Symétrie U(1) et brisure de symétrie dans les couches d'activation de réseaux de neurones convolutifs profonds

par

Louis-François BOUCHARD

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE DES SYSTÈMES
M. Sc. A.

MONTRÉAL, LE 31 AOÛT 2022

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Matthew Toews, directeur de mémoire
Département de génie des systèmes à l'École de technologie supérieure

M. Rafael Menelau Cruz, président du jury
Département de génie logiciel et des TI à l'École de technologie supérieure

M. Marco Pedersoli, membre du jury
Département de génie des systèmes à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 10 AOÛT 2022

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

AVANT-PROPOS

Dans cette thèse, Louis-François propose un modèle de propagation de l'information à travers l'espace, cohérent avec des particules de lumière dans un système optique équipé d'une lentille, c'est-à-dire les théories de Newton, Maxwell et Einstein, des particules d'information dans un réseau de neurones profond équipé d'un mécanisme de mise au point.

La théorie $U(1)$ de l'orientation de l'information est exprimée à l'aide d'opérateurs symétriques et antisymétriques, définissant les particules de bosons et de fermions en 3D telles que les photons et les électrons (physique des particules connues, Einstein, Schrödinger, Dirac, années 1900) et les particules de types "anyon" dans le plan 2D (Frank Wilczek, statistiques fractionnaires de quasi-particules en deux dimensions découvertes en 1982 et observées en 2020).

Les réponses des réseaux de neurones biologiques montrent des réponses d'orientation similaires, comprenant à la fois le traitement des neurones peu profonds dans le cortex visuel (David Hubel et Torsten Wiesel, prix Nobel 1981) et dans les couches profondes (par exemple, la drosophile mélanogaster "mouche à fruit" 1995).

Il semble naturel que les réseaux de neurones profonds soient de plus en plus analysés dans le cadre analogique de la propagation de l'information, en respectant les théories connues de l'information en 2D et 3D (à partir de 2022).

- Matthew Toews

REMERCIEMENTS

Je tiens à remercier mon superviseur pour son aide, ses conseils et les innombrables heures consacrées à m'apprendre et m'orienter. Matthew, merci pour l'opportunité que tu m'as donné de travailler avec toi pour mon PFE puis de poursuivre à la maîtrise. Ce fut ma meilleure expérience académique jusqu'à présent. Merci pour les belles discussions. Une rencontre avec toi est toujours un plaisir. Un dernier merci pour l'aide financière que tu as pu m'apporter tout au long de cette maîtrise.

Je suis également reconnaissant à Mohsen Ben Lazreg pour son travail important et nécessaire pour débuter cette thèse. Merci pour ton aide et pour avoir partagé ton code avec moi lors du début de cette maîtrise. Je suis également reconnaissant de t'avoir eu comme chargé de laboratoire dans la classe de Matthew donnant des "mini-cours" pendant les laboratoires où j'ai beaucoup appris.

Je tiens également à remercier NVIDIA pour m'avoir fourni un GPU Titan RTX pour mes recherches afin de soutenir mon travail, partageant la science sur YouTube. Ce GPU m'a aidé pendant plus d'un an, fonctionnant souvent jour et nuit pour d'innombrables expériences.

Ce travail a également été soutenu financièrement par un financement du Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) avec le programme Bourses d'études supérieures du Canada - Maîtrise (BESC M).

Symétrie U(1) et brisure de symétrie dans les couches d'activation de réseaux de neurones convolutifs profonds

Louis-François BOUCHARD

RÉSUMÉ

Nous présentons un nouveau modèle reliant les réseaux de neurones convolutifs (CNNs) à la vision biologique et à la physique fondamentale des particules. La propagation de l'information dans un CNN est modélisée via une analogie avec un système optique, où l'information est concentrée près d'un goulot d'étranglement où la résolution spatiale 2D s'effondre autour d'un point focal $1 \times 1 = 1$. Un espace 3D (t, x, y) est défini par les coordonnées (x, y) dans le plan image et la couche CNN t, où un rayon principal $(t, 0, 0)$ suis la direction de propagation de l'information à travers à la fois l'axe optique et le pixel central de l'image situé à $(x, y) = (0, 0)$, autour duquel la mise au point spatiale la plus nette possible est limitée à un cercle de confusion dans le plan image. Notre idée est de modéliser le rayon optique principal $(t, 0, 0)$ comme géométriquement équivalent au vecteur médian dans l'orthant positif $I(x, y) \in R^{N+}$ d'un espace d'activation des canaux a N dimensions, par exemple le long du vecteur de niveaux de gris (ou de luminance) (t, t, t) dans l'espace colorimétrique RVB. L'information est ainsi concentrée dans un potentiel d'énergie $E(t, x, y) = \|I(t, x, y)\|^2$, qui, en particulier pour les couches de goulot d'étranglement t des CNNs génériques, est fortement concentré et symétrique par rapport à l'origine spatiale $(t, 0, 0)$ et présente le potentiel "Sombrero" de la particule de boson bien connu. Cette symétrie est brisée dans la classification où les couches de goulot d'étranglement des modèles CNN génériques préentraînés présentent un biais cohérent spécifique à la classe vers un angle $\theta \in U(1)$ défini simultanément dans le plan image et dans l'espace des caractéristiques d'activation. Les observations initiales valident notre hypothèse à partir de cartes d'activation extraites de CNNs génériques préentraînés et d'un schéma de classification basé sur la mémoire (K-NN), sans entraînements supplémentaires ni réglages optimisés. L'entraînement à partir de zéro à l'aide d'une fonction de perte combinant un composant symétrique et un composant antisymétrique, one-hot $+U(1)$, améliore la classification pour toutes les tâches testées, y compris ImageNet, confirmant notre hypothèse de brisure de symétrie lors de la classification et agissant comme preuve de concept introduisant cette théorie. Nous soulignons également la ressemblance entre une image (réguliere, RVB) et une carte de caractéristique.

Mots-clés: Vision par Ordinateur, Reconnaissance de Formes, Réseaux de Neurones Convolutifs

U(1) Symmetry-breaking Observed in Generic CNN Bottleneck Layers

Louis-François BOUCHARD

ABSTRACT

We report on a novel model linking deep convolutional neural networks (CNN) to biological vision and fundamental particle physics. Information propagation in a CNN is modeled via an analogy to an optical system, where information is concentrated near a bottleneck where the 2D spatial resolution collapses about a focal point $1 \times 1 = 1$. A 3D space (t, x, y) is defined by (x, y) coordinates in the image plane and CNN layer t, where a principal ray $(t, 0, 0)$ runs in the direction of information propagation through both the optical axis and the image center pixel located at $(x, y) = (0, 0)$, about which the sharpest possible spatial focus is limited to a circle of confusion in the image plane. Our novel insight is to model the principal optical ray $(t, 0, 0)$ as geometrically equivalent to the medial vector in the positive orthant $I(x, y) \in R^{N+}$ of a N-channel activation space, e.g. along the greyscale (or luminance) vector (t, t, t) in RGB colour space. Information is thus concentrated into an energy potential $E(t, x, y) = \|I(t, x, y)\|^2$, which, particularly for bottleneck layers t of generic CNNs, is highly concentrated and symmetric about the spatial origin $(t, 0, 0)$ and exhibits the well-known "Sombrero" potential of the boson particle. This symmetry is broken in classification, where bottleneck layers of generic pre-trained CNN models exhibit a consistent class-specific bias towards an angle $\theta \in U(1)$ defined simultaneously in the image plane and in activation feature space. Initial observations validate our hypothesis from generic pre-trained CNN activation maps and a bare-bones memory-based classification scheme, with no training or tuning. Training from scratch using combined one-hot $+U(1)$ loss, including our antisymetric component to the symetric one-hot component, improves classification for all tasks tested including ImageNet, confirming our symmetry breaking hypothesis during classification and acting as proof of concept introducing this theory. We also point out the resemblance between an image (regular, RGB) and a feature map.

Keywords: Computer Vision, Pattern Recognition, Convolutional Neural Networks

TABLE DES MATIÈRES

Page

INTRODUCTION 1
CHAPITRE 1 TRAVAUX CONNEXES 9
Lumière, Information et Réseaux de Neurones Profonds 9
Notions Géométriques 9
Rupture de Symétrie dans les DNNs 13
Données, Granularité et Classification 15
Adaptation de Domaine, Apprentissage en Peu d'Étapes, Pré-Entraînement et Méthodes d'Encodage d'Informations 19
CHAPITRE 2 PROPAGATION DE L'INFORMATION ET RUPTURE DE SYMÉTRIE U(1) 23
2.1 Analogie de la Propagation de la Lumière et de l'Information 23
CHAPITRE 3 OBSERVATIONS ET EXPÉRIMENTATIONS AVEC LA RUPTURE DE SYMÉTRIE 29
3.1 Protocole Expérimental de nos Observations 29
3.1.1 Observation de la Rupture de Symétrie dans les Couches de Goulot d'Étranglement CNN 36
3.2 Expériences Initiales d'Entraînement avec un Encodeur-Décodeur 39
2.5 Entraînement avec les Annotations U(1) 44
CONCLUSION ET RECOMMANDATIONS 53
ANNEXE I TRAVAUX PRÉLIMINAIRES : ANALYSE DU RÉSEAU ET ENSEMBLES DE DONNÉES 57
ANNEXE II OPTIMISATIONS : PCA ET DOMAINE SPECTRAL 63
BIBLIOGRAPHIE 75
Tableau 3.1 - Résultats de classification avec annotations U(1) 50

Algorithme 3.1 - Construction de nos bases de données de pixels I et I' pour la correspondance de pixels 32
Algorithme 3.2 - Classification - K plus proches voisins avec la correspondance de pixels 33

LISTE DES FIGURES

Page

Figure 0.1 Images et CNNs 2

Figure 0.2 Comparaison entre un système de lentilles et un réseau de neurones profond. 3

Figure 0.3 Système de couleurs et information au goulot d'étranglement. 4

Figure 1.1 Fonctions symétriques et antisymétriques. 12

Figure 1.2 Rupture de symétrie $U(1)$ observée dans les activations des Réseaux de Neurones Convolutifs (Convolutional Neural Networks) (CNN). 14

Figure 1.3 Estimation subjective du niveau de granularité. 17

Figure 2.1 Système de couleurs et information au goulot d'étranglement. 27

Figure 3.1 Visualisation de la concaténation des tenseurs extraits. 30

Figure 3.2 Notre architecture de vecteurs de pixels proposée. 31

Figure 3.3 $U(1)$ rupture de symétrie observée. 37

Figure 3.4 Distributions $p(\bar{x}_{nn}|\bar{x}_i)$ des NN emplacements de pixels correspondants $\bar{x}_{nn}$ conditionnés aux emplacements l'image d'inférence $\bar{x}_i$. 39

Figure 3.5 Comparaison d'entraînements à partir de zéro. 43

Figure 3.6 Étude de la constante alpha. 47

Figure 3.7 Évolution de la précision, de la perte et des prédictions du modèle $U(1)$ au fil du temps. 49

Figure 3.8 Comparaison de la précision de la classification pour l'entraînement à partir de zéro. 51

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

CNN	Réseaux de Neurones Convolutifs (Convolutional Neural Networks).
DNN	Réseaux de neurones profonds (Deep Neural Networks).
ETS	École de Technologie Supérieure.
FFT	Transformée de Fourier Rapide (Fast Fourier Transform).
GAN	Réseau antagoniste génératif (Generative Adversarial Network).
GeM	Generalized Mean pooling.
GPU	Processeurs graphiques.
IRM	Imagerie par résonance magnétique.
KNN	K plus proches voisins (K Nearest Neighbors).
PCA	Analyse des composants principaux (Principal Component Analysis).
ReLU	Rectified Linear Unit Layer.
SIFT	Scale-Invariant Feature Transform.
VAE	Auto-encodeurs variationnels (Variational Autoencoders).

INTRODUCTION

Les systèmes de vision par ordinateur modernes impliquent l'acquisition d'images, le traitement de l'information et l'indexation de la mémoire. On sait peu de choses sur la géométrie des informations propagées dans les Réseaux de neurones profonds (Deep Neural Networks) (DNN)s, des systèmes de facto à la pointe de la technologie pour les tâches de vision par ordinateur tels que la classification et la segmentation, Krizhevsky, Sutskever & Hinton (2012a), de données d'image annotées. Cette thèse étudie la géométrie de l'information propagée dans les CNNs profonds avec des analogies simples, des observations et des expériences généralisables visant à partager une nouvelle façon d'interpréter les CNNs et leurs cartes de caractéristiques (représentation de l'information propagée).

Tout d'abord, nous souhaitons souligner cette simplification peu répandue qu'une carte de caractéristique peut être vue comme une image régulière (RVB). Effectivement, une image RVB, comme affichée à la figure 0.1, est une représentation visuelle contenant trois canaux : rouge, vert, bleu. Ces trois canaux représentent la réponse de l'image aux filtres de chacune de ces couleurs nous informant sur leur intensité. Ainsi, chaque pixel de l'image, soit leur position (x, y) dans l'espace spatial 2D peut être représentée par un vecteur RVB d'intensité, nous informant sur la couleur finale du pixel lors de l'affichage de l'image. Ainsi, un pixel d'une image RVG aura trois dimensions, chacune représentant la réponse d'un filtre pour une des trois couleurs. Similairement, comme nous allons le voir dans les travaux connexes, une carte de caractéristique, étant le produit d'une couche de convolution t, peut être vue de la même façon. Ces cartes de caractéristiques auront le même espace bidimensionnel où chaque pixel représentera une intensité de réponse de certains filtres. L'unique différence est que ces filtres ne représenteront plus des couleurs précises (taux de rouge, vert et bleu), mais des caractéristiques plus précises apprises par le réseau pour la tâche pour laquelle il a été entraîné. Ainsi, chaque pixel nous indiquera son intensité, de dimension N représentant la réponse de l'image aux N filtres utilisés à la couche de convolution précédente. Nous croyons que cette comparaison aidera

la compréhension du document et devrait être partagée plus fréquemment lors de l'introduction aux CNNs à de nouveaux étudiants.

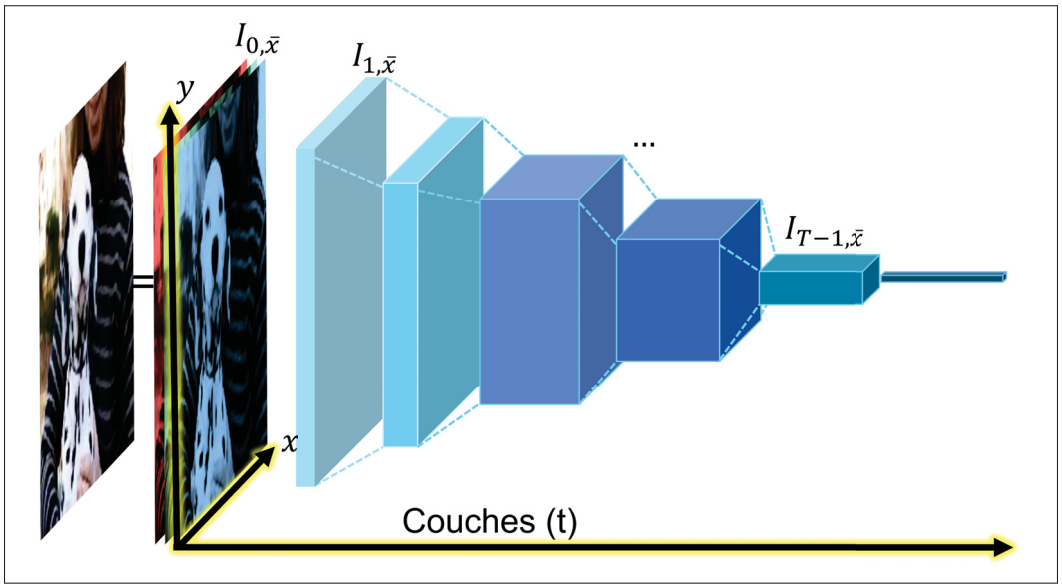

FIGURE 0.1 Visualisation d'une image RVB et d'un réseau de convolution de base, style VGG, Simonyan & Zisserman (2015), affichant notre représentation (t, x, y) de l'information propagée dans le réseau.

Afin de répondre au questionnement sur la géométrie des informations propagées dans les DNNs, nous souhaitons partager une analogie entre la propagation des informations visuelles dans les réseaux profonds et la propagation de la lumière dans un appareil optique avec une lentille et un foyer. La propagation de l'information dans des réseaux profonds *entraînés* peut être considérée comme une information propagée dans l'espace (x, y) à travers le temps t avec des canaux d'intensité $I_{(t,x,y)}$ centrés sur le même rayon principal $(t, 0, 0)$, illustré à la Figure 0.2. Notant que, dans une telle représentation, les couches les plus informatives (profondes, au goulot d'étranglement, Wang, Girshick, Gupta & He (2018b); Lazreg (2020)) condensent *l'espace image* à un point $1x1 = 1$ semblable à la lumière.

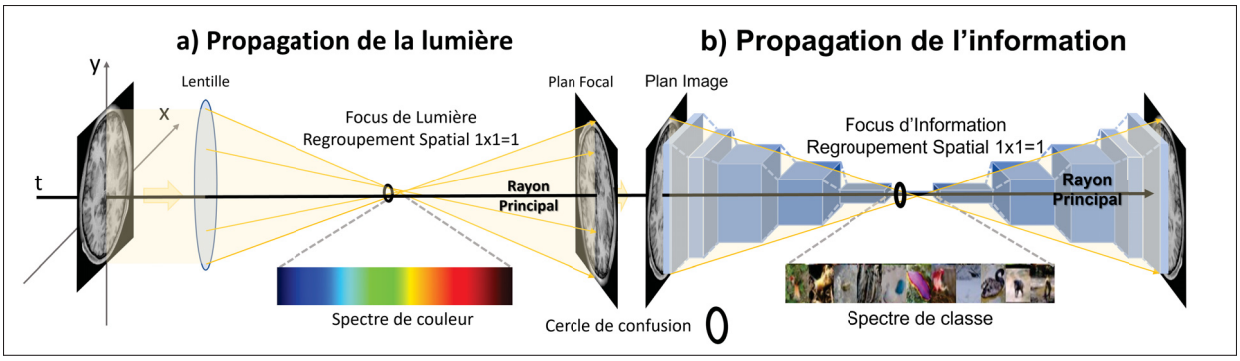

FIGURE 0.2 a) La propagation des rayons lumineux à travers un système de lentilles optiques. b) la propagation de l'information à travers un réseau de neurones profond (DNN). Les aspects partagés incluent un rayon principal commun, un spectre et un cercle de confusion. **Nous montrons ici un système encodeur-décodeur pour l'analogie, mais nos travaux mettent l'accent sur la partie gauche du système représentant l'encodeur et des tâches de classifications utilisant le "spectre de classe".** Image adaptée de Bouchard *et al.* (2022).

Une lentille mince de diamètre d et de longueur focale f peut être utilisée pour traiter une illumination d'entrée sur un spectre de fréquences. Nous accordons une attention particulière au *cercle de confusion* illustré sur la Figure 0.2 (a) dans l'encodeur d'une telle configuration (partie gauche). Ce cercle de confusion, causé par des contraintes du monde réel, ici principalement les imperfections de la lentille, y compris la diffraction et les aberrations de la lentille, limite la précision de l'information à un rayon spatial minimum. À cet instant précis t le long de l'axe principal $(x, y) = (0, 0)$ traversant le point focal et le centre de la lentille, on observe un goulot d'étranglement spatial où l'information est alors regroupée en un $1 \times 1 = 1$ foyer lumineux où un spectre de couleur peut être observé (Figure 0.2 (a)). Les encodeurs CNNs agissent de la même manière pour préparer les données pour une tâche de classification, Figure 0.2 (b, partie gauche jusqu'au spectre de classe). Notre hypothèse repose sur ce cercle de confusion généralisé aux CNNs limitant l'espace spatial minimum où condenser cette espace en $1 \times 1 = 1$ dimensions spatiales endommagerait la qualité de l'information en y ajoutant de la confusion (bruit), perdant le bénéfice qu'apporte la position (x, y) des pixels $I_{(t,x,y)}$.

Cela nous amène à notre seconde analogie introduisant notre méthode, celle-ci avec les couleurs. Les couleurs peuvent être représentées autour d'un cercle unitaire en deux dimensions $U(1)$, $x^2 + y^2 = 1$, ou avec un cône tridimensionnel (espace HSL), en utilisant un seul angle $\theta \in U(1)$, comme illustré dans la Figure 0.3 (a). Notons ici que l'angle identifiera la couleur représentant la "teinte" dans l'espace HSL lorsque notre saturation et luminance sera maximale. Nous pouvons alors simplifier le choix de la couleur par un simple angle autour du cercle affiché dans la Figure 0.3 (a). Similairement, nous suggérons qu'un simple angle $\theta \in U(1)$ extrait de l'information en fin de réseau (maximum de condensation d'information, similaire au niveau de contraste et luminance maximaux, montré dans la Figure 0.3 (a)) peut aider les réseaux profonds à mieux séparer les classes et améliorer la précision de leurs résultats, Figure 0.3 (b).

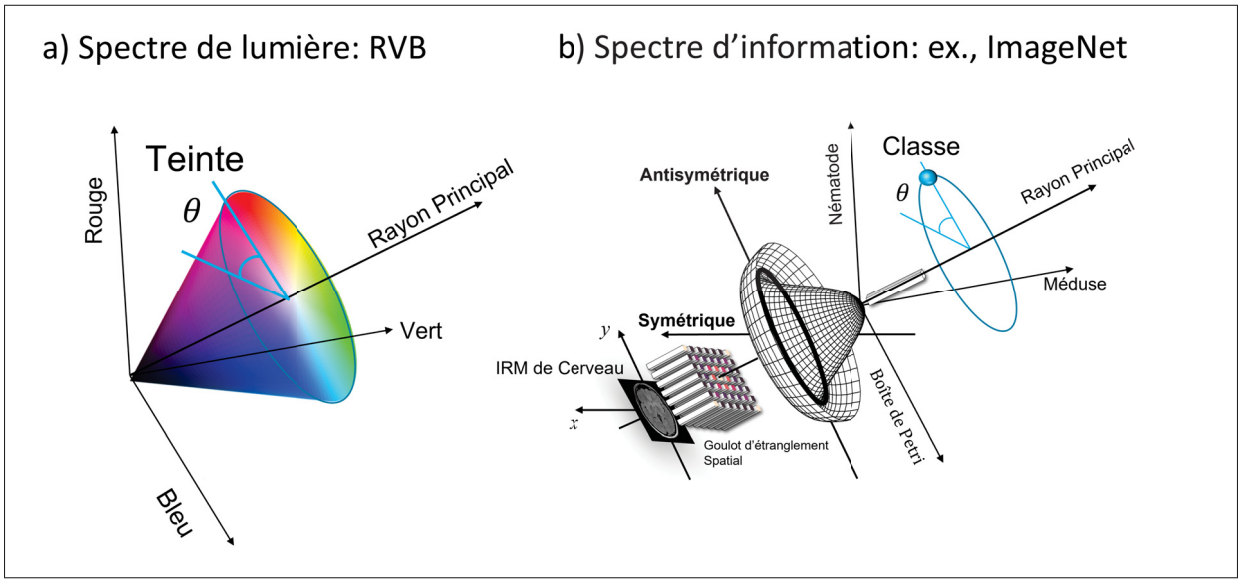

FIGURE 0.3 Illustration des systèmes de coordonnées polaires pour les données multicanaux. a) L'espace colorimétrique rouge-vert-bleu (RVB) standard à 3 canaux avec l'angle de teinte (hue) θ. b) Un sous-espace à 3 canaux de la sortie ImageNet, Deng *et al.* (2009), à 1000 canaux, suivant la couche de goulot d'étranglement spatial, montrant un angle de classe θ analogue à la teinte (hue) de couleur et défini par la classe de l'image d'entrée. Affichage d'un espace où une image d'Imagerie par résonance magnétique (IRM) est envoyé dans un réseau CNN préentraîné sur ImageNet. En b), la classe "cerveau, IRM" (non incluse dans ImageNet) correspond aux classes ImageNet les plus similaires, soit : Nématode, Boîte de Petri et Méduse. Image adaptée de Bouchard *et al.* (2022).

Nous utilisons ces analogies afin d'expliquer nos hypothèses et notre approche répondant à une question que nous jugeons essentielle : **est-ce que le positionnement spatial est important et informatif dans les couches profondes des réseaux de neurones ?** Et si oui, **comment pouvons-nous l'optimiser pour maximiser la qualité de la représentation de l'information que nous avons dans ces couches** dans des tâches telles que la classification d'image et ainsi améliorer les résultats en termes de précision et/ou quantité de données nécessaires ?

Afin de répondre à ces questions, nous apportons plusieurs **observations** appuyant ce cercle de confusion (expliquant l'espace (x, y) minimal à respecter), justifiant notre analogie avec les couleurs, confirmant notre **hypothèse** $U(1)$ **sur les réseaux profonds préentraînés** visualisés dans des essais de classification rudimentaires via la recherche classique du K plus proches voisins (K Nearest Neighbors) (KNN), Cover & Hart (1967), utilisant les informations d'activation de goulot d'étranglement comme des vecteurs de pixels désenchevêtrés (disentangled), c'est-à-dire des ensembles de vecteurs de pixels $I(t, x, y) \in R^{N+}$ dans l'espace de canaux N obtenus à partir des images d'entrée propagées pour correspondre aux vecteurs de pixels stockés en mémoire permettant des translations (dx, dy) entre les images. Puis, nous mettons de l'avant notre approche one-hot $+U(1)$ afin de représenter nos hypothèses dans l'entraînement de zéro d'un réseau CNN. Les contributions sont les suivantes :

- Tout d'abord, cette configuration d'appariement de pixels désenchevêtrés (ou correspondance de pixels) conduit à une précision de classification supérieure à celle des représentations alternatives, y compris les variantes de regroupement global (global pooling) ou les couches aplaties entières (flattened), démontrant un **potentiel d'optimisation utilisant l'espace** (x, y) omit par les techniques utilisés plus couramment.
- Deuxièmement, nous observons que dans les réseaux CNNs pré-entraînés sur Imagenet, **l'énergie d'activation au goulot d'étranglement spatial** (t) $E(t, x, y) = ||I(t, x, y)||^2$ **est hautement symétrique et concentrée près du centre de l'image** (x, y), très similaire au

potentiel dit "Sombrero" associé aux particules de boson, Goldstone (1961), voir Figure 0.3 (b) et Figure 2.1 (b,c).

- Troisièmement, nous observons **un biais constant de l'information** (x, y) **au niveau du goulot d'étranglement spatial vers une direction angulaire précise** $\theta \in U(1)$ **dans le plan image et partagée par les instances d'une classe (créant une asymétrie de classe)**, compatible avec un processus de rupture de symétrie tel que représenté sur la figure 0.3 (b) où la symétrie de l'énergie totale (contribution #2) est ainsi représentée par une anti-symétrie propre à chaque classe.
- Dernièrement, **nous proposons un système de classification "end-to-end" de base utilisant ces découvertes lors de l'entraînement de zéro d'un réseau CNN** importé sans autres modifications que l'ajout de notre composant antisymétrique $U(1)$ à la fonction de perte traditionnelle, ajoutant un composant antisymétrique au composant symétrique (one-hot) que nous jugeons essentiel. Nous démontrons les performances de ce système de la façon la plus générale possible utilisant plusieurs bases de données variées et architectures CNN, gardant des hyperparamètres de base ne visant pas les résultats les plus hauts possible, mais plutôt une généralisation maximale non affectée par l'optimisation spécifique d'autres paramètres.

Ces observations et nos premiers résultats d'expériences sur des DNNs préentraînés nous amènent à étudier l'hypothèse d'un système d'annotation de classe unitaire $\theta \in U(1)$ contraint dans un cercle unitaire $x^2 + y^2 = 1$ pendant l'entraînement du réseau. Nous démontrons également cette hypothèse dans des tests généraux où le but est simplement de **confirmer cette brisure de symétrie sur le plus grand nombre de tâches et réseaux possibles, sans toutefois viser aux meilleurs résultats possible**. Utiliser une orientation $\theta \in U(1)$ (composant antisymétrique) par classe ajoutée à la fonction de perte d'entropie croisée sur le vecteur "one-hot" (composant symétrique) lors de l'entraînement des CNNs génériques (VGG, Simonyan & Zisserman (2015), ResNet, He, Zhang, Ren & Sun (2016), Inception, Szegedy, Vanhoucke, Ioffe, Shlens & Wojna

(2016), DenseNet, Huang, Liu & Weinberger (2016), EfficientNet, Tan & Le (2019)), sans raffinement des paramètres ou augmentation de données excessives, améliore les résultats de classification dans une grande variété de tâches, y compris des objets génériques, des textures, des animaux et des catégories très spécifiques telles que les espèces d'oiseaux, les avions ou la classification des familles dans le cerveau humain IRMs, agissant comme preuve de concept pour cette théorie de brisure de symétrie "$U(1)$".

CHAPITRE 1

TRAVAUX CONNEXES

Lumière, Information et Réseaux de Neurones Profonds

Nous proposons une analogie entre la propagation de la lumière dans un système optique et le traitement par couche des CNNs où les couches sont évaluées par pas de temps discrets t dans un espace bidimensionnel (x, y) où chaque position représente l'intensité d'un pixel à dimension N représentant la réponse spatiale à des filtres (soit RVB pour une image traditionnelle ou des filtres appris lors de l'entraînement d'un réseau). Ce n'est pas la première fois que cette ressemblance entre les réseaux profonds et la lumière sont faites. Des réseaux profonds ont été utilisés pour simuler des équations différentielles, Dissanayake & Phan-Thien (1994); Chen, Rubanova, Bettencourt & Duvenaud (2018), y compris la résolution de problèmes directs (forward) et inverses et également pour simuler la propagation des ondes via des implémentations de CNN basées sur les Processeurs graphiques (GPU)s, Roska *et al.* (1995); Raissi, Perdikaris & Karniadakis (2019). D'un autre côté, des filtres optiques physiques peuvent également être utilisés pour simuler des réseaux de neurones, où la difficulté réside dans la réalisation des activations non linéaires, Sui, Wu, Liu, Chen & Gu (2020), le même défi que pour construire un DNN dans le domaine spectral afin de permettre au modèle d'empiler des couches et de représenter des fonctions plus complexes, Liu & Luk (2020); Watanabe & Wolf (2021); Rizvi, Ab Rahman, Khalil-Hani & Ayat (2021); Han & Hong (2021); Guan, Zhang, Sethares, Kijowski & Liu (2021); Han & Hong (2021); Zhu, Cui, Zhang, Li & Yang (2021).

Notions Géométriques

L'idée principale est que les informations peuvent être modélisées à travers une lentille physique ou un DNN de la même manière. Dans les deux cas, l'information se propage dans l'espace-temps. Une différence notable avec la lentille physique est que dans les DNNs l'information évolue par pas de temps discrets, de manière séquentielle et anticipée avec des transitions définies par, au

plus simple, des opérateurs de produits scalaires linéaires

$$\bar{I}_{t,\bar{x}} = \phi(I_{t-1,\bar{x}} \cdot \bar{W}_{t,\bar{x}} + \bar{b}_t) \tag{1.1}$$

entre certains pixels et un filtre de même taille, répétés sur l'image entière ainsi que plusieurs filtres, en plus des biais (b) et des activations non linéaires (par exemple tanh, Malfliet (2004), ReLu, Agarap (2018), représenté par ϕ). Cette non-linéarité nous permet d'accumuler ces produits scalaires de couche en couche et créer ces réseaux "profonds". Plusieurs réseaux ajoutent d'autres mécanismes liés aux performances tels que du sous-échantillonnage, de la mise en commun (pooling) et les connexions résiduelles introduites par Resnet He *et al.* (2016).

Nos expérimentations avec les CNNs et les informations aux couches profondes (au goulot d'étranglement) reposent sur des notions géométriques de base, l'espace image $\bar{x} = (x, y) \in R^2$ centré sur $(x, y) = (0, 0)$, un espace-temps 3D $(t, \bar{x}) \in R^+ \times R^2$, et une image RVB $(N = 3)$-canal ou image d'activation $I(t, \bar{x}) \in R^+ \times R^2 \times R^N$ sous forme de vecteur. Un CNN de T couches peut être défini comme $\phi(I_{0,\bar{x}}) = \phi^{(T)} \circ \phi^{(T-1)} \circ ... \circ \phi^{(1)}(I_{0,\bar{x}})$, où $I_{0,\bar{x}}$ est notre image d'entrée (à trois canaux RVB), $I_{t,\bar{x}}$ est notre tenseur (regroupement bidimensionnel de vecteurs $1 \times N$ où N représente le nombre de filtres utilisé par la couche de convolution précédente) d'information de goulot d'étranglement, et ϕ est notre bloc convolutif typique avec les caractéristiques susmentionnées décrit par l'équation 1.1. Nous définissons une seule variable unitaire $U(1)$, c'est-à-dire les coordonnées (x, y) pour les positions des vecteurs de pixels de goulot d'étranglement contraints au cercle unitaire $x^2 + y^2 = 1$ ou de manière équivalente à un angle $\theta = atan2(y, x)$.

Comme pour toute fonction $f(\bar{x})$, $U(1)$ peut également être interprété par la somme $f(\bar{x}) = f_s(\bar{x}) + f_a(\bar{x})$ de composantes symétriques $f_s(\bar{x})$ et antisymétriques $f_a(\bar{x})$, Figure 1.1 (a), par exemple, les composantes $cos(\bar{x})$ et $sin(\bar{x})$ de l'équation d'Euler $e^{i\theta} = \cos(\theta) + i\sin(\theta)$. Définition qui sera d'une importance cruciale dans ce travail. Ceci est dû au fait que les composantes symétriques et antisymétriques sont perpendiculaires, pouvant ainsi représenter un système de coordonnées bidimensionnel valable similaire à l'espace (x, y) traditionnel. De

telles variables unitaires sont bien connues en analyse mathématique et de plus en plus utilisées dans les architectures d'apprentissage automatique, Tang *et al.* (2021); Kiani, Balestriero, Lecun & Lloyd (2022). Pour en revenir à notre analogie avec les particules de lumière, les particules 3D fondamentales sont définies comme des champs ou des fonctions d'onde $\psi(\bar{x})$, appelés bosons $\psi_s(\bar{x})$ si la fonction d'onde est symétrique (p. ex. photon et boson de Higgs) et en fermions $\psi_a(\bar{x})$ si la fonction est antisymétrique (p. ex. électron), selon leurs spins (multiples entiers ou multiples demi-entiers), Figure 1.1, (b).

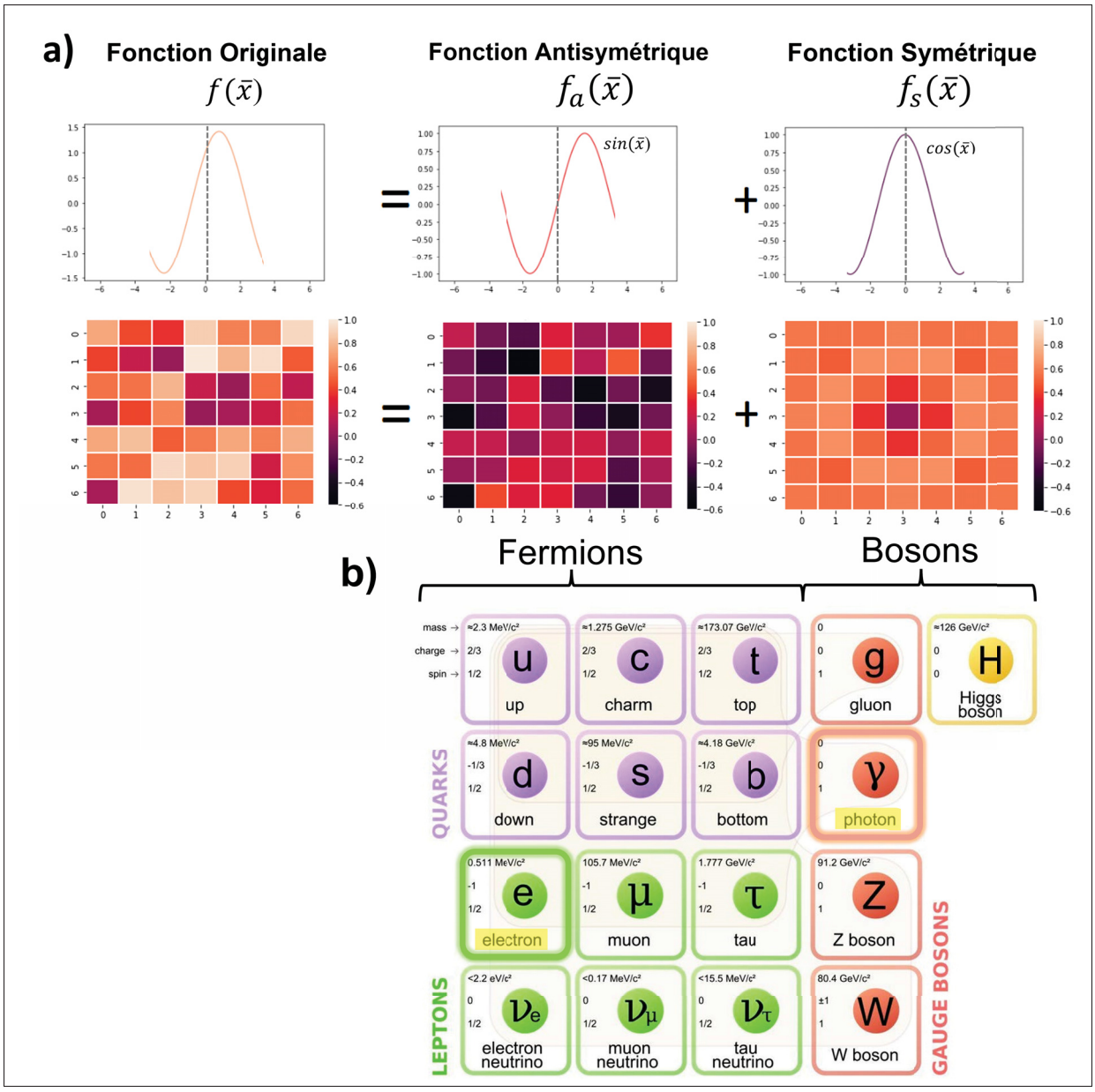

FIGURE 1.1 a) montre le lien avec les fonctions symétriques et antisymétriques à l'aide d'une fonction de base $sin(\bar{x}) + cos(\bar{x})$ et une fonction originale bidimensionnelle (tel qu'une carte de caractéristique générée par un filtre à une couche profonde d'un réseau CNN) générée aléatoirement (à gauche) reconstruite par l'addition d'une composante symétrique et antisymétrique (à droite). b) (adapté de PBS NOVA (2019)) montre les particules du modèle standard. Les Fermions forment les trois colonnes de gauche (par exemple, l'électron), asymétriques ; les bosons peuplent les deux colonnes de droite (par exemple, le photon), symétriques.

Rupture de Symétrie dans les DNNs

La rupture de symétrie est un processus par lequel un potentiel d'énergie passe d'un état symétrique à un état asymétrique, Figure 1.2 (a). Dans les réseaux de neurones profonds, les premières recherches comprenaient la rupture de symétrie de réplique, Monasson & O'Kane (1994), et les réseaux de neurones non monotones, Boffetta, Monasson & Zecchina (1993). Plus récemment, la symétrie et brisure de symétrie ont été étudiées dans les réseaux de neurones euclidiens, Smidt, Geiger & Miller (2021), et les notions de Noether de symétrie continue à partir des formulations énergétiques lagrangiennes, Tanaka & Kunin (2021).

Bien que ceci soit externe à nos travaux, nous souhaitons souligner que la rupture de symétrie a également été étudiée dans les réseaux biologiques en tant que mécanisme de sélection d'orientation au sein du cortex, Ben-Yishai, Bar-Or & Sompolinsky (1995), dynamique des attracteurs angulaire (ring attractor dynamics) dans une expérience projetant des photons individuels dans le cerveau de la drosophile et mesurant la réponse angulaire, Kim, Rouault, Druckmann & Jayaraman (2017). Plus précisément, mesurant les activations dans les couches cérébrales profondes/centrales du cortex de la drosophile melanogaster liée au réseau de boussole de la mouche déterminant le cap, Kim *et al.* (2017); Seelig & Jayaraman (2015), en réponse à des stimuli d'entrée à double photon. Comme les régions centrales du cerveau sont généralement dominées par des réseaux récurrents qui produisent souvent des modèles complexes d'activité neuronale, il est remarquable d'observer une représentation interne abstraite de haut niveau sous la forme d'un anneau topologique représentant la direction de cap en réponse aux modèles de stimulus lumineux d'entrée, comme le montre la Figure 1.2 (b, c).

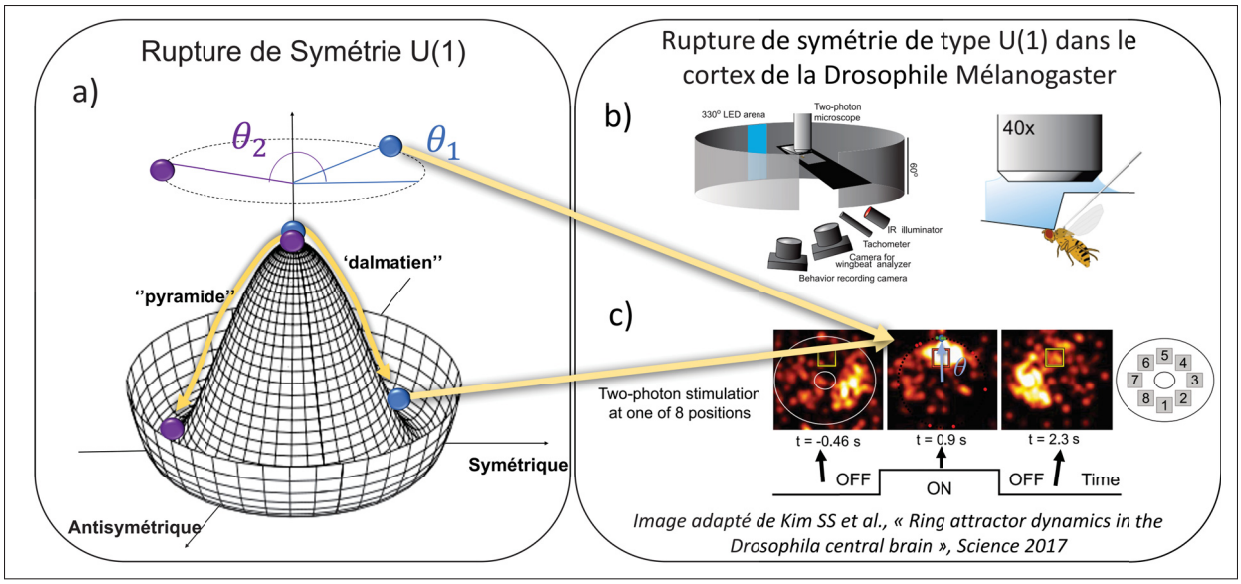

FIGURE 1.2 a) montre la rupture de symétrie $U(1)$ observée dans les activations des CNN, c'est-à-dire les correspondances NN de Densenet-201, Huang *et al.* (2016), couches vectorielles d'activation en réponse à l'entrée d'images de classes de Caltech 101, Fei-Fei *et al.* (2006), où une classe brise la symétrie totale suivant un angle θ précis. Les correspondances entre les images de même classe a) présentent des écarts angulaires constants $\theta \in U(1)$ pour des classes spécifiques (par exemple, la classe "pyramide", "dalmatien" en bleu et violet respectivement). Similairement, b) et c) montrent les réponses observées dans le cortex central des drosophiles suite à des stimulus d'entrée à deux photons générés dans un appareil expérimental spécialisé, Kim *et al.* (2017) (b). La réponse corticale (c) présente une "bosse d'activation" compatible avec une structure topologique en anneau ($U(1)$) en réponse à un stimulus spécifique "ON" (emplacement neuronal [5] dans l'image du milieu). On remarque ainsi qu'une position spatiale réelle est également représentée dans les couches profondes du cortex cérébral des drosophiles, similaire à notre observation d'une représentation angulaire observée pour une classe spécifique dans les couches profondes des CNNs.

Nos observations initiales de rupture de symétrie $U(1)$, voir Figure 1.2 (a), sont basées sur l'indexation et la classification du voisin le plus proche, Cover & Hart (1967), en utilisant spécifiquement les vecteurs d'activations spatialement localisées dans des réseaux préentraînés à la couche juste avant le goulot d'étranglement ($t = T - 1$) où l'information spatiale (x, y) devient inexistante (couche T). Cela suit l'approche d'apprentissage par transfert (transfer learning), où les réseaux préentraînés sur de grands ensembles de données génériques tels que

ImageNet, Deng *et al.* (2009), sont utilisés comme extracteurs d'information générale pour de nouvelles tâches, Kornblith, Shlens & Le (2019); Azizpour, Razavian, Sullivan, Maki & Carlsson (2015); Cimpoi, Maji, Kokkinos & Vedaldi (2016).

Données, Granularité et Classification

Nous avons testé nos hypothèses de rupture de symétrie sur 6 tâches de complexités différentes. Ces tâches varient en granularité, c'est-à-dire en taux de précision sur la tâche à accomplir. Par exemple, lorsque deux images sont très similaires (comme des IRM de cerveaux), la granularité sera très fine nécessitant un réseau détectant de minuscules différences entre deux images et donc "à grain fin". Au contraire, si la tâche est plus générale, tel que de différencier un avion d'un chat, nous aurons des grains plutôt gros où le réseau n'aura qu'à comprendre la différence entre les objets présents au lieu de définir les différences entre un objet de même classe générale. Nous incluons donc des tâches de classification à "gros grains" et à "grains fins" à partir de 5 ensembles de données uniques avec différents niveaux de granularité, comme mentionné dans Berman, Jégou, Vedaldi, Kokkinos & Douze (2019).

Nous considérons également les ensembles de données avec des classes présentes ou absentes d'ImageNet, Deng *et al.* (2009), la formation pour les tâches de classification générales (à gros grains) (par exemple, Caltech 101, Fei-Fei *et al.* (2006)) et la récupération d'instances spécifiques de catégories (grains fins) soit présentes ou absentes d'ImageNet (par exemple Stanford Dogs, Khosla, Jayadevaprakash, Yao & Fei-Fei (2011), cerveaux humains en IRMs, Van Essen *et al.* (2013)).

Les facteurs ajoutant à la complexité des tâches incluent le nombre de classes, la proportion de changement de domaine (mesurée de la même manière que dans Guo *et al.* (2020)) et la spécificité de la classe. Chacun de ces niveaux affecte la capacité de récupération d'image indépendamment à un degré différent. La Figure 1.3 montre les niveaux de granularité **subjectifs et par rapport aux résultats obtenus lors de nos expérimentations** pour les six tâches étudiées dans nos expériences de recherche en mémoire afin de donner une idée sur leurs différences de

complexités. La distance de l'origine $(0, 0, 0)$ peut agir comme un rang de complexité selon les différentes granularités. Le niveau de granularité est important à considérer pour trouver la bonne approche pour des performances maximisées (par exemple, méta-apprentissage, apprentissage par transfert, ajustement (fine-tuning), etc.). Un exemple de classification générale avec un index granulaire plus petit serait de construire un réseau capable de classer un chat par rapport à un chien où vous auriez de nombreux exemples des deux classes (par exemple, la classification du genre dans le cerveau IRMs, Van Essen *et al.* (2013)). Un indice granulaire plus élevé pourrait consister à classer les espèces de chiens, où vous auriez quelques exemples, voire un seul, de chaque espèce (par exemple, Stanford Dogs, Khosla *et al.* (2011)), ce qui créerait le besoin de trouver des informations plus localisées pour identifier un chien spécifique plutôt que d'utiliser des caractéristiques générales qui décrivent mieux les concepts de chiens et de chats.

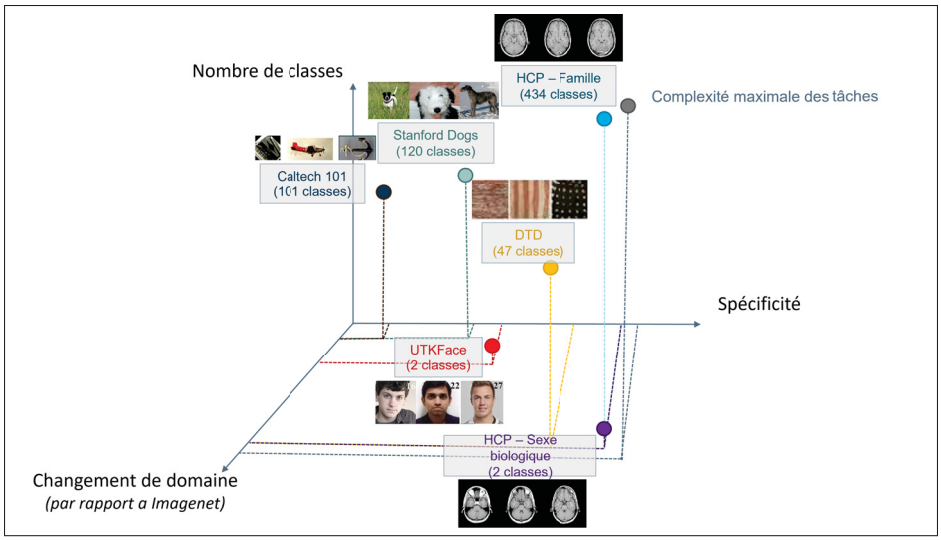

FIGURE 1.3 Estimation **subjective** du niveau de granularité et de la complexité des tâches pour les six tâches étudiées dans les expériences de recherche en mémoire avec la complexité globale de chaque tâche (distance depuis l'origine $(0, 0, 0)$). Ceci est une représentation visuelle (déterminée subjectivement et selon les résultats de nos expérimentations) non représentative de la complexité réelle de ces bases de données afin de supporter l'explication de la granularité. Ces visualisations sont malgré tout basées sur des travaux quantitatifs étudiant les différentes complexités de tâches telles que Scheidegger *et al.* (2021); Martínez-Plumed *et al.* (2022) selon des critères variables tels que le taux de précision, la quantité de données d'entraînements disponibles, la variation des images intraclasses, etc.

Tel que mentionné, une carte d'activation multicanal provenant d'un CNN peut être considérée comme un champ vectoriel de pixels (vecteurs d'intensité) en deux dimensions ayant une intensité propre résultant des filtres de la couche précédente sur lequel de futurs filtres convolutifs ainsi que sous-échantillonnage de regroupement (*down-sampling*) diffusent nécessairement des informations dans l'espace, où le degré de diffusion est déterminé par la largeur du filtre et le nombre de couches de convolutions. Les informations discriminantes pour les tâches de reconnaissance individuelles peuvent généralement consister en de subtiles variations de motifs à une résolution d'image fine, par exemple, des motifs de repliement corticaux observés entre

les membres de la famille dans des images de cerveau humain IRMs, qui ont tendance à devenir de plus en plus diffus avec le nombre et la taille des convolutions appliquées à l'image. Une approche de recherche en mémoire utilisant des réseaux préentraînés (transfer learning) est une piste d'investigation intéressante, parmi d'autres (tel que le meta-learning, le "fine-tuning" complet de réseau, le "fine-tuning" partiel, etc.), pour s'attaquer à de telles tâches. Le point fort du "transfer learning" est qu'il est facile de l'utiliser pour visualiser et comprendre les réseaux profonds, étant également l'approche la plus cohérente et la plus performante à tous les niveaux de granularité en raison de son entraînement sur un ensemble de données volumineux et *complet*, et représentant des informations de toutes les couches du réseau lorsqu'utilisé avec une architecture comportant des sauts de connections ("skip connections", permettant de propager l'information d'une couche t à une couche ultérieure $t + n$) tel que DenseNet, Huang *et al.* (2016).

Notre approche de recherche de vecteurs de pixels est un excellent compromis entre les points clés SIFT, Lowe (2004), donnant une correspondance de caractéristiques invariante puissante pour la classification d'instance (à grains fins), Toews & Wells (2009), et l'approche plus classique utilisant des CNNs préentraînés où les informations sont extraites du goulot d'étranglement, regroupées (pooled) et envoyées dans un module de classification entièrement connecté, puissant pour la classification générale, Krizhevsky, Sutskever & Hinton (2012b), mais sont des classificateurs d'instance sous-optimaux, Kaya, Hong & Dumitras (2019). De la même manière, cette configuration CNN-regroupement-classificateur peut être trompée en ne changeant qu'un pixel (adversarial attacks) à cause de ce manque d'informations de localisation, Su, Vargas & Sakurai (2019), ce qui la rend moins stable à étudier. Les informations suite au goulot d'étranglement des réseaux profonds ne contiennent pas d'informations de localisation comme les informations structurées avant le goulot d'étranglement, ce que nous priorisons avec l'approche de correspondance décrite dans la prochaine section.

Adaptation de Domaine, Apprentissage en Peu d'Étapes, Pré-Entraînement et Méthodes d'Encodage d'Informations

Il y a eu des travaux approfondis pour résoudre des tâches d'apprentissage complexes de classification en peu d'étapes (few-shot learning), y compris l'utilisation meta-apprentissage, Ravi & Larochelle (2016); Finn, Abbeel & Levine (2017); Sung *et al.* (2018); Lee, Maji, Ravichandran & Soatto (2019); Tseng, Lee, Huang & Yang (2020), où le but est d'entraîner un modèle sur une variété de tâches d'apprentissage, de sorte qu'il puisse résoudre de nouvelles tâches d'apprentissage en utilisant seulement un petit nombre d'échantillons d'apprentissage, l'utilisation de fonctionnalités d'activation, Donahue *et al.* (2014); Sharif Razavian, Azizpour, Sullivan & Carlsson (2014); Long, Cao, Wang & Jordan (2015); Nguyen *et al.* (2020); Snell, Swersky & Zemel (2017), l'entraînement à partir de zéro de réseaux profonds adaptés (par exemple, les réseaux siamois, Koch (2015), un type d'architecture dans lequel vous avez deux sous-réseaux identiques alimentés par des entrées par paires pour apprendre à reconnaître la similarité de vos données étiquetées, ou des réseaux de modules d'attention aux goulots d'étranglement (BAM, Park, Woo, Lee & Kweon (2018)) qui utilisent le canal et l'attention spatiale pour maximiser la qualité des informations sur les goulots d'étranglement dans les CNNs,), effectuant un apprentissage par transfert où nous adaptons un réseau préentraîné à une nouvelle tâche avec moins de données disponibles, Fei-Fei *et al.* (2006); Pan & Yang (2010); Yosinski, Clune, Bengio & Lipson (2014); Kornblith *et al.* (2019); Guo *et al.* (2018); Dhillon, Chaudhari, Ravichandran & Soatto (2019), ou en combinant un réseau profond avec une mémoire, Vinyals, Blundell, Lillicrap, Wierstra *et al.* (2016); Papernot & McDaniel (2018); Wang, Chao, Weinberger & van der Maaten (2019).

Les activations du goulot d'étranglement des CNNs préentraînés ont tendance à surpasser les réseaux spécialisés moins profonds et les méthodes de méta-apprentissage, Chen, Liu, Kira, Wang & Huang (2019), en particulier dans le cas de peu de données d'entraînement et d'un grand changement de domaine entre les données d'entraînement et de test, Guo *et al.* (2020). Diverses approches cherchent à adapter les modèles ImageNet, Deng *et al.* (2009), aux tâches fines en encodant les activations au niveau des couches de goulot d'étranglement, par ex. via des descripteurs (par exemple VLAD), Arandjelović, Gronat, Torii, Pajdla & Sivic (2016),

moyenne globale ou regroupement (pooling) maximal, Razavian, Sullivan, Carlsson & Maki (2016), moyenne généralisée (GeM), Radenović, Tolias & Chum (2018), regroupement maximal régional (R-MAC), Tolias, Sicre & Jégou (2016), ou encore à l'aide de couches intermédiaires modulées par des opérateurs d'attention, Noh, Araujo, Sim, Weyand & Han (2017). Un entraînement supplémentaire peut prendre en compte la fonction de coût conjointe entre les termes de classification et de récupération d'instance, Berman *et al.* (2019). Le mécanisme des activations spatialement localisées (par opposition aux descripteurs globaux) est étroitement lié aux mécanismes d'attention, Huang *et al.* (2019), y compris les réseaux non locaux, Wang *et al.* (2018b), les réseaux de compression et d'excitation, Hu, Shen & Sun (2018), architectures "Transformeurs", Vaswani *et al.* (2017b); Carion *et al.* (2020); Han *et al.* (2020), y compris les fenêtres hiérarchiquement décalées (SWIN transformers), Liu *et al.* (2021), couches minces de goulot d'étranglement, Sandler, Howard, Zhu, Zhmoginov & Chen (2018), mécanismes d'autoattention tenant compte des emplacements et des canaux, Woo, Park, Lee & Kweon (2018), corrélations intranoyaux, Haase & Amthor (2020), perceptrons multicouches incorporant l'angle d'Euler, Tang *et al.* (2021), transformeurs basés sur la correspondance, Jiang, Trulls, Hosang, Tagliasacchi & Yi (2021), et détecteurs, Sun, Shen, Wang, Bao & Zhou (2021), incorporation géométrique d'informations spatiales via des graphes, Kipf & Welling (2016); Henaff, Bruna & LeCun (2015). Alors que ces travaux recherchent généralement des solutions d'apprentissage de bout en bout adaptées aux contraintes de mémoire des GPUs, Gordo, Almazán, Revaud & Larlus (2016); Wang, Sun, Eriksson, Wang & Aggarwal (2018a), nous cherchons à démontrer notre théorie $U(1)$ via une recherche de mémoire de base en utilisant des réseaux préentraînés, ne se souciant pas de l'efficacité, la mémoire ou le calcul dans un premier lieu, mais plutôt utilisant cette méthode comme un simple outil de visualisation.

De nombreuses approches explorent le schéma d'encodage directement au lieu de sa structure, en essayant d'améliorer l'encodage "One-hot", Rodríguez, Bautista, Gonzàlez & Escalera (2018); Jaiswal, Kang, Lee & Cho (2020); Kim, Bargal, Zhang & Sclaroff (2020); Liang, Bai, Hu & Lv (2021). La plupart des travaux de visualisation se concentrent sur les filtres, Zeiler & Fergus (2014). En revanche, nos expériences avec les réseaux préentraînés ont montré une structure

d'information $U(1)$ dans le plan spatial 2D (x, y) du tenseur d'information propagé $I_{t,\bar{x}}$, qui, à notre connaissance, n'a jamais été observée ou utilisée pour améliorer la qualité des informations sur les goulots d'étranglement auparavant.

Les informations spatiales ont été étudiées sous la forme d'une régularisation géométrique basée sur des hypersphères, Mettes, van der Pol & Snoek (2019), imposant une distance radiale constante à partir de l'origine de l'espace des caractéristiques, Zheng, Pal & Savvides (2018), représentant les jetons d'entrée sous forme d'onde utilisant la magnitude et la phase, Tang *et al.* (2022), en appliquant une fonction de coût angulaire entre les prototypes, Wang, Zhou, Wen, Liu & Lin (2017), en maximisant la variance de chaque variable d'intégration et en minimisant la covariance entre les paires de variables d'intégration pendant l'entraînement, Bardes, Ponce & LeCun (2021), ou plus similaire à notre approche : utiliser des matrices unitaires dans les couches, Arjovsky, Shah & Bengio (2016); Jing *et al.* (2017); Mhammedi, Hellicar, Rahman & Bailey (2017); Sedghi, Gupta & Long (2018); Singla & Feizi (2021); Trockman & Kolter (2021); Kiani *et al.* (2022). Toutes ces approches ont pour objectifs communs (1) d'empêcher "un effondrement dans lequel les encodeurs produisent des vecteurs constants ou non informatifs" Bardes *et al.* (2021) et (2) de résoudre le problème de disparition et d'explosion de gradients (vanishing and exploding gradients) lors de la rétropropagation, mais n'étudient pas l'espace géométrique dans l'information propagée par les DNNs directement. Les approches ont également étudié la symétrie et l'antisymétrie dans les réseaux profonds, Yeh, Hasegawa-Johnson & Do (2016); Chang, Chen, Haber & Chi (2019); Dzhezyan & Cecotti (2019); Hutter (2020); Tian, Xu, Zuo, Lin & Zhang (2021), mais se concentrent principalement sur les filtres des réseaux plutôt que l'information propagée elle-même. Des travaux récents ont montré que l'exploitation des symétries et des antisymétries permet de réduire considérablement la taille de l'ensemble de données d'entraînement en gardant la même précision, Klus, Gelß, Nüske & Noé (2021). Ici, nous nous concentrons sur les composantes symétriques et antisymétriques des informations propagées naturellement produites par les réseaux profonds.

Nos expériences, principalement faites à titre d'observations, avec des réseaux préentraînés utilisant des vecteurs de pixels extraits d'informations au goulot d'étranglement dans une

mémoire ont donné des indications claires sur l'amélioration des performances des CNNs en optimisant le plan spatial 2D (x, y), mais nécessitent une mémoire, un stockage et des calculs coûteux. Nous suggérons ainsi l'entraînement à partir de zéro de notre architecture la plus optimale $U(1)$ utilisant une seule variable θ extrait de la couche du goulot d'étranglement.

Notre approche $U(1)$ décrite dans la section suivante est plus similaires au travail visant à régulariser l'annotation et/ou l'espace des fonctionnalités d'activation, y compris l'utilisation d'annotations d'entraînement à valeur réelle plutôt que sous la forme de "one-hot", Rodríguez *et al.* (2018), classificateurs basés sur l'apprentissage, Wang *et al.* (2019); Wen, Zhang, Li & Qiao (2016), réseaux prototypiques pour l'apprentissage en quelques coups (few-shot learning), Nguyen *et al.* (2020); Snell *et al.* (2017), K plus proches voisins profonds, Papernot & McDaniel (2018), basé sur la régularisation géométrique sur les hypersphères, Mettes *et al.* (2019), imposant une distance radiale constante à partir de l'origine de l'espace caractéristique, Zheng *et al.* (2018), ou une perte angulaire entre les prototypes, Wang *et al.* (2017). Nous cherchons à présenter notre théorie dans le contexte le plus large et le plus général. **Nous considérons une classification de base et évitons délibérément les modifications architecturales qui pourraient limiter la généralité de notre analyse**. Nous démontrons notre théorie générale avec des résultats en utilisant une grande variété **d'architectures**, y compris DenseNet Huang *et al.* (2016), Inception Szegedy *et al.* (2016), ResNet He *et al.* (2016), VGG Simonyan & Zisserman (2015), directement importées de TensorFlow, Abadi *et al.* (2015). Pour la même raison, nous considérons une grande variété **d'ensembles de données** de test non utilisées dans l'entraînement ImageNet Deng *et al.* (2009), avec des catégories générales (par exemple, Caltech 101 Fei-Fei *et al.* (2006)) et des instances spécifiques (par exemple, images IRMs du cerveau humain de membres de famille Van Essen *et al.* (2013), ou encore des visages Zhang, Song & Qi (2017)).

CHAPITRE 2

PROPAGATION DE L'INFORMATION ET RUPTURE DE SYMÉTRIE U(1)

Nous proposons de modéliser la propagation de l'information via un système générique de traitement de l'information visuelle analogue à l'appareil de capture optique et le traitement discret par couche dans un réseau de neurones convolutif profond, comme le montre la Figure 0.2. Nous posons une analogie entre les deux systèmes afin de développer les notions de spectre d'information, de cercle unitaire de confusion et de classification générique en tant que processus de rupture de symétrie $U(1)$ où une simple annotation d'angle peut être ajoutée à un réseau CNN de base pour améliorer la précision d'un modèle CNN. Nous brisons ainsi une symétrie circulaire visualisée dans les couches profondes des réseaux CNNs en orientant chaque classe vers un angle précis, tel que nous allons couvrir dans cette section.

2.1 Analogie de la Propagation de la Lumière et de l'Information

Le flux d'informations à travers l'espace-temps dans un système de vision par ordinateur peut être considéré comme la propagation d'information (par exemple la lumière) à travers l'espace-temps. Dans un système optique comme sur la Figure 0.2 a), les informations sont transportées par des photons rayonnant d'une scène 3D, où ces photons se déplacent le long des rayons dans l'espace 3D à une vitesse constante dans le vide. Des rayons lumineux concentrés à un foyer via une lentille ou un trou d'épingle peuvent être accumulés en une image 2D cohérente dans l'espace au niveau d'un réseau de photocapteurs RVB. Dans un système de réseau de neurones à propagation avant comme sur la Figure 0.2 b), représenté sous forme d'auto-encodeur à base de convolutions, les informations sont transportées par des vecteurs d'activation et propagent des opérations par couches, y compris des produits scalaires linéaires, des fonctions d'activation non linéaires telles que Rectified Linear Unit Layer (ReLU), Nair & Hinton (2010), regroupement (pooling) et sous-échantillonnage. Les informations concentrées sur un foyer peuvent être utilisées pour la prédiction de classe (tâche de classification, information utilisé au "bottleneck", comparable au foyer des lentilles) ou propagées davantage pour obtenir une segmentation d'image 2D

cohérente dans l'espace (tâche de segmentation, comparable au plan focal pour le système de lentilles), Long, Shelhamer & Darrell (2014).

Plusieurs aspects communs aux deux systèmes peuvent être relevés. L'information est répartie sur un plan spatial 2D (x, y), c'est-à-dire un front d'onde de lumière se propageant ou une couche profonde du réseau, et sur un spectre $I(t, x, y)$, c'est-à-dire le spectre RVB trichromatique de la lumière visible ($t = 0$) ou les canaux d'activation d'une couche du réseau ($t > 0$). Nous définissons le rayon principal comme étant perpendiculaire au plan image (x, y) et *central à la fois à l'espace image 2D et à "l'espace canal" (channel space) d'information*. Dans les deux systèmes, les informations sont acheminées vers un point focal le long du rayon principal, où la mise au point la plus fine possible est limitée à un cercle de confusion, représenté par un anneau noir sur la Figure 0.2. Le cercle de confusion est un phénomène optique bien connu résultant de la diffraction ou des imperfections de la lentille, et aggravé par la dispersion, la réfraction spécifique au canal et l'aberration chromatique. Nous émettons l'hypothèse d'un phénomène similaire dans les réseaux de neurones profonds, en raison de la fonction d'étalement des points (point spread function) des opérations de filtrage, ce qui motive notre enquête initiale sur le désenchevêtrement (x, y) des informations du goulot d'étranglement.

Dans un système optique, le spectre électromagnétique (EM) encode les informations sous forme de particules de photons, qui peuvent être modélisées comme se propageant dans l'espace 3D via les opérateurs delta de Dirac $\delta(d\bar{x})$ (c'est-à-dire les filtres de traduction) définis en 3D par la géométrie de l'appareil optique. Les photons sont des particules dites bosons définies par une fonction d'onde (ψ) symétrique $\psi(-x) = \psi(x)$, et peuvent s'accumuler à un foyer spatial selon les statistiques de Bose-Einstein, Bose (1924) [1]. Dans un réseau de neurones profond, les informations sont empêtrées dans un nombre limité de canaux d'activation $I(t, x, y) \in R^{N+}$, et propagées via des opérations de filtrage multicanaux. Une différence importante entre les deux systèmes est que dans le cas des CNNs profonds, LeCun *et al.* (1989), le spectre ou jauge du canal, Cohen, Weiler, Kicanaoglu & Welling (2019), est transformé à chaque couche afin

[1] Par opposition aux particules dites fermions telles que les électrons définis par une fonction d'onde antisymétrique $\psi(-x) = -\psi(x)$ qui ne peuvent s'accumuler que dans des paires (spin-up, spin-down) en raison du principe d'exclusion de Pauli.

d'encoder et compresser des quantités croissantes d'informations visuelles à des résolutions spatiales réduites jusqu'au goulot d'étranglement du réseau, tandis que la jauge EM est préservée pendant la propagation de la lumière.

La propagation des informations dans les deux systèmes peut être modélisée comme un champ vectoriel se propageant à travers l'espace-temps (t, x, y) via des opérateurs de produits scalaires locaux comme suit. Soit $I_{t,\bar{x}}$ un vecteur d'informations multicanal à la couche t et à l'emplacement spatial $\bar{x} = (x, y)$, par exemple une image RVB à 3 canaux ($t = 0$) ou une couche d'activation de N canaux ($t > 0$). Les informations peuvent être propagées du temps t à $t + 1$ par l'opération de produit scalaire $I_{t+1,\bar{x}} = \bar{I}_{t,\bar{x}} \cdot \bar{W}_{t,\bar{x}}$, où $\bar{I}_{t,\bar{x}}$ est un tenseur de voisinage défini dans la fenêtre spatiale entourant $(t, \bar{x})$, et $\bar{W}_{t,\bar{x}}$ sont les poids des filtres linéaires de mêmes tailles et formes que $\bar{I}_{t,\bar{x}}$. Notez que dans un réseau de neurones convolutif (CNN), LeCun *et al.* (1989), les filtres sont invariants à la translation et constants sur une couche $W_{t,\bar{x}} = W_t$, mais ils peuvent généralement varier en fonction de l'emplacement spatial, comme dans les perceptrons (unique, Rosenblatt (1958), et multicouches, Rumelhart, Hinton & Williams (1986)), ou les réseaux de transformeurs, Vaswani *et al.* (2017b), (d'où les transformeurs "éliminant la convolution"). Les fonctions d'activations introduisant la non-linéarité et les couches de sous-échantillonnages sont également essentielles pour le fonctionnement d'une telle architecture bien qu'ils aient été omis dans cette explication, comme le montre l'équation 1.1. Nous retrouvons également d'autres mécanismes aidant ces réseaux chez certaines architectures CNNs telle que des sauts de connections (skip/residual connections) avec ResNet He *et al.* (2016) permettant de propager l'information d'une couche t à une couche ultérieure $t + n$, ce qui n'est pas possible avec la lumière, mais ne pose aucune différence majeure sur l'information au bottleneck où nous faisons notre étude.

Notre modèle est illustré sur la Figure 2.1, où l'information d'activation multicanal est non négative (c'est-à-dire après activation (ReLU), Nair & Hinton (2010)), située dans l'orthant positif de l'espace d'activation $I_{t,\bar{x}} \in R^{N+}$, et peut être exprimé en coordonnées polaires (t, r, θ) centrées autour de l'origine spatiale $(x, y) = (0, 0)$ et longe le rayon principal $(t, 0, 0)$. Par exemple, les pixels de couleur RVB peuvent être représentés par des coordonnées de teinte (hue), saturation, intensité (value) (HSV), Figure 2.1 (a), où la perception des couleurs est

étroitement liée à un angle de teinte $\theta \in U(1)$. Nous proposons une analogie dans laquelle les activations CNN multi-canaux sont également représentées via un *angle de classe* $\theta \in U(1)$ selon les canaux d'intensité dominants. Nous émettons l'hypothèse qu'il s'agit d'un mécanisme principal par lequel les informations de classe sont encodées dans l'espace image, et qu'elles peuvent être identifiées et utilisées comme signal d'apprentissage dans le plan image (x, y) de couches CNN arbitraires, notamment les couches de goulot d'étranglement avec une étendue spatiale minimale. Nous observons que l'énergie scalaire $E_{t,\bar{x}} = \|I_{t,\bar{x}}\|^2 \in R^+$ est hautement symétrique, soit les distributions les plus favorables à la classification des couches d'activation d'une variété d'architectures génériques préentraînées sur le jeu de données ImageNet, Deng *et al.* (2009), en réponse aux images d'entrée non utilisées dans l'entraînement, mais comportant des classes similaires et donc un domaine similaire (par exemple, Caltech 101, Fei-Fei *et al.* (2006)). Comme le montre la Figure 2.1, l'énergie d'activation moyenne des couches de goulot d'étranglement préentraîné est généralement distribuée symétriquement autour de l'origine dans une forme de "sombrero" rappelant une fonction d'onde bosonique, causée par une antisymétrie propre à chaque classe où la somme des classes cause cette symétrie globale.

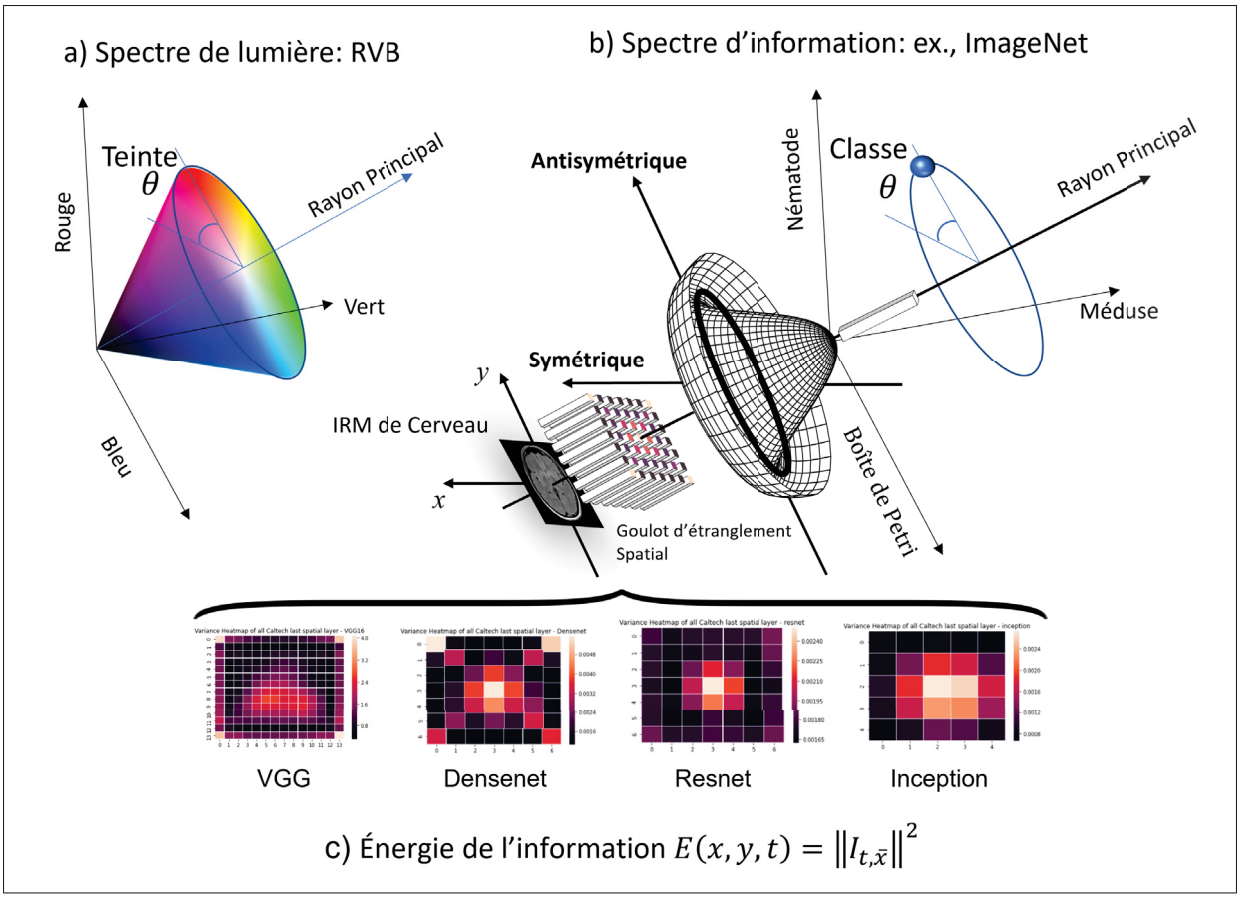

FIGURE 2.1 Illustration des systèmes de coordonnées polaires pour les données multicanaux. a) L'espace colorimétrique rouge-vert-bleu (RVB) standard à 3 canaux avec l'angle de teinte (hue) θ. b) Un sous-espace à 3 canaux de la sortie ImageNet, Deng *et al.* (2009), à 1000 canaux, suivant la couche de goulot d'étranglement spatial, montrant un angle de classe θ analogue à la teinte (hue) de couleur et défini par la classe de l'image d'entrée. (c) cartes d'énergie $E_{t,\bar{x}} = \|I_{t,\bar{x}}\|^2$ calculées à partir des couches spatiales de goulot d'étranglement de divers réseaux pré-entraînés sur ImageNet et testés sur les classes de Caltech 101, Fei-Fei *et al.* (2006), non-utilisé lors de l'entraînement. Notez la structure hautement symétrique et la similitude avec le potentiel énergétique bosonique "sombrero" bien connu, Goldstone (1961). En b), la classe "cerveau, IRM" (non incluse dans ImageNet) correspond aux classes ImageNet les plus similaires, soit : Nématode, Boîte de Petri et Méduse. Image adaptée de Bouchard *et al.* (2022).

Effectivement, cette symétrie est due au fait que l'énergie est observée sur la distribution d'une base de données totale, regroupant toutes les classes, qui est causée par une asymétrie propre à chaque classe (observée dans la section 3.1.1) où une orientation peut être apprise à partir

d'annotations de classe $U(1)$ arbitraires afin d'améliorer la classification tel que décrite dans la section 3.3.

CHAPITRE 3

OBSERVATIONS ET EXPÉRIMENTATIONS AVEC LA RUPTURE DE SYMÉTRIE

3.1 Protocole Expérimental de nos Observations

Nous commençons nos études par l'observation du comportement de l'information aux couches plus profondes des réseaux CNNs préentraînés, tels qu'étudiés par Lazreg (2020) et décrites dans l'Appendice I. Plus précisémment, nous portons nos observations selon le même procédé expérimental de Lazreg, décrit dans l'Appendice I, avec une variable importante : le désenchevêtrement des pixels. Plus précisémment, nous portons la classification via l'indexation basique du plus proche voisin des activations de réseau préentraînés (sur Imagenet, Deng *et al.* (2009), suivant le protocole décrit par TensorFlow Abadi *et al.* (2015) selon le modèle utilisé) à une approche via un processus de rupture de symétrie dans le plan de l'image où nous permettons aux pixels de se déplacer dans "l'image".

La Figure 3.2 illustre notre architecture utilisée pour nos observations, où des vecteurs individuels $I_{t,\bar{x}}$ peuvent se déplacer dans l'espace image afin de correspondre à des vecteurs similaires en mémoire extrait de classes similaires, plutôt que de correspondre directement à une seule et unique image, comme chez Lazreg et nos tests préliminaires décrits en Appendice I. Nous visualisons également $I_{t,\bar{x}}$ à l'aide de la Figure 3.1 montrant l'extraction d'un vecteur individuel surligné en jaune pour le réseau DenseNet, Huang *et al.* (2016).Comme indiqué, le mécanisme général est étroitement lié aux approches d'attention spatiale, et notre mise en œuvre est conçue pour maximiser les performances de classification de base et observer l'effet du positionnement spatial 2D (x, y) de l'information dans les couches profondes des CNN. Nous détaillons l'algorithme utilisé dans l'algorithme 3.1 pour construire la base de données et l'algorithme 3.2 pour la classification, et mathématiquement ci-bas.

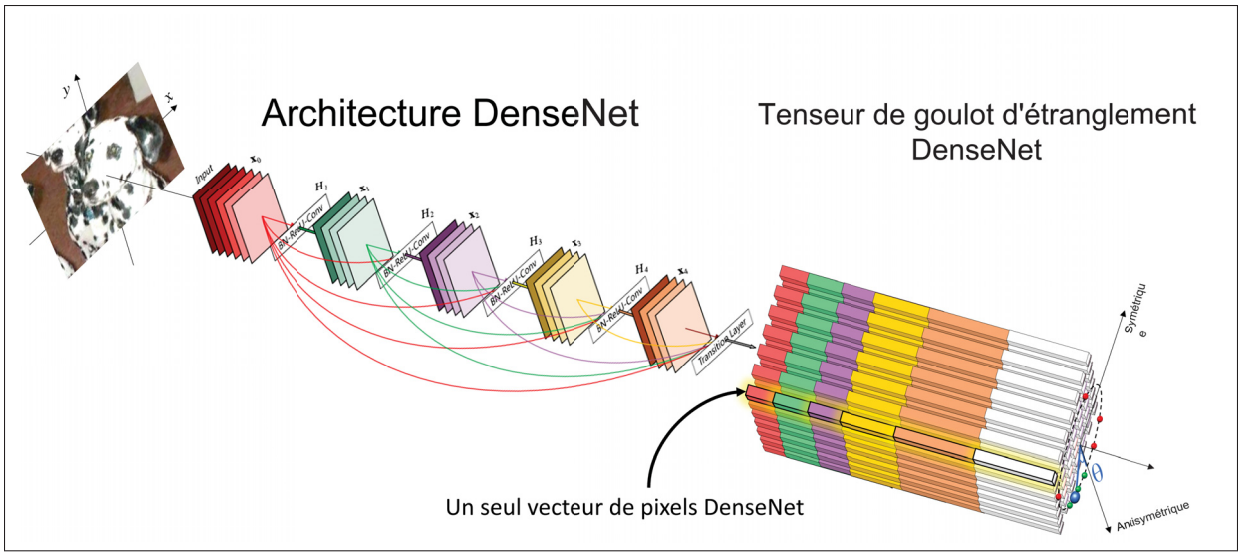

FIGURE 3.1 DenseNet, Huang *et al.* (2016), architecture et visualisation de l'extraction de l'information des tenseurs (nos tenseurs $7 \times 7 \times 1920$ de la couche $T-1$, comportant les vecteurs pixels de 1×1920 surligné en jaune). Image adaptée de Huang *et al.* (2016). Nous couvrons ce processus en détail dans l'algorithme 3.1.

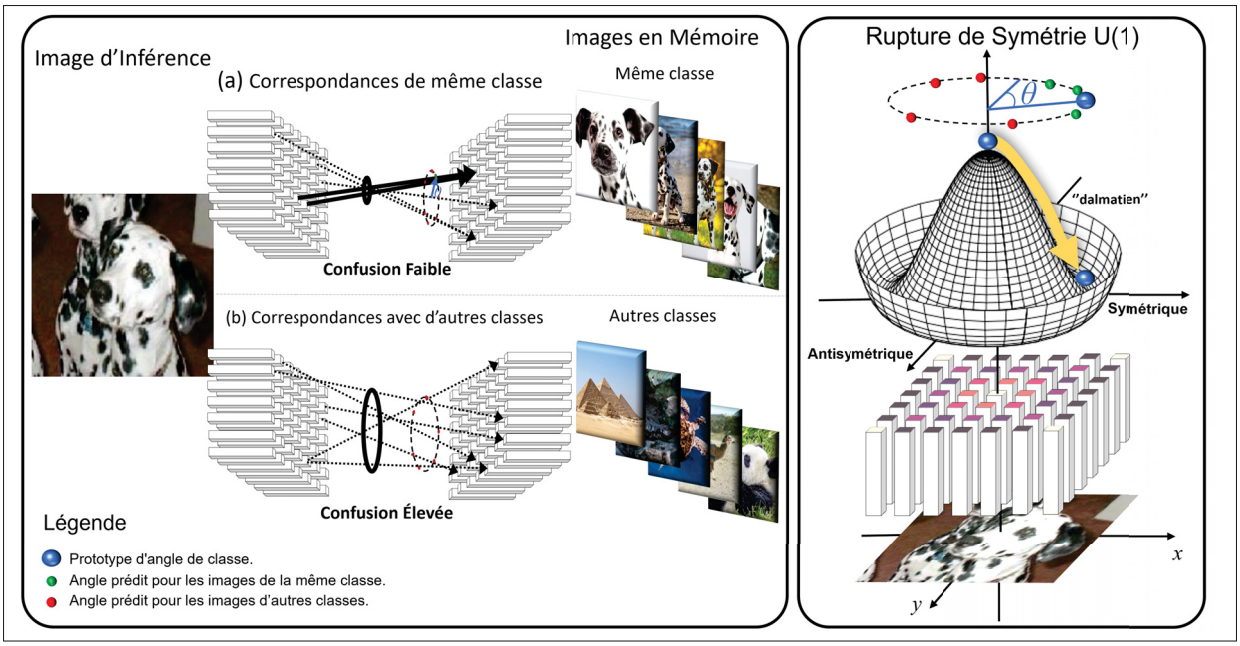

FIGURE 3.2 (À gauche) Notre architecture de vecteurs de pixels proposée où la classification est obtenue à partir des correspondances des plus proches voisins entre les vecteurs d'activation dérivés d'une image d'entrée et des images étiquetées stockées en mémoire (voir algorithme 3.2 et 3.1. En lien avec notre cercle de confusion, les correspondances entre des images de la même classe (a) présentent généralement une confusion spatiale plus faible que les correspondances avec des images de classe non liées (b), comme démontrent nos observations à venir. (À droite) Notre hypothèse selon laquelle la correspondance image-mémoire conduit à la rupture de symétrie de la fonction d'énergie de couche sur le plan (x, y) et influence la reconnaissance de classe suivant un angle θ. Image adaptée de Bouchard *et al.* (2022).

Algorithme 3.1 Construction de nos bases de données de pixels I et I' pour la correspondance de pixels

/* Construction de nos bases de données de pixels I et I' */

Entré : BD, une base de données d'images;

Sortie : N vecteurs de pixels par image séparés en base de test et mémoire;

Importer le modèle préentraîné au choix (par exemple, DenseNet-201);

for *image* $im \in BD$ **do**

  Normaliser l'image suivant la procédure des modèles Tensorflow Abadi *et al.* (2015);

  Traiter l'image im avec le CNN;

  Extraire le tenseur directement avant le *bottleneck* (couche $T-1$);

  Associer chaque vecteur de pixel avec son annotation de classe;

  Sauvegarder les vecteurs de pixels en mémoire avec son annotation et position (x, y) dans le tenseur;

end for

Séparer la nouvelle base de données de pixel en données entraînement (mémoire) et tests suivant la procédure désirée;

/* Dans le cas de Caltech, sauvegarder les pixels de 30 images aléatoires par classes pour la mémoire (I') et le reste pour les tests (I). Où I et I' contiennent les N pixels de chaque image de la base de données. */

/* Une image $im \in I$ contient tous les vecteurs de pixels du tenseur, par exemple $7x7 = 49$ pixels pour DenseNet. Donc les nouvelles bases $|I| + |I'| = 49 * |BD|$ pixels. */

Algorithme 3.2 Classification - K plus proches voisins avec la correspondance de pixels

```
/* Classification des plus proches voisins pour la base
   de données de test                                    */
```

Entré : I et I', une base de données d'entraînement et de tests contenant nos vecteurs de pixel;

Sortie : Une classification pour la base test I et l'information de nos plus proche voisins dans I';

for *Image* $im \in I$ **do**

  for *Pixel* $p \in im$ **do**

    Calculer la distance entre p et $p' \in I'$;

    ```
    /* où p' sont tous les pixels de toutes les images
       de notre base de données en mémoire.            */
    ```

    Trier les pixels p de l'image im en ordre croissant de distance pour tous les pixels p' dans I';

    Sélectionner les k-plus proches vecteurs de pixels de références en mémoire;

    Identifier la classe et position des k-plus proches vecteurs de pixels de références;

    ```
    /* Nous obtenons ainsi N × k annotations pour les
       plus proches voisins.                           */
    /* Les positions (T − 1, x, y) des pixels (plus proche
       voisins) en mémoire seront utilisés pour nos
       observations.                                   */
    ```

  end for

  Compter l'occurence de chaque classe dans nos $N \times k$ annotations pour des plus proches voisins;

  Trier l'occurence des classes en ordre descendant;

  Sélectionner la classe avec le nombre d'occurence la plus élevée pour le choix final de la classe;

end for

Comme la distance entre deux vecteurs de pixel peut être considérée comme le produit scalaire, minimiser la distance équivaut à identifier les maxima locaux du produit scalaire. La classification est obtenue en maximisant la fonction de vraisemblance $p(I|C, \{I'\})$ de la classe C associée à l'image d'entrée I à partir d'un ensemble d'exemples d'images $\{I'\}$ stocké en mémoire :

$$C^* = \underset{C}{\operatorname{argmax}}\ p(I|C, \{I'\}), \tag{3.1}$$

où C^* est une estimation du maximum de vraisemblance (ML) de la classe d'images. La fonction de vraisemblance est définie comme suit : soient $x_i, x_j \in R^2$ des emplacements de points spatiaux discrets, $x_i \in \Omega_I$ dans l'image d'inférence et $x_j \in \Omega_{I'}$ dans les images mémoire. Les points $x_j \in NN_i$ appartiennent à un ensemble de K plus proches voisins de x_i $NN_i : \{x_j : \|I_{xi} - I'_{xj}\| \leq \|I_{xi} - I'_{xk}\|\}$ identifié via l'indexation de la mémoire du vecteur d'activation, où I'_{xk} est le k^{th} plus proche voisin de I_{xi} en mémoire. La vraisemblance peut être exprimée comme une densité de noyau

$$p(I|C, \{I'\}) \propto \frac{\sum_{x_i} \sum_{x_j \in N_i} f(I_{x_i} - I'_{x_j})[C = C'_j]}{\sum_{x_j} [C = C'_j]}, \tag{3.2}$$

où $[C = C'_j]$ est le crochet d'Iverson évaluant à 1 en cas d'égalité et 0 sinon, et le dénominateur normalise la fréquence de classe sur l'ensemble de la mémoire $\sum_{x_j} [C = C'_j]$. La fonction noyau dans l'équation (3.2) est basée sur la (dis)similarité du vecteur d'activation $f(I_{x_i}, I'_{x_j})$ et est définie comme :

$$f(I_{x_i}, I'_{x_j}) = exp - \left\{ \frac{\|I_{x_i} - I'_{x_j}\|^2}{\alpha_i^2 + \epsilon} \right\}, \tag{3.3}$$

où dans l'équation (3.3), $\alpha_i = \min_{x_j \in \Omega_{I'}} \|I_{x_i} - I'_{x_j}\|$ est un paramètre de bande passante du noyau adaptatif défini comme la distance au vecteur d'activation le plus proche I_{x_i} en mémoire $I'_{x_j} \in \{I'\}$, et ϵ est une petite constante positive assurant un dénominateur non nul. Notez que les distances euclidiennes et cosinus sont équivalents $\operatorname{argmin} \|I_{x_i} - I'_{x_j}\| = \operatorname{argmin}\{1 - I_{x_i} \cdot I'_{x_j}\}$ pour les descripteurs normalisés en magnitude $\|I_{x_i}\| = \|I'_{x_j}\| = 1$, et peut être utilisé de manière interchangeable selon l'architecture de calcul à portée de main.

Tel que décrit dans l'Appendice I, les observations de la section suivante sont basées sur une variété d'architectures CNN génériques entraînées sur l'ensemble de données ImageNet, Deng *et al.* (2009), et testées dans des expériences de classification basées sur une mémoire de base utilisant divers ensembles de données spécifiquement non utilisés dans l'entraînement, y compris des objets généraux (Caltech 101, Fei-Fei *et al.* (2006) (101 classes)), des textures (DTD) (47 classes), Cimpoi *et al.* (2014), et des objets spécifiques, Van Essen *et al.* (2013); Zhang *et al.* (2017), tous séparés en base d'entraînement et de test selon les indications des auteurs.

Comme **notre objectif est de porter des observations au sujet de l'information dans les couches profondes des CNNs via les informations spatialement localisées les plus précises disponibles**, l'efficacité n'est pas une préoccupation immédiate et dépasse le cadre de notre travail ici bien que nous discutons certaines possibilitées d'optimisations dans l'Appendice II. Comme décrite dans l'algorithme 3.2, notre méthode nécessite l'utilisation de (dans le cas de DenseNet) 49 vecteurs de dimensions $1x1920$ pour chaque image en mémoire et chaque image à l'inférence où un descripteur global n'aurait besoin que d'un seul vecteur de dimensions $1x1920$. Nous avons donc un temps de recherche 49 fois plus élevé lors de la comparaison avec chaque vecteur en mémoire et devons effectuer cette recherche 49 fois plutôt qu'une pour chacun des vecteurs de pixel de notre image d'inférence. La complexité y est donc 49×49 ou N^2, où N est la quantité de pixels pour notre tenseur extrait de $N \times N \times 1920$. Nous avons tout de même étudié l'effet de la compression spatiale autant en utilisant le domaine spectrale que la compression des canaux avec l'Analyse des composants principaux (Principal Component Analysis) (PCA), F.R.S. (1901), ou encore l'utilisation d'architectures spécialisées, Sun *et al.* (2021); Jiang *et al.* (2021) tel que décrit dans l'Appendice II afin d'améliorer la précision générale sur nos différentes tâches et réduire le temps de calcul et la complexité totale, étant également des pistes de recherches futures intéressantes. Nous voulions également observer quel effet aurait ces compressions spatiales sur la correspondance de pixels, voir Appendice II pour plus d'information sur nos expérimentations de compressions.

Bien que ces optimisations ne sont pas nécessaires pour les observations portées, nous obtenons une classification efficace en utilisant la bibliothèque de méthode d'indexation "Approximate

Nearest Neighbor" (Annoy, version 1.0.3), Bernhardsson (2015), un algorithme rapide basé sur la séparation de l'espace de recherche en une "forêt" composée de D arbres (D=50 dans notre cas, utilisé par défaut dans la librairie et dans l'étude de Lazreg (2020)), réduisant la **complexité de requête** pour un élément à $O(log\ N)$ pour N éléments en mémoire, complexité qui est également influencée par la taille $|T|$ du vecteur stocké initialement où $|T| = 1x1920$ dans le cas de nos observations avec DenseNet.

3.1.1 Observation de la Rupture de Symétrie dans les Couches de Goulot d'Étranglement CNN

Des exemples de rupture de symétrie cohérente $U(1)$ observées utilisant le protocole décrit dans la section précédente sont illustrées dans la Figure 3.3 (a, b), où les distributions des correspondances de vecteurs de pixels voisins les plus proches présentent un biais angulaire cohérent pour les images d'une même classe (a). Les visualisations utilisent l'architecture $7 \times 7 = 49$ pixels extraits de DenseNet-201, Huang *et al.* (2016), avec des vecteurs de pixels $I_{t,\vec{x}} \in R^{1920}$ extraits suivant l'algorithme 3.1 et trouvant leurs plus proches voisins (correspondances) suivant l'algorithme 3.2 et affiché dans la Figure 3.3 (c) pour une image (7×7 vecteurs de pixels) d'inférence et des vecteurs de pixels en mémoire.

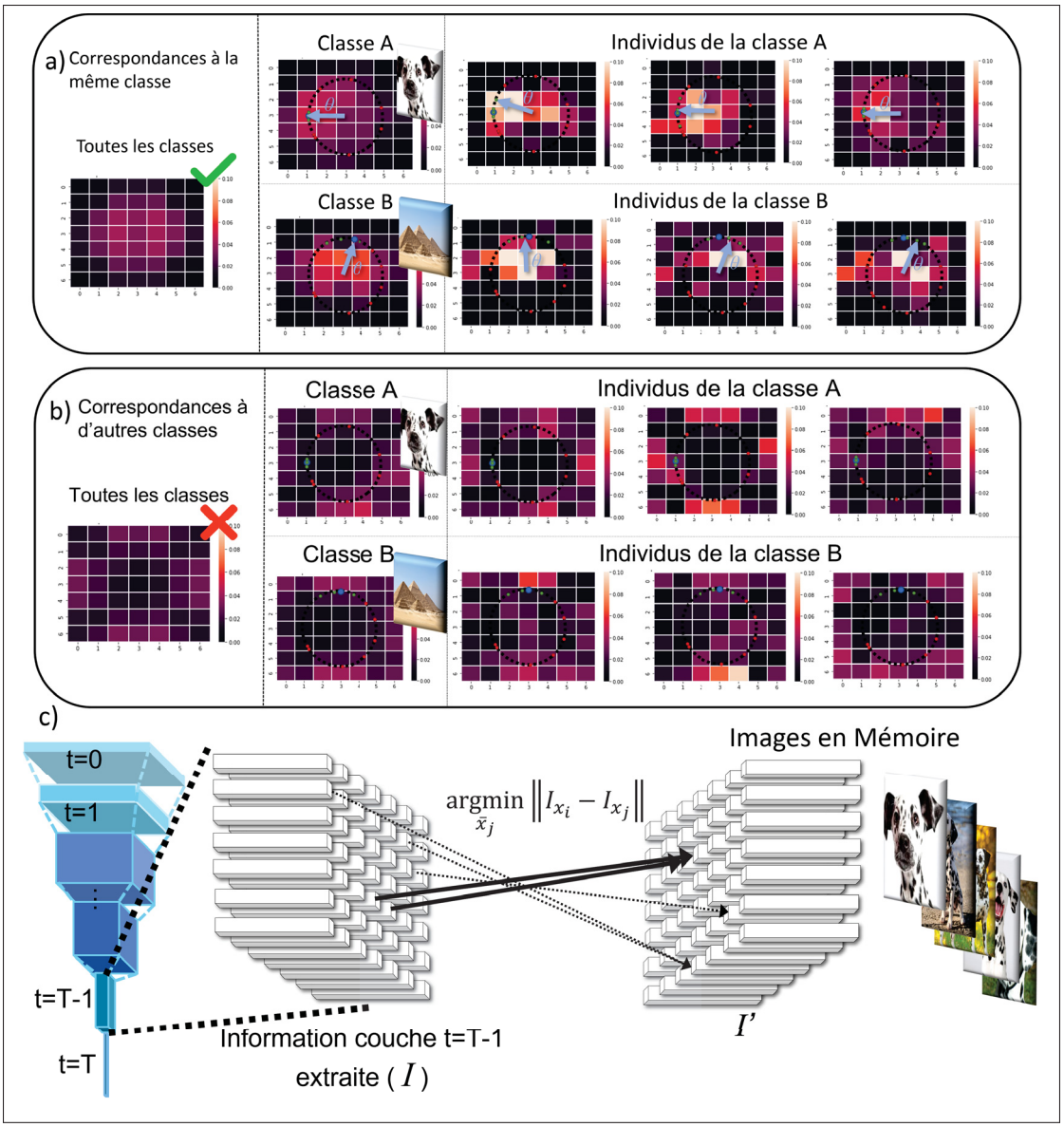

FIGURE 3.3 $U(1)$ rupture de symétrie observée dans les distributions des correspondances de vecteurs d'activation pour les classes de Caltech 101 Fei-Fei *et al.* (2006). a) Les correspondances entre les images de la même classe sont concentrées près de l'origine $(0, 0)$ et présentent des distributions angulaires constantes θ pour des classes et des individus spécifiques (par exemple, classes A et classes B, flèches bleues). Ici, l'angle θ affichée est calculée par la somme des positions (x, y) des pixels dans la carte de caractéristique, où le pixel central est $(0, 0)$, des distributions, pondérées par les probabilités de correspondances. b) Les correspondances entre les images de classe non apparentée sont dispersées à la périphérie et n'exhibent pas de distributions angulaires apparentes. c) L'algorithme 3.2 représenté visuellement affichant les correspondances entre les vecteurs de pixels extraits d'une image d'inférence et des vecteurs de pixels en mémoire de différentes images. Image adaptée de Bouchard *et al.* (2022).

Ces observations expérimentales concernant la symétrie sont dérivées d'un algorithme de classification rudimentaire du plus proche voisin (NN), où les vecteurs d'activation $I_{\bar{x}}$ des emplacements de pixels individuels $\bar{x}$ sont mis en correspondance entre les emplacements d'image d'inférence $\bar{x}_i$ et les emplacements NN $\bar{x}_{nn}$ de vecteurs stockés en mémoire, voir algorithme 3.1 et 3.2. Nous justifions l'utilisation de l'indexation vectorielle de pixels par l'observation qu'elle conduit généralement à des performances de classification supérieures par rapport à d'autres représentations, y compris la mise en commun maximale ou l'aplatissement, voir Figure 3.2 et Figure I-1 partagée dans l'Appendice I, pour un certain nombre de réseaux différents et de tâches variées de classification.

La Figure 3.3 montre les distributions des emplacements de correspondance des voisins les plus proches $\bar{x}_{nn}$. La Figure 3.4 montre les distributions $p(\bar{x}_{nn}|\bar{x}_i)$ des NN emplacements de pixels correspondants $\bar{x}_{nn}$ conditionnés aux emplacements de l'image d'inférence $\bar{x}_i$. Les emplacements correspondants $x\bar{x}_{nn}$ sont étroitement répartis autour des emplacements de pixels provenant de l'image d'inférence $\bar{x}_i$, ce qui indique que, même dans les couches de goulot d'étranglement des CNNs, les informations d'activation $I_{\bar{x}}$ sont très spécifiques à l'emplacement $\bar{x}$ et que les distributions de correspondance de classes non liées présentent une variabilité sensiblement plus élevée dans l'emplacement du voisin le plus proche $\bar{x}_{nn}$, démontrant une certaine importance à la position spatiale.

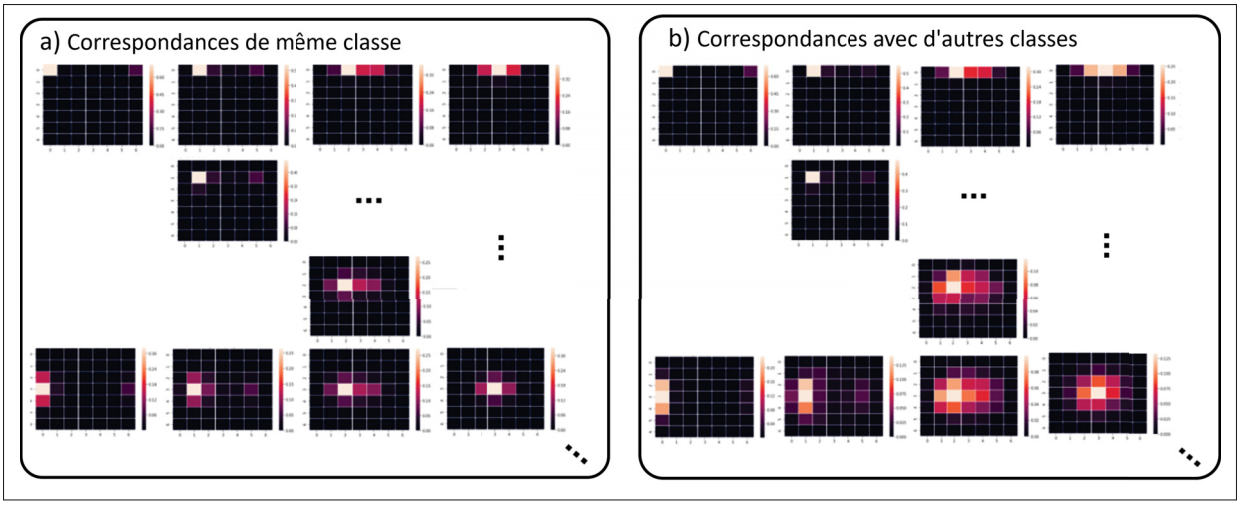

FIGURE 3.4 Distributions $p(\bar{x}_{nn}|\bar{x}_i)$ des NN emplacements de pixels correspondants $\bar{x}_{nn}$ conditionnés aux emplacements l'image d'inférence $\bar{x}_i$. La récupération de la mémoire est basée sur des vecteurs d'activation à 1920 dimensions $I_{\bar{x}} \in R^{1920+}$ de la couche de goulot d'étranglement de $7 \times 7 = 49$ pixels d'un modèle Densenet201, Huang *et al.* (2016), préentraîné sur Imagenet, Deng *et al.* (2009), et testé sur Caltech 101, Fei-Fei *et al.* (2006), contenant des images non utilisées dans l'entraînement initial. Les distributions sur $\bar{x}_{nn}$ sont affichées pour une sélection d'emplacements de pixels provenant de l'image d'inférence individuels $\bar{x}_i$. Ceux-ci sont étroitement regroupés autour des emplacements de pixels de l'image d'inférence d'origine $\bar{x}_i$ (pixels les plus brillants) et présentent une variance plus faible pour les correspondances avec les instances de la même classe (à gauche) par rapport à une classe non liée (à droite). Notez que les variations sont plus fortes dans les directions tangentielles (par opposition aux directions radiales), indiquant des déviations angulaires $\theta \in U(1)$ autour du cercle unitaire. Image adaptée de Bouchard *et al.* (2022).

3.2 Expériences Initiales d'Entraînement avec un Encodeur-Décodeur

Les U-Nets, Ronneberger, Fischer & Brox (2015), se sont révélés être des solutions puissantes pour reconstruire des images avec précision, Isola, Zhu, Zhou & Efros (2017); Karras, Laine & Aila (2019); Park *et al.* (2020); Karras *et al.* (2020a,b). Les approches récentes de Réseau antagoniste génératif (Generative Adversarial Network) (GAN) utilisent des encodeurs et décodeurs basés sur les CNN (U-Net) avec des sauts de connexion (skip connections) afin d'aider à la reconstruction d'un signal entre l'encodeur et le décodeur, Isola *et al.* (2017). Les informations sont encodées dans un code latent, une représentation optimale et dense en informations de notre image

d'entrée, en utilisant un CNN classique comme ResNet, He *et al.* (2016). Un signal est ensuite reconstruit à l'aide d'un suréchantillonnage et de convolutions transposées à partir du code latent et les informations précieuses de l'encodeur sont concaténées via des sauts de connexion pour produire des résultats de meilleure qualité, Isola *et al.* (2017).

Nous étudions si la reconstruction de notre signal $U(1)$ observé au goulot d'étranglement est possible. Pour se faire, nous construisons l'architecture d'encodeur-décodeur basé sur les CNNs la plus simple possible, similaire à Ronneberger *et al.* (2015), où nous utilisons un modèle ResNet-18, He *et al.* (2016), sans modifications et y remplace le module de classification par un module de reconstruction que nous allons décrire ci-bas afin de reconstruire les signaux observés dans les cartes d'activation 7×7 du goulot d'étranglement et y concaténer l'information de l'encodeur. Nous avons choisi d'utiliser l'architecture ResNet-18, He *et al.* (2016), pour cette expérience due au fait que ce soit une architecture vastement utilisée dans l'industrie et donc plus simple à comparer et à réimplémenter par nos paires et moins gourmande en temps de calcul étant également la variante la plus légère de la famille ResNet.

Nous prenons ainsi la représentation latente de dimensions $1 \times 1 \times 320$ (représentation finale du réseau ResNet-18 de base juste avant le module de classification complètement connecté) à partir de laquelle nous effectuons des convolutions transposées en 2D pour obtenir notre tenseur reconstruit de $7 \times 7 \times 320$, que nous concaténons avec un saut de connexion de notre encodeur à sa couche au niveau du goulot d'étranglement, donnant un tenseur reconstruit final de $7 \times 7 \times 320 + 7 \times 7 \times 320 = 7 \times 7 \times 640$. Plus précisément, nous avons la dernière couche de $1 \times 1 \times 320$ qui est envoyé dans deux couches de convolutions transposées de noyau de tailles 5×5 avec un "padding" de un, "pas" (stride) de deux et fonction d'activation ReLU, Agarap (2018), conservant le même nombre de canaux, afin de retrouver notre signal du "bottleneck" de $7 \times 7 \times 320$. Nous suivons également le protocole habituel de ResNet en y ajoutant les mêmes **normalisations par lots** retrouvées dans l'architecture de l'encodeur suite aux couches de convolutions.

Au lieu d'utiliser un réseau discriminateur comme dans l'approche GAN pour entraîner notre modèle à reconstruire nos annotations, nous nous concentrons sur la minimisation de la distance euclidienne entre la reconstruction et nos annotations enregistrées en mémoire, eq. 3.4, qui s'est avéré plus appropriée pour représenter les informations spatiales, Isola *et al.* (2017). Notons que Isola *et al.* (2017) utilise la distance de Manhattan (L1), que nous avons également comparé avant d'entrer dans nos expérimentations plus avancées et nous avons constaté que la distance euclidienne (L2) produisait des résultats légèrement supérieurs avec nos annotations de tenseurs de $7 \times 7 \times 640$.

$$L_2 = \frac{\sqrt{\sum_C (I - I_c)^2}}{activated\_tensor\_size} \tag{3.4}$$

où L_2 est la fonction de perte euclidienne représentée par la somme des distances euclidiennes entre le tenseur de caractéristiques prédis par le réseau et comparé avec tous les tenseurs de caractéristiques à l'intérieur de nos annotations de classes pour une image, normalisée suivant la taille des paramètres actifs ($activated\_tensor\_size$) dans notre tenseur, soit $7x7x10$, décrit ci-bas.

Nous avons testé diverses configurations d'annotations variant tous les paramètres possibles telles que les tailles d'annotations (spatial (x, y) et nombre de canaux), distributions utilisées pour les valeurs des scalaires utilisés, différents niveaux de parcimonie (sparsity) au niveau de nos canaux (activant et désactivant certains canaux aléatoirement dans la construction de nos annotations du au fait que les réseaux profonds semblent bien répondre à de hauts niveaux de parcimonie, Changpinyo, Sandler & Zhmoginov (2017); Hoefler, Alistarh, Ben-Nun, Dryden & Peste (2021)), ou encore donner plus d'emphase sur certains emplacements (x, y) lors du calcul de la fonction de perte (sous la forme de poids binaire 1 ou 0 pour certaines configurations spatiales décrites ci-dessous).

Plus précisémment, nous avons testé différentes tailles de cartes de caractéristiques et profondeur de canaux en variant les deux couches de convolutions transposées de notre décodeur ($7 \times 7 \times 640$ offrant les meilleurs résultats), diverses plages de valeurs et de distributions pour les valeurs dans nos annotations ($\in [-1, +1]$ uniformément distribuées donnaient les meilleurs résultats), ainsi

que différents niveaux de parcimonie (sparsity) dans nos 640 canaux où seulement une partie des canaux étaient "activés" et les autres mis à zéro (nous avons identifié qu'environ 10 canaux "activés" de manière aléatoire et tous les autres canaux réglés sur zéro donnaient de meilleurs résultats que, par exemple, activer tous les canaux). Nous avons également utilisé plusieurs schémas d'encodage spatial pour les canaux activés donnant plus d'importance à certaines positions spatiales, affichées à la Figure 3.5, pour les vecteurs de pixels. Nous avons observé que les informations autour d'un cercle unitaire fonctionnaient mieux que les informations centrées ou en périphérie, voir Figure 3.5, ce qui confirme nos observations sur l'importance de la position spatiale (x, y) dans ces cartes de caractéristiques.

Notre architecture finale d'encodeur-décodeur contient donc un encodeur ResNet-18, He *et al.* (2016), sans modifications, suivis de deux couches de convolutions transposées couplées de normalisations de lots et un saut de connexions afin de reconstruire nos annotations de $7 \times 7 \times 640$. Ces annotations sont composées de, pour chaque classe, dix canaux aléatoirement activés et les 630 autres contenant uniquement des zéros où ces dix canaux détiennent des valeurs comprises dans l'intervalle $[-1, 1]$. Suivant cette architecture, nous testons différentes configurations spatiales (x, y), affichées dans la Figure 3.5, afin d'étudier l'effet de la position dans ce tenseur de goulot d'étranglement reconstruit.

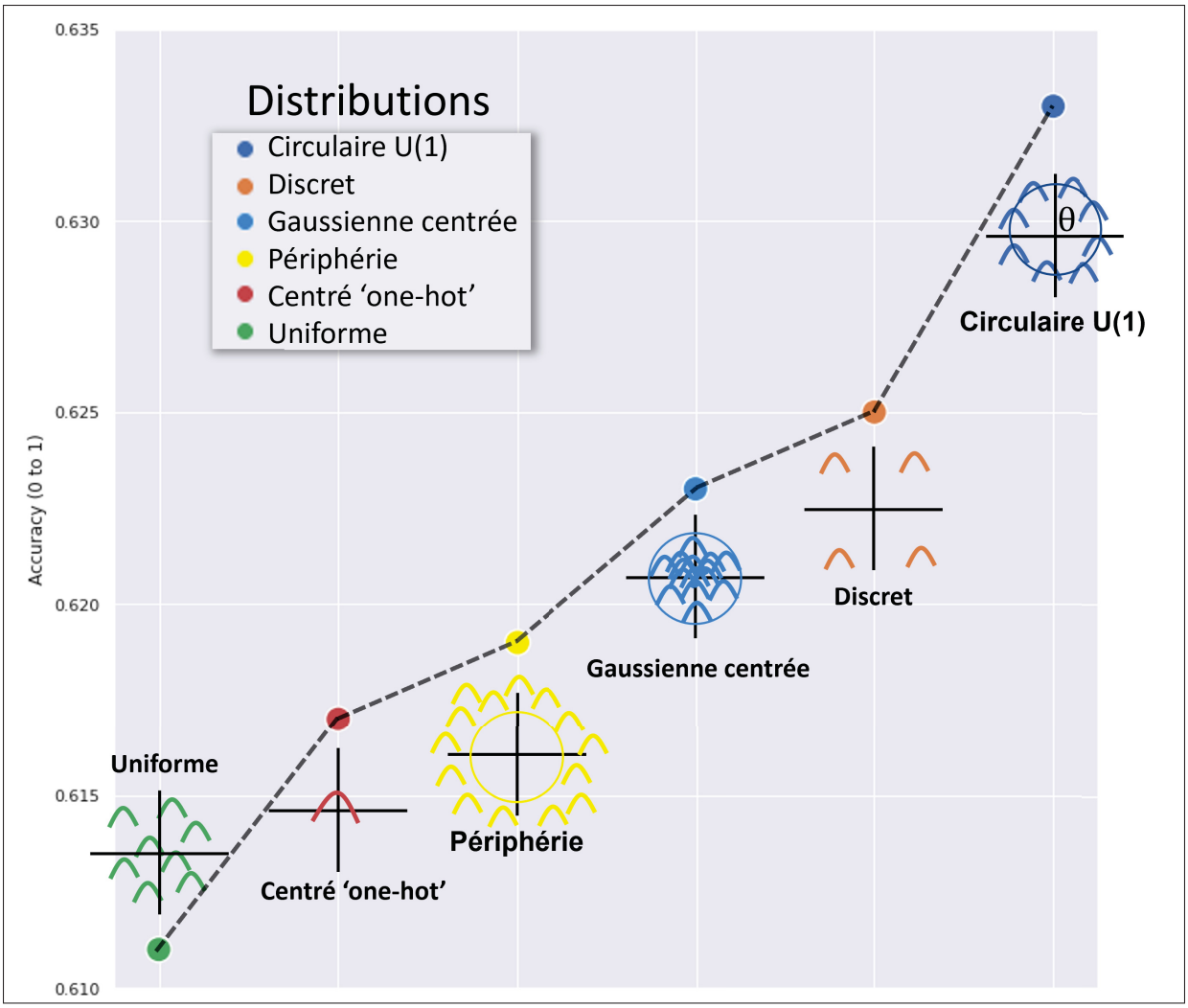

FIGURE 3.5 Comparaison d'entraînement à partir de zéro pour six des distributions spatiales d'annotations testées sur Cifar-100 Krizhevsky *et al.* à l'aide de l'architecture d'encodeur-décodeur basée sur ResNet-18, He *et al.* (2016) décrite dans cette section. La séparation de données pour la base d'entraînement et de test utilisée dans ces expérimentations est celle de base lorsqu'importée de PyTorch Paszke *et al.* (2019), spécifiée par les auteurs.

Nous entraînons le modèle sur 100 époques fixes, 28 tailles de lot, avec les ensembles d'entraînement, de validation et de test d'origine spécifié par les auteurs (selon la base de données) et une validation croisée quintuple sur un seul GPU Titan RTX. Nous utilisons l'optimiseur Adam, Kingma & Ba (2014), et le planificateur CosineAnnealingLR, Loshchilov & Hutter (2016), avec les paramètres par défaut de PyTorch, Paszke *et al.* (2019), et l'augmentation des

données de base sous la forme de retournements horizontaux et "crop" aléatoires. Sois-disant les mêmes hyper-paramètres que ceux utilisés pour nos expériences décrites dans la section suivante.

Bien que les résultats aient été prometteurs dans une architecture d'encodeur-décodeur ResNet, He *et al.* (2016) et confirmée avec DenseNet, Huang *et al.* (2016), cette solution n'est pas assez efficace, car nous ajoutons beaucoup plus de paramètres au modèle initial dû au décodeur. Nous avons donc décidé de changer d'approche plutôt que d'optimiser l'implémentation (et hyperparamètres) et généraliser cette approche aux différentes bases de données. Ayant remarqué la puissance d'un poids circulaire sur nos annotations et suite aux observations de la section précédente, nous avons décidé d'étudier une approche plus directe et optimale de nos annotations $U(1)$, représentant tout ce tenseur provenant du goulot d'étranglement à l'aide d'un seul paramètre $\theta \in U(1)$ permettant de sauver beaucoup de paramètres et enlever notre décodeur de l'équation.

Malgré le fait que cette approche ne soit ni optimisée ni utilisée pour des tests plus en profondeurs, nous jugeons ces tests pertinents et favorables à la compréhension de notre parcours vers le modèle final décrit dans la section suivante, d'où notre décision de laisser cette section dans le document.

3.3 Entraînement avec les Annotations U(1)

Étant donné le modèle de rupture de symétrie angulaire persistante observée dans le cas de l'indexation de réseau préentraîné et dans les tests à l'aide de l'architecture encodeur-décodeur, nous émettons l'hypothèse que l'orientation peut être apprise à partir d'annotations de classe aléatoires placées autour d'un cercle unitaire $U(1)$ afin d'améliorer la classification.

Nous proposons de générer des annotations de classes aléatoires distribuées uniformément autour d'un cercle unitaire $\theta \in U(1)$ dans le plan (x, y). Notre processus d'entraînement combine une fonction de perte unique standard avec une perte angulaire supplémentaire $U(1)$, et s'applique aux réseaux génériques (CNNs) et à l'entraînement à partir de zéro sans modifications excessives. La

fonction de perte utilise deux composants, un composant symétrique avec l'encodage classique "one-hot" (équation 3.6 gauche) et un composant antisymétrique basé sur nos annotations $U(1)$ (équation 3.6 droite), tous deux utilisant l'entropie croisée (de Torch, Paszke *et al.* (2019)) comme calcul de perte finale pour générer une perte d'entropie croisée combinée pour la rétropropagation, représentée ici :

$$L = Composant\ Symétrique\ +\ Composant\ Antisymétrique \tag{3.5}$$

$$L = \alpha L_{one-hot}\ +\ (1-\alpha)L_{U(1)} \tag{3.6}$$

$$L = \alpha(-\sum_{c=1}^{M} y_c \log(\frac{\exp{(x_c)}}{\sum_{i=1}^{M}\exp{(x_i)}}))\ +\ (1-\alpha)(-\sum_{c=1}^{M} y_c \log(\frac{\exp{(d_c)}}{\sum_{i=1}^{M}\exp{(d_i)}})) \tag{3.7}$$

où α est une constante déterminée expérimentalement (décrit ci-bas), M est le nombre de classes, y_c l'indicateur binaire égal à 1 si la classe c est la bonne classe et 0 sinon, x_c le résultat prédit pour one-hot à la classe c, d_c la différence euclidienne entre l'angle prédit et l'angle associé à la classe c, calculée suivant l'équation :

$$d_c = \frac{-\sqrt{(dx_p - dx_c)^2 + (dy_p - dy_c)^2}}{2 \times M} \tag{3.8}$$

où $(dx, dy)_p$ est l'angle prédit par notre réseau et $(dx, dy)_c$ l'angle de la classe c dans nos annotations. Les deux angles sont représentés sous formes de coordonnées spatiales (dx, dy) de norme unitaire $||(dx, dy)|| = 1$.

Comme stipulé ci-dessus, les annotations $U(1)$ sont paramétrées comme des coordonnées (x, y) contraintes à $x^2 + y^2 = 1^2$, et estimées via des couches entièrement connectées immédiatement après le goulot d'étranglement du réseau en parallèle avec les couches connectées pour one-hot ayant uniquement deux sorties (x, y) au lieu de N sorties pour N classes avec one-hot. Nos expériences sont conduites implémentant des architectures CNNs génériques (par exemple, EfficientNet-B0, Tan & Le (2019), ResNet-18, He *et al.* (2016)), c'est-à-dire sans aucune

modification de l'architecture interne du réseau outre l'ajout de ce composant antisymétrique parallèle à l'encodage one-hot.

Nous entraînons le modèle sur 100 époques fixes, des tailles de lot de 28, avec les ensembles d'entraînement, de validation et de test d'origine et une validation croisée quintuple sur un seul GPU Titan RTX. Nous utilisons l'optimiseur Adam, Kingma & Ba (2014), et le planificateur CosineAnnealingLR, Loshchilov & Hutter (2016), avec les paramètres par défaut de PyTorch, Paszke *et al.* (2019), et l'augmentation des données de base sous la forme de retournements horizontaux et "crop" aléatoires. Cette configuration est la même dans toutes les expériences de cette section. L'entraînement et la classification sont évalués dans diverses tâches d'apprentissage de degrés variés de granularité, y compris les ensembles de données "Describable Textures Dataset" (DTD), Cimpoi *et al.* (2014), "Caltech-UCSD Birds", Welinder *et al.* (2010), "Stanford Dogs", Khosla *et al.* (2011), "Flowers", Nilsback & Zisserman (2008), "Pets", Parkhi, Vedaldi, Zisserman & Jawahar (2012), "Indoor67", Quattoni & Torralba (2009), "FGVC-Aircraft", Maji, Rahtu, Kannala, Blaschko & Vedaldi (2013), "Cars", Krause, Stark, Deng & Fei-Fei (2013), et Imagenet, Deng *et al.* (2009).

Nous analysons également l'effet de la constante α de l'équation 3.6 sur la classification de Caltech-UCSD Birds, Welinder *et al.* (2010), utilisant les mêmes hyperparamètres et paramètres de validation et d'entraînements utilisés pour l'entraînement décrit ci-dessus, afin de trouver expérimentalement les proportions idéales pour notre constante. Nous obtenons nos meilleurs résultats lorsque α est fixé à 0.5, donnant autant de poids aux deux composants de la fonction de perte, voir Figure 3.6. Nous utilisons donc cette configuration pour les expérimentations à venir.

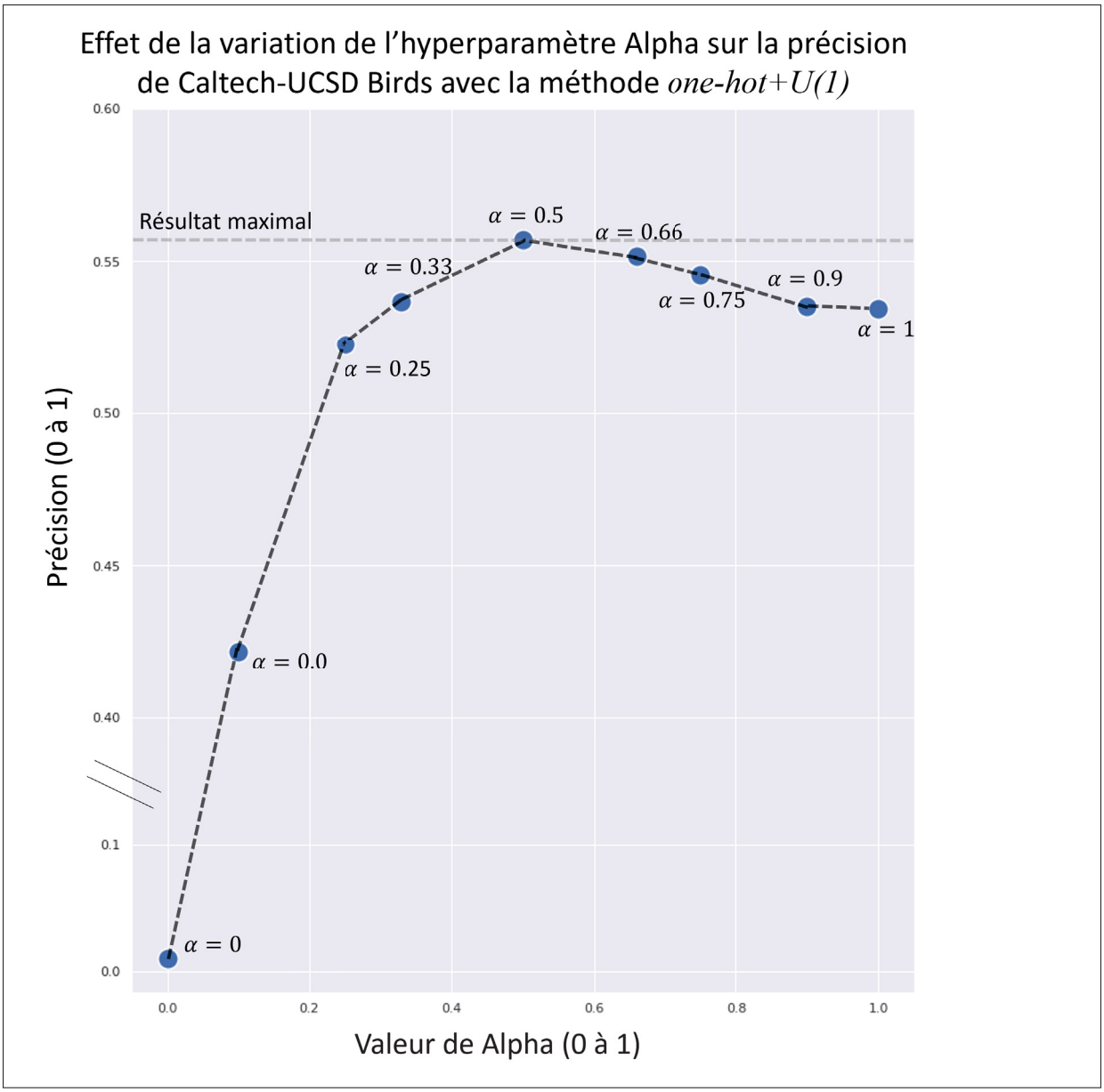

FIGURE 3.6 Étude de l'effet de la constante alpha pour un entraînement avec la méthode αone-hot $+(1-\alpha)U(1)$ sur Caltech-UCSD Birds, Welinder *et al.* (2010), utilisant les mêmes hyperparamètres et paramètres de validation et d'entraînements utilisés pour l'entraînement décrit ci-dessus.

L'entraînement avec les annotations $U(1)$ a amélioré la classification pour tous les réseaux testés. Le tableau 3.1 présente les résultats pour l'architecture de réseau EfficientNet-B0, Tan & Le (2019), qui a conduit à la plus grande précision globale parmi les réseaux testés, ainsi que pour

l'architecture ResNet largement utilisée, He *et al.* (2016), pour démontrer la généralisation de notre approche pour diverses architectures. La Figure 3.7 montre l'évolution de la précision (a), la perte (b) et des prédictions du modèle $U(1)$ (c) dans le temps (époques d'entraînement) pour les mêmes exemples de 10 classes de Caltech-UCSD Birds, Welinder *et al.* (2010), suivis au cours de l'entraînement. Notez que cette configuration est différente de celle de "Ring loss", Zheng *et al.* (2018) où un graphique similaire se trouve à la Figure 1. Ici, nous prédisons des annotations d'angle (un seul paramètre, définit avec des coordonnées (x, y)) pour chacune des classes afin d'ajouter un composant antisymétrique (dû à l'orientation d'une classe vers un angle précis plutôt que central, symétrique, dans la carte de caractéristique) à la perte symétrique de "one-hot" et aider le modèle pendant l'entraînement. Les plus grands cercles de la Figure 3.7 sont des angles de classe prédéterminés, nos annotations, que le modèle tente de prédire pour chaque image lors de l'entraînement (petits points). Chez "Ring loss", la normalisation des caractéristiques est effectuée pour contraindre la norme du vecteur extrait (x_1) sur toute sa profondeur (par exemple 512 canaux pour ResNet-64 dans leurs expériences) à "l'anneau" unitaire mis à l'échelle $(||x_1|| = 1)$ pour comparer avec un vecteur de classe unitaire (w_1). Il s'agit de trouver l'orientation globale d'un vecteur et de comparer systématiquement les vecteurs. Avec un bon schéma d'entraînement, les résultats de "Ring loss" *devraient* être séparés correctement, et **idéalement** ressemblent à notre schéma pour l'époque 100 si le modèle apprend à séparer les classes à travers toutes ses dimensions et est représenté en deux dimensions par la suite. En revanche, nous prédisons un seul angle avec le modèle au lieu d'utiliser les vecteurs complets. Par exemple, les points affichés dans Figure 3.7 sont nos prédictions et annotations de classe directement affichées sans aucun post-traitement.

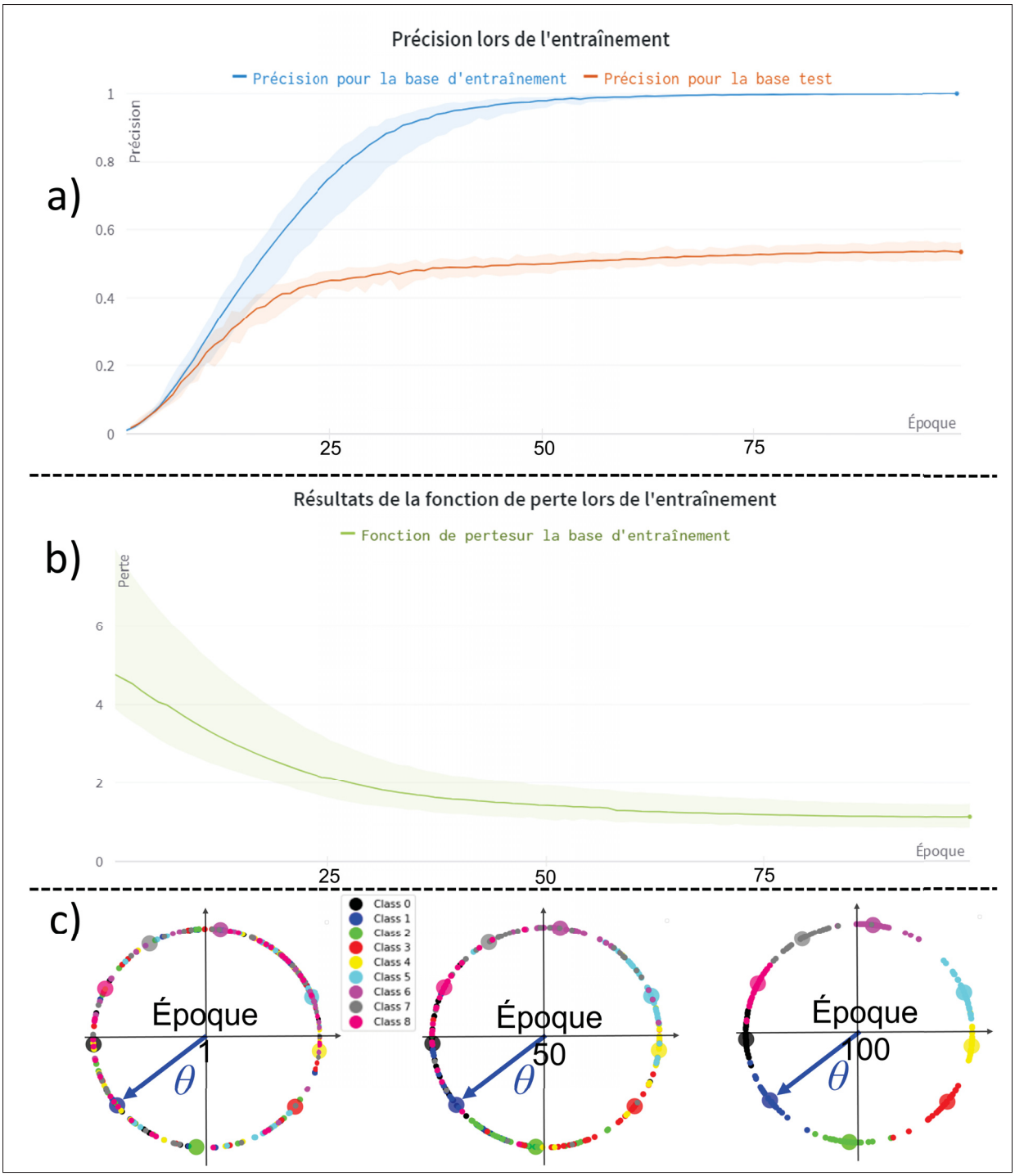

FIGURE 3.7 Évolution de la précision (a), de la perte (b) et des prédictions du modèle $U(1)$ (c) au fil du temps (époques). c) affiche les mêmes exemples (petits points de couleurs) suivis pendant l'entraînement pour 10 classes (plus gros points) de Caltech-UCSD Birds, Welinder *et al.* (2010), utilisant notre architecture de réseau EfficientNet-B0 modifiée, Tan & Le (2019) (α one-hot $+(1-\alpha)U(1)$), avec $\alpha = 0.5$.

Modèle	DTD Cimpoi *et al.*	UCSD Birds Welinder *et al.*	Stanford Dogs Khosla *et al.*	Flowers Nilsback & Zisserman	Pets Parkhi *et al.*	Indoor67 Quattoni & Torralba	FGVC-Aircraft Maji *et al.*	Cars Krause *et al.*	Imagenet Deng *et al.*
ResNet-18 He *et al.*									
One-hot	0.4153	0.3875	0.4015	0.4863	0.4090	0.4866	0.8703	0.3575	0.6350
One-hot + $U(1)$	**0.5230**	**0.5313**	**0.5754**	**0.6487**	**0.5919**	**0.6224**	**0.8920**	**0.8011**	**0.6621**
EfficientNet-B0 Tan & Le									
One-hot	0.5310	0.4907	0.5655	0.7252	0.6689	0.5458	0.8685	0.7882	0.7144
One-hot + $U(1)$	**0.5368**	**0.5569**	**0.5764**	**0.7438**	**0.6797**	**0.5477**	**0.8768**	**0.8079**	**0.7171**

TABLEAU 3.1 Résultats de la classification à l'aide d'un entraînement basé sur des annotations α one-hot+$(1-\alpha)U(1)$ où $\alpha = 0.5$ avec ResNet-18, He *et al.* (2016), démontrant la plus grande augmentation de précision générale et EfficientNet-B0, Tan & Le (2019), démontrant les meilleurs résultats. L'addition de notre composant antisymétrique $U(1)$ améliore les résultats dans tous les cas.

Comme expérience supplémentaire, nous avons testé des annotations (x, y) générés à partir de diverses distributions à 2 paramètres autres que le cercle unitaire $U(1)$ choisit, y compris des combinaisons binaires discrètes $[\pm 1, \pm 1]$, une distribution centrée ("one-hot" seul) et une distribution uniforme dans l'espace. Nous utilisons ces distributions comme plage de valeurs pour nos deux paramètres supplémentaires antisymétriques prédits par le réseau (en plus de "one-hot") selon la même configuration que décrite ci-dessus afin de tester la validité du cercle unitaire et de ce composant antisymétrique. La Figure 3.8 montre les résultats pour un ensemble de données, Caltech-UCSD Birds Welinder *et al.* (2010), où les annotations $U(1)$ conduisent à la plus grande précision parmi les alternatives. On note également qu'ajouter une composante antisymétrique au composant one-hot semble bénéfique dans tous les cas, aidant le modèle à mieux séparer les classes et améliorer sa précision.

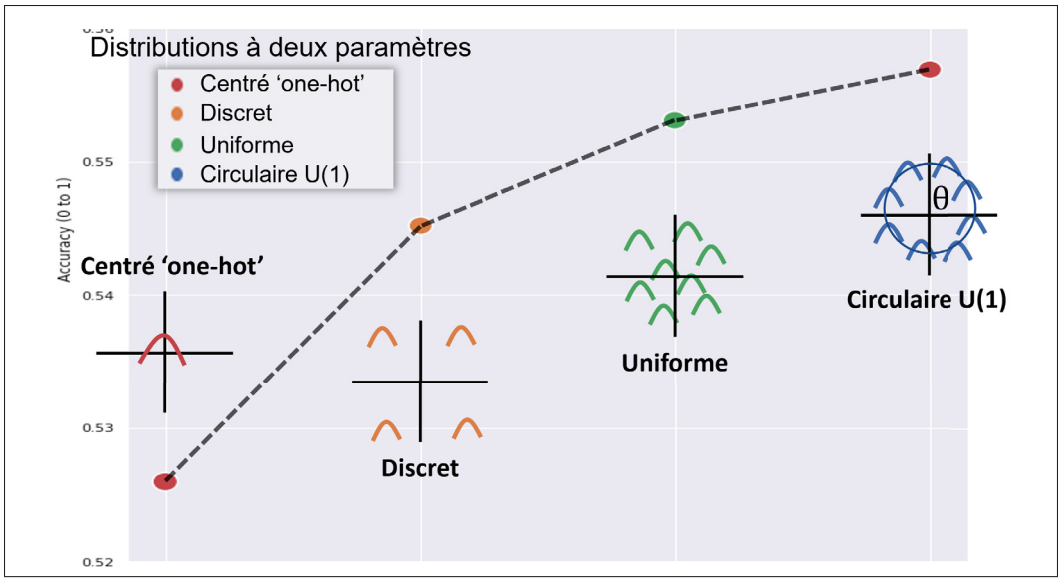

FIGURE 3.8 Comparaison de la précision de la classification pour l'entraînement à partir de zéro avec diverses configurations d'annotations (x, y) à 2 paramètres : one-hot, discret, uniforme et cercle unitaire $U(1)$. Notre $U(1)$ proposé conduit à la plus grande précision (ici en utilisant l'ensemble de données Caltech-UCSD Birds, Welinder *et al.* (2010)). Image adaptée de Bouchard *et al.* (2022).

Nous notons que **le but de ces expérimentations n'est pas d'avoir les meilleurs résultats possible ni d'optimiser nos formules** et codes en termes de temps de calcul et de précision. Nous avons plutôt opté sur le fait **d'introduire et démontrer, de façon la plus générale possible (plusieurs bases de données et réseaux), que l'utilisation d'une représentation spatiale angulaire** $U(1)$**, composant antisymétrique**, dans les couches profondes d'un réseau CNN, couplées à un composant symétrique (classification avec encodage "one-hot") est intéressante et devrait être investigué dans des recherches à venir optimisant les hyperparamètres et formules présentés dans ce document.

Effectivement, les hyperparamètres utilisés dans ces expériences pourraient être optimisés pour chaque base de données testée afin de maximiser les résultats et augmenter les performances de notre approche. Malheureusement, une telle optimisation n'était pas possible dans le cadre de ces recherches (maîtrise 2 ans) nécessitant beaucoup de temps d'entraînement et de tests (par

exemple, deux semaines pour l'entraînement de zéro sur ImageNet pour une seule validation de la validation croisée quintuple), mais sont certainement des pistes de recherches intéressantes afin de confirmer la viabilité de notre hypothèse $U(1)$ lorsque le modèle est poussé à son maximum (résultats près de l'état de l'art). Malgré tout, nous voyons un effet positif de l'ajout du composant antisymétrique dans la fonction de perte sans même une optimisation conséquente et confirmons ainsi nos observations initiales quant à l'angle de classe observé avec la correspondance de pixels et cette asymétrie créée par chaque classe individuelle.

CONCLUSION ET RECOMMANDATIONS

Nous présentons un nouveau modèle de propagation de l'information dans les couches CNN profondes à partir d'une analogie avec la propagation de la lumière dans un système optique, comprenant un point focal et un cercle de confusion inhérent dans les deux cas. Nous supposons l'équivalence de l'axe optique principal $(t, 0, 0)$ et du vecteur médian de l'espace des traits d'activation positifs. Ceux-ci sont tous deux au centre du plan image (x, y) permettant à la classe d'être caractérisée par un angle similaire à la teinte (hue) $\theta \in U(1)$ utilisée pour définir la couleur dans un espace RVB (plus précisémment HSL), défini à la fois dans le plan image et dans l'espace des caractéristiques d'activation. Deux espaces pouvant être visualisés similairement comportant un "espace" bidimensionnel important où des pixels contiennent la réponse en intensité de plusieurs filtres représentée dans nos multiples canaux.

Notre découverte la plus importante concerne les observations faites au sujet de l'information de l'espace d'image (x, y), qui, dans sa forme la plus concentrée dans les couches du goulot d'étranglement $(t = T - 1)$ du réseau juste avant la compression spatiale $1 \times 1 = 1$ à la couche T , est bien représentée par une seule variable unitaire $\theta \in U(1)$. Ces observations démontrent un processus par lequel la classification d'images implique une rupture de symétrie $U(1)$ de l'énergie $E(t, x, y) = \|I_{t,\vec{x}}\|^2$. Nos observations d'une asymétrie angulaire persistante avec la classe liée à l'image sont les premières dans la littérature, à notre connaissance, et indiquent que la rupture de symétrie se produit naturellement dans les CNNs préentraînés génériques, reliant plus étroitement l'apprentissage en profondeur aux réseaux biologiques, suggérant l'optimisation de la méthode d'entraînement vis-à-vis cette asymétrie. Nos observations angulaires sont cohérentes avec les modèles biologiques : en particulier, les déviations θ dans la Figure 3.3 induites à partir des entrées d'image, et les déviations θ dans l'orientation/la sélection de cap dans le système visuel de la mouche à fruits induites par des photons, voir Kim *et al.* (2017) Figure 3.

Nos expériences utilisent les architectures de réseaux de neurones génériques les plus connues et les ensembles de données et la classification du voisin le plus proche, Cover & Hart (1967), dans le but d'observer notre théorie $U(1)$ et de faire un premier pas confirmant cette représentation angulaire en fin de réseau. Notre méthode expérimentale de correspondance de vecteurs de pixels est efficace pour la classification basée sur une mémoire et la démonstration de notre théorie avec des architectures CNN générales préentraînées, en particulier pour des couches de goulot d'étranglement "lo-res", répondant positivement à notre interrogation sur la pertinence du positionnement spatial dans ces couches profondes. Nous sommes conscients que les résultats sont loin de l'état de l'art. Nous avons décidé de **mettre l'accent sur la généralisation de notre approche** (autant au niveau des modèles que des tâches étudiés) plutôt que sur l'optimisation de l'approche, **introduisant ce nouveau concept de fonction de perte à deux composants : symétrique (one-hot) et antisymétrique** $(U(1))$ que nous croyons pouvant être bénéfique à l'entraînement et la convergence des réseaux profonds, donnant autant une nouvelle façon de voir et comprendre l'information dans les couches profondes des CNNs, qu'une façon de la traiter et l'optimiser.

Une avenue de recherche intéressante et nécessaire serait l'optimisation de la fonction de perte et optimisations propre à toutes les bases de données afin de voir si l'approche reste tout aussi puissante dans des limites différentes poussant la qualité des résultats à l'état de l'art. Ces faibles résultats sont explicables par le fait que nous avons limité l'augmentation de données uniquement aux retournements horizontaux et avons choisi des hyperparamètres utilisés généralement par la communauté plutôt que d'affiner ces paramètres à notre réseau et nos tâches afin de ne pas influencer les résultats avec de nombreux hyperparamètres variables et garder une base solide fixe nous permettant d'**introduire cette nouvelle approche à la communauté** démontrant son potentiel généralisable, ce que nous jugeons avoir fait avec succès et **agissant comme preuve de concept pour cette théorie "**$U(1)$**"**.

Une limitation importante à noter est l'effet non négligeable de l'augmentation des données lors de l'entraînement de réseaux artificiels (profonds) (par exemple, utilisés lors de l'entraînement de "modèles préentraînés" sur de grands ensembles de données comme Imagenet), Simard, Steinkraus, Platt *et al.* (2003); Perez & Wang (2017); Shorten & Khoshgoftaar (2019). La symétrie et la brisure de symétrie observées peuvent être dues à des retournements répétitifs, de petites translations et rotations, des "crops" aléatoires centrées, etc. L'entraînement à partir de zéro utilisant notre approche avec la fonction de perte one-hot + $U(1)$ en utilisant beaucoup moins de transformations démontre la généralisation et la pertinence des annotations $\theta \in U(1)$ et des représentations unitaires récentes en général, Tang *et al.* (2021); Kiani *et al.* (2022).

Une autre piste intéressante de notre approche pour de futures recherches est le domaine de Fourier. Nous avons observé des comportements intéressants, mais nous n'avons pas pu trouver la manière optimale d'utiliser ces données dans le cadre de cette thèse. Nous pensons que l'utilisation de quelques coefficients de phase denses en informations ou d'un mécanisme d'attention ciblant ces coefficients serait un excellent moyen d'extraire des caractéristiques spatiales importantes pour mieux séparer les classes. Une autre transformation spatiale apparente pour améliorer les résultats dans le domaine spatial consiste à diviser les cartes d'activation générées en composants symétriques et anti-symétriques pour permettre au modèle de se concentrer sur ces derniers et d'améliorer les performances. Nous émettons l'hypothèse que les filtres de réseau profond discriminent automatiquement les informations générales sur l'ensemble de données des informations spécifiques à la classe, mais des recherches plus approfondies sont nécessaires pour confirmer cette hypothèse. De même, d'autres études sur les transformeurs de vision (Vision Transformers) relativement nouveaux, Dosovitskiy *et al.* (2020); Liu *et al.* (2021), constituent une prochaine étape claire pour confirmer la généralisation de notre approche aux architectures profondes. Enfin, une étude sur les canaux des informations de goulot d'étranglement serait nécessaire, car nous pourrions être en mesure de maximiser une structure sous-jacente inconnue également dans cette dimension.

Nous espérons que les analogies de lumière et de couleur, nos observations et la découverte de la rupture de symétrie $U(1)$ susciteront l'intérêt des chercheurs pour étudier la façon dont les réseaux profonds propagent l'information, ce qui pourrait conduire à des architectures plus optimisées et efficaces nécessitant moins de temps de calcul, moins de temps d'entraînement, moins de données d'entraînement et/ou une meilleure précision. De tels travaux théoriques et travaux d'observation sont relativement peu gourmands en calcul et peuvent apporter des architectures profondes plus respectueuses de l'environnement, bien que loin d'être parfaites. Ce qui reste un autre point fort de l'étude des structures sous-jacentes possibles dans les réseaux profonds actuels, plutôt que de continuer avec des réseaux plus vastes et plus de données sur lesquelles apprendre, qui nécessitent des mois de formation sur des milliers de GPUs (par exemple, GPT-3, Brown *et al.* (2020); Narayanan *et al.* (2021)). Finalement, comprendre et améliorer les DNNs pourrait également nous aider à mieux comprendre les réseaux biologiques.

ANNEXE I

TRAVAUX PRÉLIMINAIRES : ANALYSE DU RÉSEAU ET ENSEMBLES DE DONNÉES

Nos observations se sont fortement appuyées sur les résultats de Lazreg (2020). Lazreg a étudié une multitude d'architectures CNN de pointe comparant la qualité des informations extraites à plusieurs profondeurs dans le réseau. Lazreg a découvert que les informations sur les goulots d'étranglement dans l'architecture DenseNet, Huang *et al.* (2016), avaient les meilleures informations pour les capacités d'apprentissage par transfert en utilisant une approche de recherche en mémoire. Plus précisémment, Lazreg a étudier les différentes couches et différentes façon de représenter ces couches en mémoire pour ensuite performer des tâches de classifications à base de K plus proches voisins (KNN). Dans le cas de nos expérimentations préliminaires et des résultats de Lazreg, nous utilisons une approche "max-pooling" où nous envoyons l'image dans un réseau CNN préentraîné qui retire l'information à une couche t et la condense sur le plan spatial (x, y) en un seul vecteur choisissant la valeur maximum pour chaque canal, comme démontré dans l'algorithme I-1.

Algorithme-A I-1 Construction de nos bases de données de decripteurs I et I' pour la classification K-NN

```
/* Construction de nos bases de données de decripteurs I
   et I'                                                  */
```

Entré : BD, une base de données d'images;

Sortie : Un descripteur global par image séparé en base de test I et mémoire I';

Importer le modèle préentraîné au choix (par exemple, DenseNet-201);

for *image im* $\in$ *BD* **do**

  Normaliser l'image suivant la procédure des modèles Tensorflow Abadi *et al.* (2015);

  Traiter l'image im avec le CNN;

  Extraire le tenseur directement avant le *bottleneck*;

  Associer chaque tenseur avec son annotation de classe;

  Représenter le tenseur à l'aide d'un descripteur global ; `/* Tel que "max pooling" ou "average pooling" */`

  Sauvegarder les descripteurs en mémoire avec son annotation;

end for

Séparer la nouvelle base de données de descripteurs en données entraînement (mémoire) et tests suivant la procédure désirée;

```
/* Dans le cas de Caltech, sauvegarder les descripteurs
   de 30 images aléatoires par classes pour la mémoire
   (I') et le reste pour les tests (I). Où I et I'
   contiennent les descripteurs de chaque image de la
   base de données.                                       */
```

Algorithme-A I-2 Classification - K plus proches voisins avec la correspondance de pixels

```
/* Classification des plus proches voisins pour la base
   de données de test                                    */
```

Entré : I et I', une base de données d'entraînement et de tests contenant nos descripteurs globaux;

Sortie : Classification d'une base de données;

for *Image im* $\in I$ **do**

    Calculer la distance entre p et $p' \in I'$;

    /* où p' sont les descripteurs de toutes les images de notre base de données en mémoire. */

    Trier les résultats de distance des descripteurs p de l'image im en ordre croissant de distance pour tous les descripteurs p' dans I';

    Sélectionner les k-plus proches descripteurs en mémoire;

    Identifier la classe des k-plus proches descripteurs.;

    Sélectionner la classe pour le choix final de la classe.;

end for

Résultats que nous confirmons sur plusieurs tâches d'apprentissage à quelques coups (few-shot learning) dans le tableau I-1 en utilisant la même approche, soit extraire les informations de goulot d'étranglement des réseaux mentionnés à l'aide de réseaux préentraînés Tensorflow, Abadi *et al.* (2015) et regroupés (pooled). Notez qu'EfficientNet, Tan & Le (2019), (et d'autres architectures CNN comme SqueezeNet, Iandola *et al.* (2016), ResNeXt, Xie, Girshick, Dollár, Tu & He (2017) ou les récents "Vision Transformers" (par exemple ViT, Dosovitskiy *et al.* (2020) ou le "Swin Transformer", Liu *et al.* (2021))), n'existaient pas au moment de ces expérimentations préliminaires. Depuis, nous avons constaté qu'EfficientNet donnait de meilleurs résultats et était beaucoup plus efficace. Nous l'avons donc utilisé comme référence pour notre approche d'entraînement $U(1)$ à partir de zéro.

Réseau préentraîné	Couche utilisée	Dimension pré-max-pooling	UTKFace - Age	UTKFace - Genre	HCP - Genre	HCP - Famille	Caltech 101	# de Paramètres du réseau	Profondeur	Somme des classements	**Classement Final**
DenseNet-121 Huang *et al.*	conv5_-block16_concat	7 x 7 x 1024	0.814	0.782	0.806	**0.085**	**0.879**	8,062,504	121	**15**	**1**
DenseNet-201 Huang *et al.*	conv5_-block32_concat	7 x 7 x 1920	**0.835**	0.825	**0.820**	0.057	0.854	20,242,984	201	17	2
Xception Chollet	block14_sepconv2	7 x 7 x 2048	0.694	**0.833**	0.682	0.047	0.873	22,910,480	126	28	3
InceptionV3 Szegedy *et al.*	mixed10	5 x 5 x 2048	0.682	0.824	0.816	0.042	0.858	23,851,784	159	30	4
InceptionResNetV2 Szegedy, Ioffe, Vanhoucke & Alemi	conv_7b	5 x 5 x 1536	0.798	0.822	0.659	0.043	0.865	55,873,736	572	30	4
MobileNet Howard *et al.*	conv_pw_-13	7 x 7 x 1024	0.748	0.727	0.788	0.054	0.868	3,538,984	88	30	4
VGG-19 Simonyan & Zisserman	block5_-conv4	14 x 14 x 512	0.791	0.789	0.774	0.038	0.861	143,667,240	26	31	5
ResNet-50 He *et al.*	activation_-49	7 x 7 x 2048	0.676	0.714	0.765	0.068	0.892	25,613,800	-	31	5
NASNetLarge Zoph, Vasudevan, Shlens & Le	normal_concat_18	7 x 7 x 4032	0.710	0.710	0.797	0.066	0.828	88,949,818	-	36	6
VGG-16 Simonyan & Zisserman	block5_-conv	14 x 14 x 512	0.786	0.782	0.724	0.024	0.858	138,357,544	23	38	7
NASNetMobile Zoph *et al.*	normal_concat_12	7 x 7 x 1056	0.661	0.732	0.806	0.031	0.712	5,326,716	-	43	8

TABLEAU-A I-1 Résultats de la classification sur cinq tâches pour les 11 architectures CNN préentraînées disponibles dans Keras en 2019, et classement général par ordre décroissant affichant, en ordre, la couche utilisée, les dimensions de cette couche extraite (transformée en un vecteur de $1 \times N$ avec le "max-pooling"), les résultats pour nos différentes bases de données, le nombre de paramètres du réseau total, la profondeur du réseau (nombre de couches T), la somme des classements pour les différentes tâches (somme de positions où le meilleur résultat parmi les 11 modèles aura une position d'un et le modèle avec la précision la plus basse aura 11, le nombre plus bas signifie de meilleurs résultats globaux) et le classement final sur toutes nos tâches par rapport à cette somme des classements.

Notre approche de recherche en mémoire, KNN, a utilisé un K = 10 contrairement à Lazreg qui a choisi un K=3. Augmenter le nombre de voisins améliore la précision, mais a la contrepartie d'augmenter la complexité. Nous avons opté pour un plus grand nombre de voisins afin d'améliorer la qualité des résultats et d'avoir plus de données pour nos observations, puisque la complexité n'était pas un problème pour ces tests. La mesure de distance utilisée pour trouver le

voisin le plus proche est la distance euclidienne, indiquée dans l'équation ci-dessous.

$$D_i = \min_{f(I_j)} ||f(I_i) - f(I_j)||_2 \tag{A I-1}$$

Où D_i est la distance la plus basse trouvée entre le vecteur de caractéristiques testé et tous les vecteurs de caractéristiques à l'intérieur de la mémoire, min sur $f(I_j)$ choisit la valeur de distance minimale entre tous les exemples d'apprentissage à l'intérieur de la mémoire $f(I_j)$ afin de trouver notre voisin le plus proche. $f(I_i)$ est le vecteur de caractéristiques de l'image envoyée à l'inférence, et $f(I_j)$ est le vecteur de caractéristiques d'une image dans l'ensemble d'apprentissage (la mémoire).

Nous affinons ensuite cette expérience en utilisant notre gagnant clair, DenseNet, Huang *et al.* (2016) (selon le tableau I-1 avec des informations de goulot d'étranglement regroupées et démontré dans la Figure I-1 (gauche) avec notre approche de correspondance de vecteurs de pixels ("disantangled pixel vectors", voir section 3.1)) et comparons les différentes approches de mise en commun (descripteurs) (mise en commun maximale, mise en commun moyenne, GeM, Radenović *et al.* (2018)) ainsi que l'aplatissement (flattening) et notre approche de vecteur de pixels décrite dans la section 3.1, illustrée à la Figure I-1 (droite), montrant que la correspondance des vecteurs de pixels conduit à la plus grande précision parmi les alternatives, en particulier pour DenseNet. Notez que la haute précision de la correspondance des vecteurs de pixels est due au compromis entre la mémoire et la puissance de calcul (par exemple, 7x7 = 49 pixels contre 1 descripteur global (et K = 10 pour notre nombre de voisins les plus proches)).

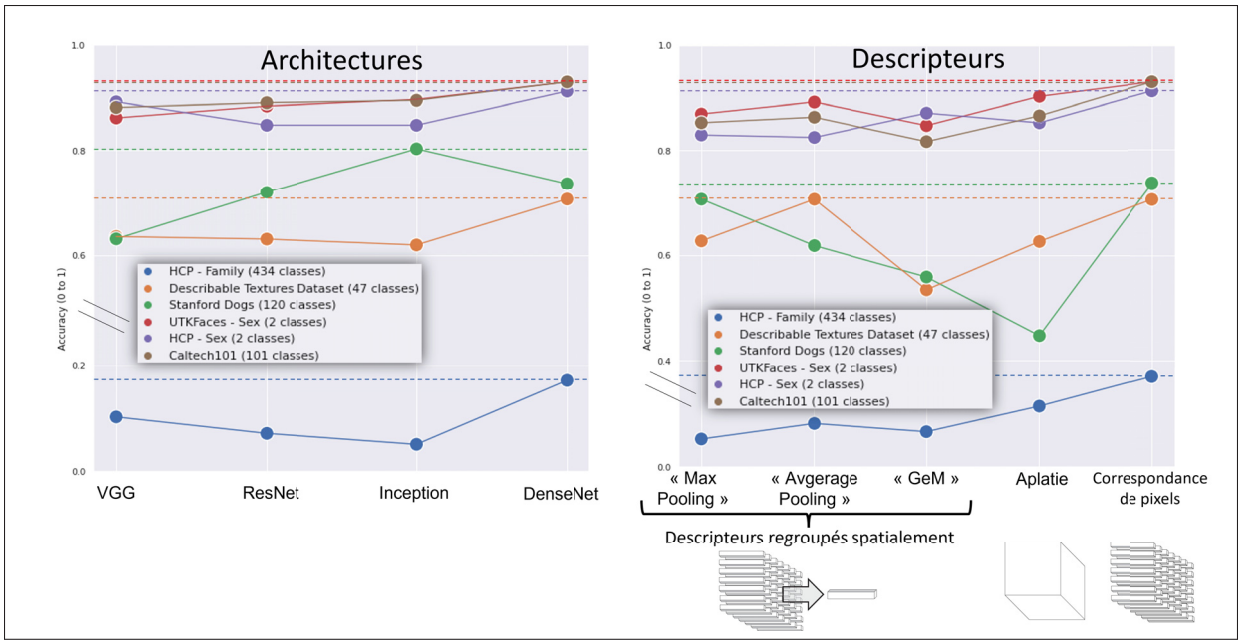

FIGURE-A I-1 Performances de classification de base des architectures (gauche) et des descripteurs (droite) sur six tâches de classification basées sur la mémoire avec ($K = 10$) voisins les plus proches. Les lignes pointillées indiquent les performances supérieures de l'architecture DenseNet, Huang *et al.* (2016), (gagnant dans 5 des 6 tâches) et des descripteurs de vecteurs de pixels (gagnant pour toutes les tâches). Image adaptée de Bouchard *et al.* (2022).

ANNEXE II

OPTIMISATIONS : PCA ET DOMAINE SPECTRAL

Un problème de notre approche de correspondance de pixel est la grande quantité de mémoire utilisée, malgré l'utilisation de la librairie Annoy, Bernhardsson (2015), nécessitant $7 \times 7 = 49$ vecteurs de pixels de 1×1920 dimensions pour chaque image de la base de données stockée en mémoire. Pareillement, comme décrit dans l'algorithme 3.2, nous devons faire $7 \times 7 = 49$ requêtes pour chaque image d'inférence plutôt qu'une, demandant beaucoup plus de temps.

Bien que notre but ait été principalement de générer des observations, nous avons également voulu étudier l'effet de la compression spatiale et par canal sur ces données, ce qui est fréquemment utilisé en machine learning Howley, Madden, O'Connell & Ryder (2005); Curran, Brys, Taylor & Smart (2015); Migenda, Möller & Schenck (2021). Ces premières expériences visent à compresser les informations par canal des vecteurs de pixels DenseNet, Huang *et al.* (2016), afin de réduire les besoins en mémoire et le temps d'inférence. Nous nous concentrons sur les tenseurs extraits du goulot d'étranglement de DenseNet, car ils sont vraiment puissants pour la classification, Lazreg (2020), et contiennent des informations de toutes les couches précédentes en raison de ses connexions de saut (skip connections) concaténées, comme le montre la figure II-1. Par exemple, un vecteur 1×1920 DenseNet-201 (201 couches) extrait contient des informations de tous les blocs de convolution précédents en raison de son architecture, les 896 premiers canaux provenant des premières couches (toutes les couches avant 169) et les 1024 derniers des 32 couches les plus profondes.

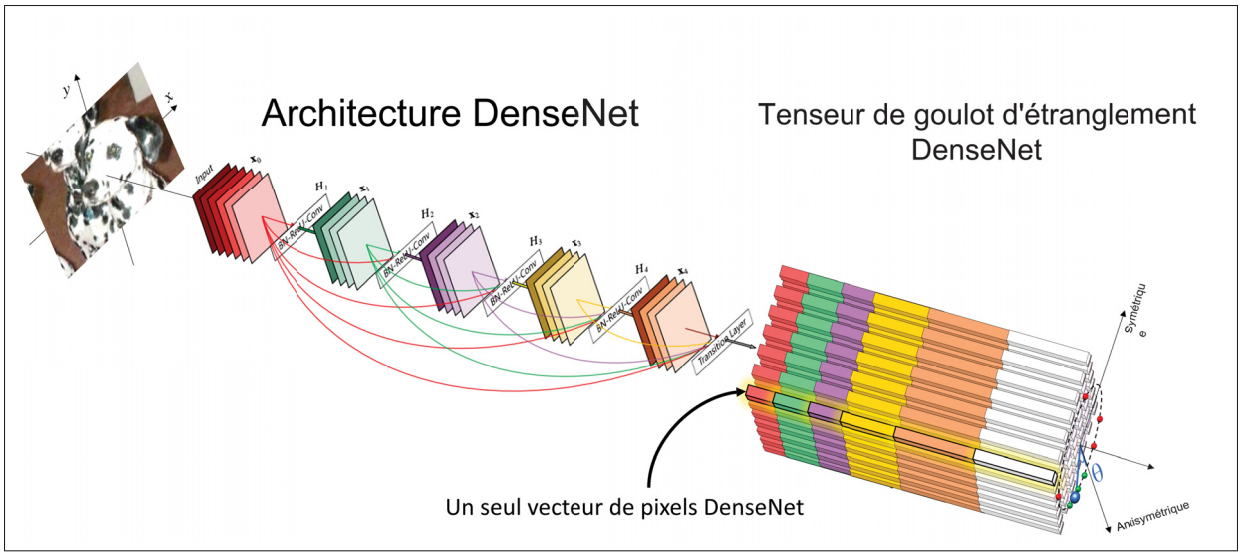

FIGURE-A II-1 DenseNet, Huang *et al.* (2016), architecture et visualisation de la concaténation des tenseurs extraits(nos tenseurs $7 \times 7 \times 1920$, comportant les vecteurs pixels de 1×1920). Image adaptée de Huang *et al.* (2016).

Le fait que l'information viennent de toutes les couches est très bénéfique. Il a été démontré que les réponses de filtre peu profonds améliorent les performances des tâches d'imagerie à grain fin, Azizpour *et al.* (2015), car elles contiennent des informations d'image localisées, Kaya *et al.* (2019), et sont également à la base de méthodes de correspondance de points clés invariants très puissantes telles que la transformation de caractéristiques invariantes à l'échelle (SIFT) Lowe (2004); Toews & Wells (2009). En revanche, les couches profondes des réseaux CNNs traditionels sont importantes pour les entités de classe générales et performent généralement mieux que l'information en début de réseau. Un meilleur compromis est l'accès à l'information sur l'ensemble du réseau (à tous les temps t), comme dans DenseNet, offrant un excellent équilibre pour la plupart des tâches, mais s'accompagne d'une charge de calcul plus lourde.

Nous avons utilisé l'algorithme PCA, F.R.S. (1901), pour extraire les informations les plus pertinentes par canal directement sur nos vecteurs de pixels, que nous appelons pixel-PCA. Pour une image particulière, pixel-PCA peut être vu comme des convolutions unidimensionnelles entre les vecteurs propres d PCA et le vecteur pixel ($7 \times 7 = 49$) de notre couche de goulot

d'étranglement. Pour tous les tests, nous avons entraîné la représentation PCA avec notre base d'entraînement uniquement et l'avons utilisé pour transformer notre base test.

Nous affichons les premiers vecteurs propres de l'algorithme PCA utilisé produits à l'aide de tous les vecteurs de pixels extraits de nos tenseurs DenseNet du goulot d'étranglement pour la base de données Caltech 101, Fei-Fei *et al.* (2006). Cette méthode de pixel-PCA est un excellent moyen pour comprendre la structure de l'information du goulot d'étranglement au niveau des canaux, alors que nous nous focalisons surtout sur l'espace spatial (x, y) dans les autres expériences.

La Figure II-2 montre la réponse moyenne pour les cinq premiers composants PCA pour les 101 classes de Caltech-101, Fei-Fei *et al.* (2006), et quelques classes spécifiques. Figure II-3 est la même représentation, mais affiche la moyenne de la variance de nos composants pixels-PCA pour la base de données complète (gauche) et des classes spécifiques (droite). La Figure II-4 montre les cinq premiers vecteurs propres utilisés par l'algorithme pour générer nos pixels-PCA de nos vecteurs de pixels extraits de Caltech 101. Nous remarquons ici que la première composante principale (vecteur propre 0) établit majoritairement les réponses périphériques, aux quatre coins extrêmes des cartes de caractéristiques de Densenet, tandis que les détails centraux de la carte de caractéristique sont déterminés via d'autres composantes principales (vecteurs propres 1, ..., N).

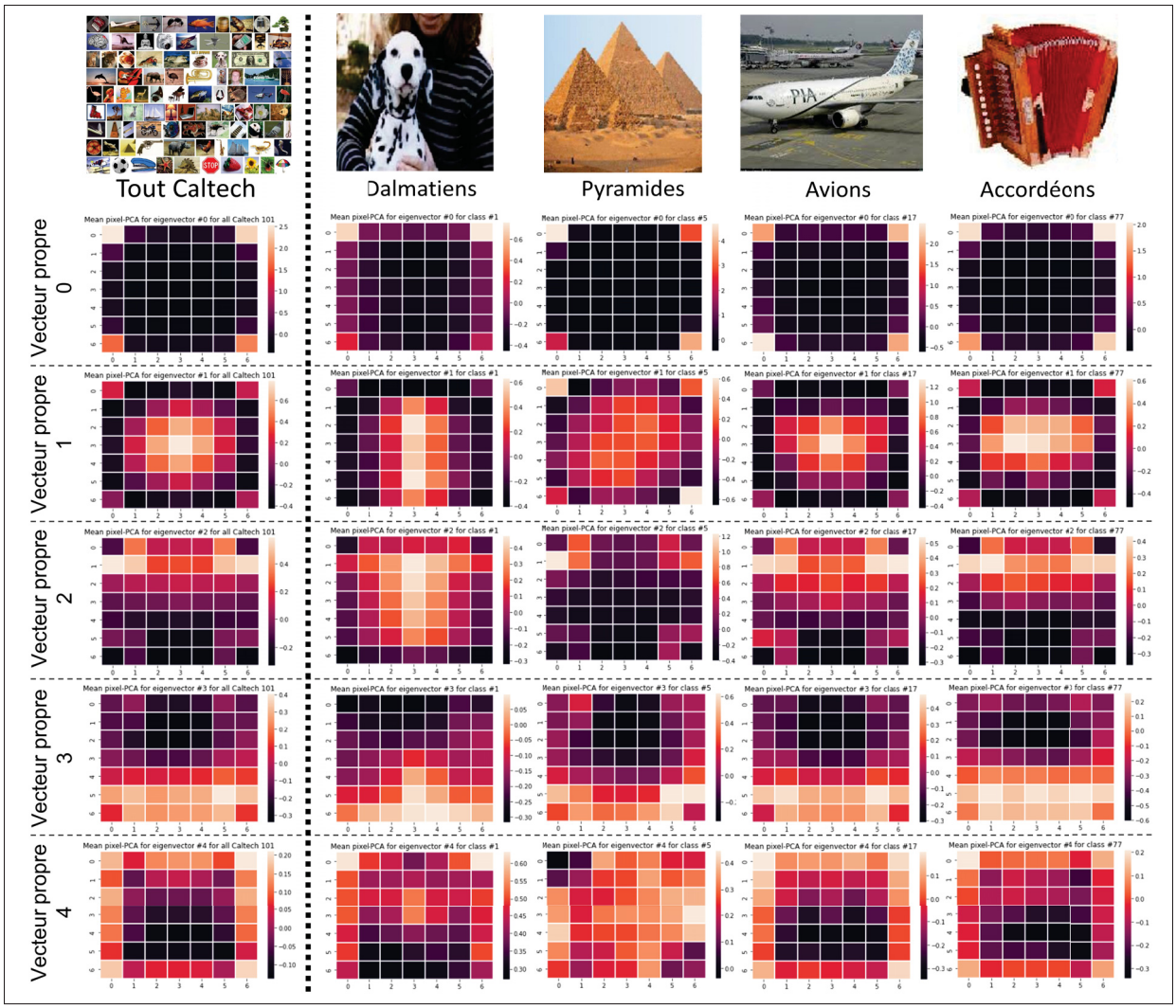

FIGURE-A II-2 Caltech-101, Fei-Fei *et al.* (2006), réponse moyenne du produit scalaire entre les cinq premiers vecteurs propres, nos pixels- PCA et nos vecteurs de pixels extraits de Caltech 101. À gauche se trouve la réponse moyenne de sur l'ensemble de la base test de Caltech-101 et à gauche la réponse moyenne de quatres classes spécifiques.

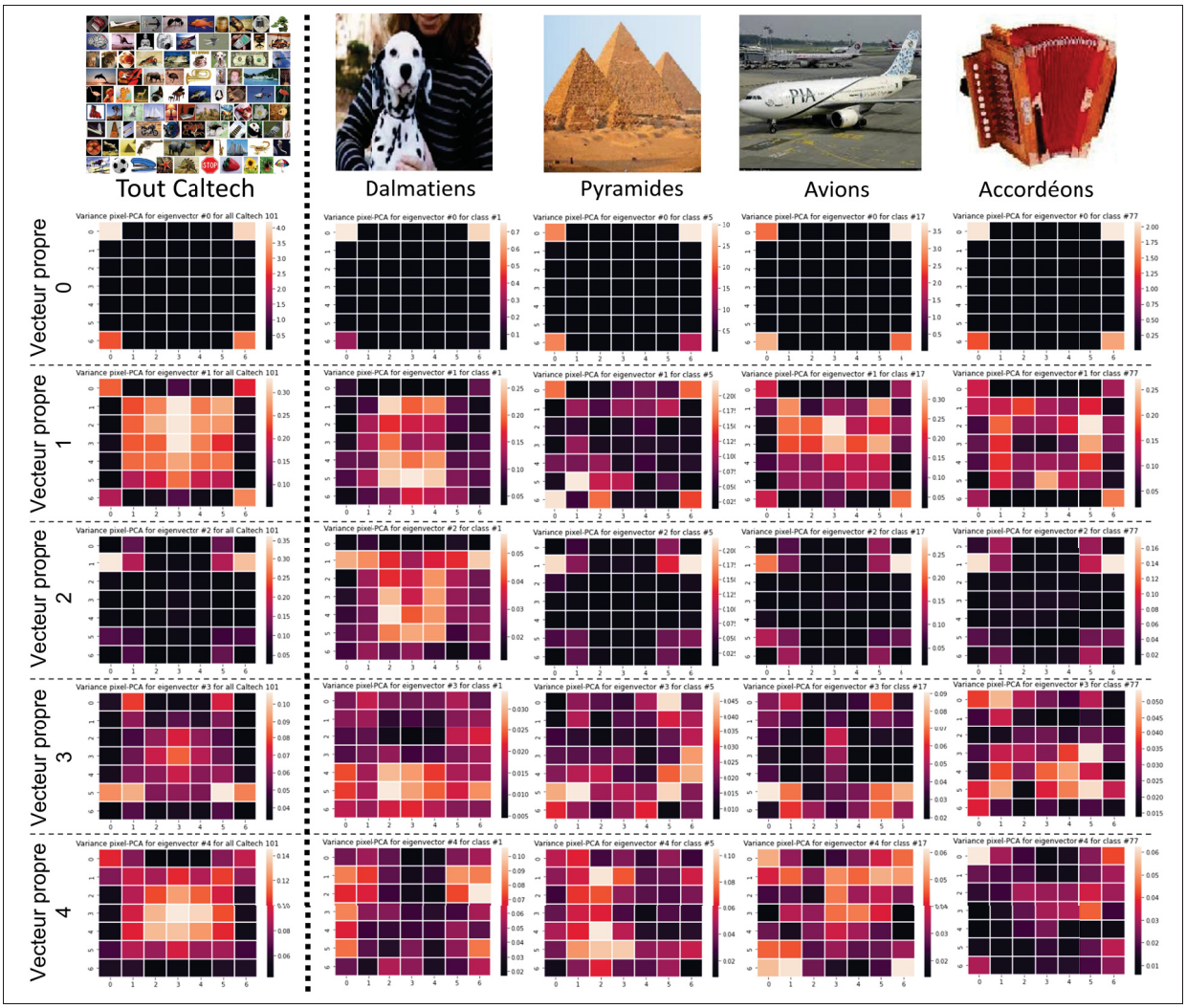

FIGURE-A II-3 Caltech 101, Fei-Fei *et al.* (2006), variance pour le produit scalaire entre les cinq premiers vecteurs propres, nos pixels- PCA et nos vecteurs de pixels extraits de Caltech 101. À gauche se trouve la variance moyenne sur l'ensemble de la base test de Caltech-101 et à gauche la variance moyenne de quatres classes spécifiques.

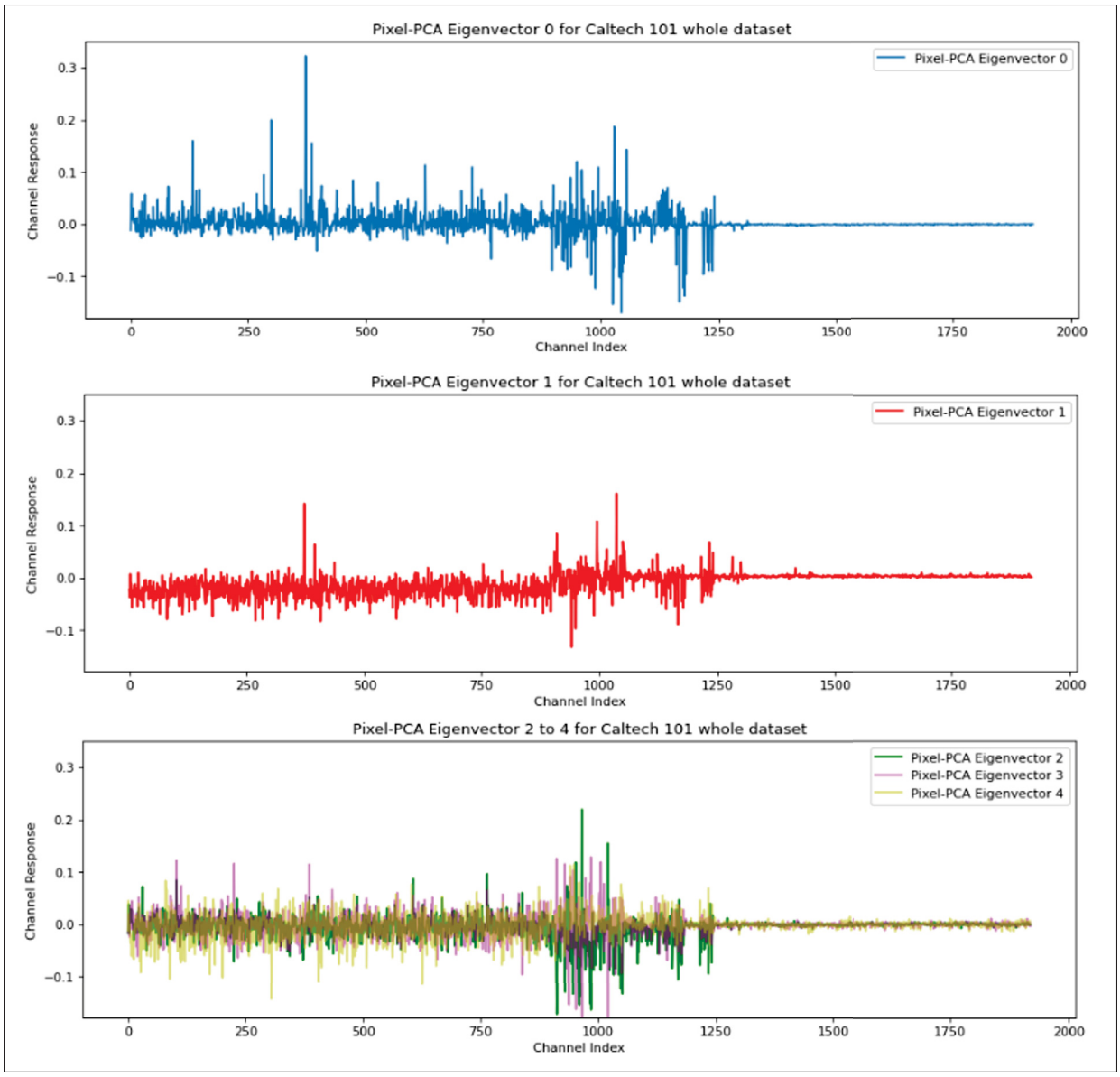

FIGURE-A II-4 Les cinq premiers vecteurs propres, pixels- PCA, de nos vecteurs de pixels extraits de Caltech 101 où la réponse est plus forte dans les canaux moins profonds et particulièrement pour certains canaux spécifiques, surtout pour le premier composant principal.

L'utilisation de PCA a permis de réduire la taille des canaux de nos pixels de $N = 1920$ à $d = 200$ avec une précision légèrement améliorée (de $90,2\%$ à $90,3\%$ sur Caltech-101, Fei-Fei *et al.* (2006)) en maximisant la variance de chaque pixel dans l'ordre décroissant des canaux, réduisant ainsi la taille de la mémoire d'un ordre de grandeur. Voir Figure II-5 pour une comparaison de

différents nombres de canaux utilisés dans l'algorithme PCA, lequel a été entraîné avec notre base d'entraînement uniquement et testé avec la base test de Caltech-101.

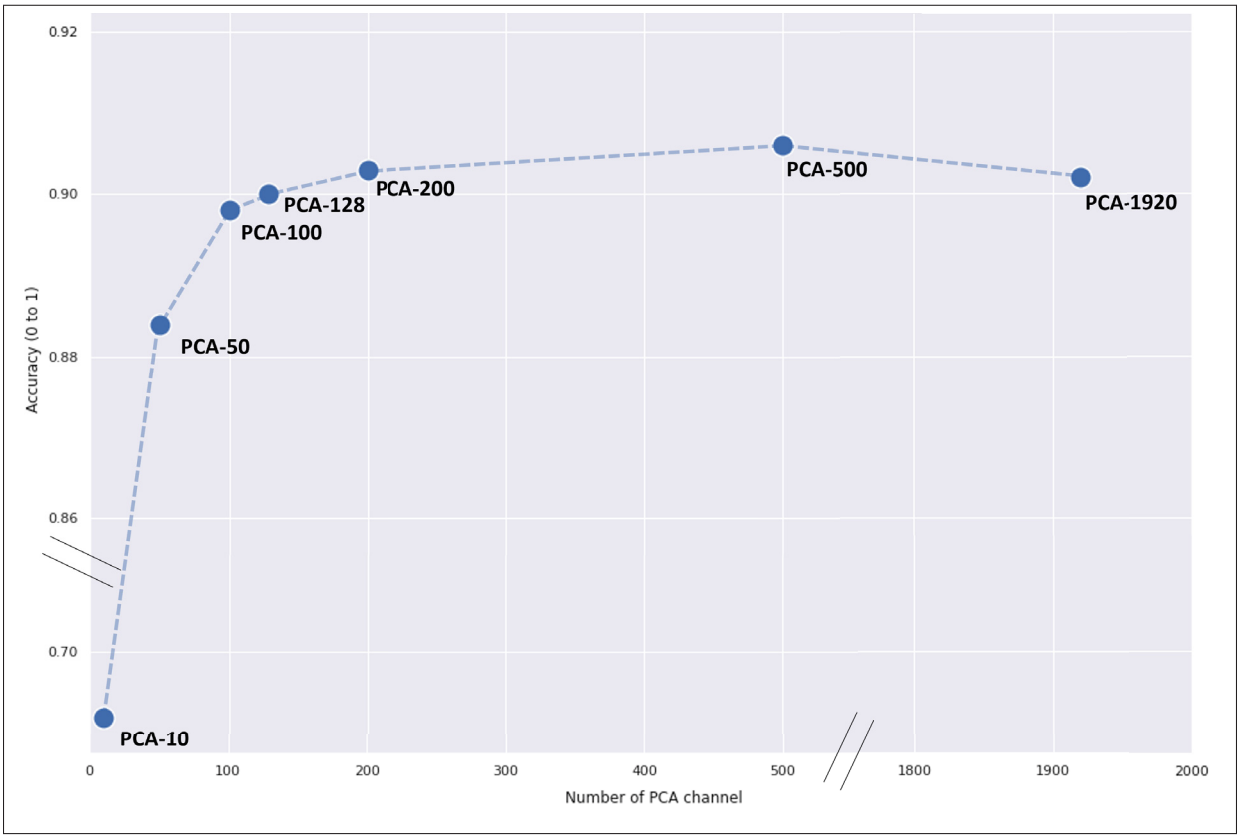

FIGURE-A II-5 Comparaison de configuration de nos pixels-PCA sur DenseNet, Huang *et al.* (2016), avec l'approche de correspondance de pixels préentraînée à pca-N canaux testée sur Caltech-101, Fei-Fei *et al.* (2006). Réduire le nombre de canaux en utilisant PCA aide à la classification jusqu'à un certain seuil (200 canaux), où la précision commence alors à chuter brutalement.

Notre deuxième intuition était d'utiliser le domaine spectral, ou domaine de Fourier, afin de réduire davantage les besoins de calcul en condensant les informations spatiales en quelques composants denses en informations, Transformée de Fourier Rapide (Fast Fourier Transform) (FFT), Nussbaumer (1981). Le domaine de Fourier (ou fréquentiel) est un puissant outil de traitement d'image utilisé pour décomposer une image en ses composantes sinusoïdales et cosinusoïdales, comme illustré sur la Figure II-6 (a) avec les coefficients de la transformée de Fourier discrète (DFT), où l'image d'entrée est l'équivalent du domaine spatial. Ici, la DFT est

simplement la transformée de Fourier échantillonnée pour les images réelles, ce qui est suffisant pour représenter nos images spatiales. La sortie DFT bidimensionnelle d'une image $N \times N$ est donnée par :

$$F(k,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) \exp^{i2\pi(\frac{km}{N}\frac{ln}{N})} \quad \text{(A II-1)}$$

où $f(m,n)$ est l'image dans le domaine spatial et le terme exponentiel est la fonction de base correspondant à chaque point (complexe) $F(k,l)$ dans le domaine de Fourier.

Ensuite, nous pouvons facilement extraire les composantes de phase (angle produit par le composant réel et imaginaire) et d'amplitude (magnitude du vecteur créé par les composants réels et imaginaires) de l'image complexe produite. Il est bien connu que la phase encode la quasi-totalité des informations nécessaires sur l'image, comme représentée sur la Figure II-6 (b), où nous pouvons toujours voir clairement l'image lors de sa reconstruction en utilisant sa propre phase et la magnitude d'une autre image. La reconstruction en une image spatiale est simplement l'inverse de l'équation A II-1 :

$$f(m,n) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l) \exp^{i2\pi(\frac{ka}{N}\frac{lb}{N})} \quad \text{(A II-2)}$$

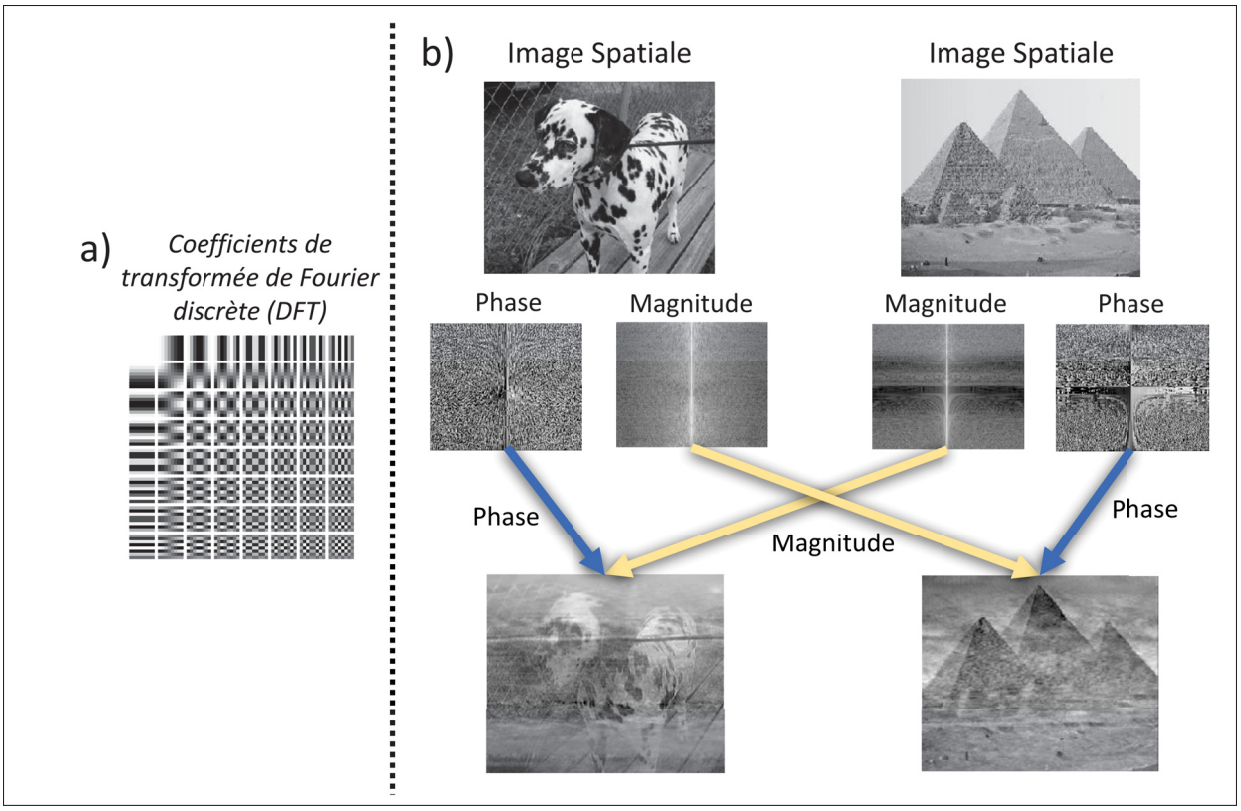

FIGURE-A II-6 a) Exemple de coefficients DFT 8×8 utilisé pour construire une image DFT à partir d'une image spatiale. b) Images spatiales transformées en images DFT avec phase et amplitude affichées et reconstruites en utilisant les composantes d'amplitude de l'autre image. Notez comment le spectre de phase θ semble contenir la quasi-totalité de la perception visuelle, alors que la magnitude ne semble pas liée.

Après des recherches plus approfondies, nous avons constaté que, comme le montre la figure II-7, cinq composantes de Fourier utilisant uniquement des valeurs de phase pouvaient être approximativement aussi informatives que les 49 pixels du domaine spatial. Nous sommes ainsi passés de l'utilisation de tenseurs de taille $(7 \times 7 \times 1920)$ à des tenseurs de taille (5×200) combinant FFT et PCA avec une perte de précision de 1.8%, soit de $90,2\%$ à $88,4\%$, pour 94 fois moins d'informations nécessaires. La Figure II-7 montre certaines des meilleures configurations de coefficients que nous avons testé sur Caltech 101, Fei-Fei *et al.* (2006), parmi tant d'autres, où nous voyons clairement que la plupart des informations sont contenues dans peu de composants de phase même dans ces cartes d'activation (comparé à la représentation

de la phase avec l'image originale, comme le montre la Figure II-6 (b)). Même si ces résultats sont encourageants, des travaux futurs sont nécessaires pour mieux comprendre comment le pixel-PCA et la représentation spectrale peuvent être optimisés avec notre approche. Au lieu de cela, nous nous concentrons sur la construction d'une architecture adaptée qui implémente efficacement la représentation $U(1)$, en commençant par un schéma d'encodeur-décodeur U-Net inspiré de ses récents succès avec les Auto-encodeurs variationnels (Variational Autoencoders) (VAE)s et les GANs en vision par ordinateur, Goodfellow *et al.* (2014); Isola *et al.* (2017); Karras *et al.* (2019); Park *et al.* (2020); Karras *et al.* (2020a,b).

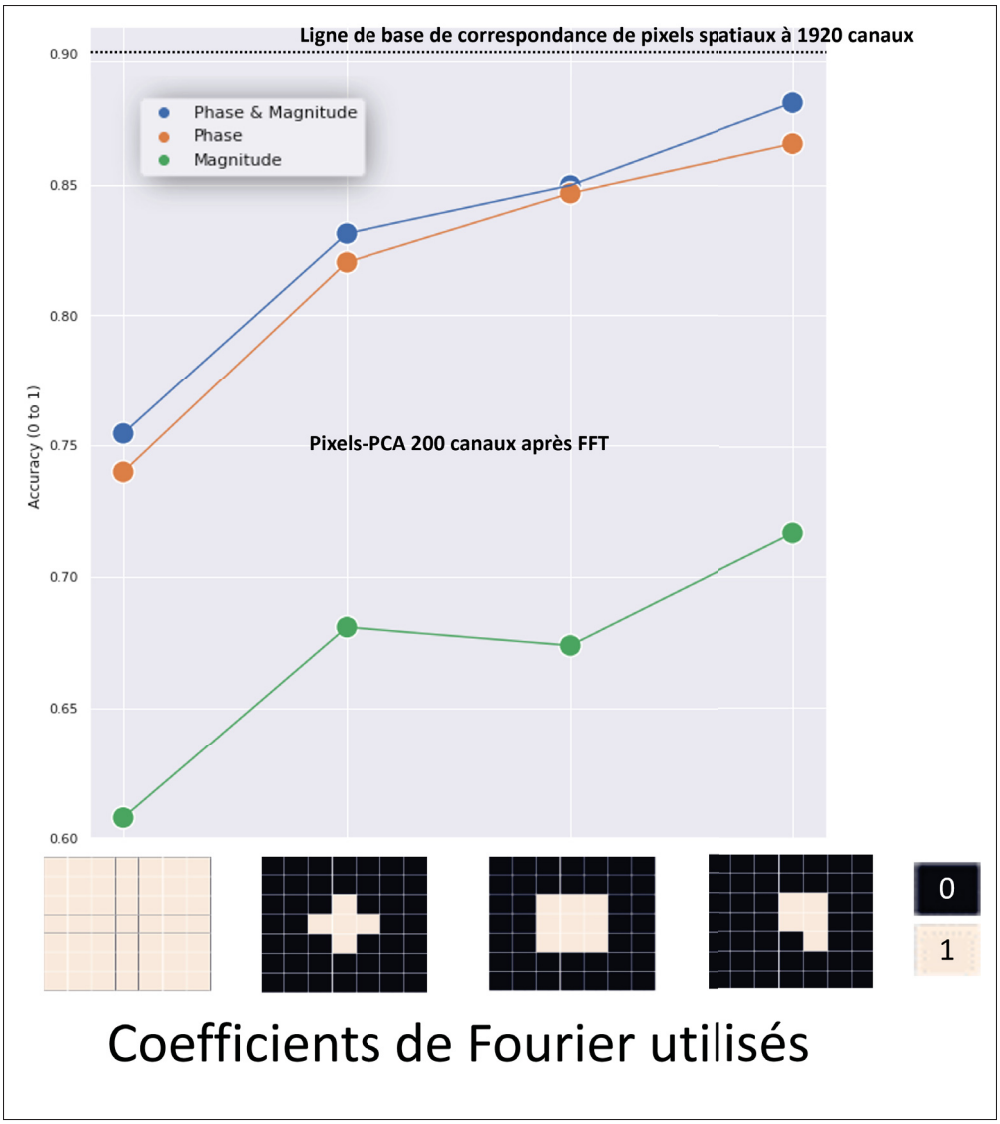

FIGURE-A II-7 Comparaison de quatre configurations de coefficients que nous avons testées sur Caltech 101, Fei-Fei *et al.* (2006) avec l'approche d'appariement de pixels à 200- PCA-canaux en utilisant la même configuration que dans nos expériences précédentes. Ci-dessous, les images spectrales avec les coefficients activés et désactivés utilisées pour les expériences dans notre base de données et au moment de l'inférence. Nous comparons également l'effet de l'utilisation de la phase, de l'amplitude ou des deux lors de l'inférence.

BIBLIOGRAPHIE

Aad, G., Abajyan, T., Abbott, B., Abdallah, J., Khalek, S. A., Abdelalim, A. A., Aben, R., Abi, B., Abolins, M., AbouZeid, O. et al. (2012). Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Physics Letters B*, 716(1), 1–29.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. & Zheng, X. [Software available from tensorflow.org]. (2015). TensorFlow : Large-Scale Machine Learning on Heterogeneous Systems. Repéré à https://www.tensorflow.org/.

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv :1803.08375*.

Arandjelović, R., Gronat, P., Torii, A., Pajdla, T. & Sivic, J. (2016). NetVLAD : CNN architecture for weakly supervised place recognition.

Arjovsky, M., Shah, A. & Bengio, Y. (2016). Unitary evolution recurrent neural networks. *International Conference on Machine Learning*, pp. 1120–1128.

Azizpour, H., Razavian, A. S., Sullivan, J., Maki, A. & Carlsson, S. (2015). Factors of transferability for a generic convnet representation. *IEEE transactions on pattern analysis and machine intelligence*, 38(9), 1790–1802.

Bardes, A., Ponce, J. & LeCun, Y. (2021). Vicreg : Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv :2105.04906*.

Bartolomei, H., Kumar, M., Bisognin, R., Marguerite, A., Berroir, J.-M., Bocquillon, E., Placais, B., Cavanna, A., Dong, Q., Gennser, U. et al. (2020). Fractional statistics in anyon collisions. *Science*, 368(6487), 173–177.

Ben-Yishai, R., Bar-Or, R. L. & Sompolinsky, H. (1995). Theory of orientation tuning in visual cortex. *Proceedings of the National Academy of Sciences*, 92(9), 3844–3848.

Berman, M., Jégou, H., Vedaldi, A., Kokkinos, I. & Douze, M. (2019). Multigrain : a unified image embedding for classes and instances. *arXiv preprint arXiv :1902.05509*.

Bernhardsson, E. (2015). Annoy on github. GitHub. Repéré à https://github.com/spotify/annoy.

Boffetta, G., Monasson, R. & Zecchina, R. (1993). Symmetry breaking in nonmonotonic neural networks. *Journal of Physics A : Mathematical and General*, 26(12), L507.

Bose. (1924). Plancks Gesetz und Lichtquantenhypothese. *Zeitschrift für Physik*, 26, 178-181.

Bouchard, L.-F., Lazreg, M. B. & Toews, M. (2022). U (1) Symmetry-breaking Observed in Generic CNN Bottleneck Layers. *arXiv preprint arXiv :2206.02220*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877–1901.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. & Zagoruyko, S. (2020). End-to-end object detection with transformers. *European Conference on Computer Vision*, pp. 213–229.

Chang, B., Chen, M., Haber, E. & Chi, E. H. (2019). AntisymmetricRNN : A dynamical system view on recurrent neural networks. *arXiv preprint arXiv :1902.09689*.

Changpinyo, S., Sandler, M. & Zhmoginov, A. (2017). The power of sparsity in convolutional neural networks. *arXiv preprint arXiv :1702.06257*.

Chauvin, L., Kumar, K., Desrosiers, C., Wells III, W. & Toews, M. (2021). Efficient Pairwise Neuroimage Analysis using the Soft Jaccard Index and 3D Keypoint Sets. *arXiv preprint arXiv :2103.06966*.

Chen, R. T., Rubanova, Y., Bettencourt, J. & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F. & Huang, J.-B. (2019). A closer look at few-shot classification. *arXiv preprint arXiv :1904.04232*.

Chollet, F. (2017). Xception : Deep learning with depthwise separable convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., & Vedaldi, A. (2014). Describing Textures in the Wild. *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Cimpoi, M., Maji, S., Kokkinos, I. & Vedaldi, A. (2016). Deep filter banks for texture recognition, description, and segmentation. *International Journal of Computer Vision*, 118(1), 65–94.

Cohen, T., Weiler, M., Kicanaoglu, B. & Welling, M. (2019). Gauge equivariant convolutional networks and the icosahedral CNN. *International Conference on Machine Learning*, pp. 1321–1330.

Cover, T. & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1), 21–27.

Curran, W., Brys, T., Taylor, M. & Smart, W. (2015). Using PCA to efficiently represent state spaces. *arXiv preprint arXiv :1505.00322*.

Deng, J., Dong, W., Socher, R., Li, L., Kai Li & Li Fei-Fei. (2009). ImageNet : A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248-255. doi : 10.1109/CVPR.2009.5206848.

Dhillon, G. S., Chaudhari, P., Ravichandran, A. & Soatto, S. (2019). A Baseline for Few-Shot Image Classification. *CoRR*, abs/1909.02729. Repéré à http://arxiv.org/abs/1909.02729.

Dissanayake, M. & Phan-Thien, N. (1994). Neural-network-based approximations for solving partial differential equations. *communications in Numerical Methods in Engineering*, 10(3), 195–201.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E. & Darrell, T. (2014). Decaf : A deep convolutional activation feature for generic visual recognition. *International conference on machine learning*, pp. 647–655.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2020). An image is worth 16x16 words : Transformers for image recognition at scale. *arXiv preprint arXiv :2010.11929*.

Dzhezyan, G. & Cecotti, H. (2019). Symnet : Symmetrical filters in convolutional neural networks. *arXiv preprint arXiv :1906.04252*.

Eisenberg, L. (2008). The Tree of Life. Repéré à https://www.evogeneao.com.

Fei-Fei, L., Fergus, R. & Perona, P. (2006). One-Shot Learning of Object Categories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4), 594–611.

Finn, C., Abbeel, P. & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *International conference on machine learning*, pp. 1126–1135.

F.R.S., K. P. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11), 559-572. doi : 10.1080/14786440109462720.

Goldstone, J. (1961). Field theories with «Superconductor» solutions. *Il Nuovo Cimento (1955-1965)*, 19(1), 154–164.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Gordo, A., Almazán, J., Revaud, J. & Larlus, D. (2016). Deep image retrieval : Learning global representations for image search. *European conference on computer vision*, pp. 241–257.

Guan, B., Zhang, J., Sethares, W. A., Kijowski, R. & Liu, F. (2021). Spectral domain convolutional neural network. *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2795–2799.

Guo, Y., Shi, H., Kumar, A., Grauman, K., Rosing, T. & Feris, R. S. (2018). SpotTune : Transfer Learning through Adaptive Fine-tuning. *CoRR*, abs/1811.08737. Repéré à http://arxiv.org/abs/1811.08737.

Guo, Y., Codella, N. C., Karlinsky, L., Codella, J. V., Smith, J. R., Saenko, K., Rosing, T. & Feris, R. (2020). A broader study of cross-domain few-shot learning. *European Conference on Computer Vision*, pp. 124–141.

Haase, D. & Amthor, M. (2020). Rethinking Depthwise Separable Convolutions : How Intra-Kernel Correlations Lead to Improved MobileNets. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14600–14609.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y. et al. (2020). A Survey on Visual Transformer. *arXiv preprint arXiv :2012.12556*.

Han, Y. & Hong, B.-W. (2021). Deep learning based on fourier convolutional neural network incorporating random kernels. *Electronics*, 10(16), 2004.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Henaff, M., Bruna, J. & LeCun, Y. (2015). Deep convolutional networks on graph-structured data. *arXiv preprint arXiv :1506.05163*.

Higgs, P. W. (1964). Broken symmetries, massless particles and gauge fields. *Phys. Lett.*, 12, 132–133.

Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N. & Peste, A. (2021). Sparsity in Deep Learning : Pruning and growth for efficient inference and training in neural networks. *J. Mach. Learn. Res.*, 22(241), 1–124.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. & Adam, H. (2017). Mobilenets : Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv :1704.04861*.

Howley, T., Madden, M. G., O'Connell, M.-L. & Ryder, A. G. (2005). The effect of principal component analysis on machine learning accuracy with high dimensional spectral data. *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pp. 209–222.

Hu, J., Shen, L. & Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141.

Huang, G., Liu, Z. & Weinberger, K. Q. (2016). Densely Connected Convolutional Networks. *CoRR*, abs/1608.06993. Repéré à http://arxiv.org/abs/1608.06993.

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y. & Liu, W. (2019). Ccnet : Criss-cross attention for semantic segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 603–612.

Hutter, M. (2020). On representing (anti) symmetric functions. *arXiv preprint arXiv :2007.15298*.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J. & Keutzer, K. (2016). SqueezeNet : AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size.

Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134.

Jaiswal, M. S., Kang, B., Lee, J. & Cho, M. (2020). MUTE : Inter-class Ambiguity Driven Multi-hot Target Encoding for Deep Neural Network Design. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 754–755.

Jiang, W., Trulls, E., Hosang, J., Tagliasacchi, A. & Yi, K. M. (2021). COTR : Correspondence Transformer for Matching Across Images. *arXiv preprint arXiv :2103.14167*.

Jing, L., Shen, Y., Dubcek, T., Peurifoy, J., Skirlo, S., LeCun, Y., Tegmark, M. & Soljačić, M. (2017). Tunable efficient unitary neural networks (eunn) and their application to rnns. *International Conference on Machine Learning*, pp. 1733–1741.

Karras, T., Laine, S. & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410.

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J. & Aila, T. (2020a). Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33, 12104–12114.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J. & Aila, T. (2020b). Analyzing and improving the image quality of stylegan. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119.

Kaya, Y., Hong, S. & Dumitras, T. (2019). Shallow-Deep Networks : Understanding and Mitigating Network Overthinking.

Khosla, A., Jayadevaprakash, N., Yao, B. & Fei-Fei, L. (2011, June). Novel Dataset for Fine-Grained Image Categorization. *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*.

Kiani, B., Balestriero, R., Lecun, Y. & Lloyd, S. (2022). projUNN : efficient method for training deep networks with unitary matrices. *arXiv preprint arXiv :2203.05483*.

Kim, D., Bargal, S., Zhang, J. & Sclaroff, S. (2020). Multi-way encoding for robustness. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1352–1360.

Kim, S. S., Rouault, H., Druckmann, S. & Jayaraman, V. (2017). Ring attractor dynamics in the Drosophila central brain. *Science*, 356(6340), 849–853.

Kingma, D. & Ba, J. (2014). Adam : A Method for Stochastic Optimization. *International Conference on Learning Representations*.

Kipf, T. N. & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv :1609.02907*.

Klus, S., Gelß, P., Nüske, F. & Noé, F. (2021). Symmetric and antisymmetric kernels for machine learning problems in quantum physics and chemistry. *Machine Learning : Science and Technology*, 2(4), 045016. doi : 10.1088/2632-2153/ac14ad.

Koch, G. R. (2015). Siamese Neural Networks for One-Shot Image Recognition.

Kornblith, S., Shlens, J. & Le, Q. V. (2019). Do Better ImageNet Models Transfer Better ?

Krause, J., Stark, M., Deng, J. & Fei-Fei, L. (2013). 3D Object Representations for Fine-Grained Categorization. *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*.

Krizhevsky, A., Nair, V. & Hinton, G. CIFAR-100 (Canadian Institute for Advanced Research). Repéré à http://www.cs.toronto.edu/~kriz/cifar.html.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012a). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25. Repéré à https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012b). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Lazreg, M. B. (2020). Recherche de l'information dans les réseaux de neurones convolutifs pré-entraînés. Repéré à http://www.matthewtoews.com/thesis/Thesis_Mohsen_BenLazreg_2020.pdf.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4), 541-551. doi : 10.1162/neco.1989.1.4.541.

Lee, K., Maji, S., Ravichandran, A. & Soatto, S. (2019). Meta-learning with differentiable convex optimization. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10657–10665.

Liang, H., Bai, H., Hu, K. & Lv, X. (2021). Polarized skylight orientation determination artificial neural network. *arXiv preprint arXiv :2107.02328*.

Liu, S. & Luk, W. (2020). Optimizing Fully Spectral Convolutional Neural Networks on FPGA. *2020 International Conference on Field-Programmable Technology (ICFPT)*, pp. 39–47.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. & Guo, B. (2021). Swin transformer : Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022.

Long, J., Shelhamer, E. & Darrell, T. (2014). Fully Convolutional Networks for Semantic Segmentation. *CoRR*, abs/1411.4038. Repéré à http://arxiv.org/abs/1411.4038.

Long, M., Cao, Y., Wang, J. & Jordan, M. (2015). Learning transferable features with deep adaptation networks. *International conference on machine learning*, pp. 97–105.

Loshchilov, I. & Hutter, F. (2016). SGDR : Stochastic Gradient Descent with Restarts. *CoRR*, abs/1608.03983. Repéré à http://arxiv.org/abs/1608.03983.

Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60, 91-. doi : 10.1023/B:VISI.0000029664.99615.94.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M. B. & Vedaldi, A. (2013). Fine-Grained Visual Classification of Aircraft. *CoRR*, abs/1306.5151. Repéré à http://arxiv.org/abs/1306.5151.

Malfliet, W. (2004). The tanh method : a tool for solving certain classes of nonlinear evolution and wave equations. *Journal of Computational and Applied Mathematics*, 164, 529–541.

Martínez-Plumed, F., Castellano, D., Monserrat-Aranda, C. & Hernández-Orallo, J. (2022). When ai difficulty is easy : The explanatory power of predicting irt difficulty. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7), 7719–7727.

Mettes, P., van der Pol, E. & Snoek, C. (2019). Hyperspherical prototype networks. *Advances in neural information processing systems*, 32.

Mhammedi, Z., Hellicar, A., Rahman, A. & Bailey, J. (2017). Efficient orthogonal parametrisation of recurrent neural networks using householder reflections. *International Conference on Machine Learning*, pp. 2401–2409.

Migenda, N., Möller, R. & Schenck, W. (2021). Adaptive dimensionality reduction for neural network-based online principal component analysis. *PloS one*, 16(3), e0248896.

Minsky, M. (1961). *Microscopy apparatus n°US3013467A*. United States.

Monasson, R. & O'Kane, D. (1994). Domains of solutions and replica symmetry breaking in multilayer neural networks. *EPL (Europhysics Letters)*, 27(2), 85.

Nair, V. & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on International Conference on Machine Learning*, (ICML'10), 807–814.

Narayanan, D., Shoeybi, M., Casper, J., LeGresley, P., Patwary, M., Korthikanti, V., Vainbrand, D., Kashinkunti, P., Bernauer, J., Catanzaro, B. et al. (2021). Efficient large-scale language model training on GPU clusters using megatron-LM. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–15.

Nguyen, V. N., Løkse, S., Wickstrøm, K., Kampffmeyer, M., Roverso, D. & Jenssen, R. (2020). Sen : A novel feature normalization dissimilarity measure for prototypical few-shot learning networks. *European Conference on Computer Vision*, pp. 118–134.

Nilsback, M.-E. & Zisserman, A. (2008, Dec). Automated Flower Classification over a Large Number of Classes. *Indian Conference on Computer Vision, Graphics and Image Processing*.

Noh, H., Araujo, A., Sim, J., Weyand, T. & Han, B. (2017). Large-scale image retrieval with attentive deep local features. *Proceedings of the IEEE international conference on computer vision*, pp. 3456–3465.

Nussbaumer, H. J. (1981). The fast Fourier transform. Dans *Fast Fourier Transform and Convolution Algorithms* (pp. 80–111). Springer.

Pan, S. J. & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345-1359. doi : 10.1109/TKDE.2009.191.

Papernot, N. & McDaniel, P. (2018). Deep k-nearest neighbors : Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv :1803.04765*.

Park, J., Woo, S., Lee, J.-Y. & Kweon, I. S. (2018). Bam : Bottleneck attention module. *arXiv preprint arXiv :1807.06514*.

Park, T., Zhu, J.-Y., Wang, O., Lu, J., Shechtman, E., Efros, A. A. & Zhang, R. (2020). Swapping Autoencoder for Deep Image Manipulation. *Advances in Neural Information Processing Systems*.

Parkhi, O. M., Vedaldi, A., Zisserman, A. & Jawahar, C. V. (2012). Cats and Dogs. *IEEE Conference on Computer Vision and Pattern Recognition*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. & Chintala, S. (2019). PyTorch : An Imperative Style, High-Performance Deep Learning Library. Dans Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. & Garnett, R. (Éds.), *Advances in Neural Information Processing Systems 32* (pp. 8024–8035). Curran Associates, Inc. Repéré à http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

PBS NOVA, F. (2019). Standard Model of Elementary Particles. Repéré à https://commons.wikimedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg.

Perez, L. & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv :1712.04621*.

Quattoni, A. & Torralba, A. (2009). Recognizing indoor scenes. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 413-420. doi : 10.1109/CVPR.2009.5206537.

Radenović, F., Tolias, G. & Chum, O. (2018). Fine-tuning CNN image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7), 1655–1668.

Raissi, M., Perdikaris, P. & Karniadakis, G. E. (2019). Physics-informed neural networks : A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707.

Ravi, S. & Larochelle, H. (2016). Optimization as a model for few-shot learning.

Razavian, A. S., Sullivan, J., Carlsson, S. & Maki, A. (2016). Visual Instance Retrieval with Deep Convolutional Networks.

Rizvi, S. M., Ab Rahman, A. A.-H., Khalil-Hani, M. & Ayat, S. O. (2021). A Low-complexity Complex-valued Activation Function for Fast and Accurate Spectral Domain Convolutional Neural Network. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, 9(1), 173–184.

Rodríguez, P., Bautista, M. A., Gonzàlez, J. & Escalera, S. (2018). Beyond one-hot encoding : Lower dimensional target embedding. *Image and Vision Computing*, 75, 21–31.

Ronneberger, O., Fischer, P. & Brox, T. (2015). U-net : Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241.

Rosenblatt, F. (1958). The Perceptron : A Probabilistic Model for Information Storage and Organization in The Brain. *Psychological Review*, 65–386.

Roska, T., Chua, L. O., Wolf, D., Kozek, T., Tetzlaff, R. & Puffer, F. (1995). Simulating nonlinear waves and partial differential equations via CNN. I. Basic techniques. *IEEE Transactions on Circuits and Systems I : Fundamental Theory and Applications*, 42(10), 807–815.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323, 533-536.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. (2018). Mobilenetv2 : Inverted residuals and linear bottlenecks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520.

Scheidegger, F., Istrate, R., Mariani, G., Benini, L., Bekas, C. & Malossi, C. (2021). Efficient image dataset classification difficulty estimation for predicting deep-learning accuracy. *The Visual Computer*, 37(6), 1593–1610.

Sedghi, H., Gupta, V. & Long, P. M. (2018). The singular values of convolutional layers. *arXiv preprint arXiv :1805.10408*.

Seelig, J. & Jayaraman, V. (2015). Neural dynamics for landmark orientation and angular path integration. *Nature*, 521, 186-91. doi : 10.1038/nature14446.

Sharif Razavian, A., Azizpour, H., Sullivan, J. & Carlsson, S. (2014). CNN features off-the-shelf : an astounding baseline for recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813.

Shorten, C. & Khoshgoftaar, T. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6. doi : 10.1186/s40537-019-0197-0.

Simard, P. Y., Steinkraus, D., Platt, J. C. et al. (2003). Best practices for convolutional neural networks applied to visual document analysis. *Icdar*, 3(2003).

Simonyan, K. & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition.

Singla, S. & Feizi, S. (2021). Skew orthogonal convolutions. *International Conference on Machine Learning*, pp. 9756–9766.

Smidt, T. E., Geiger, M. & Miller, B. K. (2021). Finding symmetry breaking order parameters with Euclidean neural networks. *Physical Review Research*, 3(1), L012002.

Snell, J., Swersky, K. & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Su, J., Vargas, D. V. & Sakurai, K. (2019). One Pixel Attack for Fooling Deep Neural Networks. *IEEE Transactions on Evolutionary Computation*, 23(5), 828-841. doi : 10.1109/TEVC.2019.2890858.

Sui, X., Wu, Q., Liu, J., Chen, Q. & Gu, G. (2020). A review of optical neural networks. *IEEE Access*, 8, 70773–70783.

Sun, J., Shen, Z., Wang, Y., Bao, H. & Zhou, X. (2021). LoFTR : Detector-Free Local Feature Matching with Transformers. *arXiv preprint arXiv :2104.00680*.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. & Hospedales, T. M. (2018). Learning to compare : Relation network for few-shot learning. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.

Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *Thirty-first AAAI conference on artificial intelligence*.

Tan, M. & Le, Q. V. (2019). EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks. *CoRR*, abs/1905.11946. Repéré à http://arxiv.org/abs/1905.11946.

Tanaka, H. & Kunin, D. (2021). Noether's Learning Dynamics : Role of Symmetry Breaking in Neural Networks. *Advances in Neural Information Processing Systems*, 34.

Tang, Y., Han, K., Guo, J., Xu, C., Li, Y., Xu, C. & Wang, Y. (2021). An image patch is a wave : Phase-aware vision mlp. *arXiv preprint arXiv :2111.12294*.

Tang, Y., Han, K., Guo, J., Xu, C., Li, Y., Xu, C. & Wang, Y. (2022). An image patch is a wave : Phase-aware vision mlp. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10935–10944.

Tian, C., Xu, Y., Zuo, W., Lin, C.-W. & Zhang, D. (2021). Asymmetric CNN for image superresolution. *IEEE Transactions on Systems, Man, and Cybernetics : Systems*.

Toews, M. & Wells, W. (2009, 06). SIFT-Rank : Ordinal description for invariant feature correspondence. 0, 172-177. doi : 10.1109/CVPRW.2009.5206849.

Tolias, G., Sicre, R. & Jégou, H. (2016). Particular object retrieval with integral max-pooling of CNN activations.

Trockman, A. & Kolter, J. Z. (2021). Orthogonalizing convolutional layers with the cayley transform. *arXiv preprint arXiv :2104.07167*.

Tseng, H.-Y., Lee, H.-Y., Huang, J.-B. & Yang, M.-H. (2020). Cross-domain few-shot classification via learned feature-wise transformation. *arXiv preprint arXiv :2001.08735*.

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E. & Ugurbil, K. (2013). The WU-Minn Human Connectome Project : An overview. *NeuroImage*, 80, 62-79. doi : https://doi.org/10.1016/j.neuroimage.2013.05.041. Mapping the Connectome.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017a). Attention Is All You Need. *CoRR*, abs/1706.03762. Repéré à http://arxiv.org/abs/1706.03762.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017b). Attention is all you need. *arXiv preprint arXiv :1706.03762*.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D. et al. (2016). Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

Wang, J., Zhou, F., Wen, S., Liu, X. & Lin, Y. (2017). Deep metric learning with angular loss. *Proceedings of the IEEE international conference on computer vision*, pp. 2593–2601.

Wang, W., Sun, Y., Eriksson, B., Wang, W. & Aggarwal, V. (2018a). Wide compression : Tensor ring nets. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9329–9338.

Wang, X., Girshick, R., Gupta, A. & He, K. (2018b). Non-local neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803.

Wang, Y., Chao, W., Weinberger, K. Q. & van der Maaten, L. (2019). SimpleShot : Revisiting Nearest-Neighbor Classification for Few-Shot Learning. *CoRR*, abs/1911.04623. Repéré à http://arxiv.org/abs/1911.04623.

Watanabe, T. & Wolf, D. F. (2021). Image classification in frequency domain with 2SReLU : a second harmonics superposition activation function. *Applied Soft Computing*, 112, 107851.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S. & Perona, P. (2010). *Caltech-UCSD Birds 200* (Rapport n°CNS-TR-201). Repéré à /se3/wp-content/uploads/2014/09/WelinderEtal10_CUB-200.pdf,http://www.vision.caltech.edu/visipedia/CUB-200.html.

Wen, Y., Zhang, K., Li, Z. & Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. *European conference on computer vision*, pp. 499–515.

Wilczek, F. (1982). Quantum mechanics of fractional-spin particles. *Physical review letters*, 49(14), 957.

Woo, S., Park, J., Lee, J.-Y. & Kweon, I. S. (2018). Cbam : Convolutional block attention module. *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19.

Xie, S., Girshick, R., Dollár, P., Tu, Z. & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks.

Yeh, R., Hasegawa-Johnson, M. & Do, M. N. (2016). Stable and symmetric filter convolutional neural network. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2652–2656.

Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. (2014). How transferable are features in deep neural networks ? *Advances in neural information processing systems*, 27.

Zeiler, M. D. & Fergus, R. (2014). Visualizing and understanding convolutional networks. *European conference on computer vision*, pp. 818–833.

Zhang, Z., Song, Y. & Qi, H. (2017). Age Progression/Regression by Conditional Adversarial Autoencoder. *CoRR*, abs/1702.08423. Repéré à http://arxiv.org/abs/1702.08423.

Zheng, Y., Pal, D. K. & Savvides, M. (2018). Ring loss : Convex feature normalization for face recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5089–5097.

Zhu, X., Cui, Z., Zhang, T., Li, Y. & Yang, J. (2021). Going Deeper in Frequency Convolutional Neural Network : A Theoretical Perspective. *arXiv preprint arXiv :2108.05690*.

Zoph, B., Vasudevan, V., Shlens, J. & Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710.