



**CENTRALE
MARSEILLE**

2A Promo entrante 2017
Parcours S8 SISN
Sciences de l'Information et
Société Numérique

par Marion FISCHER

RAPPORT DE STAGE

Modélisation du traitement visuel d'une image fovéale



Inserm

La science pour la santé
From science to health

Institut de Neurosciences des Systèmes
UMR 1106 INSERM

Encadrée par M. Emmanuel Daucé

03 Juin 2019 - 05 Août 2019

Remerciements

Je tiens à remercier tout particulièrement M. Emmanuel Daucé, mon tuteur, pour toutes les explications qu'il m'a données, sa patience et sa gentillesse.

Merci également à M. Laurent Perrinet pour avoir réfléchi avec M. Daucé et moi au fil conducteur de mon stage, et aux différentes missions qui pouvaient m'être attribuées.

Je remercie Véronique Ayala pour s'être occupé pour moi de toute la partie administrative du stage au niveau du laboratoire.

Merci à mes collègues de bureau, Lisa Otten, Anastasia Eghiaian-Loman, et Heba Elseedy pour leur bonne humeur chaque jour.

Un grand merci également à tout le reste du personnel du laboratoire pour leur accueil amical et les agréables pauses-café de 13h !

Résumé

Durant deux mois, j'ai eu l'opportunité de travailler à l'Institut de Neurosciences des Systèmes basé à la Timone à Marseille. J'ai pu aider M. Daucé et M. Perrinet sur leur projet conjoint WhereIsMyMnist qui consiste à modéliser la vision humaine pour tenter de résoudre des problèmes de classification. Ma principale mission durant ce stage a été de rajouter des fonctionnalités à un code écrit en Python modélisant la vision humaine. J'ai aussi entraîné une multitude de réseaux de neurones pour tester leurs capacités de discrimination dans des problèmes de classification et j'ai réfléchi sur les moyens d'optimiser la place en mémoire du programme.

Abstract

During two months, I had the opportunity to work in the Institut de Neurosciences des Systèmes which is localized in Marseille in France. I have helped M. Daucé and M. Perrinet with one of their projects called WhereIsMyMnist. This project consists in the modeling of the human vision to solve classifications problems. My main mission during this internship was to add functionalities to a code written in Python. I also trained lots of neuronal networks to test their abilities in classifications problems and I have thought about ways to optimize the memory used by the whole program.

Mots clés

Neurosciences Réseaux de neurones Apprentissage Classification
Pytorch Python Deep Learning

Glossaire

Recherche translationnelle : (ang. translational research) transformation des découvertes fondamentales en applications cliniques

AMU : Université d'Aix-Marseille

INSERM : Institut National de la Santé Et de la Recherche Médicale

Neurosciences computationnelles : champ de recherche des neurosciences qui tente de comprendre le fonctionnement du cerveau grâce à l'utilisation de modèles des sciences informatiques, auxquels on combine des données expérimentales et des simulations numériques.

CNRS : Centre National de la Recherche Scientifique

Electrophysiologie : étude des phénomènes électriques et électrochimiques qui se produisent dans les neurones et les fibres musculaires

Table des matières

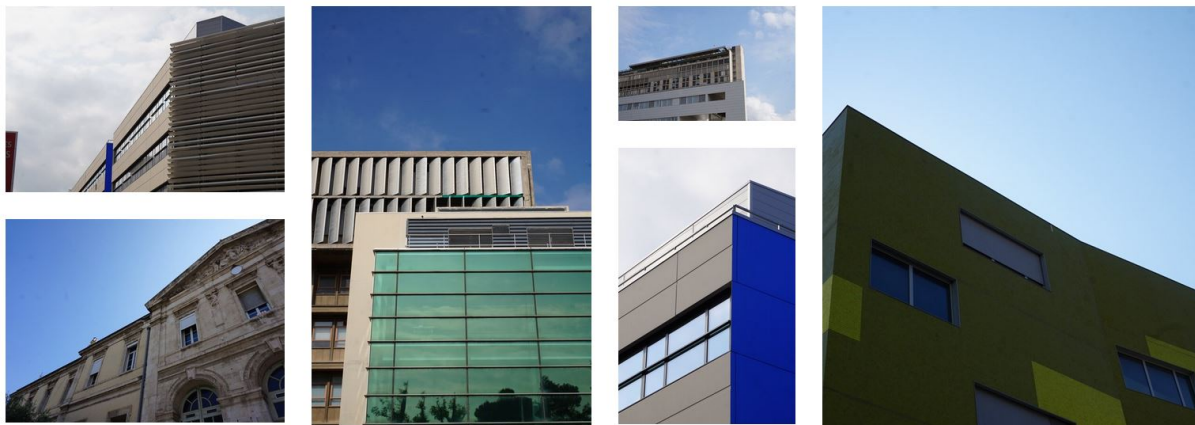
Remerciements	2
Résumé	3
Abstract	3
Mots clés	4
Glossaire	4
Introduction.....	6
Présentation du laboratoire et de ma mission	7
Le laboratoire	7
Quelques chiffres et éléments clés.....	7
Les différents groupes de recherche	8
Projet à long terme.....	9
Ma mission	10
Contexte	10
Figure représentant le mouvement effectué par l’algorithme [3]	11
Objectifs	12
Enjeux	12
Déroulement du stage.....	13
Les débuts.....	13
MNIST	13
The Chicago Face Database	16
Résultats	20
Conclusion	21
Bibliographie	22
Annexes	22

Introduction

Du 3 juin 2019 au 5 Août 2019, j'ai eu l'opportunité d'effectuer un stage dans le laboratoire Institut de Neurosciences des Systèmes basé à la Timone à Marseille. J'ai travaillé sur le projet appelé WhereIsMyMnist de M. Daucé, chercheur à l'Institut des Neurosciences des Systèmes (INS) et enseignant à l'Ecole Centrale de Marseille, et de M. Perrinet, chercheur à l'Institut de Neurosciences de la Timone. Ce projet, développé sous Python, cherche à imiter la vision humaine dans le cadre de problèmes de classification. La vision humaine est modélisée de la manière suivante : précise au centre et floue en périphérie, et des mouvements de saccade oculaire sont effectués entre des zones d'intérêt. Pour modéliser cela, des filtres Log Gabor sont utilisés et leur répartition est effectuée de manière à s'approcher le plus possible de la répartition des récepteurs de l'œil humain. Un code de plusieurs centaines de lignes était déjà écrit lors de mon arrivée, et ma tâche principale a été de le compléter pour lui ajouter de nouvelles fonctionnalités et de regarder l'influence que cela pouvait avoir sur des réseaux de neurones entraînés avec ces modifications. Les tâches de classification ont porté tout d'abord le premier mois sur des chiffres manuscrits (base de données MNIST), avant de se porter sur des classifications de genre à partir d'une base de données de visages (Chicago Face Database CFD). D'autres tâches ont aussi été effectuées comme une réflexion et des tests pratiques sur le stockage efficace des filtres nécessaires à la simulation de la vision humaine. L'intérêt de ce sujet est de voir si simuler la vision humaine ne serait pas plus économique en ressources pour effectuer des tâches de classification que les traitements actuellement disponibles pour le faire.

Présentation du laboratoire et de ma mission

Le laboratoire



Photos de l'Institut de Neurosciences des Systèmes à Marseille [1]

Quelques chiffres et éléments clés

Le laboratoire nommé Institut de Neurosciences des Systèmes (INS) se situe à la Timone dans la ville de Marseille. Dans les alentours se trouve également l'Institut de Neurosciences de la Timone (INT). Le directeur de l'INS est Viktor Jirsa. On trouve dans ce laboratoire plus de 90 chercheurs employés par Protisvalor, une filiale d'Aix-Marseille Université spécialisée dans l'accompagnement des chercheurs, et l'INSERM. L'INS présente trois avantages de taille, premièrement des recherches pluridisciplinaires, qui lui permettent d'avoir différents points de vue sur les dynamiques du cerveau. Les études peuvent aussi bien concerner le comportement de rats dans différentes situations que porter sur les neurosciences computationnelles. L'institut possède aussi un fort dynamisme en recherche translationnelle, et bénéficie de nombreuses installations, du fait de sa localisation à proximité immédiate d'un des plus grands hôpitaux français.

Les membres de l'INS peuvent être des chercheurs de l'INSERM, mais aussi des chercheurs du CNRS, ou des membres de l'établissement public de santé nommé AP-HM (Assistance publique – Hôpitaux de Marseille). Le programme de recherche de l'INS cible la compréhension des dynamiques complexes du cerveau, et ensemble, les chercheurs tentent de capitaliser les travaux expérimentaux,

les travaux théoriques, et les approches cliniques dans le but de comprendre les fonctions du cerveau mais aussi ses dysfonctionnements.

L'INS contient en son sein de nombreuses installations de recherche sur le cerveau, comme une installation permettant d'effectuer de la magnétoencéphalographie (MEG), mais aussi de l'électroencéphalographie (EEG), ainsi que de multiples laboratoires d'électrophysiologie.

Selon le Haut Conseil de l'Évaluation de la Recherche et de l'Enseignement Supérieur (HCERES), en 2017 : l'INS a des forces dans la modélisation, la théorie et la collection de données, a aussi de clairs potentiels en science translationnelle, et un énorme accès aux ressources.

Pour rester ouvert aux différents domaines liés aux neurosciences, le laboratoire accueille souvent des conférences, faites soit par une personne du laboratoire qui explique ses travaux, soit par une personne représentant un laboratoire étranger.

Les différents groupes de recherche

Les chercheurs de l'INS sont répartis en quatre équipes travaillant sur différentes thématiques.

L'équipe *Dynamical Brain Mapping* est dirigée par M. Christian Bénar, et cherche à développer des stratégies pour caractériser les dynamiques spatio-temporelles du réseau neuronal que ce soit dans le cadre de l'activité physiologique ou dans le cadre de maladies. Les stratégies les plus utilisées combinent la force de différentes technologies comme l'électroencéphalographie (EEG), la magnétoencéphalographie (MEG), l'imagerie par résonance magnétique fonctionnelle (fMRI), et les enregistrements simultanés EEG-MEG ou EEG-fMRI qui permettent d'enregistrer la même activité cérébrale sous différents points de vue. Ils comparent les résultats de ces méthodes non invasives aux résultats enregistrés directement dans les cerveaux via des électrodes des patients touchés d'épilepsie pour voir si on retrouve bien les mêmes résultats.

La seconde équipe s'appelle *Dynamics of Cognitive Processes Group* et est dirigée par M. Daniele Schön. Ils étudient les dynamiques spatio-temporelles correspondantes aux processus cognitifs. En particulier les changements de connectivité dans différents contextes : des changements lents comme lors de l'apprentissage, ou des changements rapides lors de prédictions. Ils s'intéressent aussi bien à des patients épileptiques qu'à des enfants dyslexiques, ou avec des problèmes auditifs. Ils essaient de développer des modèles cognitifs pour comprendre le processus du langage, ou anticiper les comportements ou le déclin de la mémoire. Et ils travaillent en particulier sur la récupération du langage via un entraînement lié aux rythmes et à la musique.

L'équipe *Physiology & physiopathology of neuronal networks* est dirigée par M. Christophe Bernard. Leur objectif principal est de comprendre comment les comportements physiologiques et pathologiques émergent des organisations et des réorganisations de la structure neuronale. Différents sujets sont étudiés actuellement, tels que la dynamique de la cellule dans l'apprentissage dans des conditions physiologiques et pathologiques, l'organisation anatomique des réseaux neuronaux chez des sujets sains et chez des sujets épileptiques, ou la dynamique des réseaux neuronaux et de la mémoire.

La quatrième équipe à laquelle j'ai été rattachée est nommée *Theoretical neuroscience group* et est dirigée par M. Viktor Jirsa. Cette équipe a été fondée en 1999 par M. Viktor Jirsa dans le Center for Complex Systems and Brain Sciences dans l'Université de Floride et se trouve à Aix-Marseille Université depuis 2006. Son objectif est d'avoir une compréhension plus profonde des mécanismes sous-jacents à l'émergence des fonctions cérébrales et de ses dysfonctionnements. Pour cela, une approche multiscalaire est utilisée, c'est-à-dire que les problèmes sont envisagés à plusieurs échelles, utilisant des mathématiques appliquées mais aussi des techniques computationnelles. Cela demande d'unifier différents points de vue allant de la théorie mathématique des systèmes complexes aux simulations numériques sur ordinateur.

Projet à long terme

L'un des projets à long terme auquel participe l'INS est le projet The Virtual Brain. Ce projet est mené conjointement par l'INS, l'Université de Toronto au Canada, la Charité University Medicine de Berlin, et l'entreprise CodeBox basée à Stuttgart en Allemagne. The Virtual Brain est un outil de neurosciences en libre accès permettant aux chercheurs de tester les effets d'expérimentations sur le cerveau, ou d'opérations chirurgicales. L'utilisateur peut alors visualiser en 2D ou 3D les données simulées. L'outil comptait plus de 10 000 utilisateurs en janvier 2018.

Ma mission

Contexte

M. Emmanuel Daucé, enseignant à l'Ecole Centrale de Marseille et chercheur à l'INS, ainsi que M. Laurent Perrinet, chercheur à l'INT, travaillent depuis bientôt deux ans sur un projet nommé WhereIsMyMnist. Ce projet cherche à imiter la vision humaine dans un but de classification d'objets.

Pour simuler la vision humaine, ils se sont basés sur plusieurs faits. La vision humaine est précise uniquement dans une très faible partie du champ de vision, située au centre de celui-ci, et est floue dans tout le reste du champ de vision. La concentration en capteur au niveau de la rétine explique ce fait.

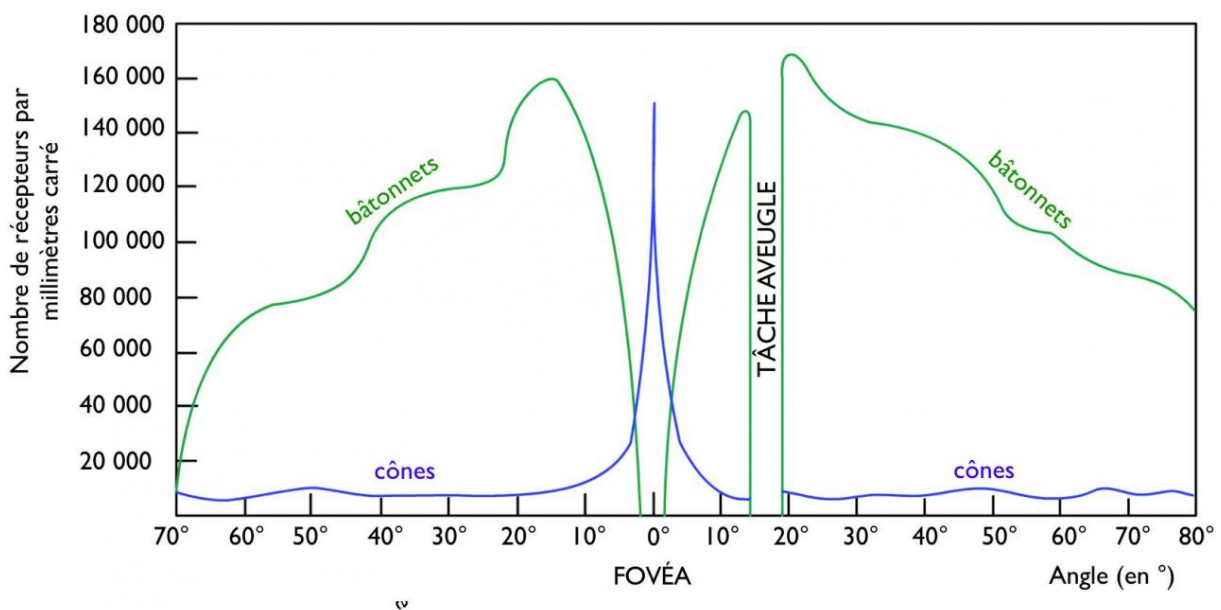


Schéma représentant la répartition des cônes et des bâtonnets sur la rétine [2]

Pour analyser leur environnement, les yeux effectuent des mouvements de saccade vers différents points d'intérêt.

M. Daucé et M. Perrinet ont donc commencé à coder sur Python un comportement qui serait similaire à celui de l'humain lors de l'exploration d'une scène. La première tâche qu'ils ont commencé à traiter est la reconnaissance d'un chiffre manuscrit écrit quelque part dans une image bruitée de 128 pixels par 128 pixels.

La simulation est initialement centrée au milieu de l'image, puis doit être capable d'effectuer une saccade vers la zone d'intérêt où se trouve le chiffre manuscrit, de telle sorte que le chiffre se trouve alors dans la zone nette du champ de vision, où le programme est capable de reconnaître le chiffre. Les lignes de code effectuant la saccade étaient déjà en partie écrites, et je ne les ai pas modifiées durant mon stage. Le code est constitué de deux grandes entités. Le « Where » gère l'image 128 pixels par 128 pixels, et est en charge d'effectuer le mouvement de saccade oculaire vers le bon endroit de l'image. Le « What » prend ensuite le relais, et analyse le carré de 28 pixels par 28 pixels central du nouveau champ de vision et essaye d'y reconnaître un chiffre dans le cas du database MNIST.

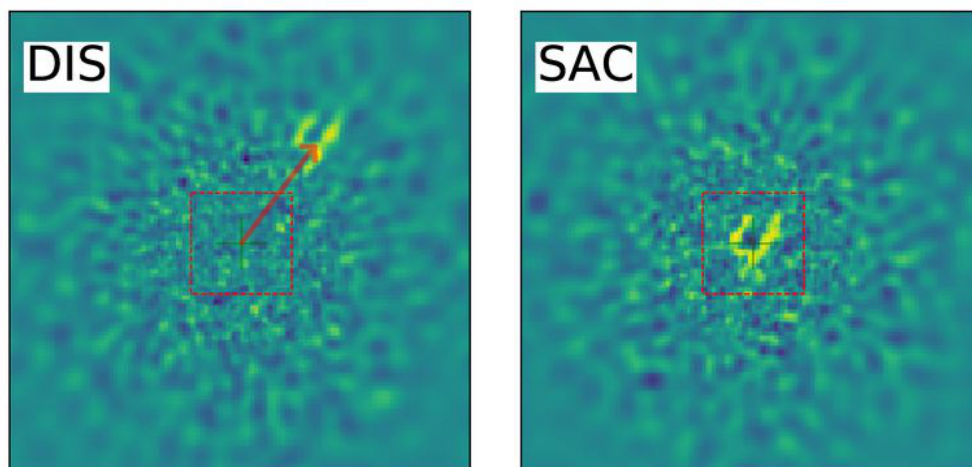


Figure représentant le mouvement effectué par l'algorithme [3]

Objectifs

Ma tâche a consisté à ajouter de multiples fonctions au code déjà existant. Plusieurs problématiques n'avaient pas encore été gérées. A part l'ajout de nouvelles fonctionnalités, il y avait aussi des questions de place en mémoire à traiter ainsi que de compatibilité avec d'autres bases de données. Nous avons commencé par traiter des chiffres manuscrits avant de passer à des visages où le but était alors de déterminer leur genre.

Enjeux

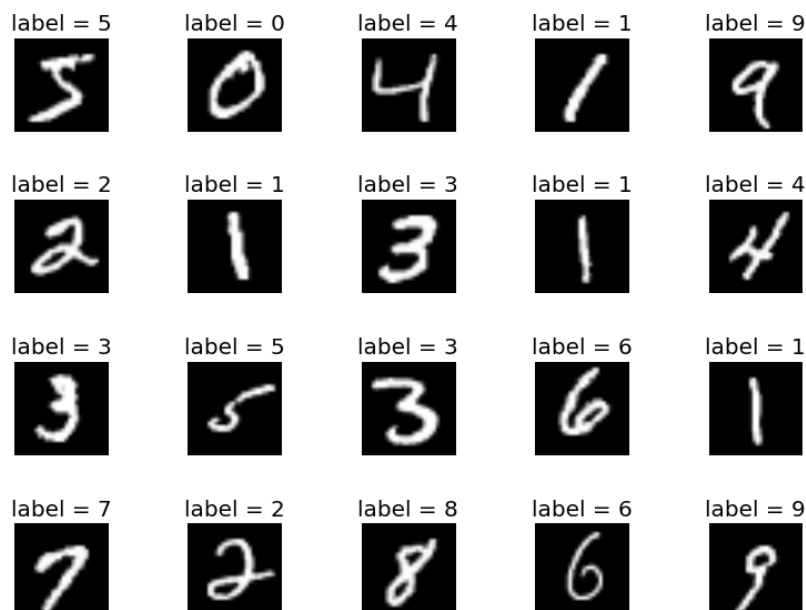
Si on arrive à optimiser le code, cela pourrait permettre d'économiser des ressources et du temps de calculs. En effet, les méthodes de classification actuelles doivent traiter la totalité de l'image pour donner un résultat, ce qui devient donc vite problématique lorsque la taille de l'image augmente.

Déroulement du stage

Les débuts

Ma première semaine de stage a été consacrée à la prise en main de différents environnements et logiciels que je ne connaissais pas. J'ai par exemple installé la plateforme Anaconda, de nombreuses librairies comme pytorch, ou torchvision, le logiciel Pycharm que l'on a lié au GitHub du projet. Je me suis familiarisée au fonctionnement de GitHub que je n'avais jamais vraiment utilisé auparavant, ainsi qu'à la création de Jupyter Notebook. Lors des premiers entraînements de réseaux de neurones, nous avons aussi pu constater que mon ordinateur chauffait trop, j'ai donc appris à utiliser et à communiquer avec Hydra, le super-calculateur de l'Ecole Centrale de Marseille. Cela a permis de faire tourner de longs calculs sans interruption et en parallèle. Durant ces premiers jours j'ai aussi pu m'habituer au code déjà écrit, et voir comment était utilisée la notion de classe en Python. J'avais déjà abordé la notion de classe durant mon S7 avec Mme Jazzar mais c'était en C++, et avec des fichiers de tailles plus raisonnables. Dans le projet WhereIsMyMnist, plusieurs fichiers contiennent des centaines de lignes, c'est donc au total un code beaucoup plus conséquent, et où les fichiers sont beaucoup plus intriqués entre eux.

MNIST

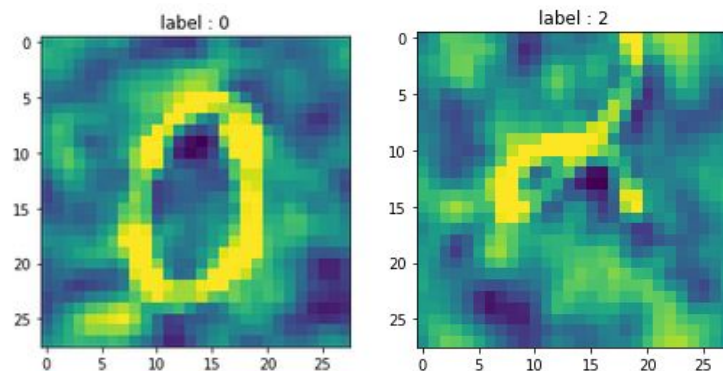


Quelques images de la database MNIST

J'ai commencé par travailler sur la database MNIST qui est composée de 60 000 chiffres manuscrits. On en voit un extrait ci-dessus. Chaque image a une taille de 28 pixels par 28 pixels, et le chiffre est centré. Nous avons utilisé ces chiffres mais en rajoutant du bruit gaussien.

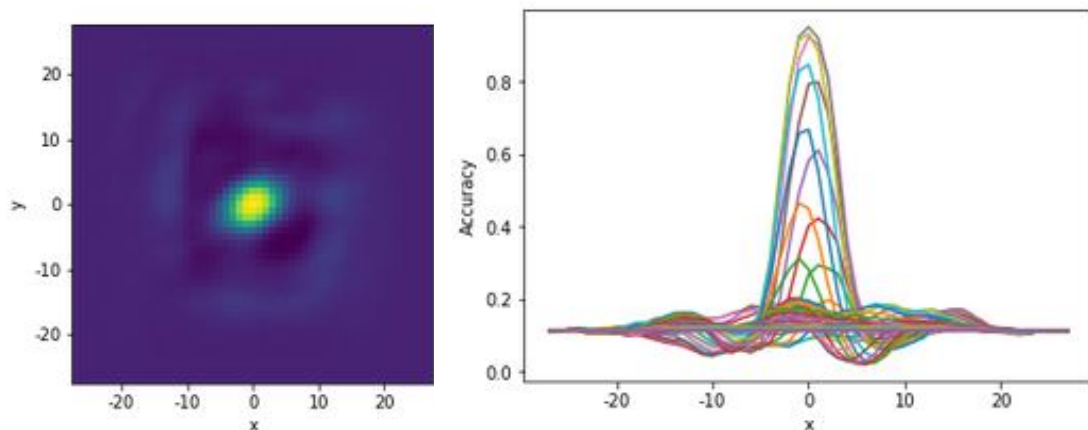
Les images traitées étant des petites images 28 pixels par 28 pixels, il s'agissait donc ici d'entraîner le What de notre modèle.

Le code déjà écrit permettait d'entraîner des réseaux de neurones sur cette base de chiffres centrés bruités, et cela menait à de très bons résultats avec un taux de classification supérieur à 95%. Cependant, si on rappelle le contexte de notre sujet, le Where effectue d'abord une saccade pour essayer de centrer le chiffre au milieu de son champ de vision. Cette saccade peut être imparfaite, et dans ce cas-là, le chiffre n'est pas parfaitement centré.



Exemple d'un zéro parfaitement centré, et d'un deux décalé vers le haut

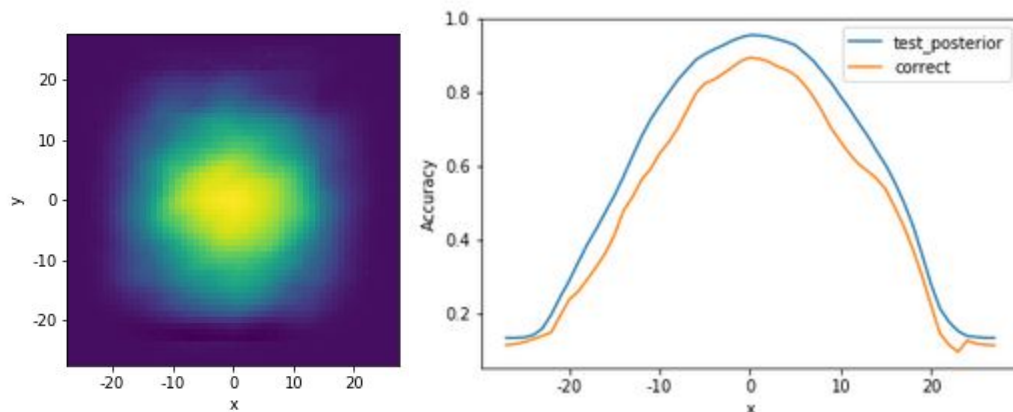
Un réseau entraîné uniquement sur des chiffres centrés se révèle très vite mauvais lorsqu'on le teste sur des chiffres légèrement décentrés, c'est ce que l'on peut constater sur les diagrammes suivants.



Accuracy Map d'un réseau entraîné sur des images centrées uniquement

Le réseau entraîné sur des images centrées est bon lorsque les images tests sont elles aussi centrées, mais son accuracy diminue très vite dès que les images sont décentrées.

Ma première vraie tâche a donc été de modifier le code pour entraîner le réseau sur des images centrées, mais aussi des images plus ou moins décentrées. Ce qui a permis d'augmenter considérablement la tache centrale du diagramme précédent.



Accuracy Map d'un réseau entraîné sur des images centrées et décentrées

Que la saccade soit excellente ou imparfaite, le réseau entraîné de cette manière est capable de reconnaître le chiffre dans un grand nombre de cas. Sur le diagramme de droite, il s'agit d'une coupe effectuée au milieu de l'image, la courbe bleue indique la probabilité que la réponse fournie soit juste d'après le réseau, et la courbe orange indique la fréquence réelle où le réseau a répondu juste. On constate que le réseau a tendance à surestimer son taux de réponses correctes quel que soit le décalage de l'image.

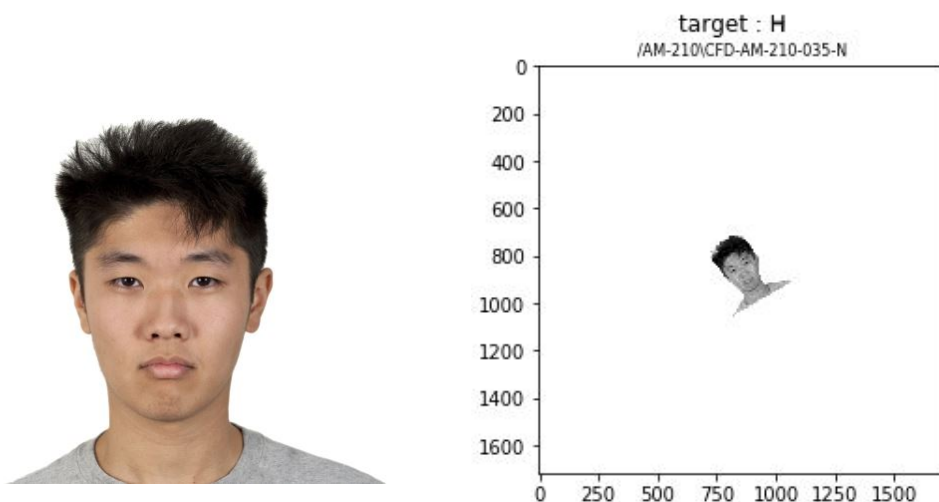
D'autres tâches ont été effectuées, comme des comparaisons de performances de différents réseaux lorsqu'on fait varier leurs paramètres.

The Chicago Face Database



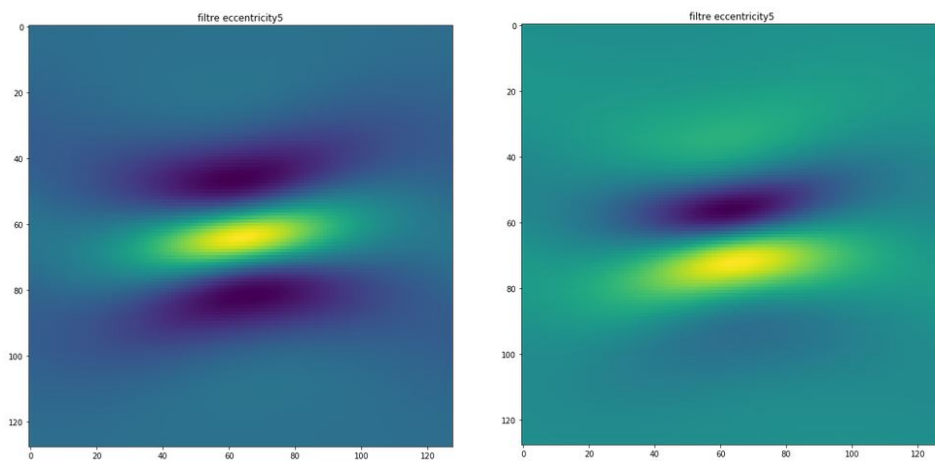
Echantillon d'images du Chicago Face Database

Courant Juillet, j'ai changé de database. Je suis passée au Chicago Face Database qui est composé de 1207 photos ayant pour taille 2444 pixels par 1718 pixels. Cette fois, on a uniquement utilisé le Where et plus du tout le What. Il a fallu modifier le code pour prendre en compte le changement de taille ainsi que le nombre de cibles car cette fois-ci on cherche à déterminer s'il s'agit d'un homme ou d'une femme, et on n'effectuera pas de saccade. J'ai écrit plusieurs transformations sous forme de classes, permettant par exemple de recouper les images d'entrée en carré, de les passer en noir et blanc ou de les tourner d'un certain angle. Il a donc fallu jongler entre les différents formats d'image, image PIL ou image Numpy, en profitant de ce que pouvait faire chacun de ses formats.

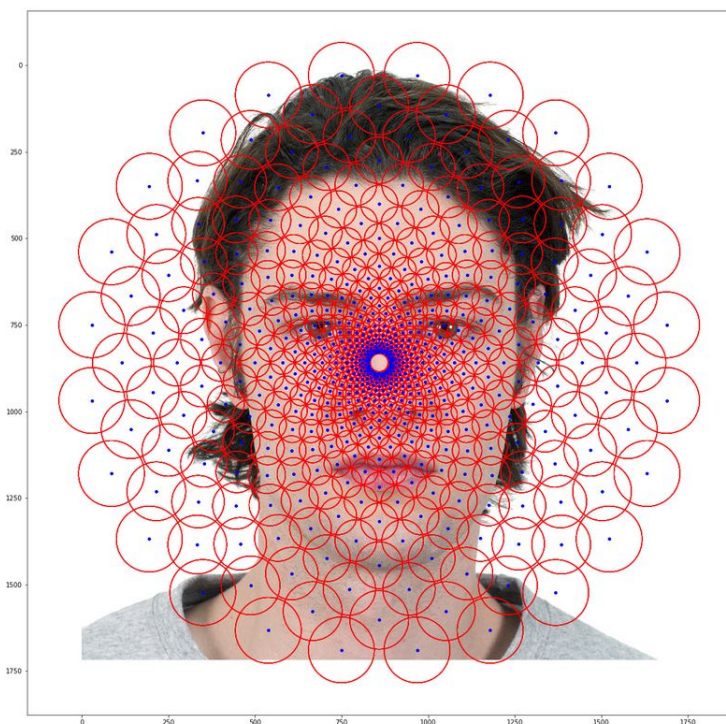


Exemple de transformations sur une image de la database

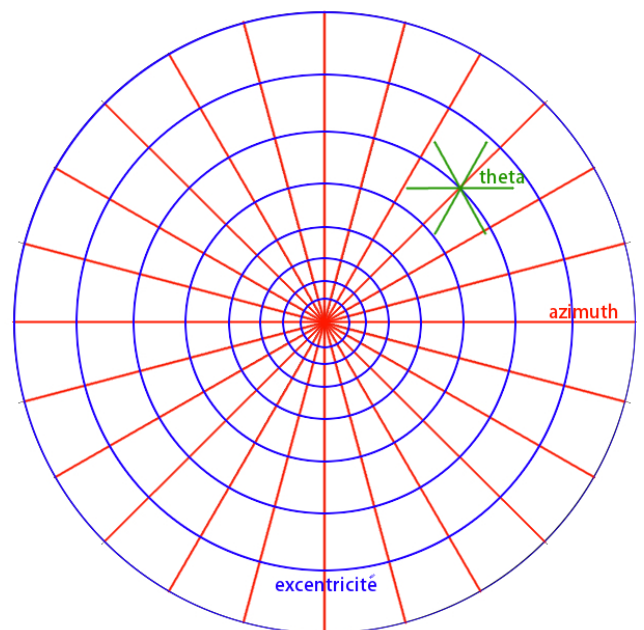
Le plus gros défi avec cette nouvelle classe n'a pas été d'effectuer ces modifications sur les images, mais de gérer l'utilisation de la mémoire lors de la reconnaissance du genre par notre algorithme. En effet, pour faire fonctionner le Where, il faut tout d'abord transformer l'image d'entrée en une image fovéale, c'est-à-dire nette au centre et floue sur les pourtours. Pour effectuer cela, on utilise des filtres Log Gabor répartis en nid d'abeille. Ils existent deux types de filtres Log Gabor, les filtres symétriques qui permettent de prendre en compte les contours, et les filtres antisymétriques qui permettent de détecter les changements de surfaces.



A gauche un filtre Log Gabor symétrique, à droite un antisymétrique



Répartition en nid d'abeilles des filtres

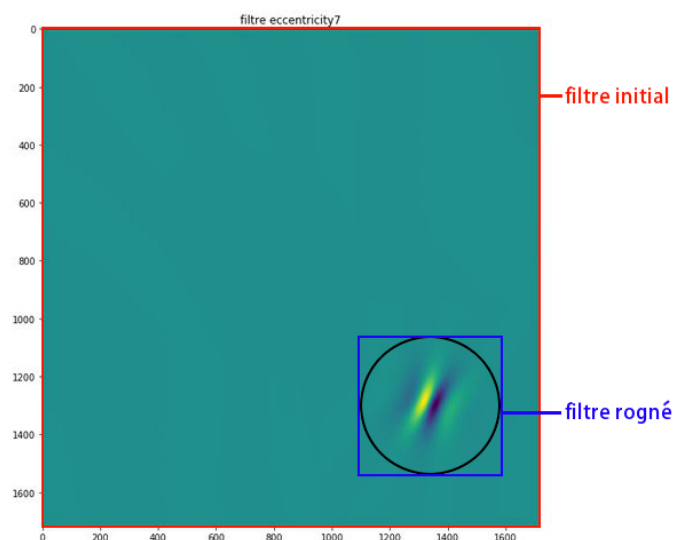


Représentation de la disposition des filtres

Sur l'image ci-dessus à gauche se trouvent 6912 filtres, les points bleus sont leurs centres, et les délimitations rouges les frontières au-delà desquelles ils peuvent être considérés comme de valeur nulle. Il y a 6912 filtre car $\text{nb_filtres} = \text{nb_azimuth} * \text{nb_excentricity} * \text{nb_theta} * \text{nb_phase} = 24 * 24 * 6 * 2$ ici, où $\text{nb_phase} = 2$ car il y a des filtres pairs et des filtres impairs. Chacun de ses filtres doit être multiplié avec l'image de la database pour obtenir l'image fovéale. Le problème de mémoire se pose donc par rapport au stockage de ces nombreux filtres.

Initialement, chaque filtre faisait exactement la même taille que l'image à traiter, et la totalité des filtres étaient stockés dans un tableau Numpy à 5 dimensions (valeur azimuth, valeur excentricité, valeur theta, valeur phase, liste des valeurs des pixels du filtre considéré). Cette méthode marchait pour des images faisant 128 pixels par 128 pixels, mais a immédiatement causé une erreur mémoire quand nous avons tenté de l'appliquer aux images de visages qui sont beaucoup plus grandes et pour lesquelles il y a davantage de filtres.

La première partie du travail a donc été de réaliser qu'il y avait beaucoup de zéros dans la matrice de chaque filtre, et que l'on pouvait donc rogner chaque matrice de filtre au minimum. Sur l'image ci-dessus à gauche, cela a consisté à prendre pour chaque filtre uniquement le carré tangent au cercle délimitant la zone efficace du filtre.



Représentation d'un filtre avant et après avoir été rogné

Rogner le filtre a aussi permis d'économiser en ressources de calcul car il suffit alors de multiplier le filtre rogné avec la portion correspondante de l'image, on gagne donc beaucoup d'opérations.

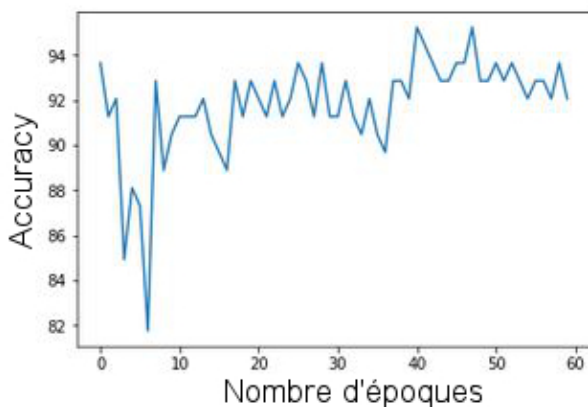
Il nous est aussi apparu que l'on pouvait stocker directement le résultat du produit scalaire entre le filtre rogné et la portion d'image correspondante au lieu de stocker tout le filtre rogné. Nous avons de plus utilisé des dictionnaires imbriqués pour stocker cette information. Au lieu de stocker 6912 images de filtres contenant 1718^2 pixels chacune, nous avons juste stocké 6912 valeurs numériques. Il a donc été possible de transformer les images en images fovéales de cette manière.

Pour réorganiser les données en dossiers utilisables pour l'apprentissage, il a fallu utiliser quelques commandes de la librairie os.

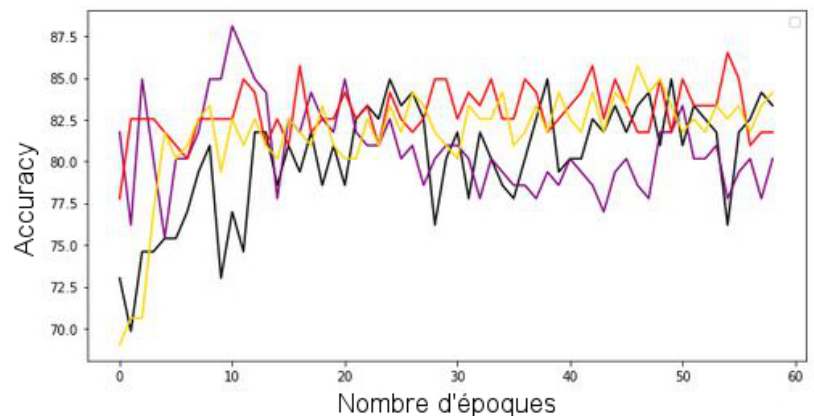
Résultats

Nous avons voulu comparer le taux de réussite obtenu par la discrimination de l'image fovéale par rapport au taux de réussite du discriminateur classique. Avec la configuration utilisée à ce moment-là, l'image fovéale se résumait en 13824 valeurs de produits scalaires, nous avons donc décidé de réduire les images servant de test au discriminateur classique de manière à ce que les images qu'il ait en entrée aient aussi environ 13824 valeurs de pixels (finalement, elles en auront 13720).

Les résultats laissent cependant à désirer, le discriminateur classique reste largement meilleur même lorsqu'on essaie d'autre paramètres pour les réseaux utilisant les images fovéales.



Résultats d'un discriminateur classique



Résultats d'un discriminateur d'images fovéales pour différents learning rate

Pour l'instant, la méthode classique semble plus efficace pour reconnaître le genre sur ces photos, on peut cependant espérer encore améliorer et optimiser la technique pour arriver à un niveau comparable. Une étape suivante intéressante pourrait être d'essayer de reconnaître les expressions faciales des personnes, la database comporte en effet des expressions heureuses bouche ouverte, heureuses bouche fermée, effrayées, neutres ou en colère. Il pourrait aussi être intéressant d'arriver à reconnaître des zones d'intérêts sur le visage pour y effectuer des saccades et reconnaître plus efficacement la cible.

Conclusion

Durant ce projet, le code aura bien avancé, plusieurs nouvelles fonctionnalités ont pu être implémentées. Il a aussi été intéressant de traiter des images issues de différentes databases, mais il reste encore pas mal de parties à optimiser pour pouvoir espérer obtenir des taux de réussite équivalent aux discriminateurs classiques. Peut-être que ce sujet pour être poursuivi par des Centraliens en projet S7, S8 ou durant un stage l'an prochain.

D'un point de vue personnel, ce stage m'aura permis d'apprendre à utiliser de nouveaux outils tels que GitHub, Pycharm, Anaconda ou Jupyter Notebook. J'ai aussi augmenté mon aisance par rapport à l'utilisation des classes. Et cette première expérience en laboratoire a été très intéressante, et riche en expériences.

Bibliographie

[1] Photos de l'INS issues du site internet

<https://ins-amu.fr/about-ins>

consulté la dernière fois le 01/10/2019

[2] Schéma de la répartition des cônes et des bâtonnets sur la rétine

<http://tpe1stdvbce.e-monsite.com/pages/retine-naturelle-humaine/1-qu-est-ce-que-des-photorecepteurs.html>

consulté la dernière fois le 01/10/2019

[3] Figure représentant le mouvement effectué par l'algorithme

Figure de M. Daucé et M. Perrinet

Self-Invertible 2D Log-Gabor Wavelets, Sylvain Fischer, Filip Sroubek, Laurent Perrinet,
International Journal of Computer Vision 75(2), 231-246, 2007

http://staff.utia.cas.cz/sroubekf/papers/gabor_07.pdf

consulté la dernière fois le 01/10/2019

A review of log-polar imaging for visual perception in robotics, V.Javier Traver, Alexandre Bernardino,
Robotics and autonomous Systems 58 (2010) 378-398, 2010

<http://marmota.dlsi.uji.es/WebBIB/papers/2010/Traver-RAS2010.pdf>

consulté la dernière fois le 01/10/2019

Annexes

Chicago Face Database (CFD)

<https://chicagofaces.org/default/>

consulté la dernière fois le 01/10/2019