

LinLED : Détection de gestes simples via une interface homme-machine utilisant des capteurs linéaires bio-inspirés

Étudiant : Madou FALL¹

Encadrants : D. Martinez², S. Viollet², J. Monnoyer^{2, 3}

¹Master 2 Traitement du Signal et des Images (TSI), Aix-Marseille Université, Marseille, France

²Institut des Sciences du Mouvement (ISM - UMR 7287), Aix-Marseille Université & CNRS

³Groupe Stellantis, Constructeur automobile multinational (Peugeot, Fiat, Citroën, ...)

Soutenance de stage de fin d'études le 04/09/2025.



INSTITUT DES SCIENCES ETIENNE DU MOUVEMENT JULES MAREY

Contexte et enjeux

- Technologie innovante : **interaction sans contact** pour un contrôle naturel
- Besoin industriel : interfaces **intuitives et réactives** embarquées dans les véhicules modernes
- Marché compétitif : **Neonode** face à **LinLED**, un acteur prometteur
- Objectif : **sécurité, ergonomie et expérience utilisateur améliorées** pour un usage fluide et fiable



Figure: Interaction sans contact dans un système automobile^a

^aSource : Groupe Stellantis

Intérêt du projet LinLED

Le projet LinLED¹ se distingue par sa précision, sa rapidité, et une gestuelle naturelle.

- LEDs et photodiodes à faible coût
- Respect de la vie privée : aucune image ni donnée personnelle enregistrée
- Technologie basée sur l'analogie et la Low-Tech

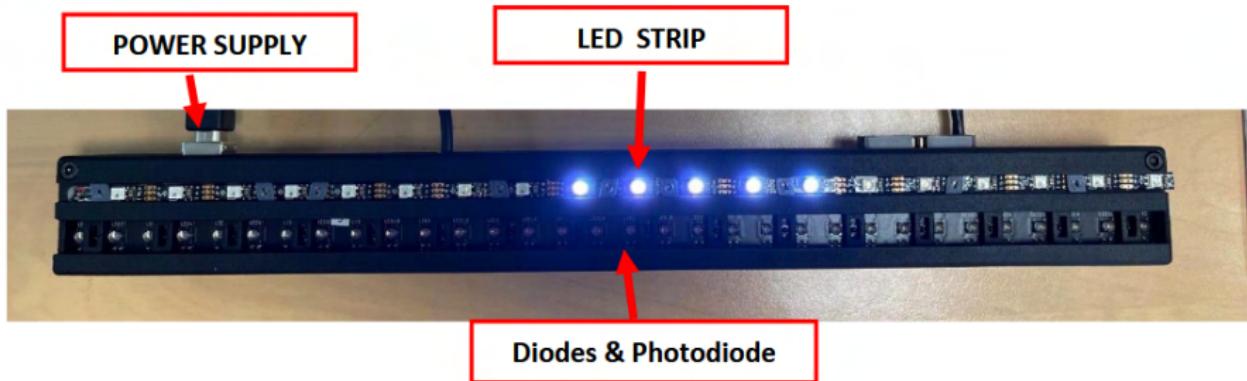


Figure: LinLED

¹Stephane Viollet, Chauvet Martin, and Ingargiola Jean-Marc (2023). "LinLED: Low latency and accurate contactless gesture interaction". In: *Companion Publication of the 25th International Conference on Multimodal Interaction*, pp. 61–65

Interagir avec une machine (comme l'écran d'une voiture),
par un simple geste de la main, grâce à des capteurs bio-inspirés,
sans aucun contact physique.

Pourquoi bio-inspirés ?



Figure: Poissons électriques : Utilisent des champs électriques pour percevoir leur environnement.²

²Myriah Haggard and Maurice J Chacron (2023). "Coding of object location by heterogeneous neural populations with spatially dependent correlations in weakly electric fish". In: *PLOS Computational Biology* 19.3, e1010938

Caractéristiques du système LinLED

- **Système** : bio-inspiré
- **Technologie** : Capteurs infrarouges (IR)
- **LEDs** : 32
- **Photodiodes** : 16

- **Portée** : Jusqu'à 500 mm
- **Latence** : 1 ms
- **Sorties analogiques** : 18 au total
- Amplificateurs à verrouillage de phase^a utilisés pour la démodulation.

^aCircuits électroniques qui extraient un signal faible noyé dans le bruit, en filtrant précisément la fréquence d'intérêt.

$$W_s = 1 \cdot w_1 + 2 \cdot w_2 + \dots \quad \text{et} \quad S = w_1 + w_2 + \dots$$

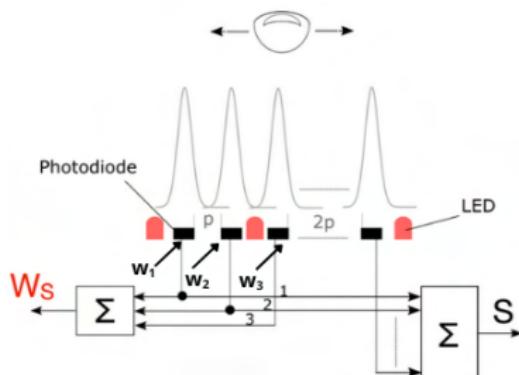


Figure: Réseau linéaire (1D) de LEDs IR et de photodiodes composant LinLED³.

³Stephane Viollet, Chauvet Martin, and Ingariola Jean-Marc (2023). "LinLED: Low latency and accurate contactless gesture interaction". In: *Companion Publication of the 25th International Conference on Multimodal Interaction*, pp. 61–65

Méthode de localisation : somme pondérée

- Chaque photodiode mesure une intensité locale.

- La somme pondérée

$$W_s = 1 \cdot w_1 + 2 \cdot w_2 + \dots$$

permet d'estimer la **position moyenne** du geste.

- **Chevauchement** : les gaussiennes des photodiodes doivent se chevaucher d'au moins 15%.
- **Monotonie et quasi-linéarité** : W_s devient monotone et presque linéaire
- **Lien avec l'hyperacuité**⁴ : localisation plus fine que l'espacement réel des photodiodes.

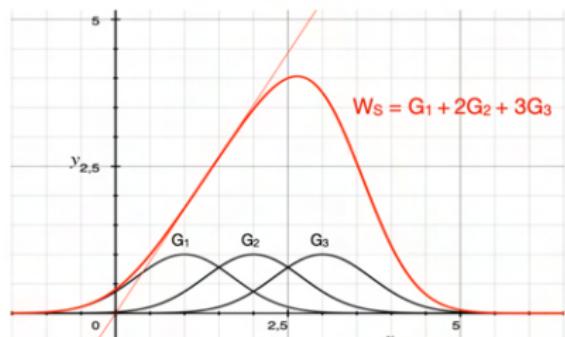


Figure: Somme pondérée de gaussiennes^a

^a<https://linled.univ-amu.fr/poster-linled>

⁴Gerald Westheimer and Suzanne P McKee (1977). "Spatial configurations for visual hyperacuity". In: *Vision research* 17.8, pp. 941–947

Problématique et défis

Problématique générale :

Comment concevoir une **interface homme-machine gestuelle**, fiable et robuste, basée sur des capteurs infrarouges bio-inspirés, et intégrable en temps réel dans un véhicule ?

Défis techniques associés :

- **Acquisition** : détecter des signaux faibles dans un environnement bruité.
- **Apprentissage** : reconnaître et différencier des gestes variés chez plusieurs utilisateurs.
- **Intégration embarquée** : implémenter les algorithmes sur microcontrôleur avec faible latence et faible consommation.

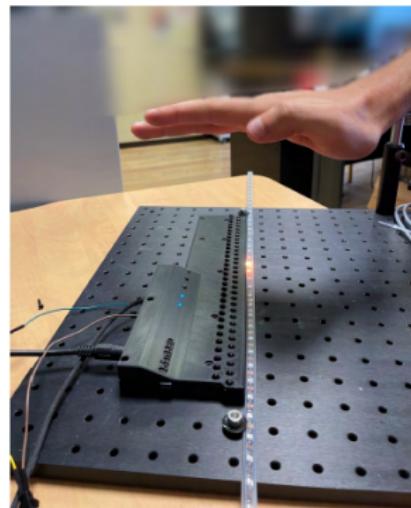


Figure: Prototype LinLED (avec bande LED pour retour visuel)

Chaîne d'acquisition des signaux



Arduino / Teensy :

- Acquisition de 16 signaux des photodiodes LinLED à 200 Hz, ADC 10 bits
- Transmission fiable vers le PC via port série

MATLAB :

- Visualisation en temps réel des signaux
- Enregistrement pour analyse ultérieure

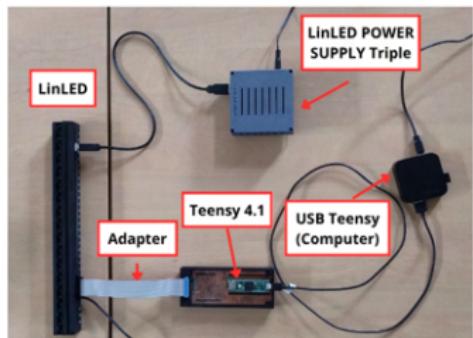
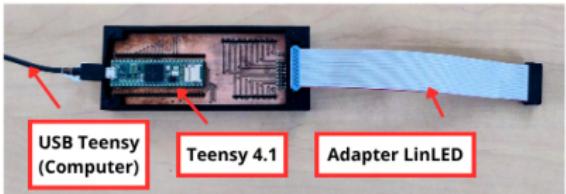


Figure: Câblage d'alimentation LinLED

Microcontrôleur Teensy :

- Prototype temps réel pour futur déploiement embarqué



Visualisation de geste enregistré

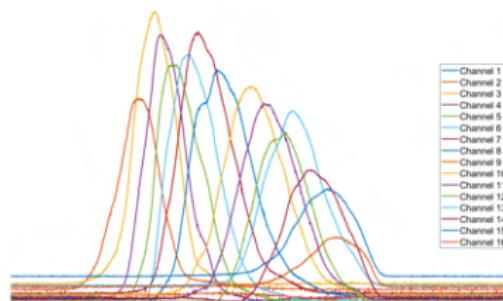
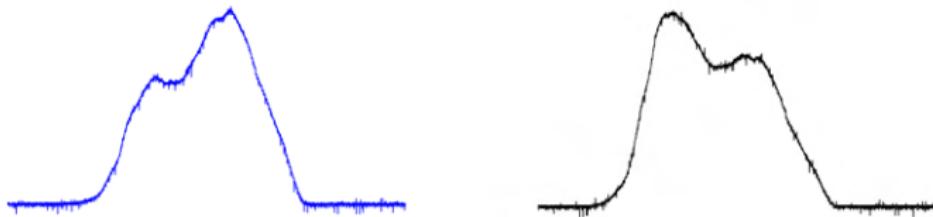


Figure: Signal issu des 16 canaux (capteurs) pendant un mouvement de la main



(a) Somme pondérée des 16 canaux

(b) Moyenne des 16 canaux

Gestes étudiés et capturés



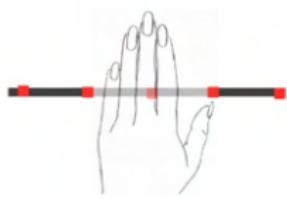
(a) **Swipe Right** : mouvement de droite à gauche.



(b) **Click** : mouvement de la main vers LinLED pour sélectionner.



(c) **Swipe Left** : mouvement de gauche à droite.



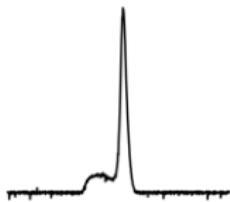
(d) **Still** : main immobile pendant 1 à 5 secondes.



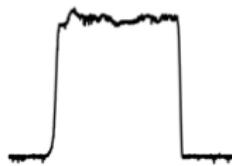
(e) **In Out** : avancer puis reculer la main devant LinLED.

Signatures temporelles des gestes

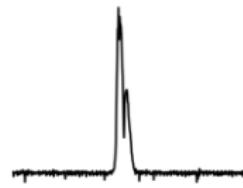
Visualisation : Courbes de la somme pondérée des 16 capteurs IR pour chaque geste enregistré.



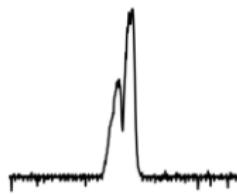
(a) Click



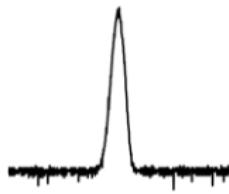
(b) Still



(c) Swipe Right



(d) Swipe Left



(e) In Out

Chaque geste produit une signature temporelle unique exploitable.

Base de données des gestes

 Collecte auprès de 20 volontaires

 Profil :

- 15 participants de moins de 30 ans
- 5 participants de plus de 30 ans

 5 gestes différents

 Chaque geste répété 10 fois par participant

 Aucun Consignes

 Accessible sur le dépôt GitHub : <https://github.com/fallmadou/LinLED-project/tree/main/database>

Apprentissage automatique des gestes

Nous avons utilisé Python pour tester plusieurs modèles, avec et sans filtrage.

Pourquoi l'auto-apprentissage^{5 6} ?

- Reconnaît bien les gestes
- S'adapte aux styles variés
- Rapide à entraîner et utiliser

Données :

- 1034 gestes enregistrés dont :
 - 827 pour l'entraînement
 - 207 pour le test

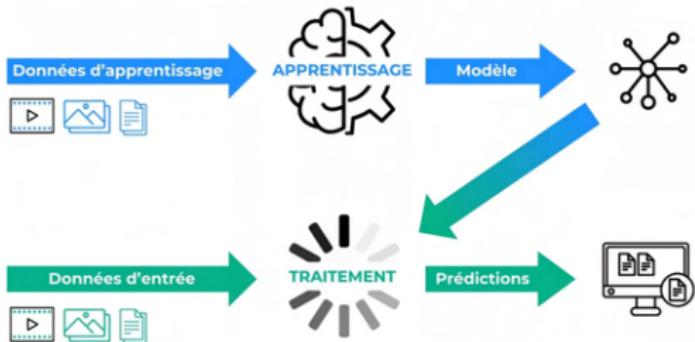
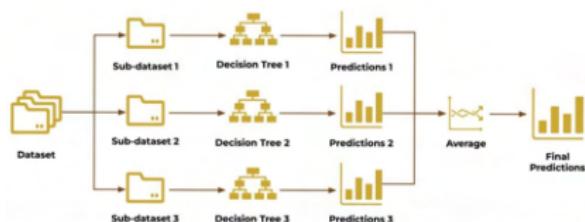


Figure: Principe de l'apprentissage automatique

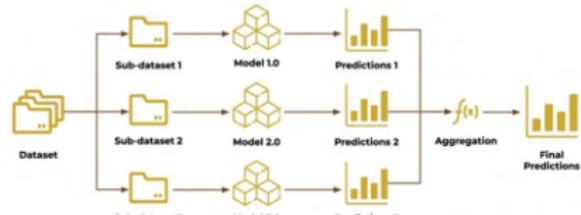
⁵Téo Sanchez (2018). "Systèmes interactifs par apprentissage actif pour la reconnaissance de gestes". In

⁶R Chellali et al. (2012). "Détection et reconnaissance des gestes emblématiques". In: *RFIA 2012 (Reconnaissance des Formes et Intelligence Artificielle)*, pp. 978–2

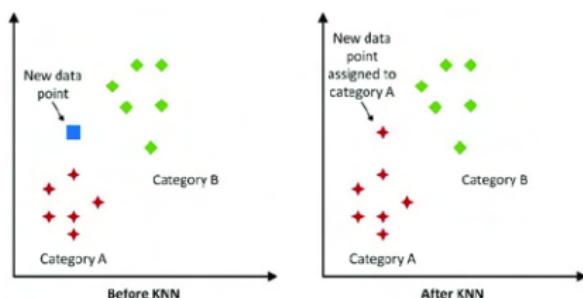
Algorithmes de classification testés



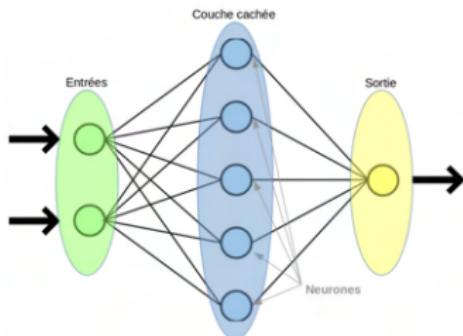
(a) Random Forest



(b) HistGradientBoosting



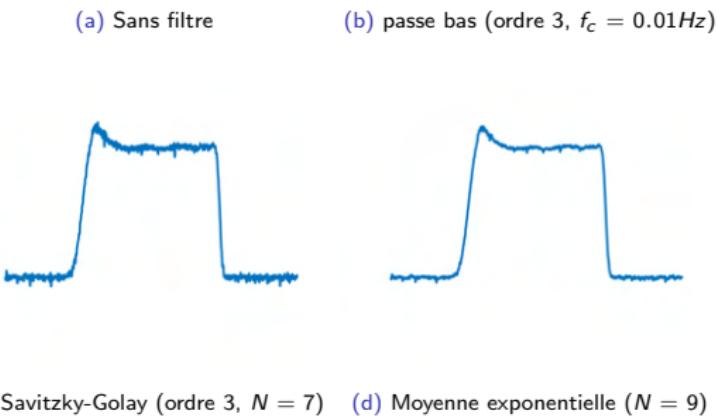
(c) K-Nearest Neighbors (KNN)



(d) MLP

Prétraitement : filtres appliqués

- **Filtre passe-bas⁷** Supprime les variations rapides (bruit) et rend le signal plus fluide.
- **Filtre de Savitzky-Golay⁸** Lisse le signal tout en conservant les détails comme les pics.
- **Moyenne exponentielle⁹** Réduit les fortes variations du signal de manière progressive.



⁷Koray S Erer (2007). "Adaptive usage of the Butterworth digital filter". In: *Journal of biomechanics* 40.13, pp. 2934–2943

⁸Ronald W Schafer (2011). "What is a savitzky-golay filter?" In: *IEEE Signal processing magazine* 28.4, pp. 111–117

⁹Masafumi Nakano, Akihiko Takahashi, and Soichiro Takahashi (2017). "Generalized exponential moving average (EMA) model with particle filtering and anomaly detection". In: *Expert Systems with Applications* 73, pp. 187–200

Pipeline d'apprentissage en Python

- ① Chargement des fichiers de la base de données
- ② Filtrage des signaux (passe-bas, Savitzky-Golay, EMA)
- ③ Segmentation des séquences de gestes (exclusion de *Neutral*)
- ④ Extraction des caractéristiques (moyenne, min, max, écart-type, A17, A18)
- ⑤ Constitution du jeu de données (X, y)
- ⑥ Apprentissage et évaluation des modèles

Code disponible sur : <https://github.com/fallmadou/LinLED-project/tree/main/Algorithm>

Chaîne de traitement des données

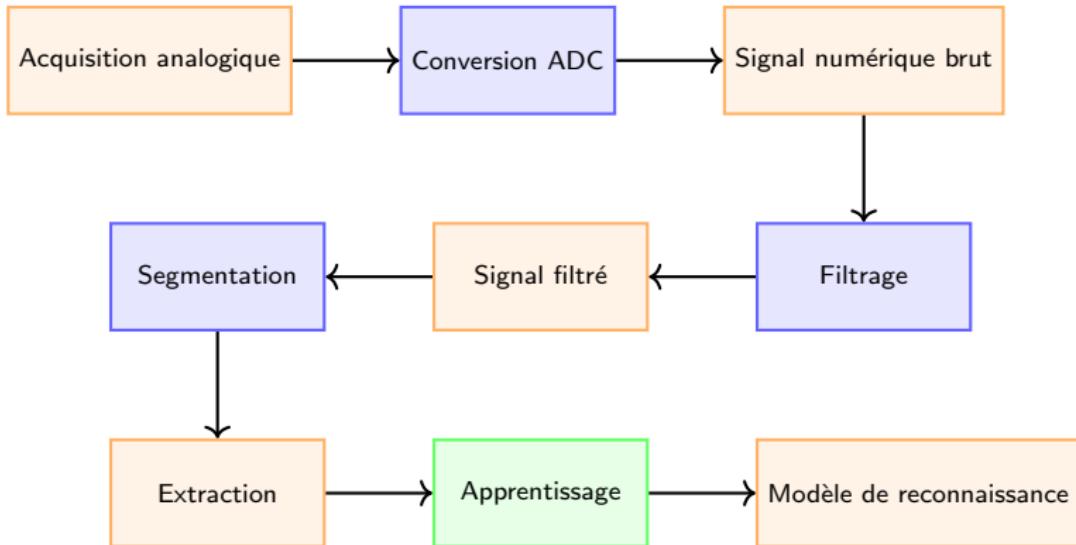


Figure: Chaîne de traitement : de l'acquisition à l'apprentissage

Analyse des performances – KNN

K-Nearest Neighbors (KNN)

Hyperparamètres :
n_neighbors = 5
weights fixée à "distance"
p = 1

Avantages

Simple à implémenter - Sensible au bruit

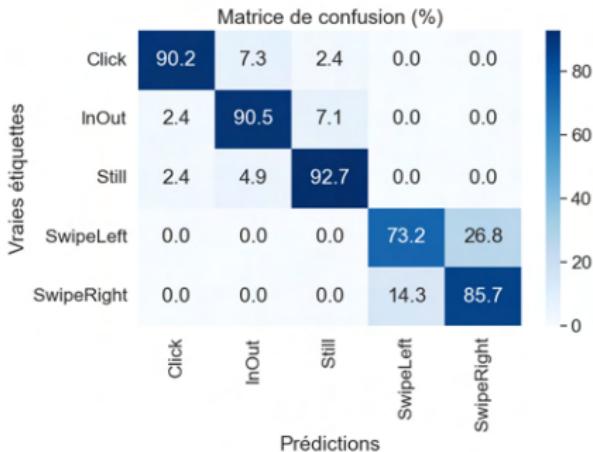


Figure: Avec filtre EMA

Filtre	Paramètres	Précision (%)	Tps inf. (ms/geste)
Sans filtrage	-	84,06	0,167
Passe-bas	ordre 3; $f_c = 0,9$ Hz	84,06	0,222
Savitzky-Golay	ordre 3; $N = 7$	83,09	0,111
EMA	$N = 6$	86,47	0,306

Analyse des performances – RF

Random Forest (RF)

Hyperparamètres :
n_estimators = 50
max_depth fixée à "None"
min_samples_split = 2
min_samples_leaf = 1.

Avantages

Robuste au bruit - Facile à embarquer

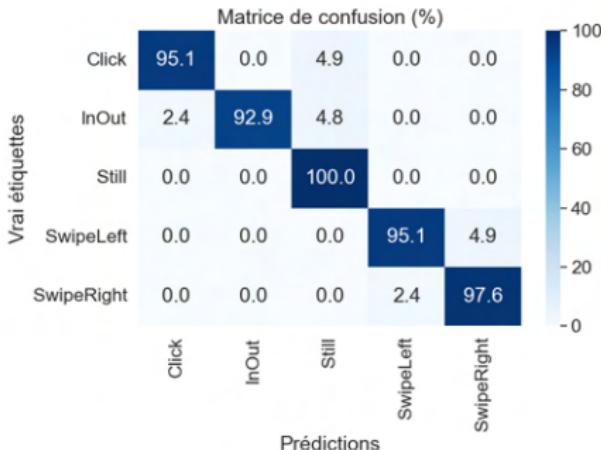


Figure: Avec filtre EMA

Filtre	Paramètres	Précision (%)	Tps inf. (ms/geste)
Sans filtrage	-	88,41	0,082
Passe-bas	ordre 3; $f_c = 0,9$ Hz	89,37	0,031
Savitzky-Golay	ordre 3; $N = 7$	91,79	0,048
EMA	$N = 5$	96,14	0,075

Analyse des performances – HGB

HistGradientBoosting (HGB)

Hyperparamètres :

```
learning_rate = 0.5  
max_iter = 50  
max_depth fixée à "None"  
l2_regularization = 0.
```

Avantages

Très précis – légèrement plus complexe

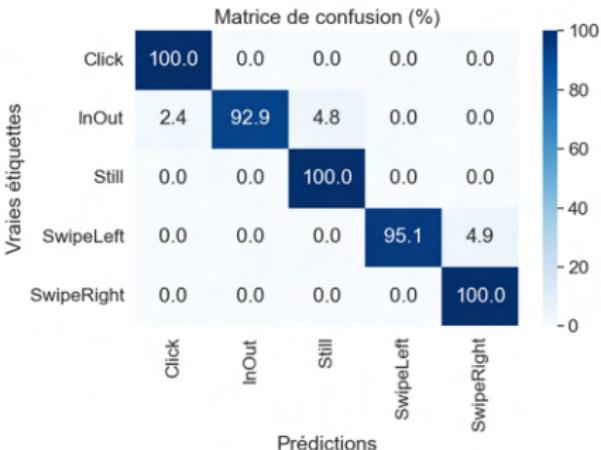


Figure: Avec filtre EMA

Filtre	Paramètres	Précision (%)	Tps inf. (ms/geste)
Sans filtrage	-	92,75	0,112
Passe-bas	ordre 3; $f_c = 0,9$ Hz	94,20	0,087
Savitzky-Golay	ordre 5; $N = 7$	92,75	0,069
EMA	$N = 5$	98,07	0,068

Analyse des performances – MLP

Perceptron Multi-Couche (MLP)

Hyperparamètres :
dropout_rate = 0,1
learning_rate = 0,001
neurones : 16 – 8
epochs = 50
batch_size = 16

Avantages

Précision élevée – Apprentissage flexible

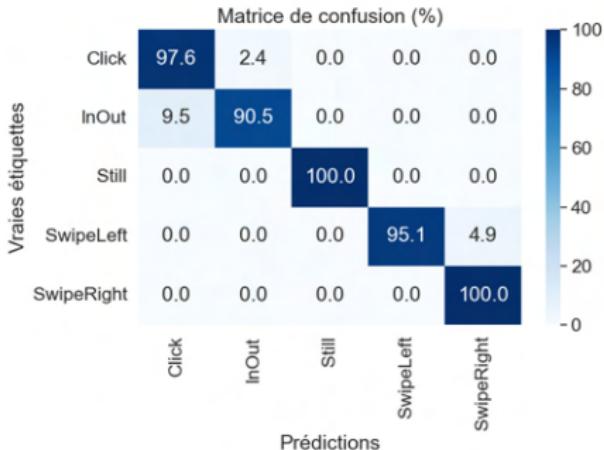


Figure: Avec filtre EMA

Filtre	Paramètres	Précision (%)	Tps inf. (ms/geste)
Sans filtrage	-	90,34	1,257
Passe-bas	ordre 3; $f_c = 0,5$ Hz	92,75	0,116
Savitzky-Golay	ordre 3; $N = 7$	90,34	1,200
EMA	$N = 9$	96,62	1,533

Sélection du modèle final

Pour sélectionner le meilleur modèle, plusieurs aspects¹⁰ ont été pris en compte :

- La précision pour bien reconnaître les gestes.
- La rapidité d'apprentissage.
- La vitesse de prédiction.
- La robustesse face au bruit dans les données.
- La simplicité d'implémentation et d'utilisation.

Choix final : Random Forest + filtre passe-bas

- Précision élevée (90%)
- Robuste aux signaux bruités
- Temps de prédiction rapide
- Facile à embarquer dans un microcontrôleur¹¹

¹⁰Zafar Takhirov et al. (2017). "Field of groves: An energy-efficient random forest". In: *arXiv preprint arXiv:1704.02978*

¹¹Simon Koschel et al. (2023). "Fast inference of tree ensembles on arm devices". In: *arXiv preprint arXiv:2305.08579*

Perspectives et suites du projet

- Élargir la base de données pour améliorer la robustesse et la généralisation du modèle.
- Déploiement sur microcontrôleur pour une utilisation en temps réel :
 - IHM dans la voiture
 - Optimisation pour du système embarqué
 - Communication avec l'IHM du véhicule



Figure: IHM

Merci pour votre attention !