/en/latest/solutions/hands.html

[36] PANDAS DEVELOPMENT TEAM. Pandas: Powerful Python Data Analysis Toolkit [en ligne]. 2024. [Consulté le 16 avril 2026]. Disponible à l'adresse : https://pandas.pydata.org/

[37] ORDNANCE SURVEY. OS Data Catalogue [en ligne]. In : data.gov.uk. Dernière mise à jour : 20 décembre 2024. [Consulté le 16 avril 2026]. Disponible à l'adresse : https://www.data.gov.uk/dataset/f5e45f2d-cad8-45df-8522-882f4c93fd62/os-data-catalogue1.

[38] CHRAIR, Amina et LAROUI, Fella Hayat. La biométrie de la main sans contact. Mémoire de Master : Génie Biomédical. Tlemcen : Université de Tlemcen, 2023. [Consulté le 16 avril 2026]. Disponible à l'adresse : https://dspace.univ-tlemcen.dz/handle/112/20998

[39] FAWCETT, Tom. An Introduction to ROC Analysis. Pattern Recognition Letters, vol. 27, n° 8, 2006, p. 861–874.

الجمهورية الجزائرية الديمقراطية الشعبية

Republique Algérienne Démocratique Populaire
وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur de la Recherche Scientifique

Université 20 août 1955 Skikda

جامعة 20 أوت 1955 – سكيكدة

Faculte des Sciences
Departement d'informatique

كلية العلوم
قسم الإعلام الآلي
الرقم: .... / ق/ إ/ أ/ ك.ع2026/

Autorisation de Dépôt de Mémoire de Master

Je soussigné : .Dr....Bouhmoun...Imene.....

Certifie que l'étudiant(e) : .Boumat....Aya.....

Spécialité : ..Intelligence...Artificielle.....

Ayant soutenu le projet intitulé : .Système..de.reconnaissance...des.gestes de la main pour le contrôle d'actions informatiques

A apporté les corrections nécessaires sur son manuscrit de Master.

Signature de l'encadreur

جامعة 20 أوت 1955 سكيكدة
قسم الإعلام الآلي
كلية العلوم

Scanné avec CamScanner