

AIX-MARSEILLE UNIVERSITÉ

MÉMOIRE DE MASTER

**Optimisation de la vision artificielle
bio-inspirée par exploration saccadique de
l'environnement**

Auteur:

Pierre ALBIGÈS

Superviseur:

Laurent PERRINET

Un mémoire présenté à

ECOLE DOCTORALE DES SCIENCES DE LA VIE ET DE LA SANTÉ

en vue de l'obtention du diplôme de

MASTER DE NEUROSCIENCES, SPÉCIALITÉ INTÉGRATIVES ET COGNITIVES

et réalisée au sein de

Institut de Neurosciences de la Timone

Durant la période : 12/03/2018 - 08/06/2018

AIX-MARSEILLE UNIVERSITÉ

Résumé

Faculté des Sciences, département de Biologie

Master de Neurosciences

Master de Neurosciences, spécialité Intégratives et Cognitives

**Optimisation de la vision artificielle bio-inspirée par exploration saccadique de
l'environnement**

by Pierre ALBIGÈS

Contents

1	Introduction	1
2	Matériel et méthodes	4
2.1	Matériel	4
2.2	Base de données MNIST	4
2.3	Carte de certitude	4
2.4	Pré-traitements de l'image	5
2.4.1	Redimensionner et replacer	5
2.4.2	Bruit écologique	5
2.5	Filtre LogPolaire	6
2.6	Modèle	6
3	Résultats	8
3.1	Résultats escomptés	8
3.2	Résultats préliminaires	9
4	Discussion et perspectives	10
	Bibliography	13
A	Figures	14
B	Code source et documents complémentaires	22

1 Introduction

Au cours de l'histoire évolutive et sous la pression de la sélection naturelle, tous nos systèmes perceptifs ont tendu (et tendent encore) vers une optimisation de leur fonctionnement, en fonction de nos besoins et de nos ressources. L'ensemble de notre système visuel, de la rétine jusqu'aux aires corticales les plus associatives, a ainsi évolué pour pouvoir réaliser une description robuste et rapide de notre environnement, nous permettant d'en intégrer les informations les plus pertinentes, d'interagir efficacement avec lui et d'en appréhender les dangers. Cette pression évolutive a notamment mené au développement de deux caractéristiques du système visuel qui vont plus particulièrement nous intéresser : l'acuité visuelle est non-uniforme et l'œil explore la scène visuelle en effectuant des saccades.⁸

L'acuité visuelle peut être définie comme l'efficacité avec laquelle les stimuli visuels peuvent être analysés. Celle-ci n'est pas fixe mais varie au sein de notre champs visuel (portion de l'espace observée par un œil immobile), qu'il est ainsi possible de séparer en deux parties : la vision centrale et la vision périphérique.⁸

La vision centrale est soutenue anatomiquement par la fovéa, une région rétinienne comprenant exclusivement des cônes. Cette composition, couplée à une forte densité de photorécepteurs et une faible convergence photorécepteurs/cellules ganglionnaires, permet à cette région de présenter l'acuité visuelle la plus importante du système visuel, ainsi qu'une bonne perception des couleurs.⁸

La composition et la densité en photorécepteurs de la rétine soutenant la vision périphérique change avec son excentricité par rapport à la fovéa, mais elle comprends majoritairement des bâtonnets. De même, le degré de convergence photorécepteurs/cellules ganglionnaires augmente avec cette excentricité (lorsque ce degré augmente, le nombre de photorécepteurs convergents vers une même cellule ganglionnaire augmente). En conséquence, l'acuité visuelle et la perception des couleurs dans la vision périphérique diminuent avec la distance de la fovéa, mais on peut y observer une importante sensibilité aux variations de luminance et de fréquence spatiale.⁸

Cette variabilité des caractéristiques de notre système visuel et notamment de son acuité, permet de fortement réduire la quantité d'informations à traiter par les réseaux nerveux en aval de la rétine, cette dernière recevant quasi-continuellement un flux d'information estimé à 10^8 bits/s, subissant une réduction de plus de 99% pour engendrer une sortie par le nerf optique estimée à 10^2 bits/s.^{3,8,9}

La variabilité de l'acuité visuelle en fonction de l'excentricité à la rétine, ainsi que l'organisation spatiale des stimuli sur celle-ci sont d'ailleurs conservées tout au long des réseaux nerveux réalisant leur traitement, formant ce que l'on nomme l'organisation rétinotopique des régions cérébrales visuelles.⁸

Mais cette réduction du flux d'informations à traiter présente au moins un inconvénient majeur. Une description précise d'un stimulus visuel ne peut être réalisée avec une grande certitude que dans une partie très réduite du champ visuel (sur un disque d'environ 2 degrés d'angle visuel chez l'Humain). Pour palier à cela, lorsqu'un agent voudra explorer son environnement visuel, il devra réaliser une série de mouvements oculaires brefs. Ces saccades oculaires permettront de placer les régions visuelles d'intérêt dans la vision centrale, pour que le système visuel en aval puisse en réaliser des descriptions précises. Par exemple, l'observation passive d'une scène (sans consigne ou recherche précise d'une cible) va impliquer la réalisation d'en moyenne 2 à 4 saccades par seconde.^{4,8}

La sélection attentionnelle et motrice de la cible à décrire fait intervenir un réseau complexe d'aires cérébrales et corticales et nécessite l'intégration de signaux *top-down* comme *bottom-up*.⁸

La modélisation du système visuel est l'un des domaines phares du développement de l'intelligence artificielle depuis ses débuts, dans les années 60. L'objectif est de s'inspirer, voir de mimer le fonctionnement des systèmes biologiques afin de permettre aux systèmes informatisés d'accéder à la compréhension de leur environnement. La vision artificielle connaît ainsi depuis plusieurs décennies l'application de nombreuses méthodes, à diverses échelles et niveaux de complexité.⁸

Les modèles à carte de saillance permettent par exemple de reconstruire l'influence qu'ont les signaux *bottom-up* sur l'orientation du regard. Ces modèles décrivant chaque point de l'espace visuel par une valeur, ceux qui ressortent le plus de l'environnement sont considérés comme portant l'intérêt le plus grand pour le système et attirent le regard. Après avoir été explorée, une

région voit sa saillance devenir nulle, car elle ne peut alors plus fournir d'information à l'agent. Les prédictions de ces modèles sont meilleures que le hasard mais ne sont pas parfaites, notamment car ils ne prennent pas en compte certaines caractéristiques des systèmes biologiques, tels que la recherche de cible ou la variabilité de l'acuité visuelle.⁸

L'étude, le développement et l'utilisation de la vision artificielle permet non seulement d'améliorer les performances des systèmes informatisés, mais aussi de mieux appréhender les zones d'ombre dans nos connaissances du fonctionnement du système visuel (notamment lorsque la modélisation d'une fonction spécifique ne permet pas de prédire le comportement naturel).⁸

Dans ce travail exploratoire nous avons tenté, via la simulation de la variabilité de l'acuité visuelle et de l'exploration saccadique de l'environnement, et en les appliquant à des réseaux nerveux artificiels, de proposer une alternative aux modèles actuels de description de l'environnement visuel qui se basent pour la plupart sur une classification pixels par pixels (ou groupes de pixels) sur l'ensemble du champ visuel. Grâce à la simulation de ces fonctions biologiques, notre modèle doit pouvoir rechercher une cible dans son environnement visuel de façon autonome et ne décrire (classifier) que les régions qui lui fourniront des informations pertinentes.

L'objectif est double: d'une part aider à l'optimisation des systèmes de vision par ordinateur en proposant une méthode neuromimétique rapide et peu coûteuse de la recherche de cible, notamment pour les systèmes embarqués pour lesquels ces caractéristiques sont primordiales, et d'autre part d'explorer les connaissances neuroscientifiques sur le sujet afin d'offrir un point de départ dans l'identification de zones d'ombre dans la compréhension de l'exploration saccadique de l'environnement ainsi que de la recherche/suivi de cible visuelle.

2 Matériel et méthodes

2.1 Matériel

L'ensemble des simulations (comprenant apprentissage et évaluations) ont été réalisées sur une machine connectée à distance via un protocole `ssh` et dont les caractéristiques sont visibles dans la [Table A.1](#).

2.2 Base de données MNIST

Dans ce travail, nous avons utilisé comme stimuli les images provenant de MNIST, une base de données de 70000 images contenant un et un seul chiffre manuscrit chacune, accompagnées d'un *label* décrivant de quel chiffre il s'agit. Cette base de données a été choisi car sa classification est considérée comme l'évaluation standard dans la cadre du développement des modèles de *machine learning*. Son utilisation courante nous permettra de facilement comparer les performances d'autres modèles à celle du notre, et sa simplicité nous permet de construire des prototypes simples mais fonctionnels qui pourront ensuite être complexifiées pour être adaptés à d'autres types de stimuli.

2.3 Carte de certitude

En amont de l'initialisation de notre modèle, nous avons créé un classifieur simple que nous entraînons pour être capable d'obtenir, dans des conditions classiques, des performances acceptables sur la base de données MNIST (99% de reconnaissance positive du chiffre contenu dans l'image). Ce modèle est ensuite évalué avec des images de 28*28 pixels contenant toujours un chiffre MNIST mais celui-ci pouvant être décalé dans cet espace. La performance du classifieur est ainsi calculée 1000 fois pour chaque position possible du chiffre dans l'espace. Nous obtenons une matrice correspondant à la certitude avec laquelle le modèle peut reconnaître le chiffre qu'on

lui impose selon sa position par rapport au centre de l'image, correspondant à son centre de fixation. Une reconstruction graphique de cette matrice est visible dans la figure A.1. Cette carte de certitude servira de base pour construire les *labels* qui seront utilisés lors de l'apprentissage automatisé de notre modèle principal.

2.4 Pré-traitements de l'image

Avant d'être utilisées par notre modèle, les images subissent un certain nombre de pré-traitements. L'objectif de ces pré-traitements est de les rendre plus écologiques, c'est à dire plus proches des stimuli que rencontrent les systèmes biologiques.

2.4.1 Redimensionner et replacer

A l'origine, les exemples MNIST sont codées en niveau de gris dans une image normalisée de 28×28 pixels (figure A.2). Afin de réduire la taille du stimulus au sein de l'image, nous introduisons cette image de 28×28 pixels dans une image vide de 128×128 pixels (figure A.3). Cette insertion se fait systématiquement à un emplacement aléatoire pour permettre de produire un stimulus utilisable dans notre tâche de détection de la position d'une cible. En parallèle, la carte de certitude construite précédemment et contenue dans une image de 54×54 pixels est introduite dans une image pseudo-vide de 128×128 pixels, au même emplacement que le stimulus (figure A.4).

Les images-hôtes sont pseudo-vides car leurs valeurs en chaque point correspond à la valeur minimale de l'image insérée, permettant de ne pas créer un cadre autour de cette dernière lorsqu'elle est intégrée.

2.4.2 Bruit écologique

Pour permettre à nos stimuli de s'approcher de ceux pouvant être reçus et traités par les systèmes biologiques, nous avons superposé à nos signaux un bruit généré de manière aléatoire et selon deux méthodes possibles. La première consiste en la génération de bruit Perlin⁷ (figure A.5), permettant à l'origine de produire automatiquement des textures à l'aspect naturel destinées à être utilisées pour des effets spéciaux numériques. La seconde consiste en la génération de bruit

MotionCloud (figure A.6), permettant d’obtenir des textures aléatoires et semblants naturelles, destinées à l’origine à être utilisées dans des études sur la perception des mouvements. C’est cette dernière que nous avons utilisé par défaut lors de l’apprentissage automatisé, mais les deux méthodes sont implantées dans nos scripts.⁵

2.5 Filtre LogPolaire

Finalement, afin de simuler une variabilité de l’acuité visuelle chez notre modèle, nous avons appliqué à nos stimuli un filtre LogPolaire (figure A.7). Ce filtre, construit avec une approche neuromimétique, est constitué d’un ensemble de filtres Gabor et vise à reproduire la forme et l’organisation réelle des champs récepteurs présents dans les régions visuelles des systèmes nerveux biologiques. De précédentes études ont montré que cette méthode présente un certain nombre d’avantages pour la modélisation des systèmes biologiques, notamment car elle est aisément modifiable pour simuler les champs récepteurs de différentes régions impliquées dans la vision (rétine, corps genouillé latéral, colliculus supérieur, V1 puis aires associatives). Le filtre LogPolaire correspond en réalité à une matrice de valeurs qui, lorsque appliquée à une image par multiplication matricielle, permet une décroissance de la résolution en fonction de l’excentricité (distance) par rapport au centre de l’image. Le résultat de l’application de ce filtre sur l’un de nos stimuli est visible sur les figures A.8 (non-bruité) et A.9 (avec un bruit *MotionCloud*). Des reconstructions alternatives (logarithmiques) sont visibles sur les figures A.10 et A.11². Une version de ce filtre dans laquelle les filtres Gabor d’un même emplacement (mais ne possédant pas la même orientation) sont moyennés est appliquée à la carte de certitude, servant de label pour l’apprentissage de notre modèle (figures A.12 et A.13).

2.6 Modèle

Le fonctionnement de notre modèle se basant sur des méthodes d’apprentissage automatisé (ou *machine learning*), son fonctionnement peut être décrit en deux temps.

Durant la première phase, dite d’apprentissage, nous fournissons au réseau nerveux artificiel à la fois une entrée (ou *input*) correspondant à l’image transformée à partir de laquelle il va devoir prédire la position d’un stimulus et un label, correspondant à une carte de chaleur représentant la position réelle du stimulus. Durant un nombre d’itérations prédéfinies, le réseau nerveux va

interpréter l'input pour produire une série de valeurs correspondant à sa prédiction de la position du stimulus. Le modèle calcul ensuite un coût, c'est à dire (de manière simplifiée) la différence entre cette prédiction et le label fourni (et donc à la justesse de la prédiction), via la méthode de l'entropie croisée binaire (*BCELoss*). Finalement selon les valeurs de coût, les matrices de poids du réseau nerveux artificiel sont mises à jour via la méthode de descente de gradient stochastique (*SGD*).

Après la phase d'apprentissage vient la phase d'évaluation (ou de test) où nous fournissons au modèle des inputs qu'il n'a jamais rencontré afin de s'assurer de ses performances et de ses capacités de généralisation (transcrire à de nouveaux stimuli ce qu'il a appris). Les inputs sont ainsi fournis seuls, sans labels, et le modèle doit produire des prédictions sur la position des stimuli. Aucun apprentissage n'est réalisé durant cette phase.

Lors de l'écriture de ce rapport, le réseau nerveux artificiel était composé de trois couches linéaires séparées par une rectification linéaire "à fuite" (*leaky_ReLU*). L'ensemble des étapes décrites jusqu'ici sont visibles sur la figure [A.14](#). Pour plus de détails sur le fonctionnement du modèle, se reporter à l'appendice [B](#).

3 Résultats

3.1 Résultats escomptés

Au terme du stade de développement actuel notre modèle devrait être capable, lorsque nous lui fournissons une image bruitée de 128×128 pixels contenant un chiffre MNIST placé au hasard, de produire une carte de probabilités représentant une prédiction précise de la position du stimulus dans l'espace. A chaque point de l'espace devrait donc correspondre une valeur comprise dans l'intervalle $[0, 1]$, pouvant être traduite par la certitude du modèle de la présence d'un stimulus en ce point. La prédiction du modèle devrait ainsi comprendre au moins une zone chaude (une seule lors d'une prédiction optimale), c'est à dire une région circulaire (dépendant de la forme des champs récepteurs) où sont rassemblées les valeurs de probabilités les plus élevées, à l'image de ce que l'on peut observer dans les labels (figure A.13). Selon les caractéristiques de notre filtre rétinien, l'acuité du modèle et donc la certitude de ses prédictions devraient diminuer avec l'excentricité par rapport au centre de la fovéa. Ainsi lorsqu'on éloigne le stimulus de cette dernière, la zone chaude devrait s'étendre (révélant une prédiction moins précise) pendant que les probabilités qu'elle contient diminuent pour se rapprocher du seuil définissant le hasard (révélant une certitude plus faible).^{2,8}

Une fois cette prédiction de la position réalisée, le modèle devrait pouvoir l'utiliser pour réaliser une saccade rapprochant la position prédite de la cible du centre de la fovéa. A noter qu'en l'absence d'organe perceptif physique, les saccades correspondent à un décalage spatial de l'image utilisée en entrée. Enfin, après avoir supposément placé la cible sur sa fovéa ou tout du moins proche de celle-ci, le modèle pourra réaliser une classification dans la partie centrale de son champs visuel. Si la prédiction de la position de la cible a été correcte et donc que la bonne partie de l'environnement visuel a été placé sur la fovéa, alors le modèle devrait être capable de classifier correctement le chiffre qu'on lui présente.⁸

3.2 Résultats préliminaires

Il est possible d'observer sur la figure A.16 une diminution de la valeur du coût d'environ 50% lors des itérations de l'apprentissage automatisé. Une diminution de cette valeur est le signe que l'apprentissage a bien lieu et plus elle est importante, plus l'apprentissage devrait être efficace. Notons qu'après une décroissance rapide (itérations 0-200) de cette valeur, nous pouvons rapidement observer (itérations 200-500) un ralentissement de celle-ci, phénomène couramment observé dans les modèles d'apprentissage automatique.

Lorsque, durant la phase d'évaluation, nous fournissons à notre modèle une image bruitée et observée au travers d'un filtre rétinien, celui-ci est capable de l'intégrer pour produire un vecteur contenant un ensemble de valeurs comprises dans l'intervalle $[0, 1]$. Ce vecteur, à partir duquel il est possible de réaliser par reconstruction un graphique (figure A.16), présente une ou plusieurs zones chaudes où sont rassemblées les valeurs les plus élevées. La position de ces zones chaudes semble concorder avec la position réelle de la cible dans l'image fournie en entrée. Notons que la reconstruction graphique que nous utilisons modifie les valeurs des données, entraînant la perte de l'intervalité $[0, 1]$ de celles-ci.

4 Discussion et perspectives

Lors de l'écriture de ce rapport, aucune analyse quantitative n'a encore été réalisée pour vérifier les résultats et leur robustesse. Malgré cela, il nous a été possible d'observer lors des évaluations une majorité de prédictions correctes (mais pas exactes; qui se rapprochent suffisamment de la véritable position de la cible pour ne pas être considérées comme des explorations hasardeuses). Un certain nombre d'erreurs ont tout de même été observées, ce qui révèle la nécessité d'optimiser le modèle si nous voulons obtenir des performances à la fois élevées et robustes. Au moins une partie des erreurs que l'on observe peut être imputée à la simplicité du réseau nerveux que nous avons construit. Cependant le fait que l'on observe une majorité de prédictions correctes malgré cette simplicité est encourageant concernant les performances que pourra montrer le modèle dans des stades développement plus avancés.

La simplicité du modèle s'accompagne d'une forte adaptabilité. En effet, l'implémentation du modèle et notamment de la production des entrées à partir d'images est actuellement adaptée à la base de données MNIST mais a été réalisée de sorte qu'utiliser une autre base de données peut se faire sans aucune modification du code, hormis le chargement des données, et surtout de sorte que le réseaux nerveux artificiel s'adapte automatiquement pour pouvoir réaliser apprentissage et évaluation sans travail supplémentaire. L'association d'une simplicité et d'une forte adaptabilité sont deux avantages importants pour notre modèle et ces caractéristiques seront primordiales dans les prochaines stade de développement, notamment lors de l'intégration dans un agent physique.

Bien que la réalisation d'une saccade pour se rapprocher de la position détectée de la cible ainsi que la classification de la région nouvellement observée soient exposées dans la partie 3.1, ces fonctionnalités n'ont pas encore été implantées lors de l'écriture de ce rapport.

Malgré le stade de développement peu avancé de notre modèle, nous avons dès aujourd'hui identifié de nombreuses étapes de développement que nous devrons probablement réaliser dans le futur pour complexifier son comportement et améliorer ses performances. Ces étapes iront

compléter les optimisations que doit recevoir notre modèle à ce stade. Ce travail futur, permettant notamment de passer d'un modèle exploratoire à une application physique en plus de continuer de répondre aux objectifs actuels, semble pouvoir être intégré au sein d'un sujet de thèse.

La première étape sera certainement d'étudier la robustesse du modèle en lui soumettant lors de l'étape d'évaluation des images vides mais pouvant être bruitées. Dans son état actuel, le modèle ne devrait pas être capable relever la différence avec les images que nous lui fournissons actuellement et devrait donc tenter de réaliser une détection, puis une classification malgré l'absence de stimulus. En réponse à ce problème, il sera possible d'ajouter une couche de neurones artificiels précédant notre réseau actuel et détectant la présence ou l'absence de stimulus dans le champs visuel, modifiant en fonction le comportement de la suite du réseau. Une solution alternative serait d'entraîner le classifieur avec une catégorie de label supplémentaire, représentant une entrée sans stimulus, et donc de lui donner la possibilité d'appréhender ce type de situations.

Nous savons que les systèmes biologiques n'accèdent pas nécessairement à leurs cibles en une seule saccade et qu'une fois qu'ils les ont atteinte ils réalisent autour d'elles des micro-saccades. De même, nous avons pu observer la présence d'erreurs lorsque le modèle réalise une prédiction, et donc lors de la saccade vers sa cible, entraînant un rapprochement incomplet de la fovéa. Ainsi une seconde étape nous semblant primordiale est l'intégration au modèle de la possibilité de réaliser plusieurs saccades à la suite, séparées par une re-évaluation de son environnement, pour lui permettre de diminuer les conséquences d'une prédiction imprécise.^{6,8}

Mais cette série de prédictions ne peut se faire sans mémoire ou l'on risque de voir le modèle osciller continuellement entre les mêmes points de son environnement visuel. Ainsi dans un même temps, il sera nécessaire d'insérer dans le modèle une forme simple de mémoire de son environnement et de ce qu'il en a exploré. Cette mémoire pourrait prendre la forme d'une carte de probabilité mise à jour au fil de l'exploration. Ainsi lorsque le modèle réaliserait une prédiction de la position de sa cible, elle irait se superposer à cette carte mnésique, laquelle serait lu pour définir le lieu présentant la plus haute probabilité de contenir cette cible. Ensuite lorsqu'une région aura été explorée, sa représentation sur la carte mnésique subira la soustraction d'une valeur pré-déterminée afin de fortement réduire la probabilité que cette région soit à nouveau explorée durant les saccades suivantes, créant une inhibition de retour (le modèle considéreras que pendant quelques itérations cette région ne peut plus lui apporter d'information supplémentaire).^{6,8,9}

A ce stade, notre modèle devrait être capable de rechercher et de détecter une cible dans son environnement visuel, puis de réaliser la ou les saccade(s) nécessaires pour la placer au plus proche de sa fovea. Mais un environnement visuel ne contenant qu'un seul stimulus, même bruité, ne représente qu'une version très artificielle de ce que peuvent observer continuellement les systèmes biologiques. Ainsi pour passer un nouveau stade de développement, il nous semble nécessaire d'intégrer à notre modèle une forme d'influence *top-down*. Cette composante permettra à notre modèle de s'approcher du comportement biologique en mimant la sélection attentionnelle et motrice de la cible à décrire. Elle influera le modèle en définissant quelle type de cible il devra rechercher dans son environnement, et pourra prendre la forme d'une matrice de poids spécifique à une classe de cibles, voir à une cible unique, qui sera implantée dans le réseau nerveux artificiel pour influencer son comportement.⁸

L'intégration de cette influence *top-down* est cruciale car elle permettra au modèle d'appréhender des environnements écologiques, et précède son intégration dans un agent physique. L'insertion du modèle d'abord dans une caméra mobile, puis dans un agent autonome, permettra notamment d'ajouter tout un panel d'actions que l'agent pourra réaliser : saccades oculaires réelles dans un premier temps, puis déplacements dans l'environnement et autour de la cible. Cette nouvelle échelle de complexité perceptif et comportementale pourra être appréhendée par le développement, à partir de notre modèle, d'un agent markovien. Les processus de décision markoviens permettent l'intégration de la perception de l'agent, de ses connaissances sur son environnement et des actions qu'il peut réaliser pour décider quelle sera l'action la plus à même d'aider l'agent à accomplir son objectif (dans nos travaux, détecter et reconnaître la cible). On peut par exemple imaginer une situation où l'agent, observant un individu de dos, décider de façon autonome que la meilleure action à réaliser pour l'identifier sera de tourner autour de celui-ci puis de repérer son visage.^{1,6,9}

En conclusion, ce travail exploratoire nous aura permis de construire un modèle de localisation de cible mettant en avant la simplicité et l'adaptabilité. Celui-ci n'est pour l'instant qu'aux premiers stades de son développement mais de nombreuses avancées pourront y être implantées dans le futur, influençant à la fois ses performances et son fonctionnement. Malgré des performances non-quantifiées, nous pensons que ce travail pourra servir de preuve de concept pour ces développements futurs.

Bibliography

- [1] Nicholas J Butko and Javier R Movellan. “Infomax control of eye movements”. In: *Autonomous Mental Development, IEEE Transactions on* 2.2 (2010), pp. 91–107.
- [2] Jeremy Freeman and Eero P. Simoncelli. “Metamers of the ventral stream”. In: *Nature Neuroscience* 14.9 (2011), pp. 1195–1204. ISSN: 10976256. DOI: [10.1038/nn.2889](https://doi.org/10.1038/nn.2889).
- [3] Philip Kortum and Wilson S. Geisler. “Implementation of a foveated image coding system for image bandwidth reduction”. In: *SPIE Proceedings* 2657 (1996), pp. 350–360. ISSN: 0277786X. DOI: [10.1117/12.238732](https://doi.org/10.1117/12.238732).
- [4] Richard J. Krauzlis, Laurent Goffart, and Ziad M. Hafed. “Neuronal control of fixation and fixational eye movements”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 372.1718 (2017), p. 20160205. ISSN: 0962-8436. DOI: [10.1098/rstb.2016.0205](https://doi.org/10.1098/rstb.2016.0205). URL: <http://rstb.royalsocietypublishing.org/lookup/doi/10.1098/rstb.2016.0205>.
- [5] P. S. Leon et al. “Motion clouds: model-based stimulus synthesis of natural-like random textures for the study of motion perception”. In: *Journal of Neurophysiology* 107.11 (2012), pp. 3217–3226. ISSN: 0022-3077. DOI: [10.1152/jn.00737.2011](https://doi.org/10.1152/jn.00737.2011). arXiv: [arXiv:1208.6467v1](https://arxiv.org/abs/1208.6467v1). URL: <http://jn.physiology.org/cgi/doi/10.1152/jn.00737.2011>.
- [6] J Najemnik and Wilson S. Geisler. “Optimal eye movement strategies in visual search”. In: *Nature reviews. Neuroscience* 434 (2005), pp. 387–391. URL: <http://dx.doi.org/10.1038/nature03390>.
- [7] Ken Perlin. “An image synthesizer”. In: *Computer Graphics* 19.3 (1985), pp. 287–296. DOI: [10.1145/325334.325247](https://doi.org/10.1145/325334.325247).
- [8] John S. Werner and Leo M. Chalupa, eds. *The new visual neurosciences*. MIT Press. 2014, p. 1675. ISBN: 9780262019163.
- [9] Li Zhaoping. *Understanding vision : theory, models and data*. Oxford Uni. 2014, p. 383. ISBN: 9780199564668.

A Figures

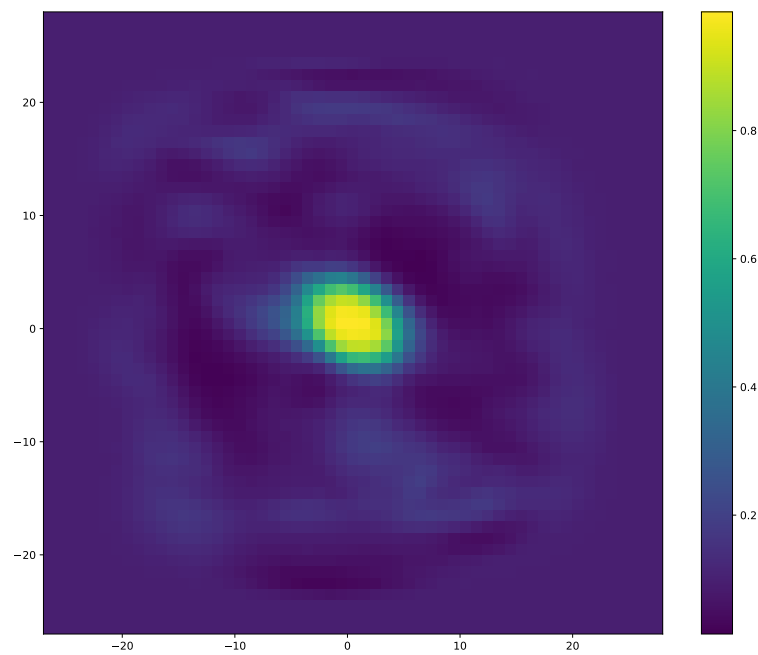


FIGURE A.1: Reconstruction en carte de chaleur (55×55 pixels) de la matrice de certitude

Système d'exploitation	Processeur	Mémoire vive	Carte graphique
Ubuntu 16.04.4	Intel Xeon E5-1607 (3,1GHz)	40 GB	NVIDIA GeForce GTX1060

TABLE A.1: Matériel utilisé pour réaliser les modélisations

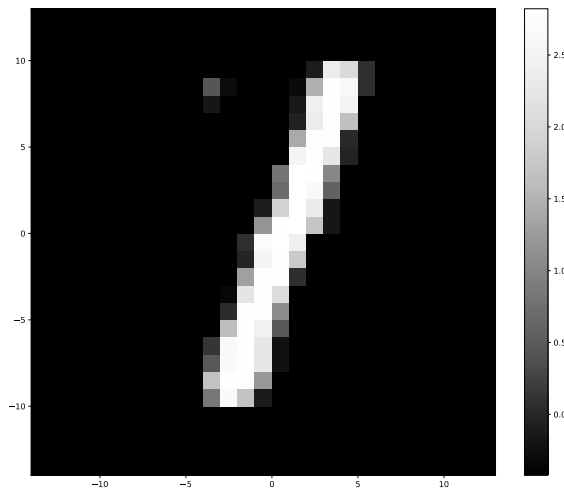


FIGURE A.2: Reconstruction en carte de chaleur (28×28 pixels) d'une image provenant de la base de données MNIST

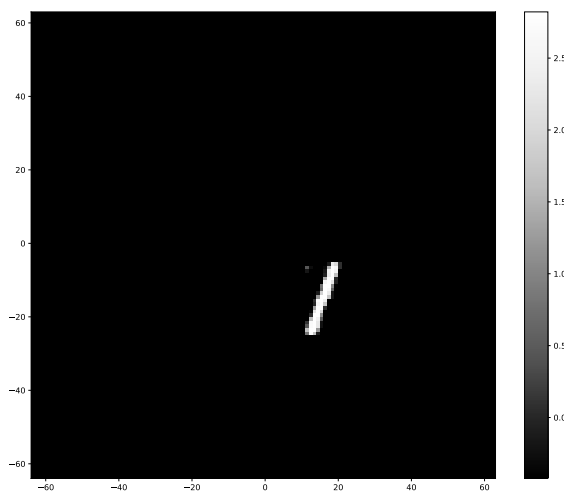


FIGURE A.3: Reconstruction en carte de chaleur d'une image provenant de la base de données MNIST, après transformation pour la placer dans une image de dimension 128×128 pixels

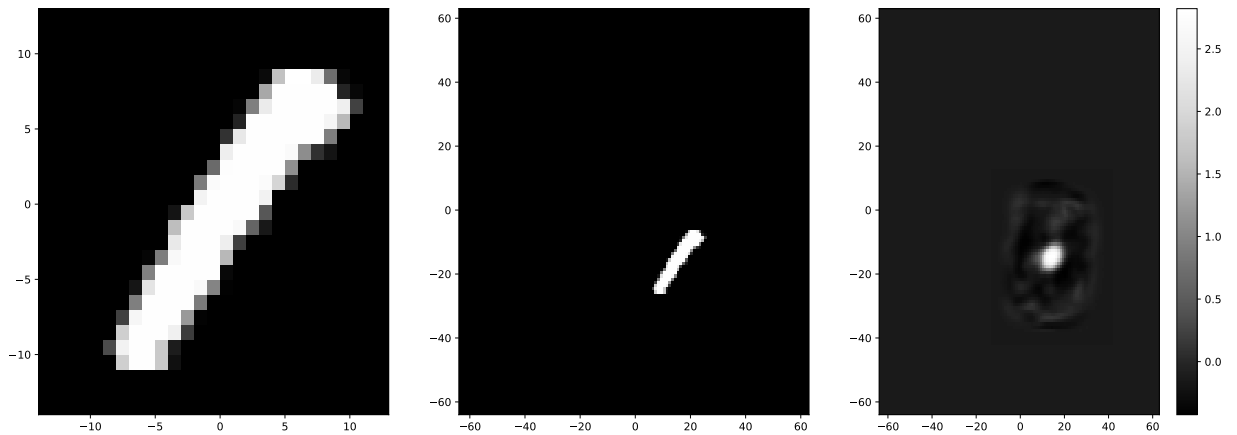


FIGURE A.4: Illustration en cartes de chaleur de la construction automatique d'un couple stimulus/label par la modèle. L'emplacement du chiffre MNIST dans l'image 128×128 pixels est choisi au hasard, et cette même position est reprise pour placer la carte de certitude dans l'autre image 128×128 pixels

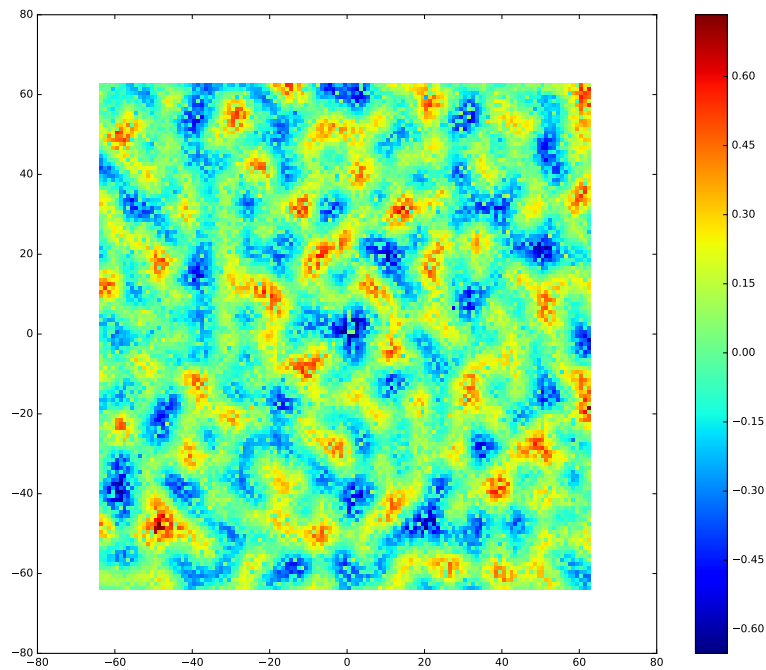


FIGURE A.5: Reconstruction en carte de chaleur (128×128 pixels) d'un bruit Perlin généré automatiquement et aléatoirement⁷

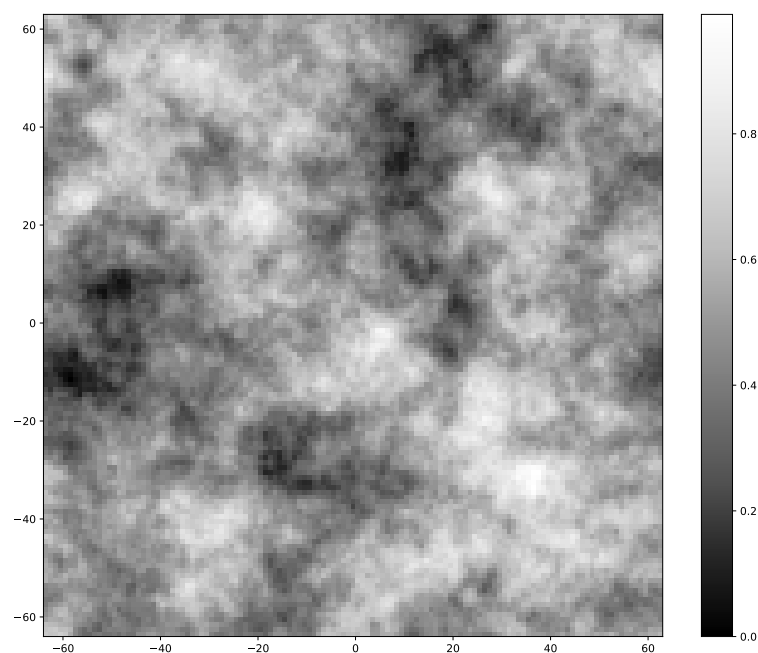


FIGURE A.6: Reconstruction en carte de chaleur (128×128 pixels) d'un bruit Motion-Cloud généré automatiquement et aléatoirement⁵

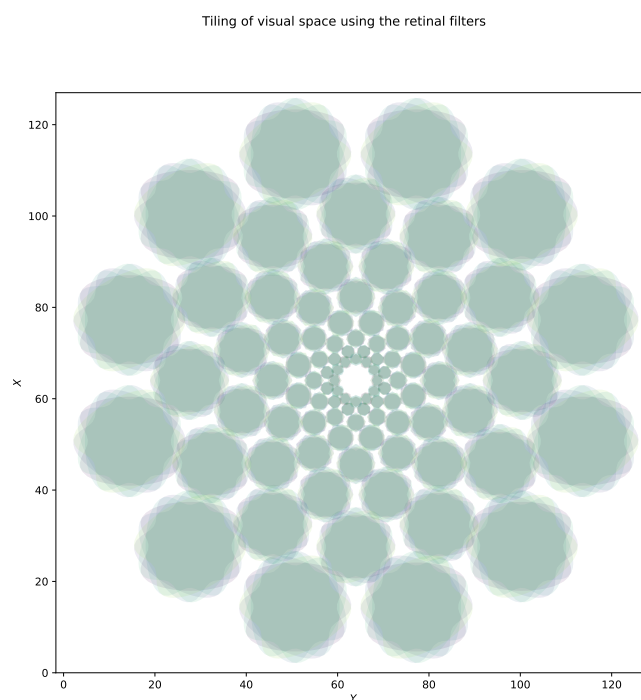


FIGURE A.7: Schéma (128×128 pixels) représentant le filtre LogPolaire pour les paramètres ($\theta=6$, $\text{azimuth}=12$, $\text{eccentricity}=8$, $\text{phase}=2$, $\rho=1.61803$). Chaque ovoïde représente un champs récepteur, modélisé par un filtre Gabor²

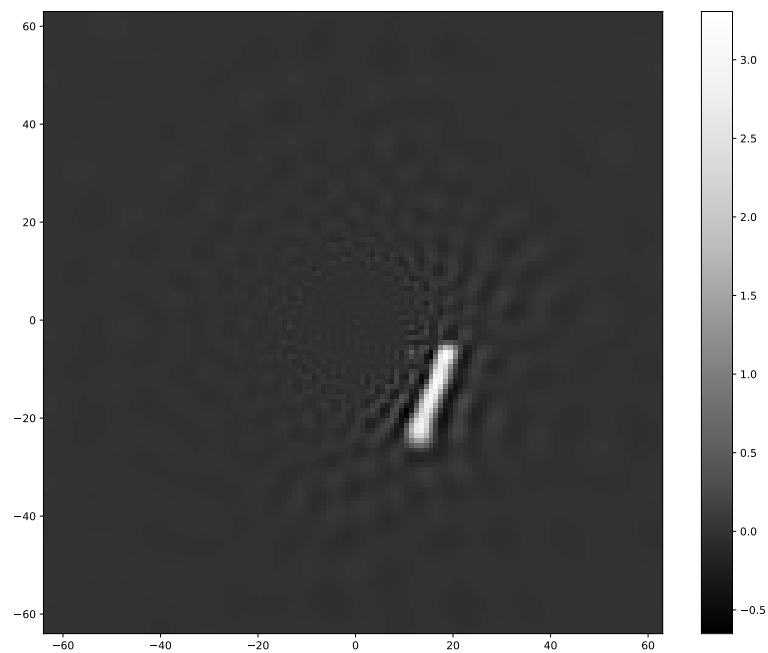


FIGURE A.8: Reconstruction en carte de chaleur (128×128 pixels) d'un stimulus non-bruité après passage dans le filtre rétinien LogPolaire

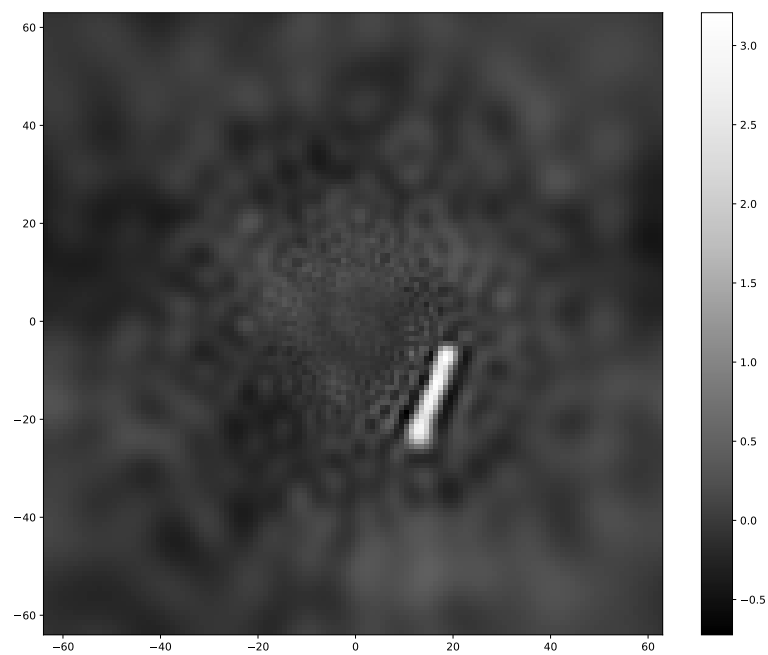


FIGURE A.9: Reconstruction en carte de chaleur (128×128 pixels) d'un stimulus bruité par la méthode MotionCloud après passage dans le filtre rétinien LogPolaire

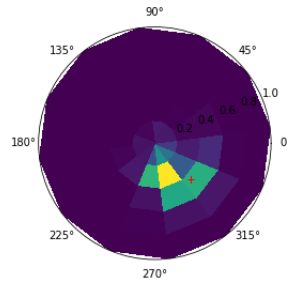


FIGURE A.10: Reconstruction en graphique logarithmique d'un stimulus non-bruité après passage dans le filtre rétinien LogPolaire. La croix rouge représente la position du stimulus

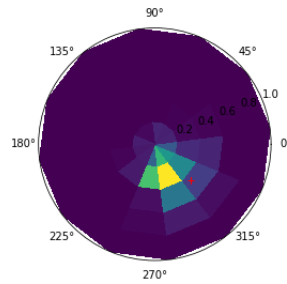


FIGURE A.11: Reconstruction en graphique logarithmique d'un stimulus bruité par la méthode MotionCloud après passage dans le filtre rétinien LogPolaire. La croix rouge représente la position du stimulus

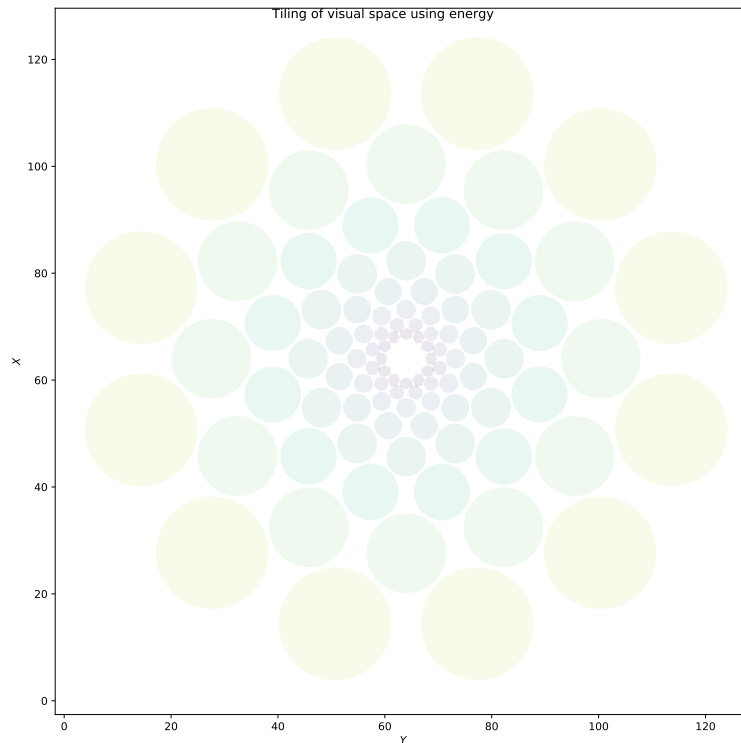


FIGURE A.12: Schéma (128×128 pixels) représentant le filtre LogPolaire énergétique pour les paramètres (azimuth=12, eccentricity=8, $\rho=1.61803$). Chaque cercle représente un champ récepteur

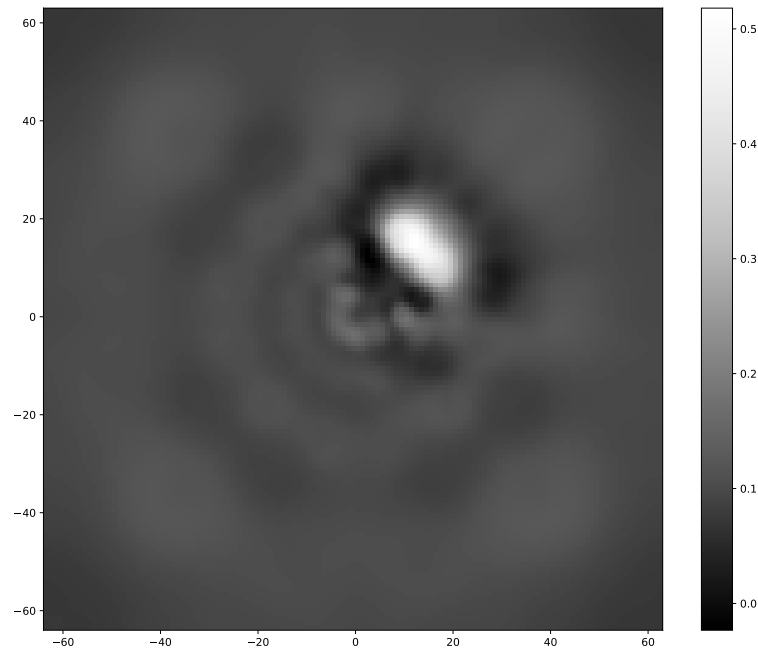


FIGURE A.13: Reconstruction en carte de chaleur (128×128 pixels) de la carte de certitude après passage dans le filtre rétinien LogPolaire

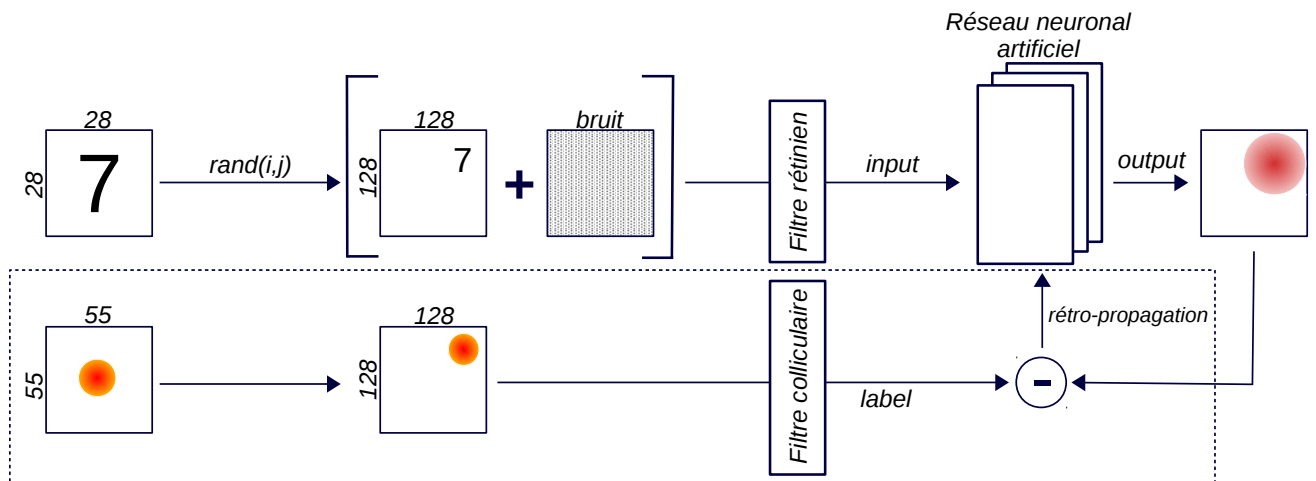


FIGURE A.14: Schéma décrivant les étapes nécessaires à la production de nos entrées (ou *input*) et de nos labels, puis celles pour que notre modèle réalise des prédictions (sorties ou *output*) de la position des stimuli. La partie entourée en pointillés, correspondant à la production et l'intégration du label, n'est réalisée que lors de la phase d'apprentissage

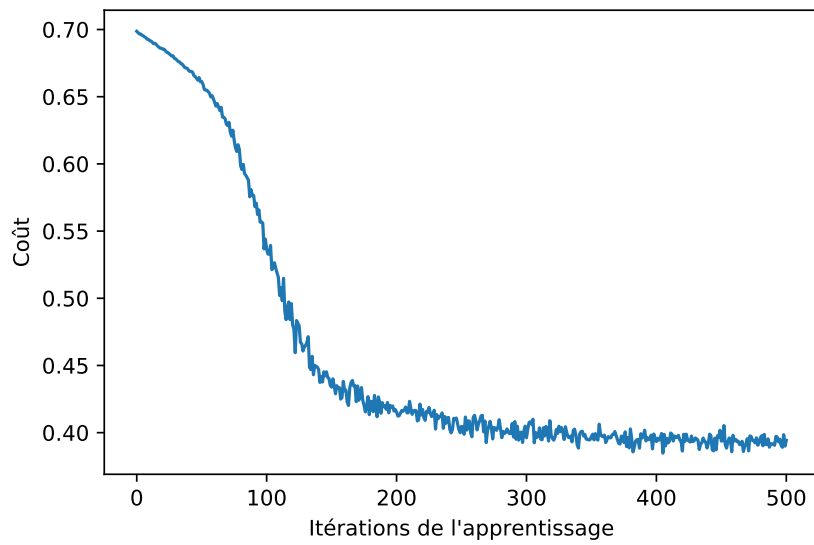


FIGURE A.15: Variations de la valeur de coût du modèle au cours de la première époque d'apprentissage, pour un paramètre d'apprentissage $\alpha = 0.05$

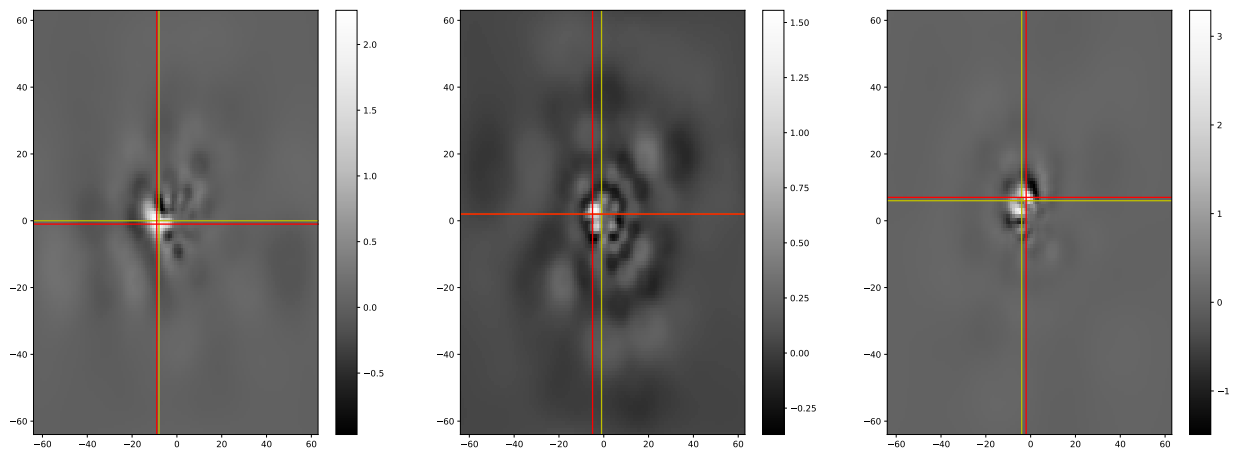


FIGURE A.16: Reconstructions en carte de chaleur (128×128 pixels) de la sortie du modèle lors de la phase d'évaluation. Chaque figure représente un essai différent. Les croix jaunes représentent la position réelle de la cible dans l'environnement visuel, tandis que les croix rouges représentent sa position prédite par le modèle

B Code source et documents complémentaires

L'ensemble du code source du modèle sous forme de notebooks jupyter et de scripts python, de ce rapport au format \LaTeX ainsi que de l'ensemble des autres documents issus de ce travail (dont les notes personnelles au format Markdown) sont entièrement disponibles **en ligne** ou en contactant directement l'**auteur**.

```
class Net(torch.nn.Module):
    def __init__(self, n_feature, n_hidden1, n_hidden2, n_output):
        super(Net, self).__init__()
        self.hidden1 = torch.nn.Linear(n_feature, n_hidden1)
        self.hidden2 = torch.nn.Linear(n_hidden1, n_hidden2)
        self.predict = torch.nn.Linear(n_hidden2, n_output)

    def forward(self, data, do_leaky_relu=True):
        if not do_leaky_relu:
            data = F.relu(self.hidden1(data))
            data = F.relu(self.hidden2(data))
        else:
            data = F.leaky_relu(self.hidden1(data))
            data = F.leaky_relu(self.hidden2(data))
        data = F.sigmoid(self.predict(data))
        return data
```