

RAPPORT DE STAGE

ANALYSE DES MOUVEMENTS DE LA SOURIS À L'AIDE DES RÉSEAUX DE NEURONES À APPRENTISSAGE PROFOND

Étudiante:
Rûmeysa CAN

Tuteur de Stage:
Anton DOGADOV

Enseignant référent :
Thibault LEGIGAN



Table des matières

1. Introduction

1.1 Contexte général

1.2 Problématique

1.3 Objectifs du stage

1.4 Approche méthodologique

2. Méthodes

2.1 Les ondes corticales

2.2 DeepLabCut : suivi du comportement

2.3 Réseaux de neurones pour la classification d'images

2.4 Résultats expérimentaux

- Résultats avec ResNet-101
- Résultats avec EfficientNet-B0
- Résultats avec ResNet-50
- Comparaison des performances

3. Discussion

3.1 Interprétation des résultats

4. Conclusion

5. Bibliographie / Références

1. Introduction

1.1 Contexte

L'équipe de Daniel E. Shulz travaille sur le traitement des informations sensori-motrices et la plasticité cérébrale chez les rongeurs. Ils étudient comment les neurones traitent ces informations en utilisant différentes méthodes comme l'électrophysiologie, l'imagerie, l'optogénétique et des tests comportementaux. Leur but est de comprendre comment le cerveau organise ces fonctions et comment il apprend à partir des sensations et des mouvements. Ils s'intéressent notamment au codage sensoriel, à la prédiction des sensations, à l'apprentissage sensorimoteur et à la création de neuroprothèses. En utilisant la souris comme modèle, ils cherchent à mieux comprendre les mécanismes cérébraux liés à ces processus.



1.2 Problématique

Comment le cerveau, en particulier le cortex sensorimoteur, organise-t-il et modifie-t-il son activité pour permettre à un animal d'apprendre à générer des ondes cérébrales spécifiques en réponse à une récompense, grâce à l'usage de ses mouvements corporels ?

1.3 Les objectifs

Cette approche méthodologique ouvre une voie unique pour mieux comprendre comment le réseau de neurones coordonne le flux d'informations au sein des réseaux corticaux lorsque l'animal exécute activement une tâche.

Dans ce rapport de stage, je vais détailler les expériences réalisées et les résultats obtenus, ainsi que les implications de ces découvertes pour le domaine de la recherche en neurosciences et des interfaces cerveau-machine.

Nous allons essayer de comprendre comment la souris apprend à montrer une activité cérébrale pour recevoir une récompense avec l'aide des mouvements de son corps. Ces connaissances pourraient ouvrir de nouvelles perspectives pour les technologies futures permettant une meilleure interaction entre l'homme et la machine en utilisant l'activité cérébrale.

1.4 Approche méthodologique

Pour répondre à la problématique, plusieurs méthodes ont été utilisées pendant mon stage.

D'abord, on a suivi en temps réel les ondes cérébrales chez la souris grâce à une protéine appelée GCaMP6f, qui rend visible l'activité des neurones avec une caméra rapide. Cela permet de voir comment les ondes se déplacent dans différentes parties du cortex, comme les zones sensorielles et motrices.

Ensuite, on a utilisé DeepLabCut, un outil basé sur l'intelligence artificielle, pour suivre précisément les mouvements des souris. Cet outil aide à analyser le comportement de l'animal sans avoir à le faire manuellement.

Enfin, pour analyser les images obtenues, j'ai travaillé avec différents réseaux de neurones (ResNet et EfficientNet) qui servent à classer et reconnaître les motifs dans ces images.

En combinant ces techniques, on a pu étudier comment les mouvements du corps sont liés à l'activité cérébrale et à l'apprentissage chez la souris.

2. Méthodes

2.1 Interface cerveau-machine

Dans cette étude, l'équipe s'est concentrée sur une échelle de mesure mésoscopique, c'est-à-dire entre 0,1 et 1 mm, pour observer les dynamiques de l'activité cérébrale. Ils ont découvert que des ondes cérébrales synchronisées se propagent à travers le cortex sous deux formes principales : soit sous forme de zones localisées appelées « blobs » (dépolarisations stationnaires), soit sous forme d'ondes itinérantes qui se déplacent à travers le réseau cortical. Ces ondes semblent jouer un rôle important dans la transmission et le traitement des informations, mais leur fonctionnement reste encore partiellement compris.

Pour mieux comprendre ces ondes et voir s'il est possible de les contrôler, ils ont mis en place une expérience avec des souris maintenues immobiles. Ces souris, génétiquement modifiées pour exprimer une protéine fluorescente appelée GCaMP6f, permettent de visualiser l'activité cérébrale via des variations de calcium dans les neurones. Une fenêtre optique de 6 mm a été posée sur leur cerveau, au niveau des zones somatosensorielles et motrices primaires gauches.

Grâce à cette technique d'imagerie du calcium, ils ont pu suivre en temps réel la trajectoire des ondes mésoscopiques qui se déplacent dans le cortex. Ils ont ensuite conditionné la remise d'une récompense en eau à la production d'ondes spécifiques suivant certaines trajectoires.

Ces résultats montrent que les souris peuvent apprendre à contrôler ces ondes cérébrales, ce qui ouvre des perspectives prometteuses pour le développement d'interfaces cerveau-machine, où l'activité cérébrale serait utilisée pour commander des dispositifs externes.

2.2 GcaMP

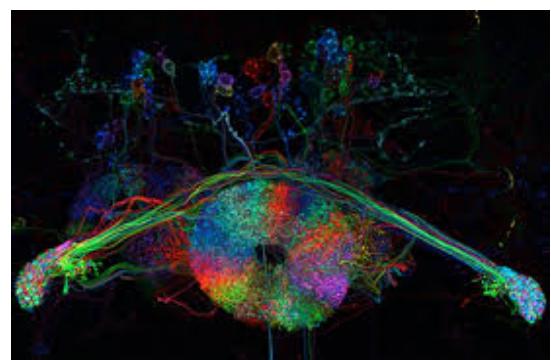
GCaMP (Indicateur de Calcium Génétique) est une protéine fluorescente qui permet de surveiller en temps réel les variations de calcium à l'intérieur des cellules neuronales. Elle a été développée par Junichi Nakai et est très utilisée en neurosciences pour étudier l'activité cérébrale.

Le calcium joue un rôle important dans la communication entre les neurones. Quand un neurone est actif, la concentration de calcium augmente à l'intérieur de la cellule.

GCaMP est une protéine modifiée génétiquement qui émet de la fluorescence lorsqu'elle se lie au calcium. En combinant cette protéine avec des techniques d'imagerie adaptées, les chercheurs peuvent détecter ces variations de fluorescence et les interpréter comme un signe d'activité neuronale.

En d'autres termes, GCaMP fonctionne comme un capteur fluorescent qui permet de visualiser l'activité cérébrale en temps réel. Cette technologie offre ainsi la possibilité d'étudier le cerveau dans différents contextes, comme la réponse à des stimulations sensorielles, la dynamique cérébrale pendant un comportement, ou encore l'organisation des réseaux neuronaux impliqués dans des fonctions cognitives.

GCaMP a changé la façon dont les neuroscientifiques étudient le cerveau, car elle permet des mesures non invasives et rapides de l'activité cérébrale. Cela ouvre la voie à une meilleure



compréhension des mécanismes du cerveau et des processus liés au comportement et à la cognition.

2.3 Les ondes corticales

Pendant mon stage, j'ai eu la chance de travailler sur un projet passionnant qui consistait à suivre en temps réel les ondes corticales chez la souris.

Pour cela, une installation a été mise en place afin de détecter ces ondes grâce à la protéine GCaMP6f, avec un enregistrement à 100 images par seconde. Ils ont réussi à réduire les interférences liées aux signaux hémodynamiques en utilisant un éclairage en deux couleurs, bleu (480 nm) et vert (530 nm).

Les souris étaient privées d'eau jusqu'à ce qu'elles génèrent des ondes dans leur cortex cérébral. Le but de l'entraînement était de leur apprendre à produire des ondes qui se déplacent d'une zone de départ vers une zone cible spécifique, soit dans le cortex sensoriel primaire (S1), soit dans le cortex moteur (M1) (voir Fig. 1b, c).

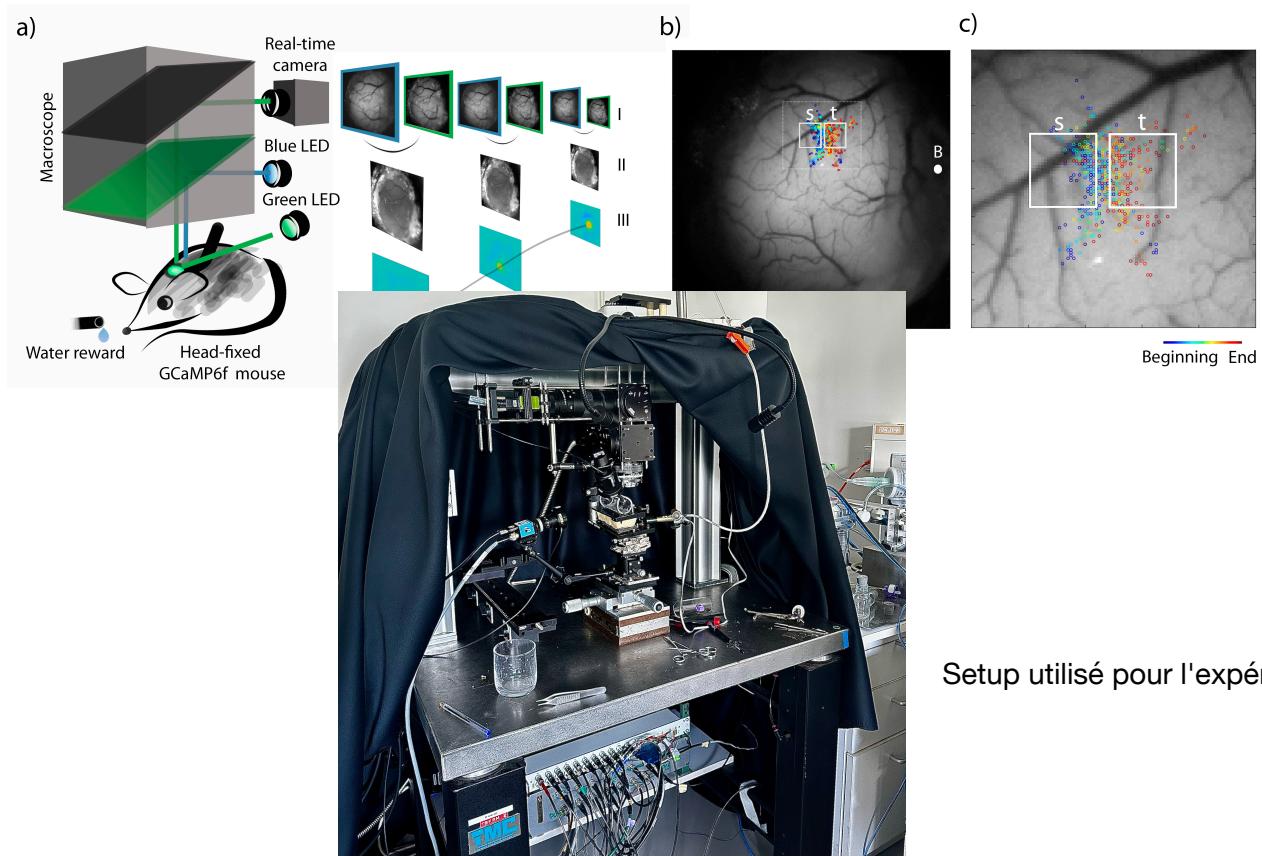
Ces ondes avaient une taille mésoscopique, c'est-à-dire comprise entre 0,1 et 1 mm.

L'objectif était de récompenser les souris lorsqu'elles produisaient ces ondes particulières. Une fenêtre de 500 ms était définie pour chaque onde : si la souris léchait pendant ce temps, elle recevait une récompense en eau.

Pour mesurer et suivre les ondes corticales, une caméra 2D a été utilisée. Cette technique a permis d'extraire les ondes en mouvement dans le champ de vision, couvrant donc la gamme de tailles entre 0,1 et 1 mm.

Le suivi des coordonnées x et y de chaque onde identifiée a été réalisé sur une certaine période, ce qui a permis de capturer l'évolution de l'activité cérébrale dans le temps.

L'entraînement a été réalisé sur plusieurs jours afin que les souris s'habituent au protocole expérimental et développent la capacité à générer des ondes mésoscopiques.



2.4 DeepLabCut

DeepLabCut (DLC) est une plateforme avancée d'apprentissage automatique spécialement conçue pour analyser les mouvements et la posture, aussi bien chez les animaux que chez les humains.

Pendant mon stage, j'ai découvert que DLC est une solution puissante et polyvalente, qui permet de réaliser une analyse automatisée et très précise du suivi des mouvements. Son fonctionnement repose sur le deep learning, utilisant des réseaux de neurones artificiels pour suivre des points d'intérêt précis dans des vidéos.

Pour commencer, j'ai créé un jeu de données annotées manuellement, en identifiant et en marquant huit points d'intérêt sur chaque image vidéo : les pattes avant et arrière, le nez, la queue, ainsi que deux supports.

Ensuite, j'ai entraîné le modèle DLC avec ce jeu de données, ce qui a permis au système de reconnaître et de suivre automatiquement ces points dans de nouvelles vidéos.

DeepLabCut s'est révélé être un outil très utile, offrant une alternative rapide et fiable aux méthodes traditionnelles souvent complexes d'analyse du mouvement. Son utilisation m'a permis de simplifier le traitement des données, ce qui m'a donné plus de temps pour interpréter les résultats et approfondir mes analyses.

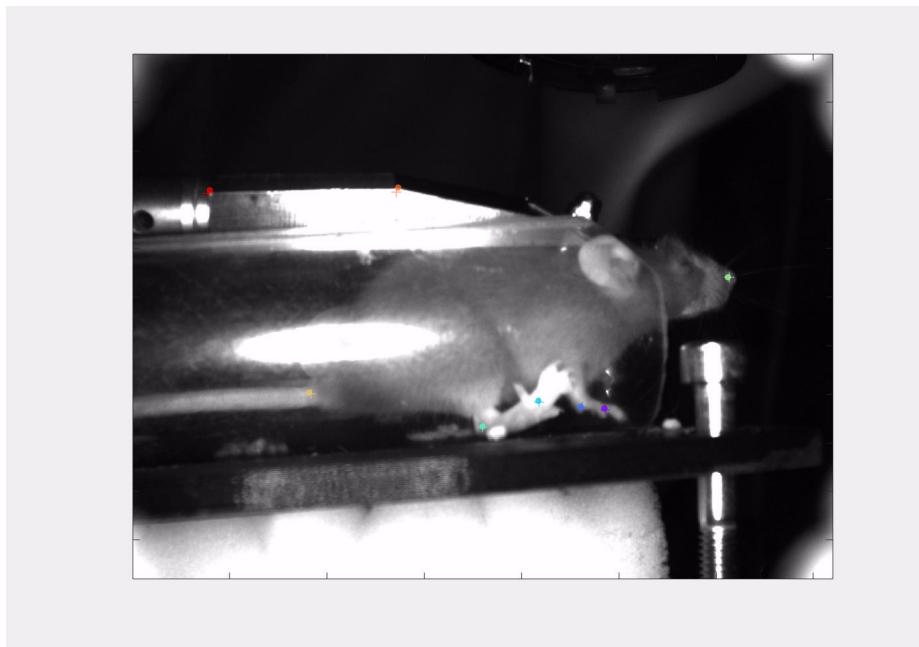


Image labellisée sur DeepLabCut

2.5 L'utilisation de DeepLabCut (DLC)

Pendant mon stage, j'ai eu l'occasion d'utiliser DeepLabCut (DLC) pour analyser précisément les mouvements dans le cadre de mon projet.

Pour commencer, j'ai collecté plusieurs vidéos montrant les mouvements spécifiques que je voulais étudier. Ensuite, j'ai annoté ces vidéos à la main en repérant et marquant les points clés du mouvement, comme les articulations ou d'autres repères anatomiques.

Avec cette base de données annotée, j'ai entraîné le modèle DLC, qui repose sur un réseau de neurones à convolution. Cette étape était importante, car c'est pendant l'entraînement que le système apprend à détecter automatiquement les points d'intérêt dans de nouvelles vidéos.

Après l'entraînement, j'ai testé le modèle sur des vidéos de validation, non utilisées lors de l'apprentissage, pour vérifier sa précision. J'ai ensuite ajusté les paramètres et refait plusieurs entraînements jusqu'à obtenir des résultats satisfaisants.

Une fois le modèle correctement entraîné et validé, j'ai pu utiliser DLC pour suivre automatiquement les points d'intérêt dans de nouvelles vidéos. Cela m'a permis de réaliser une analyse rapide et automatisée, réduisant ainsi considérablement le temps nécessaire pour traiter un grand volume de données.

Les résultats obtenus ont apporté des informations détaillées sur les mouvements étudiés, enrichissant mon analyse et améliorant la compréhension des processus que je cherchais à étudier.

L'utilisation de DeepLabCut a donc été un élément essentiel de notre projet, car elle a permis d'obtenir des analyses de mouvement précises et fiables, tout en me libérant du temps pour approfondir l'interprétation des résultats et développer de nouvelles connaissances.

2.6 Utilisation de MATLAB

Au cours de mon stage, j'ai également utilisé MATLAB pour analyser les données issues des vidéos de souris. Après avoir annoté manuellement les points d'intérêt sur les images extraites des vidéos grâce à DeepLabCut, ces données labellisées ont été utilisées dans MATLAB pour effectuer des analyses plus approfondies.

Le code écrit par mon tuteur permettait de traiter les coordonnées des points suivis (comme les pattes, le nez ou la queue) afin de visualiser les mouvements de la souris sous forme de courbes. Cela nous a permis d'observer l'évolution des positions dans le temps et de mieux comprendre les comportements moteurs liés à l'activité cérébrale.

MATLAB a donc joué un rôle essentiel dans la visualisation des résultats, en transformant des données brutes issues de DeepLabCut en graphiques interprétables. Cette étape était cruciale pour interpréter les comportements moteurs des souris dans le contexte des ondes corticales enregistrées.

2.7 Les réseaux de neurones

Au cours de mon stage, j'ai eu l'opportunité d'explorer différents réseaux de neurones utilisés pour la classification d'images, notamment ResNet-50, ResNet-101 et EfficientNet-B0. Ces modèles font partie des architectures les plus avancées dans le domaine de l'apprentissage automatique et de la vision par ordinateur.

Les réseaux ResNet-50 et ResNet-101 appartiennent à la famille des réseaux résiduels (ResNets), reconnus pour leur grande profondeur. Grâce à leur structure particulière, ils permettent d'entraîner des modèles très profonds sans perte de performance. Cela les rend particulièrement adaptés aux tâches complexes de reconnaissance d'images.

De son côté, EfficientNet-B0 est un modèle plus léger, conçu pour optimiser le rapport entre précision et consommation de ressources (mémoire et temps de calcul). Il parvient à de très bonnes performances tout en étant plus rapide et moins gourmand que les ResNets, ce qui le rend intéressant lorsque les ressources informatiques sont limitées.

J'ai réalisé plusieurs essais avec ces trois réseaux, en les entraînant sur différentes bases de données d'images. ResNet-101 a souvent donné les meilleurs résultats en termes de précision, probablement grâce à sa plus grande profondeur. Cependant, EfficientNet-B0 a montré un excellent compromis entre performance et rapidité. Enfin, ResNet-50 s'est avéré robuste et efficace, tout en étant moins exigeant que ResNet-101.

Cette exploration des réseaux de neurones m'a permis de mieux comprendre les avantages et les limites de chaque architecture, et de choisir le modèle le plus adapté en fonction des contraintes du projet.

3. Discussion

3.1 Interprétation des résultats

Résultats avec ResNet-101

Les résultats obtenus avec le réseau ResNet-101 donnent des indications intéressantes sur ses performances. Après 150 000 itérations, l'erreur d'entraînement était de 3,54, tandis que l'erreur de test était de 3,53. Cette proximité entre les deux erreurs suggère une certaine stabilité du modèle et une bonne généralisation sur les données non vues.

Lorsque le nombre d'itérations a été porté à 500 000, l'erreur d'entraînement a légèrement diminué à 3,41, mais l'erreur de test est restée quasiment inchangée (3,54). Cela pourrait indiquer que le modèle a atteint un plateau, et qu'augmenter davantage les itérations n'apporte plus d'amélioration significative sur des données nouvelles.

Résultats - ResNet101							
DLC_resnet101_MOUSEPROJECT_Jul21_shuffle1_150000-results							
Training iterations	% Training dataset	Shuffle number	Train error (px)	Test error (px)	p-cutoff used	Train error with p-cutoff	Test error with p-cutoff
150000	95	1	3.54	3.53	0.6	3.52	3.53
DLC_resnet101_MOUSEPROJECT_Jul21_shuffle1_500000-results							
Training iterations	% Training dataset	Shuffle number	Train error (px)	Test error (px)	p-cutoff used	Train error with p-cutoff	Test error with p-cutoff
500000	95	1	3.41	3.54	0.6	3.41	3.54

Résultats avec EfficientNet-B0

Concernant le réseau EfficientNet-B0, l'entraînement a donné de bons résultats dans un premier temps : après 200 000 itérations, l'erreur d'entraînement était de 2,69. Toutefois, après 450 000 itérations, une forte hausse de l'erreur d'entraînement a été observée, atteignant 13,02.

Cette dégradation des performances, accompagnée d'une augmentation de l'erreur de test, suggère un surapprentissage (overfitting). Le modèle semble s'être trop spécialisé sur les données d'entraînement, perdant ainsi sa capacité à bien généraliser. Ce comportement met en évidence la nécessité de trouver un équilibre entre la durée d'entraînement et la capacité de généralisation du réseau.

Résultats - EfficientNet-B0

DLC_effnet_b0_MOUSEPROJECT_Jul21_shuffle1_200000-results

Training iterations	% Training dataset	Shuffle number	Train error (px)	Test error (px)	p-cutoff used	Train error with p-cutoff	Test error with p-cutoff
200000	95	1	2.69	3.91	0.6	2.69	3.91

DLC_effnet_b0_MOUSEPROJECT_Jul21_shuffle1_450000-results

Training iterations	% Training dataset	Shuffle number	Train error (px)	Test error (px)	p-cutoff used	Train error with p-cutoff	Test error with p-cutoff
450000	95	1	13.02	11.15	0.6	9.24	6.68

Résultats avec ResNet-50

Le réseau ResNet-50 a été entraîné sur 1 030 000 itérations, et les résultats ont été particulièrement stables. L'erreur d'entraînement a atteint 3,23, et l'erreur de test était très proche, à 3,27.

Ces résultats montrent que ResNet-50 a bien appris les données d'entraînement, tout en maintenant une très bonne performance sur des données nouvelles. Cela suggère une capacité de généralisation efficace et une robustesse du modèle, même après un grand nombre d'itérations.

Résultats - ResNet50

DLC_resnet50_MOUSEPROJECT_Jul21_shuffle1_1030000-results

Training iterations	% Training dataset	Shuffle number	Train error (px)	Test error (px)	p-cutoff used	Train error with p-cutoff	Test error with p-cutoff
1030000	95	1	3.23	3.27	0.6	3.23	3.27

Après avoir comparé les performances des trois modèles, ResNet-50 se distingue comme le plus fiable dans le cadre de cette étude. Il offre un bon compromis entre précision, stabilité et capacité de généralisation.

Même si chaque modèle présente des avantages spécifiques — par exemple la profondeur de ResNet-101 ou la légèreté d'EfficientNet-B0 — ResNet-50 semble être le plus adapté à nos besoins pour notre expérience.

Cela dit, le choix optimal du réseau dépendra toujours du contexte, de la nature des données, des ressources disponibles et des objectifs recherchés.

4. Conclusion

Ce stage, le tout premier de mon parcours, a été une expérience extrêmement enrichissante et formatrice. Il m'a permis de découvrir concrètement le monde de la recherche en neurosciences et d'explorer des outils technologiques avancés tels que l'imagerie calcique avec GCaMP6f, le logiciel DeepLabCut, MATLAB, ainsi que les réseaux de neurones pour la classification d'images.

J'ai pu suivre des ondes corticales, analyser des mouvements de souris et comparer les performances de modèles d'intelligence artificielle, ce qui m'a donné un aperçu concret des applications de la neurotechnologie.

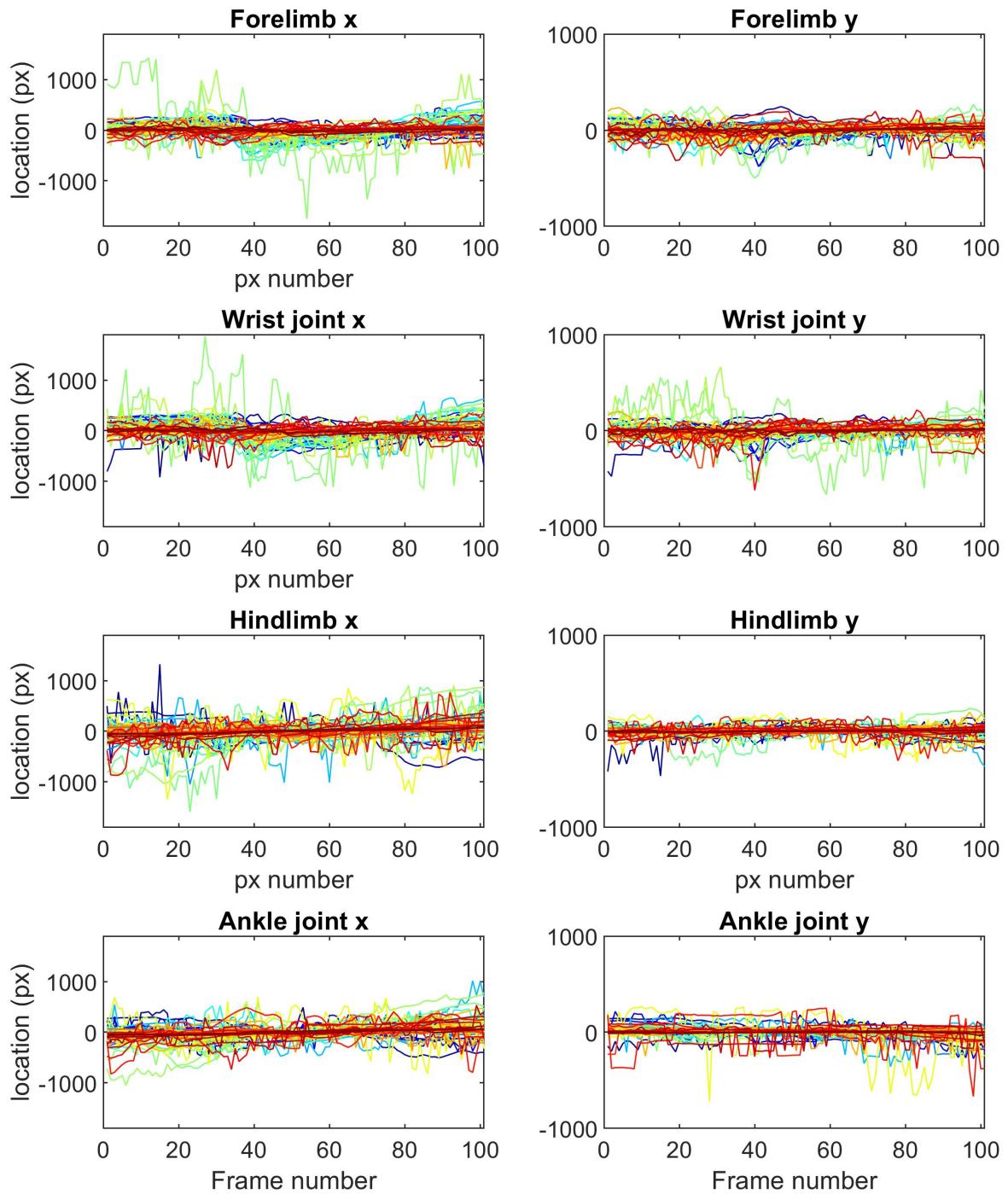
Ce stage a également confirmé mon envie de devenir neuroingénieur. J'ai compris à quel point l'interaction entre le cerveau, l'informatique et les mathématiques peut ouvrir de nouvelles perspectives pour mieux comprendre et, peut-être un jour, interagir avec l'activité cérébrale de manière directe et précise. J'ai encore beaucoup à apprendre, mais cette première expérience m'a donné des bases solides et une forte motivation pour la suite.

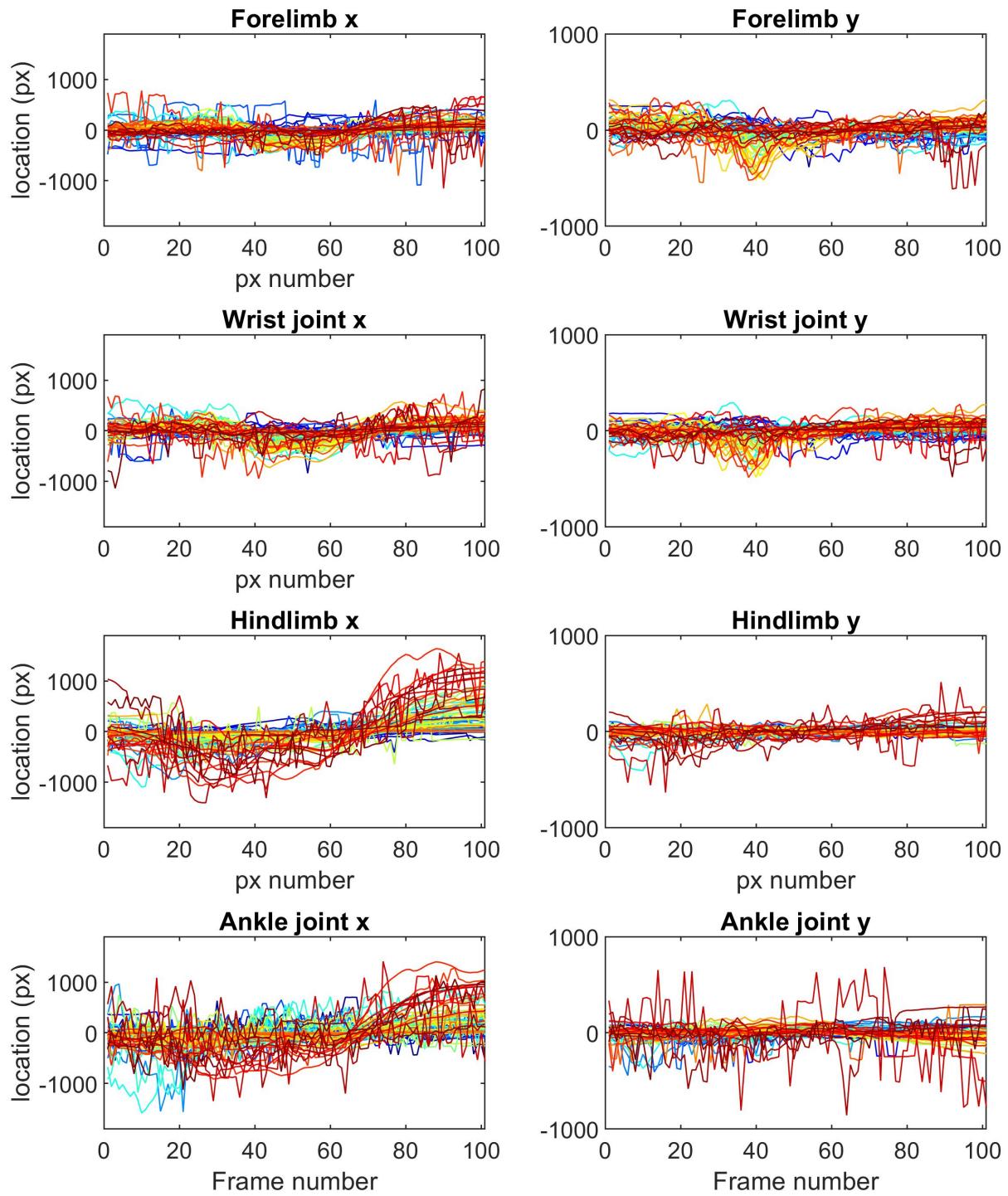
Je tiens à remercier chaleureusement toute l'équipe de recherche qui m'a accueilli, encadré et guidé tout au long de ce stage. Leur disponibilité, leurs conseils et leur bienveillance ont grandement contribué à la réussite de cette expérience. Grâce à eux, j'ai pu progresser et prendre confiance dans un environnement scientifique stimulant.

5. Bibliographie / Références

- [1] L. Muller et al., Cortical travelling waves: Mechanisms and computational principles, *Nat Rev Neurosci.* 19 (2018) 255–268. <https://doi.org/10.1038/nrn.2018.20>.
- [2] Y. Ma et al. Resting-state hemodynamics are spatiotemporally coupled to synchronized and symmetric neural activity in excitatory neurons, *Proc Natl Acad Sci U S A.* 113 (2016) E8463–E8471. <https://doi.org/10.1073/pnas.1525369113>.
- [3] A. A. Dogadov, D. E. Shulz, I. Ferezou, V. Ego-Stengel et L. Estebanez, Operant conditioning of cortical waves through a brain-machine interface, *bioRxiv* (2024). <https://doi.org/10.1101/2024.09.10.612344>

6. Annexes





M Thibaut Legigan,
Maître de conférences
Université Sorbonne Paris Nord

Saclay,
Le 7 septembre 2023

Objet : Lettre d'évaluation pour Rûmeysa Can

Monsieur Legigan,

Rûmeysa Can a effectué un stage sous ma supervision du 10/07/2023 au 04/08/2023. Durant son stage, Rûmeysa a utilisé un réseau de neurones d'apprentissage profond pour étudier les mouvements de la souris dans le cadre d'un projet visant à étudier l'activité corticale d'une souris transgénique exprimant le rapporteur de calcium GCaMP6f dans les neurones corticaux excitateurs. Dans son travail, Rûmeysa a utilisé la boîte à outils DeepLabCut pour Python ainsi que Matlab pour analyser les données d'imagerie grande vitesse acquises sur ces souris. Elle a démontré ses connaissances en biologie et en programmation et a pu se familiariser avec des outils génétiques tels que Cre-Lox, ainsi qu'observer une procédure chirurgicale sur des souris et l'acquisition de données comportementales et corticales. Rûmeysa a pu utiliser les méthodes les plus avancées de traitement vidéo appliquées à des données réelles. Rûmeysa s'est montrée très travailleuse et motivée, et les résultats de son stage sont directement utilisés dans notre projet de recherche.

Veuillez agréer, Monsieur, mes salutations distinguées.



Anton Dogadov, PhD

Chercheur postdoctoral
Institut des Neurosciences Paris-Saclay (NeuroPSI)
Département de Neurosciences Intégratives et Computationnelles (ICN)

