Auto-filtrage des entrées par le complément spectral et des normalisations fréquentielles et l'utilisation de filtres en loi de puissance pour les réseaux de neurones convolutifs

par

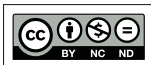
Vincent ROUGEAU-MOSS

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

MONTRÉAL, LE 18 MAI 2021

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

Vincent Rougeau-Moss, 2021



Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

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. Jean-Marc Lina, président du jury
Département de génie électrique à l'École de technologie supérieure

M. Ismail Ben Ayed, examinateur externe
Département de génie des systèmes à l'École de technologie supérieure

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

LE 22 AVRIL 2021

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je remercie tous les professeurs que j'ai eus jusqu'à maintenant au cours de ma carrière d'étudiant et qui m'ont aidé à vouloir toujours aller plus loin et à persévérer dans mes études. Je remercie tout particulièrement mon professeur de mémoire Matthew Toews pour m'avoir soutenu durant ma maîtrise et m'avoir notamment amené à explorer des voix différentes de celles habituelles de la littérature. Bien que certains de ses concepts et idées aient été différents de celles de certains chercheurs qui pourraient s'y opposer, l'innovation peut amener des surprises étonnantes et la découverte de nouveaux sujets de recherche particulièrement intéressants. Je remercie également spécialement ma professeure de mathématiques de troisième secondaire pour m'avoir amené à davantage prendre au sérieux mes études et qui sans qui, je n'aurais probablement pas eu la conviction de faire les efforts nécessaires pour réaliser mes rêves. Je remercie également profondément tous ceux qui m'ont aidé par rapport à ma dyslexie orthographique et mon trouble d'attention avec hyperactivité et qui sans qui, je n'aurais probablement jamais réussi à me rendre jusqu'en maîtrise et réaliser ce rêve.

Je remercie également tous mes amis et tous mes collègues de l'ÉTS qui m'ont soutenu de diverses manières durant la maîtrise, le baccalauréat et avant, ainsi que ceux du laboratoire du LIVIA pour la bonne ambiance de travail et certains échanges d'idées. Je remercie également tout particulièrement mon meilleur ami d'enfance Mathieu Gauthier-Daviault pour tous nos bons moments passés ensemble depuis notre rencontre en maternelle.

Je remercie également toute ma famille de m'avoir toujours appuyé de diverses manières dans mes études et dans ma vie et pour m'avoir notamment appris à toujours persévérer malgré les difficultés. Je remercie également mes parents Richard et Louisette de m'avoir épaulé dans mes études et de m'avoir toujours apporté leur aide quand j'en avais le plus besoin. Je remercie également ma soeur Gabrielle, mes deux grand-mères décédées Lucille Gougeon et Jocelyne (Roy) Rougeau, mon grand-père George Rougeau, mon grand-père décédé Léon Moss, mes tantes Gemma Roy et Jacinthe Rougeau, ma tante décédée Sr Rose-Aimée Roy ainsi que mon oncle Gilles Rougeau pour tout l'amour, le bonheur et leur soutien qu'ils m'ont apportés de diverses manières dans cette vie ou dans l'autre. Je remercie également mon chat récemment

décédé Soleil, mes deux nouveaux chats Comète et Kuybi, mes anciens animaux de compagnies comme Toutoune, Achille, Max, Mélissa, ainsi que les oiseaux et les écureuils autour de chez moi qui ont égayé mes journées, remonté le moral et aidé à vouloir continuer à persévérer malgré les difficultés de la vie.

Je remercie également tous ceux qui m'ont aidé ou influencé de diverses manières jusqu'à maintenant dans ma vie, que ce soit directement ou indirectement, de leur vivant ou non, et sans qui je ne serais pas devenu celui que je suis dorénavant.

Auto-filtrage des entrées par le complément spectral et des normalisations fréquentielles et l'utilisation de filtres en loi de puissance pour les réseaux de neurones convolutifs

Vincent ROUGEAU-MOSS

RÉSUMÉ

Ce mémoire porte sur l'utilisation de procédés de prétraitement des images à l'aide du domaine de Fourier, afin d'améliorer les taux de précision des CNN (*convolutional neural network*) de classification d'images. Pour ce faire, trois nouvelles méthodes sont proposées. La première est la SFSC (*Self-Filtering by the Spectral Complement*) qui consiste à l'utilisation d'un auto-filtrage des entrées combiné à des combinaisons de normalisation et moyennage fréquentiels, afin de simplifier les images et les rendre plus simples à analyser par les CNN. La deuxième et troisième méthode portent sur un filtrage des images d'entrées ou de la sortie de la SFSC, à l'aide de filtres en loi de puissance qui évoluent au cours de l'apprentissage. L'efficacité de nos nouveaux procédés est vérifiée au travers d'expérimentations sur un total de 14 architectures uniques, ayant des poids aléatoires ou prédéfinis, pour trois types de tâche de classification d'images : la classification générale d'images naturelles (CIFAR), la classification de textures (DTD) et la classification fine des émotions du visage (KDEF). La forte polyvalence et l'adaptabilité de la SFSC lui ont permis d'améliorer tous les modèles utilisés, à l'exception du DenseNet et du Wide ResNet sur CIFAR, qui ont eu de légères pertes globales de moins de 1 % de précision. Quant aux filtres en loi de puissance, ceux-ci ne sont pas adaptés aux modèles pré-entraînés et fonctionnaient généralement moins bien que la SFSC, bien qu'ils l'aient dépassé dans certains cas. De futures recherches pourront utiliser nos méthodes et nos conclusions afin de les améliorer, les utiliser autrement avec les CNN ou développer de nouveaux procédés utilisant le domaine fréquentiel avec les CNN.

Mots-clés: CNN, apprentissage profond, Fourier, prétraitement des images, analyse fréquentielle, normalisation, classification d'images

Self-filtering of inputs by their spectral complement and frequency normalizations and the use of power-law filters for convolutional neural networks

Vincent ROUGEAU-MOSS

ABSTRACT

This thesis studies the use of image preprocessing methods using the Fourier domain, in order to improve the accuracy rates of the CNNs (convolutional neural networks) for image classification. For this purpose, three new methods are proposed. The first is the SFSC (Self-Filtering by the Spectral Complement) which consists of the use of a self-filtering of the inputs combined with combinations of normalization and frequency averaging, in order to simplify the images and make them easier to analyze by CNNs. The second and third method consist in filtering the input images or the output of the SFSC using power-law filters which evolve during learning. The efficiency of our new processes is verified by experiments on a total of 14 unique architectures, with random or pre-trained weights, for three types of image classification tasks : the general classification of natural images (CIFAR), the classification of textures (DTD) and the fine-grained classification of facial emotions (KDEF). The high versatility and adaptability of the SFSC allowed it to improve all models used except DenseNet and Wide ResNet on CIFAR which exhibited slight overall losses of less than 1 % accuracy. As for the power-law filters, they are not suitable for pre-trained models and generally perform less well than the SFSC, although they have exceeded it in some cases. Future research may be based on our methods and conclusions in order to improve them, to use them differently with CNNs or to develop new methods using the frequency domain with CNNs.

Keywords: CNN, deep learning, Fourier, preprocessing of images, frequential analysis, normalization, image classification

TABLE DES MATIÈRES

Page

INTRODUCTION 1

CHAPITRE 1 REVUE DE LITTÉRATURE 7
1.1 Les CNN utilisant le domaine de Fourier 7
1.1.1 Présentation de la transformée de Fourier discrète 8
1.1.2 L'utilisation du domaine de Fourier avec les CNN 10
1.1.2.1 L'utilisation de filtres fréquentiels idéaux carrés et de couches de mise en commun fréquentielles avec les CNN 10
1.1.2.2 Définitions des différentes couches des CNN dans le domaine de Fourier 12
1.2 L'utilisation des ondelettes avec les CNN 16
1.2.1 Présentation de la décomposition en ondelettes discrètes 16
1.2.2 Présentation des méthodes de CNN avec des ondelettes 18
1.3 L'utilisation des filtres et des ondelettes de Gabor avec les CNN 25
1.3.1 Présentation des filtres et des ondelettes de Gabor 25
1.3.2 L'utilisation des filtres et ondelettes de Gabor avec les CNN 27
1.4 L'utilisation de la *shearlet transform* avec les CNN 31
1.4.1 Présentation de la *shearlet transform* 31
1.4.2 L'utilisation de la *shearlet transform* comme méthode d'extraction des caractéristiques pour des CNN de la classification d'images 32
1.5 L'utilisation de la *scattering transform* avec les CNN 35
1.5.1 Présentation de la *scattering transform* 35
1.5.2 La *scattering transform* comme méthode d'extraction des caractéristiques des images d'entrées des CNN 37
1.6 Conclusion sur la revue de littérature 38

CHAPITRE 2 L'AUTO-FILTRAGE PAR LE COMPLÉMENT SPECTRAL (SFSC) 43
2.1 Préliminaire : La transformée de Fourrier discrète 43
2.2 Définition de l'auto-filtrage par le complément spectral (SFSC) 45
2.3 Les normalisations qui prennent en compte les différentes dimensions 46
2.4 L'apprentissage des normalisations et du choix du calcul de moyenne 49
2.4.1 La couche de normalisation par les maximums (MN) 49
2.4.2 La couche sur le choix du moyennage effectué (AC) 52

CHAPITRE 3 LES FILTRES FRÉQUENTIELS PASSE-HAUT EN LOI DE PUISSANCE 55
3.1 Illustration sur l'effet de γ et du MN sur les filtres fréquentiels en loi de puissance 57
3.2 Présentation des configurations finales des filtres fréquentiels 62

CHAPITRE 4 RÉSULTATS EXPÉRIMENTAUX 65
4.1 Détails sur les conditions d'expérimentations 66
4.1.1 Détails d'implémentations : CIFAR 66
4.1.2 Détails d'implémentations : DTD 70
4.1.3 Détails d'implémentations : KDEF 73
4.2 Résultats expérimentaux de la SFSC 75
4.2.1 Présentation des poids appris par la SFSC 75
4.2.2 Présentation d'exemples et analyse des effets de la SFSC sur les images 79
4.2.3 Présentation des résultats pour les modèles originaux et ceux avec la SFSC 86
4.3 Résultats expérimentaux pour les filtres fréquentiels en loi de puissance 91
4.3.1 Présentation des poids appris par les filtres fréquentiels 91
4.3.2 Présentation des résultats obtenus par les filtres fréquentiels 97
4.4 Expérimentations supplémentaires sur la SFSC et les filtres fréquentiels 104
4.4.1 La SFSC : Apprentissage partiel et formes sans apprentissage 105
4.4.2 La SFSC : Uniquement normaliser en fréquences les entrées ou uniquement utiliser l'AC 107
4.4.3 Les filtres fréquentiels : Comparaison par rapport à leur forme sans apprentissage 110

CONCLUSION ET RECOMMANDATIONS 113

ANNEXE I REPRÉSENTATION FRÉQUENTIELLE DES COUCHES DE CONVOLUTION DES MODÈLES 117

RÉFÉRENCES BIBLIOGRAPHIQUES 163

LISTE DES TABLEAUX

Page

Tableau 2.1 L'ensemble des maximums qu'il est possible d'utiliser pour les normalisations max-min effectuées sur $M_{n,c,u,v}$ selon les trois dimensions (N, C, S) où $S = (H, W)$ 47

Tableau 2.2 L'ensemble des calculs de moyennes possibles sur les trois dimensions (N, C, S) où $S = (H, W)$ conservera sa forme initiale 48

Tableau 3.1 Configurations finales des filtres fréquentiels de forme ω^{γ} 63

Tableau 4.1 Taux d'apprentissage initiaux pour les filtres fréquentiels sur CIFAR avec, en gras, les valeurs qui diffèrent de celles utilisées par les CNN originaux 68

Tableau 4.2 Tableau de la moyenne des poids appris sur les 5 exécutions aux poids aléatoires différents avec les modèles sur CIFAR (10 exécutions pour notre CNN de référence). Les poids des MN et les écarts entre leur poids maximal et minimal sont en % 76

Tableau 4.3 Tableau de la moyenne des poids appris sur les 10 *splits* pour les 2 cas de DTD. Les poids des MN et les écarts entre leur poids maximal et minimal sont en % 76

Tableau 4.4 Tableau de la moyenne des poids appris sur les 10 *splits* pour KDEF. Les poids des MN et les écarts entre leur poids maximal et minimal sont en % 76

Tableau 4.5 Résultats des CNN sur CIFAR avec 5 exécutions aux poids aléatoires différents (10 exécutions pour notre CNN de référence). Les "+" indiquent avec l'augmentation de données et les chiffres entre parenthèses indiquent les différences par rapport aux modèles originaux 87

Tableau 4.6 Résultats des CNN pré-entraînés et *from scratch* sur DTD avec les statistiques reliées à leur exécution sur les 10 *splits*. Les chiffres entre parenthèses indiquent les différences par rapport aux modèles originaux 87

Tableau 4.7 Résultats des CNN sur KDEF avec les statistiques reliées à leur exécution sur les 10 *splits*. Les chiffres entre parenthèses indiquent les différences par rapport aux modèles originaux 88

Tableau 4.8 Tableau des moyennes des poids appris par les filtres fréquentiels sur les 5 exécutions aux poids aléatoires avec les différents modèles sur CIFAR (10 exécutions pour notre CNN de référence). Les poids des MN et les écarts entre leur poids maximal et minimal sont en % 93

Tableau 4.8 Tableau des moyennes des poids appris par les filtres fréquentiels sur CIFAR (suite) 94

Tableau 4.9 Tableau des moyennes des poids appris par les filtres fréquentiels sur les 10 *splits* de DTD *from scratch*. Les poids des MN et les écarts entre leur poids maximal et minimal sont en % 94

Tableau 4.10 Tableau des moyennes des poids appris par les filtres fréquentiels sur les 10 *splits* de KDEF. Les poids des MN et les écarts entre leur poids maximal et minimal sont en % 95

Tableau 4.11 Résultats des CNN avec les filtres fréquentiels sur CIFAR avec 5 exécutions aux poids aléatoires différents (10 exécutions pour notre CNN de référence). Les "+" indiquent l'utilisation de l'augmentation de données 98

Tableau 4.11 Résultats des CNN avec les filtres fréquentiels sur CIFAR (suite) 99

Tableau 4.12 Résultats des CNN avec les filtres fréquentiels sur DTD *from scratch* avec les statistiques reliées à leurs exécutions sur les 10 *splits* 100

Tableau 4.13 Résultats des CNN avec les filtres fréquentiels sur KDEF avec les statistiques reliées à leurs exécutions sur les 10 *splits* 100

Tableau 4.14 Résultats du CNN de référence avec les différentes formes de la SFSC partiellement apprises et sans apprentissage (poids fixes) sur CIFAR avec 10 exécutions par configurations. Les "+" indiquent les cas avec de l'augmentation de données 106

Tableau 4.15 Résultats du CNN de référence pour les différents cas avec uniquement Norm0 #2 ($\bar{M}$) et uniquement l'AC sur les différents cas de CIFAR avec 10 exécutions. Les "+" indiquent les cas avec de l'augmentation de données 109

Tableau 4.16 Résultats du CNN de référence avec les filtres fréquentiels fixes ou appris sur CIFAR sur les 10 exécutions. Les "+" indiquent les cas avec de l'augmentation de données 111

LISTE DES FIGURES

Page

Figure 1.1 Représentation fréquentielle des filtres d'ondelettes de Haar et Rbio3.1 pour 1, 2 et 3 niveaux de DWT et de WPG ainsi que le découpage fréquentiel de la DTCWT pour 2 niveaux et 6 orientations 19

Figure 1.2 Exemple de représentation du découpage fréquentiel effectué par une banque de filtres de Gabor réelle avec 6 orientations et 4 fréquences 27

Figure 1.3 Exemple de découpage fréquentiel en 2D effectué par la *cone-adapted shearlet transform* avec à gauche un découpage selon 1 niveau et à droite un découpage selon 3 niveaux 33

Figure 1.4 Schéma du découpage fréquentiel effectué par la *scattering transform* pour le premier niveau (a) et le deuxième (b) niveau de décomposition avec l'utilisation de 6 angles entre 0 et π (L = 12) et 3 niveaux d'échelle (J = 3) 36

Figure 2.1 Les étapes du calcul de l'équation 2.5 49

Figure 3.1 La forme globale des filtres fréquentiels de forme ω^γ 58

Figure 3.2 Exemple de résultats obtenues par l'application de différents filtre de forme ω^γ pour différentes valeurs de γ 59

Figure 3.3 Comparaison des effets relatifs a l'utilisation des MN sur les filtres fréquentiels de forme ω^γ 61

Figure 3.4 Exemple de résultats obtenues par les différents filtres de $\omega^{0,5}$ et ω^1 avec l'utilisation des différents MN de la figure 3.3 61

Figure 4.1 Architecture de notre CNN de référence 69

Figure 4.2 Les résultats intermédiaires obtenus par notre CNN de références sur CIFAR-100+ avec différentes images ainsi que la moyenne globale des résultats obtenus sur l'ensemble de la base de données d'apprentissage de CIFAR-100+ 80

Figure 4.3 Les résultats intermédiaires obtenus par le AlexNet+BN sur DTD *from scratch* avec différentes images ainsi que la moyenne globale des résultats obtenus sur l'ensemble de la base de données d'apprentissage de DTD, *split* #7 81

Figure 4.4 Les résultats intermédiaires obtenus par le ResNet-50 pré-entraîné (sur ImageNet) sur DTD avec différentes images ainsi que la moyenne globale des résultats obtenus sur l'ensemble de la base de données d'apprentissage de DTD, *split* #3 82

Figure 4.5 Les résultats intermédiaires obtenus par le ResNet-34 sur KDEF avec différentes images ainsi que la moyenne globale des résultats obtenus sur l'ensemble de la base de données d'apprentissage de KDEF, *split* #4 83

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AC	Average Choice
ANN	Artificial Neural Networks
BN	Batch Normalization
CIFAR	Canadian Institute for Advanced Research
CMA	Cumulative Moving Average
CNN	Convolutional Neural Network
DTCWT	Dual-Tree Complex Wavelet Transform
DTD	Describable Textures Dataset
DWT	Discrete Wavelet Transform
EMA	Exponential Moving Average
FC	Fully Connected layer
GPU	Graphics Processing Units
IN	Instance Normalization
KDEF	Karolinska Directed Emotional Faces
LIVIA	Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle
LN	Layer Normalization
LSTM	Long Short-Term Memory
MN	Max Normalization
MRA	Multi-Resolution Analysis
RAM	Random Access Machine
ReLU	Rectifier Linear Unit
RNN	Recurrent Neural Network
SFSC	Self-Filtering by the Spectral Complement

SGD	Stochastic Gradient Descent
SMA	Simple Moving Average
SN	Switchable Normalization
TF	Transformé de Fourier
TFD	Transformé de Fourier Discrète
TFDI	Transformé de Fourier Discrète Inverse
TFI	Transformé de Fourier Inverse
WPD	Wavelet Packet Decomposition
WRN	Wide ResNet

INTRODUCTION

Les réseaux de neurones convolutifs (CNN) sont des réseaux de neurones artificiels (ANN) qui utilisent des couches de convolutions comme principaux moyens pour extraire les caractéristiques des données qu'ils reçoivent en entrées. Ces couches de convolutions effectuent des convolutions dans le domaine spatial, entre leurs filtres numériques et leur entrée, afin d'obtenir les réponses de ces filtres qui indiquent par leur intensité leur niveau de correspondance, tout en étant localisés dans le domaine spatial. Bien que les CNN soient capables d'extraire ces informations pour accomplir leurs tâches, différentes recherches ont montré que l'utilisation de procédés de prétraitement des entrées, à l'aide du domaine spectral, peut les aider à combler certaines lacunes ou leur fournir des informations qui leur seraient autrement plus difficilement obtenables.

Parmi ces procédés de prétraitement, outre ceux liés à des cas particuliers, il y a notamment plusieurs recherches qui ont porté sur l'utilisation des ondelettes discrètes, des filtres et d'ondelettes de Gabor, de la *shearlet transform* ainsi que de la *scattering transform*, comme méthode de prétraitement des images. Le plus souvent, ces méthodes fournissent plusieurs versions modifiées des trois canaux de couleur des images en ayant chacune, comme caractéristique, la mise en évidence de certaines régions du spectre fréquentiel. Bien que ces méthodes soient à la base des méthodes d'extraction des caractéristiques, elles sont également soutenues par des recherches qui indiquent que les CNN utilisent l'ensemble du spectre fréquentiel, et non uniquement les basses ou les hautes fréquences (Tsuzuku & Sato, 2019 ; Yin, Gontijo Lopes, Shlens, Cubuk & Gilmer, 2019a) (voir Annexe I). Par ailleurs, d'autres recherches ont montré que les ANN semblent d'abord apprendre les basses fréquences avant de tendre vers l'apprentissage des moyennes et des hautes fréquences (Rahaman et al., 2019 ; Xu, Zhang & Xiao, 2019 ; Xu, 2018). L'utilisation de filtres qui mettent en évidences les fréquences habituellement apprises plus tardivement pourrait donc favoriser une convergence de départ plus rapide et de meilleure qualité.

Malgré tout, plusieurs de ces méthodes de prétraitement nécessitent des modifications plus ou moins importantes des modèles pouvant aller de l'utilisation d'un plus grand nombre de canaux d'entrées, jusqu'au besoin de concevoir des ensembles de modèles ou des architectures très particulières. Cependant, effectuer ce genre de modification n'est pas toujours possible, notamment si l'on souhaite utiliser des modèles pré-entraînés en raison de la faible taille de la base de données et de sa difficulté pouvant rendre l'apprentissage particulièrement difficile. Certaines de ces méthodes présentent également des faiblesses importantes. Dans le cas des ondelettes discrètes, celles-ci ont notamment comme défaut d'avoir des filtres passe-haut très clairsemés (*sparse*) qui fournissent des images peu informatives et difficiles à analyser pour les CNN. Dans le cas de leur filtre passe-bas, celui-ci peut, dans bien des cas, être remplacé par une couche de mise en commun par valeur moyenne (*average pooling*) en raison de son équivalence avec le filtre passe-bas de l'ondelette de Haar, qui obtient le plus souvent des performances similaires aux autres ondelettes. Dans le cas de la *scattering transform*, il a été démontré que celle-ci devenait moins efficace par rapport à des couches de convolution standards, notamment lors de l'utilisation de grandes bases de données ou d'images aux caractéristiques complexes comme celles des images naturelles (Oyallon, Belilovsky & Zagoruyko, 2017 ; Oyallon, Belilovsky, Zagoruyko & Valko, 2018 ; Ren, Zhou & Guo, 2019) ou de textures (Rodriguez et al., 2020). Finalement, bon nombre de recherches utilisant ces méthodes d'analyse spectrale ont effectué une seule mesure par configuration ou n'ont pas expérimenté leur méthode avec diverses formes d'architectures et divers types de bases de données. Par conséquent, il est souvent difficile de déterminer si celles-ci pourraient donner de bons résultats dans d'autres contextes que celui de départ et si les mesures sont fiables et reproductibles. Ceci est d'autant plus vrai lorsque les chercheurs ont effectué leurs mesures sur de petites bases de données puisqu'elles présentent, le plus souvent, des écarts-types pouvant être relativement élevés.

De l'autre côté, plusieurs recherches ont également porté sur l'utilisation du domaine de Fourier, non pas uniquement pour accélérer les calculs des couches de convolution, mais pour notamment

effectuer des attaques en fréquences, mieux comprendre les CNN ou tenter d'effectuer l'ensemble des calculs des CNN dans le domaine fréquentiel. Cependant, peu de ces recherches ont tenté des approches visant principalement à améliorer les performances de leurs modèles, en exploitant les propriétés de ce domaine, au lieu de l'utiliser comme une simple représentation différente des images par rapport au domaine spatial.

Par rapport à ces différentes recherches, on peut se demander s'il serait possible d'utiliser les données d'entrée pour s'auto-filtrer afin de ne pas appliquer des filtres qui sont, soit définis selon des procédures manuelles (*hand-crafted*) avec plusieurs hyperparamètres à optimiser, soit définis par les couches de convolutions. En effet, les filtres correspondant le mieux aux données sont les données elles-mêmes et les relations entre leurs dimensions peuvent être prises en compte afin de mettre en évidences les régions du spectre qui sont les plus importantes pour les modèles. De plus, l'utilisation d'un auto-filtrage permet d'utiliser des filtres de la même taille que les entrées et qui couvrent l'ensemble du spectre fréquentiel. Ceci permet de ne plus être limité à la couverture de petites régions locales dans le domaine spatial, ni de devoir effectuer un découpage fréquentiel afin d'envoyer séparément les différentes régions du spectre aux CNN. Le modèle peut également rester intact en n'ayant pas plusieurs versions filtrées des entrées si le seul filtre utilisé est les entrées elles-mêmes. Les modèles n'ont donc pas à être adaptés en fonction de la méthode choisie et l'utilisation de modèles pré-entraînés devient également possible, à l'aide d'un procédé qui modifie leur entrée au lieu de le faire sur leur sortie, comme le font le plus souvent les chercheurs dans la littérature.

Dans la cadre de cette recherche, nous proposons trois nouvelles méthodes de prétraitement des images utilisant le domaine de Fourier et qui sont notamment basées sur ces questionnements et ces problématiques. Ces méthodes visent également à répondre à certaines des faiblesses observées dans plusieurs recherches sur des méthodes de prétraitement utilisant le domaine spectral et ont trois objectifs :

- les nouvelles méthodes visent à obtenir des performances plus élevées, ou au moins semblables aux modèles originaux, qui peuvent être de diverses formes d'architecture et être utilisées sur divers types de bases de données. Les expérimentations doivent également être validées au travers de plusieurs exécutions effectuées avec chacune des configurations, tout en visant à isoler la contribution liée à ces méthodes par rapport aux autres facteurs possibles.
- les nouvelles méthodes ne doivent nécessiter aucune modification des caractéristiques des architectures ou des algorithmes d'apprentissage lié aux poids des modèles auxquels elles sont rattachées. Par conséquent, les modèles de références sont les originaux tels que décrits par leurs auteurs. Les méthodes peuvent également être utilisées avec des modèles ayant des poids prédéfinis où seules leurs entrées seront modifiées.
- comme pour plusieurs méthodes utilisant des filtres prédéfinis avec les ondelettes discrètes, les filtres et ondelettes de Gabor, la *shearlet transform* ou la *scattering transform*, les nouvelles méthodes doivent également posséder des formes fixes sans apprentissage.

Cette recherche s'est limitée à l'utilisation de nos méthodes en tant que procédé de prétraitement des images 2D des entrées de CNN effectuant de la classification d'images naturelles sur CIFAR (*Canadian Institute for Advanced Research*) (Krizhevsky, Hinton et al., 2009), de textures sur DTD (*Describable Textures Dataset*) (Cimpoi, Maji, Kokkinos, Mohamed & Vedaldi, 2014) et des émotions du visage sur KDEF (*Karolinska Directed Emotional Faces*) (Lundqvist, Flykt & Öhman, 1998). Selon certains chercheurs, la classification d'images naturelles nécessite le plus souvent des modèles relativement profonds (He, Zhang, Ren & Sun, 2016b ; Hu, Shen & Sun, 2018 ; Huang, Liu, Van Der Maaten & Weinberger, 2017 ; Xie, Girshick, Dollár, Tu & He, 2017) alors que la classification de textures pourrait être suffisante avec peu de profondeur (Andrearczyk, 2017). Dans le cas de la classification des émotions du visage, les modèles doivent extraire des informations qui sont plus spécifiques par rapport à la reconnaissance générique puisqu'il ne s'agit pas de classifier des entités de différentes classes, mais de plutôt reconnaître les traits des visages propres à chacune des émotions.

Cette recherche s'est également limitée à l'utilisation de taille de lots (*batch size*) par processeur graphique (GPU) qui sont supérieurs à 32 et qui sont standards par rapport à la littérature (Keskar, Mudigere, Nocedal, Smelyanskiy & Tang, 2016 ; Luo, Ren, Peng, Zhang & Li, 2018b ; Luo, Zhang, Ren, Peng & Li, 2019b ; Wu & He, 2018). Étant donné le deuxième objectif, les expérimentations effectuées n'ont pas cherché à optimiser les architectures ou les procédures d'apprentissage des poids des modèles utilisant nos trois nouvelles méthodes. Elles ont été développées dans le but d'être des nouvelles façons de faire au lieu d'être des améliorations de méthodes existantes.

Malgré les différentes limitations de cette recherche, nous pensons que nos méthodes peuvent être appliquées dans des contextes plus larges dans de futures recherches. Ceci comprend leur utilisation combinée à certaines méthodes de la littérature et leurs exécutions avec d'autres bases de données, architectures et sur d'autres tâches que la classification d'images. Parmi ces autres tâches possibles, les CNN effectuant de la segmentation d'images pourraient être améliorés en raison de leur exploitation des contours des images qui sont associés aux hautes fréquences et que ces informations deviennent plus accessibles à l'aide de nos méthodes.

Le premier chapitre porte sur la revue de littérature et présente diverses recherches ayant directement utilisé le domaine de Fourier ou ayant porté sur les ondelettes discrètes, les filtres et ondelettes de Gabor, la *shearlet transform* et la *scattering transform*. Les recherches présentées utilisent le plus souvent des CNN pour de la classification d'images, mais elles ne se limitent pas uniquement à des procédés de prétraitement des images d'entrées de leur modèle.

Le deuxième chapitre présente notre première méthode qui utilise le principe de l'autocorrélation des images afin de les filtrer par leur magnitude inversée et de supprimer certaines de leurs informations. Ceci permet, le plus souvent, de rendre la convergence plus rapide et d'atteindre de meilleures performances. Cette façon de faire a l'avantage de n'utiliser aucune forme prédéfinie et elle est, par conséquent, uniquement basée sur le contenu fréquentiel des entrées des CNN.

Le troisième chapitre présente notre seconde et notre troisième méthode qui utilisent des filtres fréquentiels passe-haut en loi de puissance où leur forme est basée sur la définition des dérivées dans le domaine de Fourier. Comme pour la première méthode, la forme de ces filtres est inversée par rapport aux entrées et évolue au cours de l'apprentissage. Ces deux méthodes servent également de méthodes comparatives par rapport à la première méthode et elles fournissent certains éléments de réponses au sujet d'autres recherches de la littérature ayant également utilisé des filtres aux formes prédéfinies sur leurs entrées.

Le quatrième chapitre présente les résultats obtenus par nos trois méthodes et inclut également une dernière section sur des expérimentations supplémentaires visant à répondre à certaines interrogations possibles relativement à nos méthodes.

Nous finissons ce mémoire par la conclusion ainsi que des recommandations et des suggestions sur les travaux futurs pouvant être effectués pour poursuivre les recherches sur nos trois nouvelles méthodes ou s'en inspirer pour en développer de nouvelles.

CHAPITRE 1

REVUE DE LITTÉRATURE

Ce chapitre présente une revue de l'état de l'art de différentes recherches ayant porté sur des sujets connexes à ceux liés à nos trois nouvelles méthodes de prétraitement des images d'entrées à l'aide du domaine fréquentiel de Fourier. Étant donné que nos méthodes ont été utilisées pour de la classification d'images avec des CNN, la plupart des recherches qui seront présentées portent également sur cette utilisation. Chacune des sections débute par une brève description des transformations ou méthodes utilisées et est suivie d'une vue d'ensemble des travaux effectués par les chercheurs. Plusieurs de ces recherches ne se limitent pas uniquement à des méthodes de prétraitement des images d'entrées des CNN, mais celles-ci sont malgré tout incluses.

Ce chapitre couvre principalement les sujets suivants par rapport aux CNN utilisant des images comme données d'entrée :

- l'utilisation du domaine de Fourier avec les CNN afin d'effectuer des couches de mise en commun ou d'effectuer l'ensemble de ses opérations dans le domaine fréquentiel ;
- les ondelettes discrètes fixes ou apprises pour le filtrage des entrées ou des sorties des couches des CNN ;
- l'utilisation des filtres et ondelettes de Gabor pour initialiser ou influencer l'apprentissage effectué par la première, la deuxième ou l'ensemble des couches de convolution des CNN ;
- l'utilisation de la *shearlet transform* comme méthode d'extraction des caractéristiques pour des CNN, en l'utilisant sur les entrées ou dans les premières couches des CNN ;
- la *scattering transform* comme méthode de prétraitement des entrées pouvant remplacer les premières couches de convolution des CNN en raison de son grand nombre de filtres et de la présence d'une opération non linéaire.

1.1 Les CNN utilisant le domaine de Fourier

Étant donné que notre recherche ainsi que toutes les autres méthodes exposées dans cette revue de littérature utilisent ou ont des liens avec le domaine de Fourier, cette première section décrit la

transformée de Fourier discrète et certaines de ses propriétés. Par la suite, différentes recherches ayant utilisé ce domaine pour de la classification d'images réelles 2D sont présentées.

1.1.1 Présentation de la transformée de Fourier discrète

La transformée de Fourier (TF) est une transformation linéaire qui permet d'exprimer un signal ou une fonction définie dans le domaine temporel ou spatial, comme étant une somme pondérée d'ondes sinusoïdales à différentes fréquences. La transformée de Fourier discrète (TFD) est la version discrète de cette transformation pour des signaux numériques. Dans les deux cas, la représentation obtenue par ces transformations est définie selon des valeurs complexes où, en format polaire, la magnitude décrit l'amplitude totale de la sommation des ondes sinusoïdales aux différentes fréquences, tandis que la phase décrit leurs déphasages totaux, et donc, leurs orientations globales aux différentes fréquences. De l'autre côté, la transformée de Fourier inverse (TFI) permet de revenir au domaine spatial ou temporel sans causer de perte d'information.

Dans le cadre du traitement d'images, la magnitude est reliée à la manière dont sont distribuées les intensités des pixels dans le domaine spatial, alors que la phase décrit le contenu des images en décrivant l'orientation des lignes et des arêtes. Par conséquent, modifier la phase des images change leur contenu et ce qui est représenté, alors qu'une modification de la magnitude change uniquement l'aspect visuel des images. Par conséquent, de manière générale pour le traitement d'images, les filtres définis dans le domaine de Fourier ne modifient que la magnitude des images pour, par exemple, éliminer ou atténuer certaines plages de fréquences afin de mettre davantage en évidence celles qui ont été conservées. On note également que pour des images naturelles, la forme de la magnitude se présente selon une exponentielle inversée où la majorité de l'énergie se trouve dans les basses fréquences, et la magnitude à la fréquence de 0, DC, correspond à la moyenne globale des intensités des pixels dans le domaine spatial. Ceci est dû au fait que les basses fréquences sont associées aux variations lentes de luminosité ou de couleurs dans le domaine spatial, alors que les hautes fréquences sont associées aux variations rapides d'intensité, tel qu'au niveau des lignes et des arêtes. Ainsi, un filtre passe-bas va atténuer ou éliminer les hautes fréquences, rendant ainsi l'image plus floue en ayant moins de détails. De

l'autre côté, un filtre passe-haut va plutôt conserver les hautes fréquences, donnant ainsi des couleurs plus grisâtres (neutres) et potentiellement trop de détails, ce qui peut donner un aspect visuel d'images bruitées.

Parmi les nombreuses propriétés mathématiques que possède la TFD, quelques-unes d'entre elles sont particulièrement utiles pour mieux comprendre son utilisation avec les CNN. Premièrement, la convolution dans le domaine spatial devient une multiplication (*element-wise multiplication*) dans le domaine fréquentiel, et vice-versa. Cependant, pour que cette multiplication dans le domaine de Fourier corresponde bien à une convolution linéaire dans le domaine spatial, il est nécessaire de faire une complétion de zéros (*zero-padding*) sur les entrées avant de les mettre dans le domaine de Fourier. En effet, sans cela, cette multiplication correspondrait plutôt à une convolution circulaire, soit une convolution linéaire à laquelle un effet *aliasing* a été ajouté (Ayat, Khalil-Hani, Ab Rahman & Abdellatef, 2019 ; Gonzalez & Woods, 2006). Cette propriété d'appliquer des filtres sur les images à l'aide d'une multiplication dans le domaine de Fourier est notamment utilisée pour les couches de convolution des CNN où les calculs de convolution deviennent particulièrement lents sur de grosses images ou lors de l'utilisation de filtres de grande taille. Le fait de mettre les images d'entrées et les filtres dans le domaine de Fourier afin de les multiplier ensemble, puis de remettre leur résultat dans le domaine spatial, permet de réduire le nombre d'opérations et les temps de calcul des couches de convolution (Lin, Ding, Tu, Huang & Liu, 2019 ; Popa, 2017 ; Popa & Cernăzanu-Glăvan, 2018). Bien que ces opérations de changement de domaine soient rapides à effectuer et nécessaires étant donné que le reste des opérations effectuées dans les CNN est généralement dans le domaine spatial, ceci engendre des coûts supplémentaires non négligeables. Ainsi certains chercheurs qui sont cités dans la section 1.1.2.2 se sont penchés sur cette problématique pour tenter de minimiser ces nombreux transferts de domaines.

Deuxièmement, pour des entrées réelles, il y a la symétrie paire (ou *conjugate symmetry*) qui est une symétrie entre les deux côtés de l'axe imaginaire. Par conséquent, on peut représenter et utiliser le domaine fréquentiel en utilisant une représentation qui n'utilise qu'un seul des deux côtés de l'axe imaginaire, donnant ainsi une taille correspondante à la moitié de l'image dans le

domaine spatial. Ceci permet alors d'effectuer les calculs plus rapidement sans pour autant avoir une représentation incomplète ou approximative.

Troisièmement, en raison des définitions mentionnées précédemment sur la magnitude et la phase, la magnitude est invariante à la translation, mais ne l'est pas par rapport à la rotation et les variations reliées aux couleurs des pixels dans le domaine spatial comme la luminosité, le contraste ou la saturation. De l'autre côté, la phase est invariante aux changements des couleurs des pixels, mais varie en fonction de la translation et de la rotation.

La prochaine section présente différentes recherches ayant utilisé la transformée de Fourier discrète avec des CNN.

1.1.2 L'utilisation du domaine de Fourier avec les CNN

Le domaine de Fourier a été utilisé sous diverses formes avec des CNN effectuant de la classification d'images où son utilisation ne se limite pas, par exemple, à accélérer les calculs dans les couches de convolution (Chung, Cheng & Liu, 2019 ; Highlander & Rodriguez, 2016 ; Mody, Ghone, Mathew & Jones, 2017) ou uniquement aider à mieux comprendre les CNN ou leur comportement par rapport à certains types d'attaque en fréquences (Jo & Bengio, 2017 ; Rahaman *et al.*, 2019 ; Tsuzuku & Sato, 2019 ; Wang, Wu, Huang & Xing, 2020 ; Yin *et al.*, 2019a).

1.1.2.1 L'utilisation de filtres fréquentiels idéaux carrés et de couches de mise en commun fréquentielles avec les CNN

Plusieurs recherches ont porté sur des méthodes de mise en commun (*pooling*) dans le domaine fréquentiel utilisant des filtres idéaux passe-bas carrés (Rippel, Snoek & Adams, 2015 ; Zhang & Ma, 2018) (même lorsque les chercheurs indiquent utiliser des filtres idéaux passe-haut (Pratt, Williams, Coenen & Zheng, 2017)). Basé sur le fait que la majorité de l'énergie dans les images naturelles est concentrée dans les basses fréquences, l'aspect visuel des images obtenues par ces méthodes semble plus fidèle aux images d'entrée qu'avec des méthodes de mise en

commun spatial habituel, comme le *max pooling* ou le *average pooling*. On note cependant que les gains de performance semblent très légers lors de leurs tests de comparaison effectués avec de petits modèles (Zhang & Ma, 2018). Étant donné que ce sont des filtres idéaux, ceux-ci induisent des *ringing artifacts* dans le domaine spatial qui peuvent nuire aux modèles. Les hautes fréquences seront également complètement éliminées en étant mises à zéro alors que, par exemple, un *average pooling* ne fera que les atténuer. En effet, le filtre passe-bas LL de l'ondelette discrète de Haar (figure 1.1a) correspond à un *average pooling* avec une fenêtre de 2×2 pixels. Le fait de complètement enlever les hautes fréquences peut avoir pour conséquence de limiter le potentiel maximal de ces modèles, quand on sait que, selon certaines études, les réseaux de neurones apprennent d'abord les basses fréquences avant de lentement tendre vers l'apprentissage des hautes fréquences (Rahaman *et al.*, 2019 ; Xu *et al.*, 2019 ; Xu, 2018). Aussi, l'ensemble du spectre fréquentiel reste important pour les CNN, même sur des images naturelles où la majorité de l'énergie est concentrée dans les basses fréquences (Tsuzuku & Sato, 2019 ; Yin *et al.*, 2019a) (voir Annexe I).

D'autre part, d'autres chercheurs ont utilisé ce type de méthode de mise en commun dans le domaine de Fourier uniquement avec la dernière couche de convolution afin d'envoyer la magnitude de sa sortie aux couches entièrement connectées (FC) (Chidester, Zhou, Do & Ma, 2019 ; Ryu, Yang & Lim, 2018). Selon eux, cette façon de faire aiderait à améliorer l'invariance à la translation des modèles (Ryu *et al.*, 2018) ou, par exemple, conserver certaines propriétés d'invariance à la rotation dans l'ensemble de leur modèle (Chidester *et al.*, 2019). Bien que cette façon de faire semble améliorer leurs modèles, par rapport à une couche de mise en commun global (*global pooling*), elle augmente le nombre de paramètres pour la première FC, induisant ainsi des coûts supplémentaires. Ceci est d'autant plus vrai si plusieurs sorties auxiliaires sont utilisées et les comparaisons effectuées ne prennent pas nécessairement en compte ces différences de tailles de modèles (Ryu *et al.*, 2018).

Finalement, on note également que ces filtres idéaux carrés sont également utilisés dans certaines recherches visant à mieux comprendre les CNN ou leur comportement par rapport à certains types d'attaques de CNN (Jo & Bengio, 2017 ; Wang *et al.*, 2020 ; Yin *et al.*, 2019a). Cependant,

leurs analyses sont potentiellement légèrement biaisées parce que les filtres idéaux sont parmi les pires types de filtres fréquentiels pour le traitement d'images en raison des artefacts ajoutés aux images dans le domaine spatial et qui sont également visibles dans leurs exemples (Rippel *et al.*, 2015 ; Wang *et al.*, 2020 ; Yin *et al.*, 2019a ; Zhang & Ma, 2018). L'utilisation de filtres fréquentiels n'induisant pas d'artefact dans les images serait probablement préférable pour éviter d'induire ce genre de biais dans leur analyse.

En somme, l'utilisation de filtres n'induisant pas ce genre d'artefact permettrait probablement aux chercheurs d'obtenir des analyses plus fiables et plus précises sur les CNN, tout en utilisant des méthodes de mise en commun spectrale pouvant donner de meilleures performances. Malgré tout, en raison du fait que les CNN possèdent généralement une forte capacité d'adaptation et de généralisation, il est probable que les conclusions de ces différentes recherches restent similaires même avec d'autres types de filtres fréquentiels. On note également que, pour les chercheurs qui utilisent des méthodes de mise en commun fréquentielle à l'intérieur des CNN, ils doivent continuellement changer de domaine, avant et après avoir appliqué leur méthode. Ceci induit des coûts de calcul supplémentaires qui s'ajoutent à ceux effectués par les couches de convolution. La prochaine sous-section présente différentes façons de faire qui peuvent être utilisées dans le cas où l'on voudrait effectuer l'ensemble ou la majorité des opérations des CNN, dans le domaine de Fourier.

1.1.2.2 Définitions des différentes couches des CNN dans le domaine de Fourier

Plusieurs chercheurs ont tenté différentes approches pour effectuer la majorité ou l'ensemble de leurs opérations avec des images réelles mises sous la forme de valeurs complexes. Bien que seuls certains d'entre eux ont pris en compte le fait que la convolution spatiale devient une multiplication dans le domaine de Fourier (Ayat *et al.*, 2019 ; Guan, Zhang, Sethares, Kijowski & Liu, 2019 ; Li, You & Robles-Kelly, 2018 ; Pratt *et al.*, 2017 ; Vasilyev, 2015 ; Watanabe & Wolf, 2020), d'autres chercheurs ayant utilisé des CNN spatiaux sur des entrées complexes (simulées ou à l'aide de la DFT) présentent malgré tout des recherches intéressantes à ce sujet (Guberman, 2016 ; Popa, 2017 ; Popa & Cernăzanu-Glăvan, 2018 ; Wang, Zhang,

Chai & Xia, 2018). Bien que la convolution devient une multiplication dans le domaine de Fourier, la définition des autres couches des CNN peut être, de son côté, plus difficile à correctement caractériser dans le domaine de Fourier. Ceci comprend notamment les couches de mise en commun (*pooling*), la fonction d'activation non linéaire *rectifier linear unit* (ReLU), les couches de normalisation (BN), les couches entièrement connectées (FC) et la fonction de perte à minimiser *softmax*.

Pour les couches de mise en commun (*pooling*), certains chercheurs ont utilisé la méthode de mise en commun avec le filtre idéal présenté précédemment (Ayat *et al.*, 2019 ; Pratt *et al.*, 2017 ; Watanabe & Wolf, 2020), tandis que d'autres ont proposé l'idée d'utiliser un sinus cardinal (sinc), mais ne présentent pas nécessairement de résultats par rapport à leur utilisation avec des CNN (Vasilyev, 2015). De l'autre côté, certains chercheurs ont également utilisé des couches de mise en commun par valeur maximale ou moyenne standard sur des entrées mises dans le domaine de Fourier (Popa, 2017 ; Popa & Cernăzanu-Glăvan, 2018). Cependant, ce genre d'opération n'est pas nécessairement appropriée puisque les principes de localité des images ne sont pas respectés dans le domaine de Fourier (Guberman, 2016) et que cela peut également induire certains artefacts dans les images (Bouman, 2020 ; Popa & Cernăzanu-Glăvan, 2018).

En ce qui concerne la normalisation avec BN (*batch normalization*), Wang *et al.* (2018) avaient proposé l'utilisation d'une matrice de covariance 2×2 afin de garantir une variance égale entre les parties réelles et imaginaires, mais ils utilisaient un CNN spatial combiné à une simulation d'entrée complexe pour leurs expérimentations. De l'autre côté, Ayat *et al.* (2019) avaient plutôt mis à zéro les DC des canaux des entrées, puisque ceux-ci correspondent aux moyennes globales de chaque canal, et ils ont considéré que le calcul de la division par l'écart-type serait équivalent entre les deux domaines (Watanabe & Wolf, 2020). Dans les deux cas, leur utilisation semble améliorer les performances de leurs modèles, notamment en aidant à stabiliser les calculs numériques ainsi que la convergence de leurs modèles.

Pour ce qui est de la fonction d'activation ReLU, plusieurs façons de faire ont été présentées dans la littérature puisqu'elle est souvent l'une des principales difficultés auxquelles ces recherches

doivent faire face. Certains chercheurs ont d'abord exprimé le fait que ReLU serait, en principe, une convolution dans le domaine de Fourier, car elle peut également être exprimée dans le domaine spatial par une multiplication avec une fonction *Heaviside* (Li *et al.*, 2018 ; Vasilyev, 2015). Cependant, son utilisation selon cette définition implique de devoir faire ReLU dans le domaine spatial, ce qui enlève tout intérêt de l'utiliser tel quel (Vasilyev, 2015). De l'autre côté, l'utilisation d'une convolution avec des poids appris pour éviter ce problème a pour conséquence de n'avoir aucune non-linéarité à l'intérieur du CNN (Li *et al.*, 2018). D'autres ont plutôt proposé des approches de ReLU où seule une portion du spectre fréquentiel serait conservée avec notamment : modReLU pour préserver la phase et modifier la magnitude (Arjovsky, Shah & Bengio, 2016), zReLU qui conserve uniquement le premier quadrant (Guberman, 2016 ; Wang *et al.*, 2018) et CReLU qui applique séparément des ReLU sur les valeurs réelles et imaginaires (Popa, 2017 ; Popa & Cernăzanu-Glăvan, 2018 ; Wang *et al.*, 2018). Cependant, zReLU et CReLU ont le principal problème d'effectuer séparément leurs opérations sur les parties réelles et imaginaires, ce qui peux avoir pour conséquence de modifier la phase (Arjovsky *et al.*, 2016), et ils ne respectent pas la propriété de la symétrie paire du domaine de Fourier. On note également que leurs expérimentations pour de la classification d'images portaient sur des CNN spatiaux et qu'elles ne présentent généralement pas de gain significatif de performance qui justifierait leur utilisation par rapport à un CNN standard sur des entrées réelles du domaine spatial (Guberman, 2016 ; Popa, 2017 ; Popa & Cernăzanu-Glăvan, 2018 ; Wang *et al.*, 2018). Finalement, d'autres ont tenté des approches basées sur un polynôme d'ordre deux approximant la forme de ReLU dans le domaine spatial tout en utilisant une convolution dans le domaine de Fourier (SReLU) (Ayat *et al.*, 2019), ou l'utilisation des premières et deuxièmes harmoniques dans le domaine de Fourier (2SReLU) (Watanabe & Wolf, 2020). Bien que ces approches respectent davantage les propriétés du domaine de Fourier, SReLU implique, elle aussi, un calcul de convolution dans le domaine de Fourier. De l'autre côté, les principes relatifs à 2SReLU perdent de leur valeur par la présence du filtre idéal passe-bas pour leur couche de mise en commun (Watanabe & Wolf, 2020). On note également que ces méthodes ont le défaut de généralement causer des baisses de performances pouvant être significatives, même sur des bases des données très simples avec de très petits CNN, montrant ainsi que leur utilisation

à plus grande échelle pourrait occasionner des pertes de performances plus importantes par rapport à leur équivalent dans le domaine spatial. Au final, bien que certaines de ces approches présentent des gains en termes de vitesse de calcul, notamment lorsqu'elles sont combinées à une convolution spatiale qui devient une multiplication dans le domaine de Fourier, elles semblent au mieux égaler leur version spatiale. La plupart d'entre elles ne sont également pas conçues dans un but d'obtenir de meilleures performances par rapport à leur forme dans le domaine spatial. De plus, la plupart de leurs expérimentations ne sont effectuées qu'avec une seule mesure par expérimentation, ce qui ne permet pas de mesurer avec exactitude les écarts de performances en raison de l'aléatoire présent dans les CNN.

Finalement, pour ce qui est des couches entièrement connectées (FC) et la fonction *softmax*, certains chercheurs ont préféré revenir dans le domaine spatial ou utiliser une seule sortie commune aux parties réelles et imaginaires (Ayat *et al.*, 2019 ; Lin *et al.*, 2019 ; Wang *et al.*, 2018 ; Watanabe & Wolf, 2020). Par ailleurs, d'autres ayant notamment utilisé des CNN spatiaux ont plutôt effectué les opérations séparément sur les valeurs réelles et imaginaires (Popa, 2017 ; Popa & Cernăzanu-Glăvan, 2018). Au final, les définitions exactes pour ces opérations ne semblent pas encore parfaitement élucidées pour leur utilisation qui respecteraient parfaitement les propriétés du domaine de Fourier, bien que l'intérêt réel d'approfondir cela peut justement être assez faible (Ayat *et al.*, 2019 ; Vasilyev, 2015).

Pour finir, bien qu'effectuer les calculs des couches de convolution spatiale dans le domaine de Fourier aide à réduire le nombre d'opérations et les temps de calcul, définir l'ensemble des autres couches des CNN dans le domaine de Fourier présente, de son côté, des difficultés supplémentaires. D'une part, plusieurs de ces recherches portent surtout sur des CNN spatiaux avec des entrées réelles qui sont simulées comme étant complexes et ne présentent pas de gains significatifs de performances par rapport aux modèles standards. D'autre part, les chercheurs ayant tenté de respecter les propriétés du domaine de Fourier, pour de la classification d'images, présentent surtout des expérimentations sur de petits modèles et des bases de données très simples. Ceci ne permet donc pas de clairement déterminer les performances de ces méthodes pour des utilisations à plus grande échelle. Dans tous les cas, il reste que la plupart des chercheurs

semblent au mieux atteindre des performances similaires à leur équivalent dans le domaine spatial avec des entrées réelles. D'autres recherches pourraient cependant tenter d'exploiter ce domaine pour améliorer les performances des CNN par rapport à leur forme de base, comme certains chercheurs cités dans la section précédente ont essayé de faire. Cependant, leurs recherches ont principalement porté sur des méthodes de mise en commun ou de modification des sorties des dernières couches de convolution (Chidester *et al.*, 2019 ; Rippel *et al.*, 2015 ; Ryu *et al.*, 2018 ; Zhang & Ma, 2018).

Dans les prochaines sections, nous présentons différentes méthodes d'extraction des caractéristiques qui sont utilisées avec des CNN pour de la classification d'images et toutes possèdent des représentations dans le domaine de Fourier. Pour celles qui seront présentées, elles découpent le spectre fréquentiel en plusieurs sections qui seront ensuite analysées par des CNN.

1.2 L'utilisation des ondelettes avec les CNN

1.2.1 Présentation de la décomposition en ondelettes discrètes

Une ondelette (*wavelet* en anglais) est par définition un signal oscillant de durée limitée et ayant une valeur moyenne de zéro et une norme non nulle (Matlab, 2020c). Son nom réfère au fait qu'elle correspond à une petite onde ou une brève oscillation. Les ondelettes ont initialement été créées dans le but d'avoir une transformation qui permet une localisation dans le temps et dans les fréquences des variations dans un signal, étant donné que la transformée de Fourier perd l'information relative à la localisation spatiale, une fois que les données sont mises dans le domaine fréquentiel (Wikipedia, 2020b). Par conséquent, les ondelettes s'appliquent dans le domaine temporel ou spatial à l'aide d'une convolution entre un signal d'entrée et une ondelette, laquelle agit comme un filtre permettant de mettre en évidence certaines régions du spectre fréquentiel. Les coefficients de l'ondelette sont alors calculés à l'aide d'une ondelette mère définissant la forme de base du signal utilisé ainsi qu'un paramètre relié à son échelle (*scale*) et un autre relié à sa translation. Ces paramètres permettent alors d'ajuster la transformation en ondelettes en fonction des informations qu'on souhaite obtenir. La discrétisation de ces deux

paramètres change selon la transformée en ondelettes utilisées (Matlab, 2020b), mais dans le cas des images numériques, c'est la transformée en ondelettes discrètes (DWT) qui est la plus utilisée avec des CNN de classification d'images.

La DWT est une transformation linéaire qui permet une décomposition et une reconstruction identique d'un signal au travers d'une analyse multirésolution (MRA). La décomposition sur un niveau d'un signal 1D consiste à appliquer séparément un filtre passe-bas (L) et un filtre passe-haut (H), puis à faire une décimation par deux, sur ces deux signaux. Cette décimation s'effectue en lisant une donnée sur deux ou, dans le cas d'un signal analogique, en divisant la fréquence d'échantillonnage par deux. Le signal filtré par le passe-bas représente alors une approximation du signal original alors que celui filtré par le passe-haut met en évidence les détails des coefficients. La reconstruction va, de son côté, inverser la décimation en suréchantillonnant (*oversampling*) les parties passe-bas et passe-haut qui seront ensuite filtrées par des ondelettes de reconstructions, avant que les deux parties soient remises ensemble. La DWT 2D, qui est notamment utilisée pour les images 2D, se calcule de la même façon, mais avec l'utilisation de quatre filtres, soit un filtre passe-bas (LL) et trois filtres passe-haut selon trois orientations différentes, soit l'horizontal (LH), la verticale (HL) et la diagonale (HH). La décimation utilisée lors de la décomposition s'applique également selon les deux axes et aura pour effet de diviser par deux la taille des images, selon la hauteur et la largeur.

La DWT sur plusieurs niveaux porte également le nom de banque de filtres et consiste à décomposer de manière successive la partie passe-bas à chaque étape. Cette méthode est notamment utilisée par la compression d'images JPEG2000 qui exploite la représentation très éparpillée (*sparse*) des sous-bandes des filtres passe-haut afin d'effectuer la compression des images (Ashwini, Amutha & Harini, 2017 ; Dragotti, 2015). De l'autre côté, la décomposition en paquet d'ondelettes (WPD) consiste à décomposer les parties passe-haut et passe-bas de chaque étape. Cependant, celle-ci n'est généralement pas utilisée avec des CNN de classification d'images, potentiellement en raison des représentations très éparpillées obtenues par ces sous-bandes supplémentaires des hautes fréquences. Finalement, la *dual-tree complex wavelet transform* (DTCWT) consiste essentiellement à effectuer séparément une DWT sur les valeurs

réelles et une autre sur les valeurs imaginaires (Oulhaj, Jennane, Amine, El Hassouni & Rziza, 2018). Elle fournit davantage d'orientations pour les parties passe-haut par rapport à la DWT standard et sa représentation est approximativement invariante au décalage (*shifting*) (Lu, Wang, Zhang, Won & Yoon, 2018a ; Oulhaj *et al.*, 2018). D'autres transformations d'ondelettes existent, mais elles ne sont généralement peu ou pas représentées dans la littérature avec des CNN de classification d'images 2D.

La figure 1.1 présente les représentations fréquentielles dans le domaine de Fourier des filtres de décomposition des ondelettes de Haar et de *Reverse biorthogonal 3.1* (rbio3.1) pour un à trois niveaux de DWT et de WPD, dans le cas où la décomposition s'effectue sans décimation par deux. Dans le cas de la WPD, en raison des propriétés d'associativité et de commutativité de la convolution, qui font en sorte que l'ordre des opérations n'a pas d'importance, seules les combinaisons de filtres uniques sont conservées. De l'autre côté, la figure 1.1e présente un schéma sur les régions couvertes par les sous-bandes passe-haut de la DTCWT sur deux niveaux, à l'aide de six orientations qui sont de ± 15, 45 et 75 degrés.

La prochaine section présente diverses recherches ayant utilisé des ondelettes discrètes avec des CNN qui sont le plus souvent utilisés pour de la classification d'images.

1.2.2 Présentation des méthodes de CNN avec des ondelettes

Dans la littérature, les ondelettes discrètes sont utilisées sous diverses formes avec les CNN, telles que leur utilisation pour réduire la taille des images d'entrées, la compression et la décompression des sorties des couches des CNN ou l'extraction de caractéristiques des images d'entrées ou des sorties de certaines couches. Bien que la plupart des chercheurs aient utilisé les ondelettes sous la forme de filtres fixes, certains d'entre eux ont également tenté l'utilisation de paramètres qui sont reliés à la forme des ondelettes utilisées (Cotter & Kingsbury, 2018 ; De Silva, Vithanage, Xavier, Piyatilake & Fernando, 2020 ; Rodriguez *et al.*, 2020). Il a aussi certaines recherches avec des CNN-RNN (*recurrent neural network*) (Dong, Zhang & Shao,

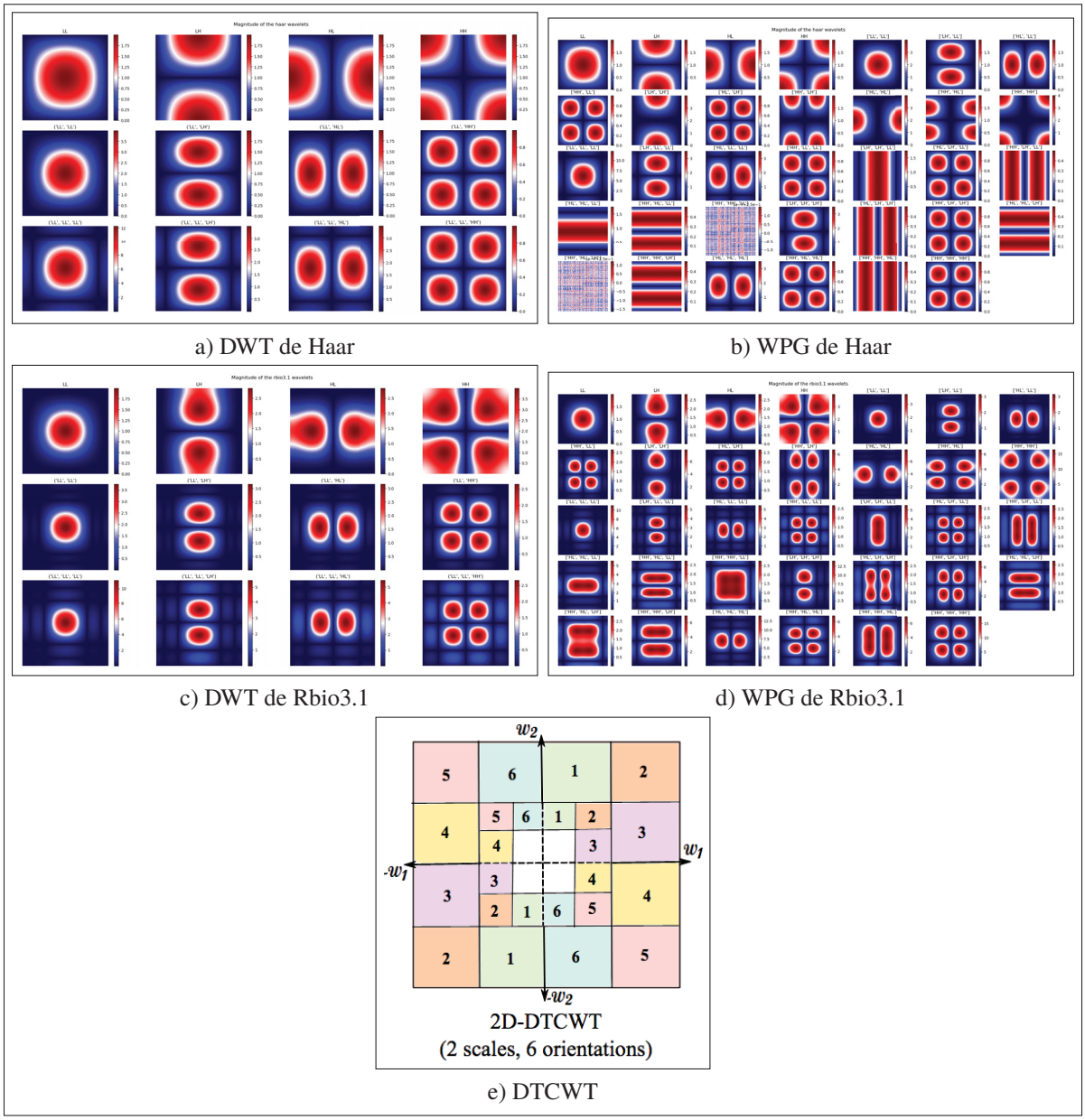


Figure 1.1 Représentation fréquentielle des filtres d'ondelettes de Haar et Rbio3.1 pour 1, 2 et 3 niveaux de DWT et de WPG ainsi que le découpage fréquentiel de la DTCWT pour 2 niveaux et 6 orientations

Adaptée avec la permission de *Springer-Verlag London Ltd., part of Springer Nature*, Oulhaj *et al.* (2018), *Copyright © 2018*

2019 ; Yin, Zhang & Shao, 2019b) et d'autres ayant utilisé la DTCWT (Cotter & Kingsbury, 2018 ; Dong *et al.*, 2019 ; Lu *et al.*, 2018a).

Certains chercheurs ont uniquement utilisé le filtre passe-bas LL de la DWT afin d'obtenir une version approximative et de taille réduite des images qui sont envoyées aux CNN, comme le ferait également une interpolation (Chen, Lin, Zuo, Luo & Zhang, 2017a ; Das & Dutta, 2019 ; Kausar, Wang, Idrees & Lu, 2019 ; Khatami et al., 2020 ; Levinskis, 2013 ; Venkataswamy, Omair Ahmad & Swamy, 2018). L'utilisation d'images plus petites leur permet surtout de réduire fortement leurs temps de calcul et les quantités de ressources utilisés, étant donné que les images seront réduites par deux, selon la hauteur et la largeur, après chacun des niveaux de décomposition. Leurs performances obtenues sont généralement similaires ou légèrement plus basses par rapport à l'utilisation des images originales (Chen *et al.*, 2017a ; Kausar *et al.*, 2019 ; Levinskis, 2013 ; Venkataswamy *et al.*, 2018). Cependant, certains d'entre eux ont plutôt obtenu de légères améliorations reliées au fait d'avoir moins de bruit et de détails dans leurs images médicales (Das & Dutta, 2019 ; Kausar *et al.*, 2019 ; Khatami *et al.*, 2020), bien que cela puisse aussi dépendre de l'ondelette utilisée (Kausar *et al.*, 2019). Cependant, dans un contexte plus large, certaines recherches avaient montré que, mis à part l'interpolation *nearest neighbour*, elles sont toutes plutôt équivalentes pour les CNN pour des images naturelles (Chrabaszcz, Loshchilov & Hutter, 2017). Il semble en être de même avec des ondelettes spécialisées pour de l'interpolation d'images (Dumic, Grgic & Grgic, 2007). En raison de cela et du fait que la plupart de ces chercheurs se servant de ces CNN n'effectuent qu'une seule exécution par mesure et qu'ils ne comparent généralement pas leurs résultats par rapport à des méthodes d'interpolation, leurs recherches présentent globalement peu d'arguments justifiant l'utilisation des ondelettes par rapport aux méthodes d'interpolation standard. De l'autre côté, certains chercheurs ont plutôt proposé des algorithmes pour mettre davantage de l'avant les arêtes des images à l'aide des ondelettes (De Silva, Fernando, Piyatilake & Karunarathne, 2019) ou des méthodes plus avancées d'interpolation avec les ondelettes (Asamwar, Bhurchandi & Gandhi, 2010 ; Asamwar *et al.*, 2010 ; Pragada & Sivaswamy, 2008 ; Soltan, El-Rabaie, El-Fishawy, Abd El-samie & Elkhamy, 2010 ; Velisavljević, 2008). Il a aussi d'autres chercheurs qui ont plutôt proposé des méthodes ayant des couches apprises pour générer des images de taille réduite et qui peuvent obtenir de meilleures performances par rapport aux ondelettes sur les images naturelles, tout en ayant des temps de calcul similaires (Chen *et al.*, 2017a). Ainsi, bien que l'utilisation telle quelle du filtre

LL de la DWT comme méthode d'interpolation des images d'entrées des CNN ne présente que peu d'arguments solides par rapport aux méthodes d'interpolation standard, d'autres façons de faire plus avancées présentent, de leur côté, des arguments plus solides. Cependant, le besoin réel d'avoir des méthodes plus avancées dépend en bonne partie de la base de données utilisée et de la tâche effectuée avec les CNN.

Par la suite, d'autres chercheurs ont plutôt tenté des méthodes de mise en commun (*pooling*) sur les sorties des couches des CNN à l'aide des ondelettes (Chaabane, Mellouli, Hamdani, Alimi & Abraham, 2017 ; Guo, Cao & Zou, 2018 ; Li, Shen, Guo & Lai, 2020 ; Liu, Zhang, Zhang, Lin & Zuo, 2018b ; Lu *et al.*, 2018a ; Rossetto & Zhou, 2019 ; Williams & Li, 2018). Le plus souvent, leur mise en commun avec les ondelettes (*wavelet pooling*) consiste à uniquement envoyer la sortie du filtre passe-bas LL de la DWT sur un niveau à la couche suivante (Chaabane *et al.*, 2017 ; Guo *et al.*, 2018 ; Li *et al.*, 2020 ; Rossetto & Zhou, 2019 ; Williams & Li, 2018). Selon certains, le fait de plutôt utiliser l'ensemble des sous-bandes (Liu *et al.*, 2018b) amènerait des coûts plus élevés et des performances moindres par rapport à une augmentation de la profondeur du modèle utilisant la méthode précédente, tout en ayant un nombre de paramètres totaux similaires (Li *et al.*, 2020). On note cependant qu'utiliser uniquement le filtre passe-bas de Haar de la DWT est équivalent à utiliser une mise en commun par valeur moyenne (*average pooling*) avec une fenêtre de 2×2 et un pas (*stride*) de 2 pixels ayant des valeurs multipliées par 2 (0,5 au lieu de 0,25) et un calcul de gradient pouvant être différèrent. Bien que certains d'entre eux se sont uniquement comparés par rapport au *max-pooling* (Chaabane *et al.*, 2017 ; Lu *et al.*, 2018a ; Rossetto & Zhou, 2019), d'autres ont également pris en compte d'autres méthodes de mise en commun ou de *down-sampling* (Guo *et al.*, 2018 ; Li *et al.*, 2020 ; Williams & Li, 2018). Selon certains chercheurs, toutes les méthodes de mises en commun seraient plutôt équivalentes sur un plan général, mais certaines peuvent être meilleures que d'autres, selon les circonstances (Williams & Li, 2018). Par ailleurs, l'ajout d'un certain seuil semble aider à améliorer les performances, bien que ces chercheurs n'avaient utilisé qu'une seule base de données et un seul modèle (Guo *et al.*, 2018). Finalement, bien que la recherche de Li *et al.* (2020) semble indiquer que l'utilisation des ondelettes puisse être meilleure que les méthodes de mise en

commun standard ou des couches de convolution avec un pas (*stride*) de deux pixels, ceux-ci ont effectué des modifications sur plusieurs de leurs modèles de références et ils n'ont pas mesuré leurs performances (LiQiufu, 2020). Par conséquent, il est difficile de correctement évaluer l'ensemble de leurs différences de performances obtenues puisque celles-ci dépendent également d'autres facteurs que l'utilisation ou non de leur méthode. Malgré tout, leurs résultats montraient notamment que toutes leurs ondelettes (dont celle de Haar) étaient plutôt équivalentes, indiquant ainsi, indirectement, qu'utiliser Haar ou des *average-pooling* serait, dans un certain sens, une bonne solution dans le cas de bases de données d'images naturelles. Par conséquent, dans un sens global, les arguments présentés dans ces différentes recherches ne semblent globalement pas assez solides pour pleinement justifier l'utilisation de leur *wavelet pooling* par rapport à d'autres méthodes habituellement utilisées. Ceci est principalement dû à un manque de comparaison et de prise en compte des différents facteurs pouvant influencer les performances obtenues, et le fait que le filtre passe-bas de Haar donne des performances similaires aux autres ondelettes, tout en étant équivalent à un *average-pooling*.

D'autre part, plusieurs chercheurs présentent des méthodes utilisant des ensembles de modèles pour analyser les sous-bandes de la DWT effectuées sur les images d'entrées (De Silva *et al.*, 2020 ; Huang & Gai, 2020 ; Rasti, Mehridehnavi, Rabbani & Hajizadeh, 2018 ; Sinha, Psaromiligkos & Zilic, 2019 ; Tran, Meunier & Cheriet, 2019 ; Williams, 2017). Plusieurs d'entre eux utilisent un CNN par sous-bandes avec le plus souvent un seul niveau de DWT (Rasti *et al.*, 2018 ; Sinha *et al.*, 2019 ; Tran *et al.*, 2019 ; Williams, 2017). D'autres chercheurs ayant utilisé plusieurs niveaux de DWT regroupent les parties hautes fréquences de chaque niveau pour les envoyer à différents CNN, tandis qu'un autre CNN analyse uniquement la partie des basses fréquences du dernier niveau de décomposition (De Silva *et al.*, 2020 ; Williams, 2017). Bien que leurs performances soient intéressantes, aucun de ces chercheurs ne semble présenter de comparaison parfaitement équitable pouvant démontrer que l'amélioration de performance est surtout reliée au fait d'avoir utilisé des ondelettes. En effet, certains d'entre eux comparent un modèle par rapport à un ensemble de plusieurs modèles identiques (Williams, 2017), alors que d'autres n'ont pas pris en compte le fait qu'une architecture ayant plusieurs chemins en

parallèle peut donner de meilleures performances qu'une seule voie commune (Xie *et al.*, 2017) (Rasti *et al.*, 2018 ; Sinha *et al.*, 2019). On note aussi que certaines recherches avaient montré qu'individuellement, les performances reliées aux sous-bandes des hautes fréquences de la DWT sont généralement bien moindres que ceux des images originales ou de celles des sous-bandes des basses fréquences (Kausar *et al.*, 2019 ; Williams, 2017). Ceci vient notamment du fait que ces filtres passe-haut donnent généralement des images aux aspects visuels fortement bruité et très éparpillé qui, bien qu'utiles pour la compression, peuvent rendre l'information plus difficilement accessible pour les CNN qui utilisent le plus souvent des filtres spatiaux de petite taille. Ainsi, il est possible que de tels ensembles de CNN utilisant uniquement des copies des images d'entrées réussissent mieux que ceux avec les sous-bandes d'ondelettes, en raison du fait que les images originales contiennent davantage d'information que les sous-bandes de la DWT. De la même manière, l'utilisation de couches de convolution standard pourrait donner de meilleures performances que les ondelettes tout en ayant des temps de calcul pouvant être similaires (Chen *et al.*, 2017a). Étant donné que ces recherches ne présentent généralement pas de comparaison permettant d'isoler la contribution relative aux ondelettes, par rapport aux restes de leur architecture, il est difficile de croire que leurs bonnes performances sont surtout reliées à leur utilisation des ondelettes et que l'utilisation d'autres procédés plus simples ne donnerait pas de meilleures performances.

Finalement, plusieurs autres façons de faire aux formes plus variées et moins représentées dans la littérature ont également été effectuées par les chercheurs. On note d'abord certains cas ayant utilisé une DWT sur plusieurs niveaux sur les images d'entrées, et dont les sous-bandes de chaque niveau sont concaténées avec les sorties des couches ayant les mêmes tailles spatiales (Fujieda, Takayama & Hachisuka, 2017,1 ; Mewada, Patel, Chaudhari, Mahant & Vala, 2020). Cependant, ceux-ci ne présentent, encore une fois, aucune comparaison par rapport à des modèles aux architectures équivalentes et sans ondelette, et ces modèles sont relativement très lourds en paramètres et se font dépasser par d'autres cas plus simples, avec ou sans analyse spectrale (Rodriguez *et al.*, 2020). De l'autre côté, d'autres chercheurs présentent plutôt des méthodes utilisant des ondelettes sur les sorties des dernières couches de convolution de leur

CNN (Li *et al.*, 2020 ; Rodriguez *et al.*, 2020), bien que certaines méthodes peuvent également s'appliquer sur les entrées d'un CNN (Rodriguez *et al.*, 2020). Cependant, ces chercheurs n'ont pas expérimenté de CNN sans ondelette aux architectures similaires et ayant la même profondeur globale ou le même nombre de paramètres. On note également que les performances obtenues par Rodriguez *et al.* (2020) avec le même CNN que Oyallon *et al.* (2017) présentaient un nombre de paramètres bien plus élevés et des pertes de performances significatives par rapport au modèle d'origine sans analyse spectrale. Ainsi, certaines comparaisons et améliorations de ces méthodes restent à venir avec ces approches plus innovantes qui s'appliquent sur les sorties des dernières couches de convolution des CNN. Finalement, d'autres chercheurs ont tenté des approches visant à remplacer une ou quelques couches de convolution (Dong *et al.*, 2019), voire l'ensemble des couches de convolution d'un CNN (Cotter & Kingsbury, 2018), en utilisant notamment la DTCWT afin d'avoir accès à davantage d'orientations pour les hautes fréquences qu'avec la DWT habituelle. Dans le cas de Cotter & Kingsbury (2018), leurs expérimentations de prototypage avec l'utilisation de paramètres pour leur DTCWT présentaient cependant généralement de légères baisses de performances par rapport à leur CNN de référence. De l'autre côté, la méthode de Dong *et al.* (2019) présente certaines améliorations de performances par rapport à leur modèle standard, mais leurs expérimentations ne permettent cependant pas de clairement séparer la contribution relative aux ondelettes par rapport à celle reliée aux autres aspects de leur architecture. Bien d'autres formes d'utilisation des ondelettes avec des CNN de classification d'images sont également possibles, mais l'ensemble de ces recherches présente au moins un certain aperçu de diverses autres façons de faire présentées dans la littérature.

En fin de compte, diverses formes d'utilisation des ondelettes avec des CNN ont été présentées par les chercheurs avec notamment de l'interpolation des images d'entrées, des méthodes utilisées sur les sorties de certaines couches de convolution et des ensembles de modèles. Cependant, de manière générale, peu d'entre eux ont clairement isolé l'apport ajouté par les ondelettes par rapport aux autres variables influençant leurs mesures, tout en montrant que d'autres procédés standard plus simples ne pourraient pas obtenir des performances similaires ou plus élevées. Étant donné que le filtre passe-bas de l'ondelette de Haar est équivalent à un *average pooling*,

plusieurs de ces recherches n'ayant utilisé que le filtre passe-bas de la DWT peuvent se rapporter à cette méthode de mise en commun standard. D'autre part, ceux ayant utilisé les différentes sous-bandes de la DWT avec des ensembles de modèles auraient potentiellement de meilleures performances avec des copies des images d'entrées qui fourniraient ainsi globalement plus d'information pour leurs CNN. On note également que la plupart de ces chercheurs ne prennent qu'une seule mesure par expérimentation et peu de ces recherches ont expérimenté leur méthode sur diverses formes d'architectures et différentes sortes de base de données. Par conséquent, il est souvent difficile d'évaluer la fiabilité de ces méthodes et leur capacité de fonctionnement dans d'autres circonstances que celles présentées dans leur recherche. Malgré tout, il est possible que de futures recherches réussissent à mieux répondre à ces diverses problématiques afin d'ainsi clairement démontrer que les ondelettes peuvent avoir un apport ajouté important qui ne peut pas être aisément substitué par des procédés plus standards. Cependant, les ondelettes possèdent certaines faiblesses telles que le nombre limité d'orientations de la DWT et le fait qu'elle est une opération linéaire pouvant donc plus difficilement substituer des opérations non linéaires. Certaines de ces faiblesses sont corrigées par les procédés qui sont présentés dans les prochaines sections.

1.3 L'utilisation des filtres et des ondelettes de Gabor avec les CNN

1.3.1 Présentation des filtres et des ondelettes de Gabor

Les filtres et les ondelettes de Gabor sont des filtres linéaires passe-bandes qui sont notamment connus pour être particulièrement appropriés pour la représentation et la discrimination de textures tout en ayant, selon certains chercheurs, des similitudes avec certaines parties du système visuel des mammifères (Daugman, 1985). Leur utilisation se fait généralement à l'aide de plusieurs filtres où chacun met en évidence des plages de fréquences spécifiques selon différentes orientations et l'ensemble de ces filtres correspond à une banque de filtres ou d'ondelettes de Gabor. On note également que de nombreux chercheurs indiquent que certains des filtres appris par la première couche de convolution des CNN ressemblent à des filtres de Gabor,

à la fin de leur apprentissage (John, Boyali & Mita, 2017 ; Luan, Chen, Zhang, Han & Liu, 2018 ; Ma, Luo & Yang, 2020 ; Ma *et al.*, 2020 ; Sarwar, Panda & Roy, 2017 ; Yosinski, Clune, Bengio & Lipson, 2014). Plusieurs chercheurs indiquent également que les filtres et les ondelettes de Gabor seraient notamment capables de tolérer certains degrés de rotation et de déformation des images et seraient relativement robustes aux variations de luminosité (Kwolek, 2005 ; Lu, Wu & Lu, 2018b ; Luan *et al.*, 2018 ; Patrick, Weyori & Mighty, 2020 ; Ren, Guo, Di, Han & Li, 2016 ; Sarwar *et al.*, 2017).

Au niveau de leurs calculs, les filtres et les ondelettes de Gabor sont définis dans le domaine spatial et s'obtiennent par la modulation d'une fonction gaussienne avec une onde sinusoïdale (ondelettes de Gabor) ou une onde plane sinusoïdale (filtre de Gabor 2D) (Fogel & Sagi, 1989 ; Wikipedia, 2020a). Les filtres obtenus sont complexes avec une partie réelle et une partie imaginaire qui représentent des directions orthogonales. Puisque les valeurs de ces filtres sont complexes, certains chercheurs les utilisant avec des CNN vont notamment se servir de la norme (la magnitude) du résultat obtenu après la convolution des filtres avec les images (Yang, Wu & Zhang, 2018), tandis que d'autres utilisent uniquement la partie réelle des filtres (Hosseini, Lee, Kwon, Koo & Cho, 2018 ; John *et al.*, 2017 ; Liu, Ding, Wang & Zhang, 2018a ; Luan *et al.*, 2018 ; Ma *et al.*, 2020 ; Pandya et al., 2018 ; Patrick *et al.*, 2020). Cependant, pour leur utilisation avec des CNN pour de la classification d'images, aucun de ces chercheurs ne semble avoir utilisé, par exemple, les parties réelles et imaginaires séparément ou les utiliser sous un format polaire, comme le font certains chercheurs utilisant d'autres méthodes spectrales présentées dans cette revue de littérature. On note également que la très grande majorité des chercheurs qui les utilisent avec des CNN se servent des filtres de Gabor et rares sont ceux utilisant plutôt des ondelettes de Gabor (Lu *et al.*, 2018b).

La figure 1.2 illustre un exemple de découpage fréquentiel effectué par une banque de filtres de Gabor réelle avec six orientations et quatre fréquences. En raison du fait que ces filtres de Gabor sont réels, il y a une symétrie entre les deux côtés de l'axe imaginaire vertical. La section suivante présente différentes utilisations des filtres et ondelettes de Gabor avec des CNN pour de la classification d'images.

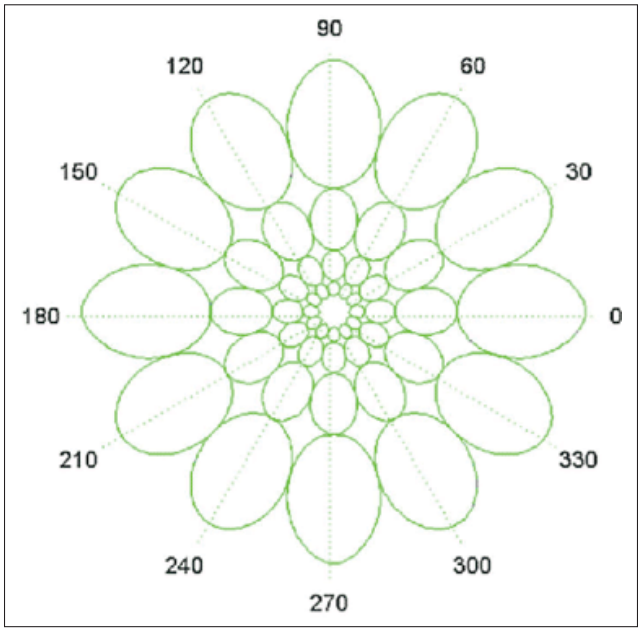


Figure 1.2 Exemple de représentation du découpage fréquentiel effectué par une banque de filtres de Gabor réelle avec 6 orientations et 4 fréquences
Adaptée avec la permission de l' *International Federation for Medical and Biological*, Nava *et al.* (2014), *Copyright © 2014*

1.3.2 L'utilisation des filtres et ondelettes de Gabor avec les CNN

Dans l'ensemble, les chercheurs utilisant les filtres ou les ondelettes de Gabor avec des CNN pour de la classification d'images semblent, soit les utiliser comme une méthode d'extraction des caractéristiques qui vise à remplacer ou modifier la première couche de convolution des CNN, soit les utiliser pour initialiser ou influencer l'apprentissage de l'ensemble ou d'une partie des couches de convolution. Pour la suite des choses, afin de simplifier la nomenclature étant donné que la majorité d'entre eux utilisent des filtres de Gabor, c'est ce terme qui sera employé par la suite, dans le présent texte, étant donné que l'utilisation des filtres et des ondelettes de Gabor avec des CNN semble être globalement identique.

Plusieurs chercheurs utilisent les filtres de Gabor comme une méthode d'extraction des caractéristiques des images, ce qui est équivalent à une première couche de convolution ayant des poids fixes et initialisés à ceux des filtres de Gabor (Chen et al., 2017b ; Hosseini *et al.*,

2018 ; Kwolek, 2005 ; Lu *et al.*, 2018b ; Xiong, Lu & Jiang, 2018 ; Yao, Chuyi, Dan & Weiyu, 2016). Certains d'entre eux ont alors effectué une comparaison équitable entre l'utilisation d'une couche de convolution standard aux poids aléatoires, mais modifiés durant l'apprentissage, et l'utilisation des filtres de Gabor aux poids fixes (Huang et al., 2019 ; Sarwar *et al.*, 2017). Ceux-ci semblaient surtout indiquer que l'utilisation des filtres de Gabor pour la première couche de convolution pouvait améliorer les performances, notamment dans le cas de petites bases de données d'apprentissage (Huang *et al.*, 2019), mais semblait généralement causer des baisses de performances sur des bases de données de plus grande taille (Sarwar *et al.*, 2017). D'autres se sont également servis des filtres de Gabor avec l'utilisation d'autres procédés, tels qu'une couche de mise en commun (Patrick *et al.*, 2020) ou un algorithme d'encodage (Ren *et al.*, 2016). Cependant, ceux-ci n'effectuent pas nécessairement une comparaison équitable par rapport à un CNN standard (Patrick *et al.*, 2020). Aussi, de manière plutôt logique, l'ajout d'une couche de convolution aux poids fixes devant le modèle améliore le plus souvent les performances. Cependant, leur comparaison par rapport à leur CNN de référence devient alors inéquitable, notamment en raison d'une couche de convolution supplémentaire et d'un nombre de canaux d'entrée largement plus élevé pour la première couche de convolution par rapport au CNN standard (Hosseini *et al.*, 2018 ; Lu *et al.*, 2018b ; Xiong *et al.*, 2018). Il est alors difficile de savoir si l'amélioration des performances vient réellement des filtres de Gabor ou si, par exemple, l'utilisation de plusieurs copies des images d'entrées ou d'une couche de convolution supplémentaire, suffirait à améliorer les performances.

D'autres chercheurs ont par la suite repris le concept afin d'utiliser les filtres de Gabor comme une méthode d'initialisation des poids de la première couche de convolution (Adjobo, Mahama, Gouton & Tossa, 2019 ; John *et al.*, 2017). Pour ceux-ci, ils semblent que l'utilisation des méthodes d'apprentissage habituelles serait suffisante, bien que certains ont tenté des versions plus complexes visant à contraindre les poids à rester proches de leur forme initiale (John *et al.*, 2017). De manière générale, leurs résultats montraient que ces méthodes d'initialisation des poids de la première couche de convolution pouvaient être potentiellement meilleures que celles habituelles, surtout pour les petites bases de données d'apprentissage. À l'opposé, d'autres ont

montré qu'il était également possible d'utiliser des filtres spatiaux aux formes prédéfinies dans la première couche et d'apprendre uniquement les paramètres liés à la génération des filtres, tels que ceux utilisés dans les équations mathématiques des filtres de Gabor (Ma *et al.*, 2020). Il n'est alors plus nécessaire d'effectuer de l'optimisation des hyperparamètres définissant les filtres aux formes prédéfinies utilisées, étant donné que c'est le modèle qui les déterminera au cours de son apprentissage. Leurs expérimentations semblaient montrer que cette façon de faire fonctionnait aussi bien avec les petites que les grandes bases de données, et même avec des modèles pré-entraînés sur de grandes bases de données d'apprentissage. Malgré tout, cette façon de faire était uniquement appliquée à la première couche de convolution et elle peut nécessiter une combinaison de plusieurs filtres aux formes prédéfinies. Le choix de ces formes et la quantité de chacune d'entre elles peuvent parfois nécessiter des ajustements et le besoin d'effectuer plusieurs expérimentations avant de trouver la bonne configuration à utiliser.

D'autres ont également utilisé des filtres de Gabor dans l'ensemble ou dans une partie des couches de convolution de leur modèle. Dans certains cas, les filtres de Gabor sont utilisés comme un moyen de réduire le nombre de paramètres du modèle, tout en ayant une baisse de performances maximale résultante autour de 1 à 2 %. Pour ce faire, ils utilisent des poids fixes pour la première couche et un mixte de poids appris et de poids fixes pour la deuxième couche de convolution (Sarwar *et al.*, 2017). De l'autre côté, d'autres ont plutôt utilisé des filtres qui résultaient de la multiplication des poids appris des couches de convolution, avec des filtres fixes de Gabor (Huang *et al.*, 2019 ; Liu *et al.*, 2018a ; Luan *et al.*, 2018). Sur un plan théorique, les filtres qui sont alors utilisés sont le résultat d'une opération *AND* entre les filtres appris et ceux de Gabor, résultant ainsi en des filtres de Gabor aux poids modifiés et adaptés en fonction de ceux appris par le CNN. Bien que cette façon de faire peut donner de meilleures performances, autant pour les petites que les grandes bases de données d'apprentissage, elle a le principal défaut de générer un très grand nombre de filtres et le besoin d'utiliser des convolutions 3D dans le cas de l'utilisation d'images 2D (Liu *et al.*, 2018a ; Luan *et al.*, 2018). Ceci vient du fait que chacun des filtres de la couche de convolution sera multiplié par chacun des filtres de Gabor, générant ainsi un nombre de filtres total relativement très élevé et un besoin d'adapter

l'ensemble de l'architecture en conséquence. D'autres façons de faire reliées à l'utilisation des filtres de Gabor avec les CNN existent, mais ces recherches qui ont été ici présentées donnent un bon aperçu des différentes façons de faire qui ont été relevées dans la littérature.

En fin de compte, plusieurs façons de faire sur l'utilisation des filtres de Gabor avec des CNN pour de la classification d'images ont été couvertes dans la littérature. Bien que certaines d'entre elles les utilisent uniquement comme une méthode d'extraction des caractéristiques des images d'entrées (Hosseini *et al.*, 2018 ; Lu *et al.*, 2018b ; Xiong *et al.*, 2018), d'autres sont allés plus loin en répondant chaque fois à certaines interrogations reliées à d'autres recherches. En effet, certains chercheurs ont d'abord montré qu'on pouvait utiliser des filtres de Gabor fixes en remplacement de la première ou de certains filtres de la deuxième couche de convolution. D'autres ont ensuite poussé le concept plus loin en les utilisant comme une méthode d'initialisation des poids de la première couche de convolution, avant d'ensuite proposer un apprentissage effectué sur les hyperparamètres des filtres de Gabor utilisés afin de ne plus avoir à manuellement les définir. De l'autre côté, d'autres se sont plutôt penchés sur leur utilisation dans l'ensemble du CNN dans le but d'ainsi utiliser des filtres de Gabor qui seront influencés par ceux appris par le CNN. Il est également possible que de futures recherches poursuivent cette façon de faire en y ajoutant un apprentissage sur les filtres de Gabor utilisés au lieu de les laisser avec des formes fixes. Sous une vue d'ensemble, les filtres de Gabor semblent donc pouvoir améliorer les CNN, autant pour les petites que les grandes bases de données d'apprentissage, et peuvent également s'intégrer avec des CNN pré-entraînés sur de grandes bases de données d'apprentissage.

Bien que toutes ces recherches soient très intéressantes, un grand nombre d'entre elles présentent certaines lacunes méthodologiques importantes, dont le fait de n'avoir qu'une seule exécution effectuée par expérimentation, ou ne pas avoir nécessairement fait une comparaison parfaitement équitable par rapport à un modèle standard. Étant donné que la plupart de ces modèles hybrides présentent de faibles écarts de performance par rapport à leurs modèles de référence, il est souvent difficile de déterminer si ces gains de performances sont réellement significatifs. Ceci est d'autant plus vrai pour les expérimentations de petites bases de données où les écarts-types des mesures peuvent être relativement élevés même pour des taux de précision de 90 à 100

%. On note également que rares sont ceux qui ont expérimenté leurs méthodes sur plusieurs architectures ou formes d'architecture, donnant alors l'impression qu'il est possible que celles-ci ne puissent fonctionner qu'avec ce type d'architecture ou ce type de modèle. Malgré tout, il semble que l'utilisation des filtres de Gabor avec les CNN pour de la classification d'images soit un sujet plutôt prolifique et certaines recherches présentent de bonnes méthodologies. Il est donc possible que certaines façons de faire deviennent des méthodes standard d'utilisation avec les CNN dans le futur.

1.4 L'utilisation de la *shearlet transform* avec les CNN

1.4.1 Présentation de la *shearlet transform*

La *shearlet transform* est une méthode d'analyse multi-échelle qui a été introduite par Guo, Kutyniok & Labate (2006) et qui a surtout été présentée comme une extension des ondelettes ayant une implémentation plus simple que les *curvelets*, tout en étant *rotation-invariant*. Pour l'analyse et le traitement des images, celles-ci sont avantageuses par rapport aux ondelettes, notamment en raison de son encodage efficace des caractéristiques anisotropes, telles que des arêtes courbées, ce que les ondelettes ne sont pas capables d'efficacement représenter, en raison de leur propriété isotrope et de leur nombre limité d'orientations (Easley, Labate & Lim, 2008 ; Guo *et al.*, 2006 ; Hermessi, Mourali & Zagrouba, 2018 ; Matlab, 2020d ; Wikipedia, 2020c). Par rapport aux *curvelets* qui utilisent des rotations, les *shearlet* se servent de cisaillements comme le font également certaines implémentations numériques des *curvelets* (Guo *et al.*, 2006). Cependant, les *curvelets* seraient, selon certains chercheurs, moins appropriées pour des images naturelles parce que ce type d'images possède des détails à tous les niveaux d'échelles (Wikipedia, 2018). La *shearlet transform* est notamment utilisée dans diverses applications de traitement d'images comme le débruitage, la compression, la restauration, la séparation d'images et l'extraction des caractéristiques (Matlab, 2020d ; Shearlab, 2020).

Quant à son calcul, la *shearlet transform* possède une forme continue ainsi qu'une forme discrète qui discrétise celle continue, et elles sont toutes les deux utilisables pour des calculs numériques.

Son calcul repose notamment sur l'utilisation d'une ondelette mère (ex. une ondelette de Meyer), de matrices et d'opérations de dilatation, de cisaillement et de translation (Guo *et al.*, 2006) qui permettent de faire un découpage trapézoïdal du domaine fréquentiel, selon différentes orientations et échelles. En raison de certains défauts reliés à la forme classique, la plupart des chercheurs utilisent la *cone-adapted shearlets* qui a pour principe de séparer les basses et les hautes fréquences à chaque niveau de décomposition, afin d'effectuer un découpage selon plusieurs orientations uniquement sur la partie des hautes fréquences. La partie des basses fréquences sera, de son côté, ensuite utilisée pour être décomposée à nouveau par le même processus au niveau suivant (Easley *et al.*, 2008). On note également que le nombre d'orientations utilisé peut varier entre chacun des niveaux de décompositions, où un nombre plus élevé d'orientations peut être utilisé pour les niveaux de décomposition plus élevés (Hermessi *et al.*, 2018 ; Poojitha & Sharma, 2019 ; Rezaeilouyeh, Mollahosseini & Mahoor, 2016). Les différentes sous-bandes obtenues peuvent ensuite être utilisées de différentes manières comme celles qui seront présentées dans la prochaine section qui porte sur leur utilisation avec des CNN de classification d'images.

La figure 1.3 présente des exemples de découpages fréquentiels effectués par la *cone-adapted shearlets transform* pour des entrées réelles qui possèdent donc une symétrie entre les deux côtés de l'axe horizontal et imaginaire ξ_2. La figure de gauche présente un découpage pour un niveau de décomposition tandis que celle de droite l'est pour trois niveaux de décomposition. Dans la figure de gauche, la zone R représente les basses fréquences tandis que les zones C_h et C_v sont pour les hautes fréquences.

1.4.2 L'utilisation de la *shearlet transform* comme méthode d'extraction des caractéristiques pour des CNN de la classification d'images

L'utilisation de la *shearlet transform* avec des CNN utilisés pour de la classification d'images semble relativement rare par rapport à d'autres transformations comme les ondelettes, les filtres et ondelettes de Gabor, ou la *scattering transform* qui sera présentée par la suite. Différentes méthodes ont été utilisées par les chercheurs pour l'utilisation de la *shearlet transform* avec

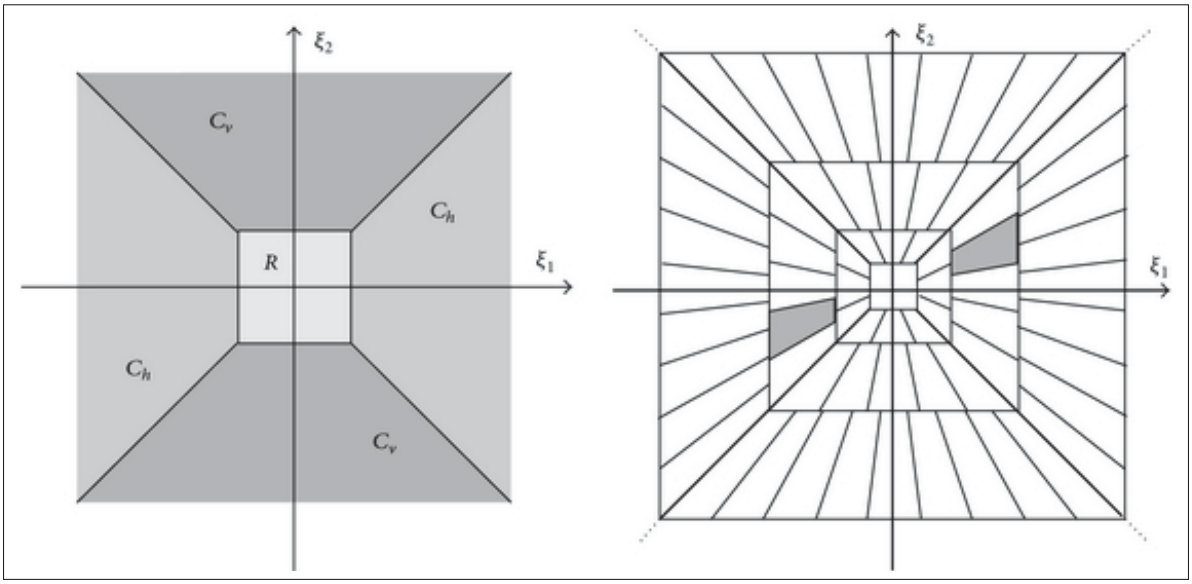


Figure 1.3 Exemple de découpage fréquentiel en 2D effectué par la *cone-adapted shearlet transform* avec à gauche un découpage selon 1 niveau et à droite un découpage selon 3 niveaux
Tirée de Matlab (2020d)

des CNN de classification, telle que l'utilisation d'ensemble de modèles analysant séparément les magnitudes et phases de chaque sous-bande (Rezaeilouyeh *et al.*, 2016), leur utilisation dans les premières couches de convolution (Yang, Yang, Lu & Liu, 2020) ou la mise en tenseur de l'ensemble des sous-bandes pour les envoyer en entrée (Poojitha & Sharma, 2019), avec parfois les images originales (Jabason, Ahmad & Swamy, 2018). Étant donné que les coefficients obtenus par la *shearlet transform* sont généralement complexes, plusieurs chercheurs choisissent le format polaire afin d'utiliser la magnitude et la phase des coefficients (Poojitha & Sharma, 2019 ; Rezaeilouyeh *et al.*, 2016), tandis que d'autres semblent utiliser la forme spatiale des images traitées (Yang *et al.*, 2020).

Cependant, plusieurs d'entre eux n'effectuent pas de comparaison parfaitement équitable par rapport à des CNN sans la *shearlet transform*, ce qui peut nécessiter d'avoir au moins deux modèles de références au lieu d'un seul. En effet, plusieurs d'entre eux se limitent à un modèle de référence ne possédant pas la *shearlet transform* et qui sert donc à vérifier si la transformation améliore leur modèle d'origine sur certains aspects. Cependant, celui-ci aura, par exemple,

moins de canaux d'entrées pour ces couches de convolution (Jabason *et al.*, 2018 ; Yang *et al.*, 2020) ou sera comparé par rapport à un ensemble de plusieurs modèles (Rezaeilouyeh *et al.*, 2016). L'utilisation d'un deuxième modèle de référence peut alors servir à vérifier si cette transformation apporte de meilleures performances par rapport à une répétition des images d'entrées (Jabason *et al.*, 2018) ou l'utilisation de couches de convolutions ayant autant de filtres que la transformation choisit (Yang *et al.*, 2020). Dans le cas d'ensemble de modèles, un autre ensemble de modèles utilisant uniquement les images originales peut être utilisé pour avoir une comparaison plus équitable (Rezaeilouyeh *et al.*, 2016). Il est également intéressant de noter que dans les expériences de Poojitha & Sharma (2019), leurs modèles utilisant la *shearlet transform* avaient obtenu les pires performances parmi leurs quatre modèles, incluant notamment deux modèles ayant les images d'origines. Aussi, ceux-ci indiquaient que l'utilisation d'un RNN-LSTM (*long short-term memory*) semblait préférable par rapport à l'utilisation d'un CNN pour traiter les sous-bandes de la *shearlet transform*.

En somme, l'utilisation de la *shearlet transform* avec des CNN pour de la classification d'images semble rare par rapport aux autres transformations ayant des points en communs avec les ondelettes. Plusieurs d'entre eux n'ont cependant pas effectué de comparaison pouvant prouver que cette transformation pouvait être réellement meilleure par rapport à d'autres façons de faire nécessitant uniquement les images originales, comme cela a également été souvent le cas avec les ondelettes. Le fait qu'il semble y avoir assez peu de modèles hybrides de classification avec cette transformation peut cependant être un signe que l'utilisation d'autres transformations pourrait être préférable pour ce type de tâche avec des CNN. Malgré tout, des modèles de CNN hybrides utilisant la *shearlet transform* sont utilisés dans plusieurs autres tâches impliquant des images, telles que : la fusion d'images (Feng, Lu, Bai, Cao & Yin, 2020 ; Hermessi *et al.*, 2018 ; Hou, Zhou, Nie, Liu & Ruan, 2019 ; Vishwakarma, Bhuyan & Iwahori, 2018), le débruittage (Shahdoosti & Rahemi, 2019), la restauration d'images (Goossens, Luong & Philips, 2019), la *densely-sampled light field (DSLF) reconstruction* (Gao, Bregovic, Koch & Gotchev, 2020) ou la détection d'arêtes (Andrade-Loarca, Kutyniok, Oktem & Petersen, 2019).

1.5 L'utilisation de la *scattering transform* avec les CNN

1.5.1 Présentation de la *scattering transform*

La *wavelet scattering transform* est une transformation qui a été introduite par Mallat (2012) et qui donne une représentation qui possède de nombreux points en commun avec les premières couches de convolution des CNN, à la principale différence d'utiliser des filtres fixes qui n'ont pas besoin d'être adapté en fonction de la base de données ou de leur tâche d'utilisation. Cette représentation possède cependant quelques hyperparamètres qui peuvent nécessiter certaines optimisations en fonction de la tâche, bien que plusieurs recherches utilisent les mêmes valeurs. Aussi, contrairement à, par exemple, les ondelettes, les ondelettes de Gabor et la *shearlet transform*, cette transformation possède également une opération non linéaire utilisée au cours des étapes de la transformation et non, par exemple, uniquement à la fin.

La *scattering transform* se calcule en plusieurs niveaux où les sorties d'un niveau deviennent les entrées du niveau suivant. À chaque niveau, on effectue d'abord la convolution des entrées avec une famille d'ondelettes complexes, suivie du calcul de leur module (leur magnitude). Les coefficients du niveau actuel sont alors obtenus en y appliquant un filtre passe-bas gaussien ayant une fenêtre spatiale d'échelle 2^J. Cependant, ce sont les coefficients obtenus avant l'application du filtre passe-bas qui sont utilisés pour le niveau suivant. On répète donc ensuite ce processus pour les niveaux qui suivent.

Les coefficients du niveau (ou ordre) 0 sont obtenus par la convolution du signal d'entrée initial avec le filtre passe-bas. Pour la famille d'ondelettes, celle-ci est généralement obtenue à l'aide d'une ondelette mère de Morlet qui a été dilatée et mise selon différentes orientations, de manière à obtenir un total de J niveaux d'échelle (ou octaves) et L orientations entre $[0, 2\pi]$ (Andreux et al., 2020 ; Bruna & Mallat, 2013 ; Matlab, 2020a ; Oyallon, Mallat & Sifre, 2013 ; Oyallon *et al.*, 2017). Ceci permet alors de faire un découpage de fréquences qui est présenté dans la figure 1.4. Étant donné la forme circulaire du découpage sur des images qui sont généralement mises sous une forme carrée, les très hautes fréquences dans les coins ne sont pas prises en

compte par la transformation. Aussi, étant donné qu'à mesure qu'on effectue des niveaux de décomposition, l'énergie des coefficients de la transformée converge vers zéro, leur niveau d'énergie devient inférieur à 1 % après le deuxième niveau de décomposition (Bruna & Mallat, 2013). Par conséquent, la plupart des chercheurs se limitent à maximum deux niveaux de décomposition et utilisent alors tous les coefficients obtenus par les niveaux d'ordre 0, 1 et 2, ce qui donne un total de $3(1 + JL + \frac{1}{2}J(J-1)L^2)$ sous-bandes pour une image d'entrée en couleur où l'on applique la *scattering transform* séparément pour chacun des canaux (Oyallon *et al.*, 2017). Malgré tout, d'autres chercheurs ont également réussi à obtenir d'excellents résultats en utilisant uniquement les coefficients d'ordre 0 et 1, ce qui permet de notamment diminuer le nombre de sous-bandes et les temps de calcul (Oyallon *et al.*, 2018). Aussi, en raison des sous-échantillonnages effectués, les images obtenues auront une taille spatiale divisée par 2^J selon les axes de la hauteur et de la largeur (Bruna & Mallat, 2013 ; Oyallon *et al.*, 2017).

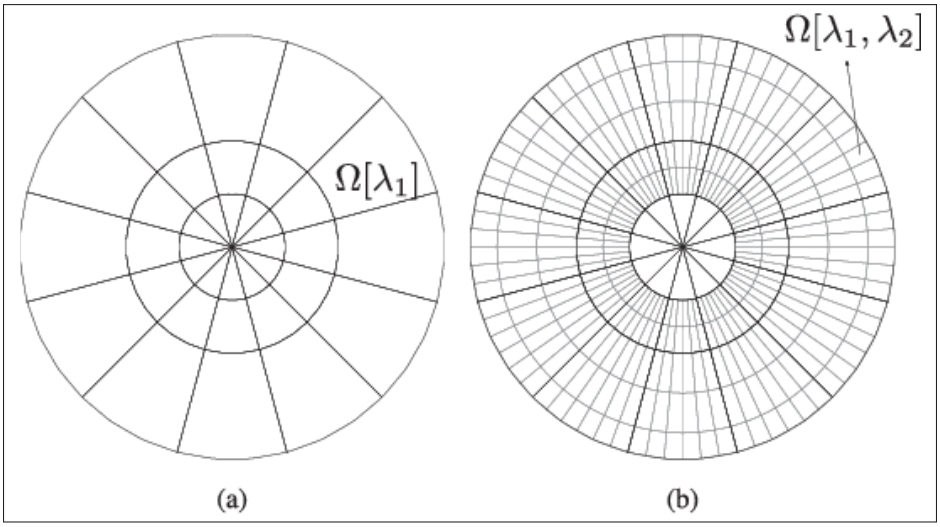


Figure 1.4 Schéma du découpage fréquentiel effectué par la *scattering transform* pour le premier niveau (a) et le deuxième (b) niveau de décomposition avec l'utilisation de 6 angles entre 0 et π (L = 12) et 3 niveaux d'échelle (J = 3)
Tirée de Bruna & Mallat (2013) *Copyright © 2013, IEEE*

Les propriétés principales que possèdent la *scattering transform* sont notamment le fait d'être invariant aux translations de tailles inférieures à 2^J, l'obtention de coefficients plus discriminants

que ceux obtenables avec Fourrier à partir du deuxième niveau de décomposition, le fait que c'est une représentation qui linéarise les petites déformations des images (condition de Lipschitz) et qu'elle est non expansive et presque complète (Bruna & Mallat, 2013 ; Mallat, 2012 ; Oyallon *et al.*, 2017). Aussi, la consommation de mémoire et les temps de calcul des modèles hybrides peuvent être significativement plus faibles par rapport à leur version de référence sans cette transformation, surtout dans le cas où l'on se limite aux ordres 0 et 1 (Oyallon *et al.*, 2018).

La figure 1.4 présente un schéma du découpage fréquentiel effectué par la *scattering transform* pour le premier et le deuxième niveau avec l'utilisation de six angles entre $[0, \pi]$ (L = 12) et trois niveaux d'échelle (J = 3). La prochaine section présente les différentes recherches ayant utilisé cette transformation avec des CNN pour de la classification d'images.

1.5.2 La *scattering transform* comme méthode d'extraction des caractéristiques des images d'entrées des CNN

De manière générale, la plupart des chercheurs utilisent la *scattering transform* comme une méthode qui vise à remplacer les premières couches de convolution des CNN en étant l'équivalent de quelques couches de convolution aux poids fixes (Manjusha, Kumar & Soman, 2018 ; Oyallon *et al.*, 2017,1 ; Ren *et al.*, 2019). L'un de leurs objectifs est de mesurer les bienfaits que peuvent apporter la *scattering transform* par rapport à l'utilisation de couches apprises. Leurs recherches montrent jusqu'à maintenant que leur utilisation avec des CNN apporte surtout des gains de performances dans le cas de bases de données d'apprentissage de petite taille ou ayant des caractéristiques simples et géométriques, comme dans le cas de bases de données d'images manuscrites (Bruna & Mallat, 2013 ; Oyallon & Mallat, 2015 ; Oyallon *et al.*, 2017,1).

Cependant, ces méthodes donnent de moins bons résultats lorsque ces conditions ne sont pas respectées, comme lors de l'utilisation de grandes bases de données d'images naturelles comme CIFAR ou ImageNet (Deng et al., 2009) (Oyallon *et al.*, 2017,1 ; Ren *et al.*, 2019 ; Zheng & Zhu, 2019). Pour ces bases de données, les modèles de références sans la *scattering transform* obtiennent, à chaque fois, de meilleures performances que ceux avec cette transformation, même lorsque les modèles hybrides ont davantage de paramètres. D'autre part, il semble que l'utilisation

de la *scattering transform* d'ordre deux avec le CNN de Oyallon *et al.* (2017) ne soit pas adéquat pour de petites bases de données de textures comme KTH-TIPS2-b (Mallikarjuna et al., 2006) (Rodriguez *et al.*, 2020). Bien que cela semble surtout relié à des problèmes de surapprentissage (*overfitting*), le fait que des modèles ayant bien plus de paramètres aient obtenu de meilleurs performances indiquent également que les couches de convolution était, d'une certaine manière, plus efficace que cette transformation. Cependant, davantage d'expérimentations serait nécessaire pour vérifier cette hypothèse. On note également qu'un grand nombre de recherches relatives à la *scattering transform* ont porté sur son utilisation sans des CNN et avec, à la place, l'utilisation de méthodes de réduction du nombre de coefficients ou d'encodage, suivie d'un classificateur. Cependant, leurs performances dépassent rarement celles que peuvent atteindre des CNN ou des CNN hybrides, surtout lorsque la base de données d'apprentissage est relativement grande et que les caractéristiques des images sont plus complexes, comme avec des images naturelles (Bruna & Mallat, 2013 ; Dadashnialehi, Bab-Hadiashar & Hoseinnezhad, 2017 ; Oyallon & Mallat, 2015 ; Oyallon *et al.*, 2013,1 ; Zarka, Thiry, Angles & Mallat, 2019).

Par conséquent, en ce qui concerne les performances obtenues, l'utilisation de couches de convolutions apprises semble plus efficace dans de telles situations en raison de leur capacité à aller chercher des informations que la *scattering transform* ne peut pas obtenir. Malgré tout, il est possible que de futures recherches sur des modèles hybrides utilisant la *scattering transform* présentent de meilleures performances que leurs modèles équivalents sans celle-ci. Ceci pourrait notamment être effectué avec, par exemple, l'utilisation d'autres ondelettes complexes que celle de Morlet (Dadashnialehi *et al.*, 2017 ; Ren *et al.*, 2019), des modifications sur la manière d'effectuer cette transformation (Oyallon & Mallat, 2015 ; Sifre & Mallat, 2013), ou des modèles intégrant autrement la *scattering transform*, comme certains de ceux présentés pour les ondelettes discrètes (section 1.2) ou les ondelettes et filtres de Gabor (section 1.3).

1.6 Conclusion sur la revue de littérature

En somme, diverses recherches ont été effectuées dans la littérature par rapport à l'utilisation directe du domaine de Fourier avec les CNN ainsi que sur l'utilisation de certains procédés

connexes à ce domaine et qui sont utilisés pour des CNN hybrides avec notamment, l'utilisation des ondelettes discrètes, des filtres et des ondelettes de Gabor, de la *shearlet transform* ou de la *scattering transform*.

Bien que ces procédés avec des CNN hybrides soient à la base des méthodes d'extraction de caractéristiques, les chercheurs ont tenté de prouver que celles-ci pouvaient améliorer les performances des CNN de classification d'images en les utilisant de différentes manières avec leurs modèles. Cependant, dans les faits, peu d'entre eux y sont réellement arrivés. En effet, bien que la plupart présentent des résultats indiquant des améliorations de performances, un grand nombre de leurs recherches ont des faiblesses méthodologiques importantes. Ces erreurs concernent le plus souvent le fait que les comparaisons par rapport à leurs modèles de référence n'isolent pas toujours les impacts liés à leur méthode par rapport aux autres facteurs également présents et qui concerne le plus souvent des architectures différentes entre les deux modèles. Dans certains cas, d'autres procédés standards et plus simples que ceux proposés pourraient donner des résultats similaires, voire même meilleurs, sans pour autant nécessairement utiliser beaucoup plus de ressources. À cela s'ajoute également le fait que peu de chercheurs ont effectué plus d'une mesure par configuration ou ont effectué des expérimentations avec diverses formes d'architectures et divers types de bases de données d'images. Or, sans cela, il est souvent difficile de déterminer si leurs mesures sont fiables et reproductibles et si leurs méthodes obtiendraient de bonnes performances dans d'autres contextes que celui de départ. Malgré tout, certaines recherches avec notamment les filtres et ondelettes de Gabor ainsi que la *scattering transform*, ont su dépasser ces faiblesses méthodologiques. Bien que la *scattering transform* semble surtout appropriée pour les bases de données d'apprentissage simples ou de taille limitée, il est possible que, dans le futur, des chercheurs obtiennent de bonnes performances avec les CNN, même dans le cas des bases de données d'images naturelles de grande taille. Dans le cas des filtres et ondelettes de Gabor, les chercheurs les ont utilisés de diverses manières avec leur modèle et il est possible que certains d'entre eux deviennent un jour de nouveaux standards pour l'utilisation des CNN de classification d'images.

De l'autre côté, pour les recherches ayant directement utilisé le domaine de Fourier, plusieurs d'entre elles visent à notamment mieux comprendre les CNN, à effectuer des attaques en fréquences ou réduire la consommation des ressources des modèles. Ce sont donc surtout les chercheurs ayant étudié des couches de mises en commun avec des filtres idéaux carrés ou l'utilisation de la magnitude de la sortie de la dernière couche de convolution, qui ont tenté des approches pour améliorer les performances de leur CNN de classification d'images à l'aide du domaine de Fourier. Cependant, les premiers seraient probablement mieux avec des filtres qui ne causent pas de *ringing artefact*, alors que les seconds ne prennent pas en compte le nombre de paramètres beaucoup plus élevé que peut avoir la première couche entièrement connectée (FC) vu que leur méthode peut remplacer une couche de mise en commun global (*global pooling*). Il y reste donc une grande place à de l'innovation pour des couches qui exploiteraient le domaine de Fourier non pas pour reproduire celle du domaine spatial, mais pour effectuer des opérations qui seraient difficiles, voire impossibles, à effectuer dans le domaine spatial. L'idéal serait également d'avoir des méthodes où l'architecture des modèles n'a pas à être modifiée et qui peuvent être utilisées autant avec des modèles aux poids aléatoires, qu'avec ceux ayant des poids pré-entraînés sur de grandes bases de données d'apprentissage. En effet, ces derniers sont notamment souvent utilisés pour les petites bases de données ayant beaucoup de surapprentissages, mais relativement peu de recherches ont réussi à les améliorer en modifiant leur entrée au lieu de leur sortie (Ma *et al.*, 2020).

Afin de répondre à plusieurs de ces problématiques, cette recherche présente trois nouvelles méthodes de prétraitement des entrées des CNN qui ne nécessitent aucune modification de l'architecture ou des algorithmes d'apprentissage liés à leurs poids. Celles-ci ont été expérimentées avec un total de 14 architectures uniques pour de la classification d'images naturelles sur CIFAR, de texture sur DTD, ainsi que des émotions du visage sur KDEF, tout en ayant toujours cinq ou dix mesures par configuration expérimentée.

Ces méthodes sont notamment basées sur deux volets des recherches de la littérature. Le premier est le fait que l'ensemble du spectre fréquentiel est utilisé par les CNN, mais que les régions principalement utilisées peuvent changer selon l'architecture et la base de données exploitées

(Tsuzuku & Sato, 2019 ; Yin *et al.*, 2019a) (voir Annexe I). Le second volet est le fait que, selon certaines recherches, il semble que les réseaux de neurones apprennent d'abord les basses fréquences avant de lentement tendre vers l'apprentissage des hautes fréquences (Rahaman *et al.*, 2019 ; Xu *et al.*, 2019 ; Xu, 2018). En prenant en compte ces deux volets et une volonté de ne pas avoir plusieurs versions filtrées des images d'entrées, ces méthodes visent à notamment atténuer les basses fréquences afin que les CNN puissent les apprendre plus rapidement et accèdent donc plus aisément aux informations liées aux moyennes et hautes fréquences. Ceci devrait alors leur permettre de converger mieux et plus rapidement tout en étant moins affecté par les variations des couleurs des basses fréquences qui ne représentent qu'une portion de ce dont les CNN ont réellement besoin.

La première méthode utilise les images d'entrées pour y soustraire les informations liées à l'autocorrélation de leur magnitude à l'aide d'un filtre correspondant à leur magnitude inversée. Ceci permet d'effectuer un filtrage propre au spectre fréquentiel de chaque image, mais qui prend également en compte les relations entre l'ensemble des dimensions des entrées utilisé. Cette seconde portion est gérée par une nouvelle couche de normalisation fréquentielle ainsi qu'une couche de moyennage des dimensions. Étant dans le domaine de Fourier, ces couches prennent en compte les relations entre les fréquences des entrées et peuvent être considérées comme étant des modifications globales selon l'ensemble du domaine fréquentiel et spatial, au lieu d'être des modifications selon des régions locales dans le domaine spatial ou de Fourier.

Les deux autres méthodes concernent plutôt l'utilisation des filtres passe-haut en loi de puissance dont leur forme globale est, comme pour la première méthode, selon un spectre inversé par rapport aux images d'entrées. Comme certaines des recherches de la littérature, celle-ci intègre l'apprentissage de paramètres pour que leur forme puisse évoluer au cours de l'apprentissage.

Le prochain chapitre présente la première méthode, tandis que celui d'après porte sur la deuxième et troisième méthode développée au cours de cette recherche.

CHAPITRE 2

L'AUTO-FILTRAGE PAR LE COMPLÉMENT SPECTRAL (SFSC)

Ce chapitre présente la première méthode développée au cours de cette recherche que nous appelons la *Self-Filtering by the Spectral Complement* (SFSC) qui est une nouvelle méthode de prétraitement consistant à utiliser le complément spectral des images dans le domaine de Fourier (soit un filtre correspondant à leur magnitude inversée), afin d'y soustraire l'équivalent de l'autocorrélation de leur magnitude. Ceci permet de simplifier les représentations spatiales des images afin d'aider les modèles à atteindre de meilleurs taux de précision. Étant donné que la forme de base de cette méthode est relativement simpliste, une nouvelle couche de normalisation dans le domaine de Fourier ainsi qu'une couche effectuant des calculs de moyennage sur les dimensions des entrées sont également utilisées. Ces couches visent à ainsi améliorer les performances et la polyvalence de la SFSC pour son utilisation sur différentes bases de données et diverses architectures.

Une section sur les préliminaires est d'abord présentée pour certains aspects mathématiques reliés à la transformation dans le domaine de Fourier avant d'ensuite présenter les différents aspects relatifs à cette méthode et ses différentes couches.

2.1 Préliminaire : La transformée de Fourrier discrète

Définissons une donnée d'entrée f par un tenseur 4D de forme (N, C, H, W), indiquant respectivement le nombre d'échantillons (*batch size*), le nombre de canaux, la hauteur et la largeur d'un canal. La transformée de Fourier discrète (TFD) par canal de f est définie comme suit :

$$F_{n,c,u,v} = \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} f_{n,c,x,y} e^{-j2\pi(\frac{ux}{H}+\frac{vy}{W})} \tag{2.1}$$

où $j^2 = -1$ est l'unité imaginaire complexe. F est un tenseur complexe ayant des dimensions spatiales de taille (H, W) et peut être défini en fonction de chaque canal, de chaque image, en

coordonnées polaires comme étant :

$$F_{u,v} = R_{u,v} + jI_{u,vs} = M_{u,v}e^{j\phi_{u,v}} \tag{2.2}$$

où M est la magnitude et ϕ est l'angle de phase sont définis par :

$$\begin{aligned} M_{u,v} &= \left|F_{u,v}\right| = \sqrt{R_{u,v}^2 + I_{u,v}^2} \\ \phi_{u,v} &= arctan\left(\frac{I_{u,v}}{R_{u,v}}\right) \end{aligned} \tag{2.3}$$

Telle que définie par les propriétés de Fourier, la convolution dans le domaine spatial devient une multiplication dans le domaine de Fourier. Aussi, dans le cas de valeurs réelles, celles-ci ont la propriété de la symétrie paire/hermitienne (*even symmetry*) ou la symétrie par le conjugué (*conjugate symmetry*) :

$$F_{n,c,u,v} = F^*_{n,c,H-u,W-v} \tag{2.4}$$

où $*$ est l'opérateur du conjugué. Cette propriété permet de notamment réduire le nombre d'opérations nécessaires pour utiliser F en ayant uniquement besoin de la moitié de sa dernière dimension qui aura la forme $\lfloor \frac{W}{2} \rfloor + 1$, au lieu de W.

Afin que les multiplications effectuées dans le domaine de Fourier puissent correspondre à des convolutions linéaires en évitant d'avoir des erreurs de bouclage (*wrap-around error*), une complétion de zéros (*zero-padding*) de la même taille que les dimensions spatiales (H, W) est utilisée avant d'effectuer la TFD. L'entrée f aura donc la forme $(N, C, 2H, 2W)$ avant l'application de la TFD et cette complétion de zéros est enlevée après avoir fait la TFDI (TFD inverse) (Gonzalez & Woods, 2006). Cependant, en prenant en compte la propriété de la symétrie paire, F sera de taille $(N, C, 2H, W + 1)$ par rapport à celle originale de f, ce qui donne, au final, une consommation de mémoire d'approximativement deux images superposées (au lieu de quatre sans l'utilisation de la symétrie paire).

2.2 Définition de l'auto-filtrage par le complément spectral (SFSC)

Le concept de la SFSC consiste à ne modifier que les magnitudes des entrées dans le domaine de Fourier afin de les remplacer par les résultats obtenus par l'équation suivante :

$$\bar{M}_{n,c,u,v}(1 - \widehat{M}_{n,c,u,v}) = \bar{M}_{n,c,u,v} - \bar{M}_{n,c,u,v}\widehat{M}_{n,c,u,v} \tag{2.5}$$

où $\bar{M}$ et $\widehat{M}$ sont deux versions différentes de la magnitude de Fourier M des entrées qui ont été normalisées ($\bar{M}$ & $\widehat{M} = M \in [0, 1]$). Cette normalisation, qui sert a avoir des valeurs entre [0,1], a pour objectif initial de faire en sorte que $\bar{M} - \bar{M}\widehat{M} \geq 0$, afin de ne pas inverser la phase en raison d'une magnitude négative. D'autre part, la partie $(1 - \widehat{M})$ correspond au complément spectral, soit une version inversée et normalisée de M.

Sous sa forme la plus simple, la normalisation effectuée peut avoir pour définition d'être simplement $M_{n,c,u,v}$ divisée par son maximum global (*global max*). Ceci permet alors d'appliquer une normalisation max-min où le minimum est approximé comme étant à zéro, puisque c'est généralement le cas pour les magnitudes des hautes fréquences dans les images naturelles en raison de la définition de la TFD (équation 2.1). Dans une telle situation et où l'on appliquerait la même normalisation à $\bar{M}$ et $\widehat{M}$, l'équation 2.5 devient alors l'équation suivante :

$$\begin{aligned} \widehat{M}_{n,c,u,v}(1 - \widehat{M}_{n,c,u,v}) &= \widehat{M}_{n,c,u,v} - \widehat{M}_{n,c,u,v}\widehat{M}_{n,c,u,v} \\ &= \widehat{M}_{n,c,u,v} - \widehat{M}^2_{n,c,u,v} \geq 0 \because \widehat{M} \in [0, 1] \end{aligned} \tag{2.6}$$

La magnitude originale normalisée $\widehat{M}$ est donc soustraite par le résultat obtenu en multipliant celle-ci par elle-même dans le domaine de Fourier, ce qui correspond à la convolution de l'image par elle-même dans le domaine spatial. Le résultat obtenu par $\widehat{M}^2$ met alors en évidence les régions où les magnitudes sont les plus élevées et présente un aspect visuel similaire à une autocorrélation. Ces régions sont habituellement associées aux très basses fréquences comme celles autour de DC, soit la moyenne globale des intensités dans le domaine spatial, mais également de potentielles régions bruitées dans les images qui auraient des magnitudes plus

élevées que celles avoisinantes. Le fait d'enlever des informations en soustrayant $\widehat{M}^2$ de $\widehat{M}$ a donc pour effet de diminuer les valeurs à ces régions, réduisant ainsi globalement les écarts d'échelles de valeurs de magnitude entre les basses, moyennes et hautes fréquences, et donc également les gradients des variations des magnitudes, tout en aidant à réduire leurs intensités à des zones potentiellement bruitées dans les images.

Au bout du compte, l'équation 2.5 a pour effet de principalement réduire le niveau d'intensité des magnitudes dans les images en abaissant particulièrement celles qui sont les plus élevées, que ce soit sur le plan global ou local des fréquences dans le domaine de Fourier. Cependant, contrairement aux méthodes présentées dans la revue de littérature, lesquelles utilisent des filtres ou des méthodes prédéfinies afin d'appliquer le même procédé global, peu importe le contexte et les entrées utilisées, la SFSC se sert des données d'entrées pour s'auto-filtrer et aller spécifiquement aux endroits où chacune d'entre elles a des magnitudes plus élevées que les autres. Le filtrage effectué se fait à l'aide d'un filtre de la même taille que les images dans le domaine spatial, et effectue une modification globale sur l'ensemble des fréquences au lieu d'être effectué selon un découpage en plages de fréquences ou selon des modifications locales dans le domaine spatial. Par ailleurs, ces motivations sont également similaires à celles d'autres recherches ayant également utilisé les données d'entrées pour s'autodécrire, bien que leurs méthodes soient très différentes de la SFSC (Deselaers & Ferrari, 2010 ; Shechtman & Irani, 2007). De plus, comme mentionné plus haut dans l'équation 2.5, il y a au moins deux normalisations différentes.

2.3 Les normalisations qui prennent en compte les différentes dimensions

Comme mentionné précédemment pour l'équation 2.5, $\bar{M}$ et $\widehat{M}$ ont utilisé des normalisations différentes sur la magnitude M des images d'entrées pour mettre celles-ci entre [0,1]. En effet, une normalisation utilisant uniquement le maximum global (*global max*) sera une normalisation de l'ensemble du spectre selon la fréquence ayant la magnitude maximale parmi les $NCHW$ fréquences de M qui a une forme (N, C, H, W). Ceci considère peu les relations existantes entre les différentes dimensions de M et limite les performances maximales atteignables par la SFSC. Néanmoins, l'utilisation de certains principes basés sur les couches de normalisations spatiales

telles que la *batch normalization* (BN) (Ioffe & Szegedy, 2015a), la *instance normalization* (IN) (Ulyanov, Vedaldi & Lempitsky, 2016) et la *layer normalization* (LN) (Ba, Kiros & Hinton, 2016) peuvent permettre d'y remédier. Puisque la magnitude de chaque canal des images est spécifique aux dimensions spatiales (H, W), tel que spécifié dans l'équation 2.1, un regroupement de ces dimensions peut être effectué afin de faire des manipulations selon les dimensions (N, C, S) où $S = (H, W)$. Toutes les combinaisons possibles des dimensions (N, C, S) pour calculer les maximums pouvant être utilisés pour appliquer une normalisation max-min sur M pour obtenir $\bar{M}$ et $\widehat{M}$, sont représentées dans le tableau 2.1.

Tableau 2.1 L'ensemble des maximums qu'il est possible d'utiliser pour les normalisations max-min effectuées sur $M_{n,c,u,v}$ selon les trois dimensions (N, C, S) où $S = (H, W)$

Name	**Shape**	**Description**
#0 Global max	(1,1,1)	The maximum spectral magnitude
#1 IN max	(N,C,1)	1 max/channel/image
#2 LN max	(N,1,1)	1 max/image
#3 element-wise LN max	(N,1,S)	1 max/pixel/image
#4 BN max	(1,C,1)	1 max/channel
#5 RGB max image	(1,C,S)	1 max/pixel/channel
#6 Gray max image	(1,1,S)	1 max/pixel

De plus, il est également possible d'appliquer une moyenne sur certaines des dimensions de l'entrée afin de légèrement simplifier le calcul pour la multiplication dans l'équation 2.5, tout en aidant à légèrement réduire le niveau de bruit et des variations dans les images. Par exemple, un moyennage des trois canaux de couleurs des images va avoir pour effet de traiter de la même manière ceux-ci, à l'aide d'un seul canal en nuance de gris. Dans un tel cas, ceci peut être avantageux si, par exemple, les trois couleurs contiennent des informations similaires ou que certaines variations légères de bruits ne sont présentes que dans certains des canaux. Dans un tel cas, la version en nuance de gris aura globalement moins de bruits tout en ayant toujours le même contenu global que lorsque les canaux de couleurs sont séparés.

Afin de garder la forme des dimensions $S = (H, W)$ inchangée, le moyennage n'est possible que sur les différentes combinaisons des dimensions (N, C) qui sont présentées dans le tableau

2.2. Mathématiquement et expérimentalement confirmés, certains des calculs de maximum du tableau 2.1 deviennent équivalents après avoir effectué ces moyennes. De l'autre côté, d'autres causeront plutôt une division par eux-mêmes s'ils sont appliqués seuls, ce qui aura pour effet de mettre à zéro les magnitudes des images dans l'équation 2.5. Les maximums qui provoquent une division par eux-mêmes ne sont donc pas représentés dans le tableau 2.2. Le cas #-1 correspond à la situation où l'on n'applique pas de moyennage étant donné que parfois, il peut être préférable de ne pas en faire pour garder les informations séparées. Il convient également de noter qu'étant donné que les calculs de moyenne font changer les valeurs et font nécessairement en sorte que les valeurs maximales ne soient plus exactement à 1, l'application d'une autre normalisation pour les remettre à 1 a été utilisée dans le cadre de cette recherche. Cependant, celle-ci est uniquement utilisée lorsqu'un moyennage a été effectué puisque cela est inutile autrement.

Tableau 2.2 L'ensemble des calculs de moyennes possibles sur les trois dimensions (N, C, S) où $S = (H, W)$ conservera sa forme initiale

Dimensions used	**Output shape**	**Possible maximum of the table 2.1**
#-1 Nothing	(N,C,S)	all them - no averaging
#0 (N)	(1,C,S)	{#0 *or* #2, #1 *or* #4, #3 *or* #6}
#1 (C)	(N,1,S)	{#0 *or* #4, #1 *or* #2, #5 *or* #6}
#2 (N,C)	(1,1,S)	{#0 *or* #1 *or* #2 *or* #4}

Notre méthode finale relative au calcul de l'équation 2.5 est représentée dans la figure 2.1 où $\bar{M}$ et $\widehat{M}$ reçoivent tous les deux des normalisations différentes et tel que mentionné précédemment, une deuxième normalisation est effectuée pour $\widehat{M}$, si un moyennage est effectué sur celui-ci. Aussi, le moyennage n'est effectué que pour $\widehat{M}$ étant donné qu'on ne souhaite pas non plus complètement supprimer les informations relatives aux différentes dimensions de M, mais de plutôt ajuster différemment les informations qui seront enlevées par la partie relative à $(1 - \widehat{M})$ dans l'équation 2.5. Ceci correspond au complément spectral, soit une version inversée et complémentaire du spectre normalisé.

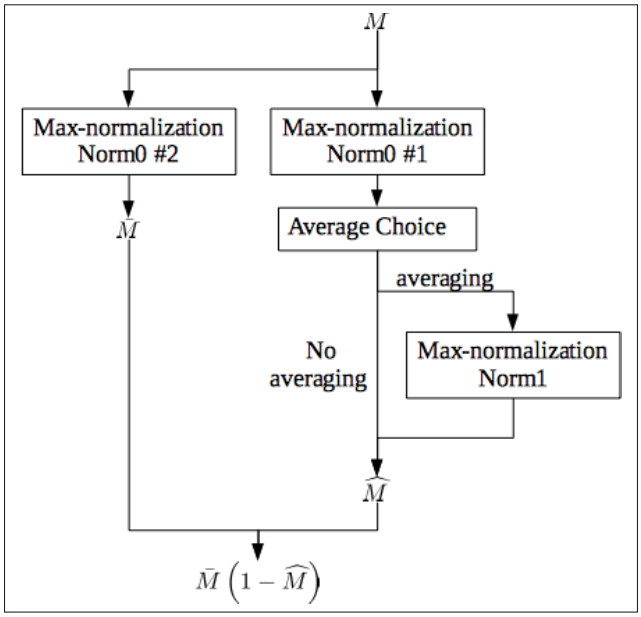


Figure 2.1 Les étapes du calcul de l'équation 2.5

2.4 L'apprentissage des normalisations et du choix du calcul de moyenne

Tel que présenté à l'aide des tableaux 2.1 et 2.2, le nombre de combinaisons de maximum et de moyennage qu'il est possible d'effectuer pour les trois calculs représentés dans la figure 2.1 est relativement très élevé et les meilleures combinaisons pour une situation peuvent devenir sous-optimales ou mauvaises dans une autre situation. Une solution possible est de plutôt effectuer des apprentissages pour utiliser une combinaison des différents maximums ainsi qu'un apprentissage sur le choix du moyennage effectué.

2.4.1 La couche de normalisation par les maximums (MN)

Basé sur le *switcheable normalization* (SN) (Luo, Ren, Peng, Zhang & Li, 2018a ; Luo, Zhang, Ren, Peng & Li, 2019a), qui utilise une *Z-score normalization* pondérée combinant les couches de normalisation BN, IN et LN pour normaliser les sorties des couches de neurones dans le domaine spatial, nous avons développé une couche de *Max Normalization* (MN) qui effectue une combinaison des différentes possibilités de maximum présentées dans le tableau 2.1. Le

calcul de chacun de ces maximums se fait en allant chercher les valeurs maximales selon les dimensions mises à 1 dans la colonne *Shape* du tableau 2.1. Ceci se traduit par les équations suivantes :

$$\begin{aligned} p_0 &= \max_{N,C,S}(h) & p_1 &= \max_{S}(h) & p_2 &= \max_{C,S}(h) \\ p_3 &= \max_{C}(h) & p_4 &= \max_{N,S}(h) & p_5 &= \max_{N}(h) \\ p_6 &= \max_{N,C}(h) \end{aligned} \tag{2.7}$$

desquelles on peut déduire des simplifications en raison des dimensions communes entre certaines de ces équations :

$$\begin{aligned} p_2 &= \max_{C,S}(h) = \max_{S}(p_3) = \max_{C}(p_1) & p_4 &= \max_{N,S}(h) = \max_{N}(p_1) = \max_{S}(p_5) \\ p_6 &= \max_{N,C}(h) = \max_{N}(p_3) = \max_{C}(p_5) & p_0 &= \max_{N}(p_2) = \max_{C}(p_4) = \max_{S}(p_6) \end{aligned} \tag{2.8}$$

où l'on définit $h_{n,c,u,v}$ et $\hat{h}_{n,c,u,v}$ comme étant respectivement un pixel ou une fréquence de l'entrée h du MN avant et après la normalisation, et p comme étant la liste des différents maximums. Le calcul du MN consiste alors à effectuer le calcul défini par les équations suivantes :

$$\hat{h}_{n,c,u,v} = \frac{h_{n,c,u,v}}{\sum_k w_k p_k} \tag{2.9}$$

$$w_k = \frac{\log(1 + \lambda_k)}{\sum_k \log(1 + \lambda_k)} \tag{2.10}$$

où λ_k et w_k sont respectivement les poids numéro k avant et après la normalisation en vecteur unitaire et $\lambda = \vec{1}$ à l'initialisation. À l'opposé du SN, le MN utilise la fonction d'activation *softplus* au lieu de la fonction *softmax* pour effectuer sa normalisation du vecteur de poids en vecteur unitaire, dans le but de faire en sorte que chacun des poids w_k représente alors une pondération en % de l'importance accordée à ce poids. De ce fait même, $\sum_k w_k = 1$ et $w \in [0, 1]$ et pour qu'un ou plusieurs poids augmentent, d'autres doivent nécessairement descendre (ou vice-versa). Ceci oblige donc le modèle à devoir effectuer des choix sur quels poids seront

maximisés et minimisés, tout en ayant une échelle pour les valeurs de w qui reste indépendante de celle utilisée par λ.

L'utilisation de la fonction d'activation *softplus* au lieu de *softmax* vient principalement d'expérimentations effectuées sur les fonctions d'activation utilisées par les quatre calculs de la figure 2.1 et c'est celle-ci qui semble préférable pour les trois MN. Ceci vient notamment du fait que la fonction *softplus* possède un gradient plutôt lent pour les plages de valeurs des λ_k, ce qui permet de notamment éviter de prendre des décisions trop hâtives au début de l'apprentissage. Le fait qu'elle ne prenne pas en compte l'ensemble des valeurs du vecteur de poids comme le font notamment *softmax* ou *softmin*, semble expérimentalement contribuer à favoriser l'utilisation de davantage de poids maximaux que ceux-ci. En effet, les fonctions *softmax* et *softmin* ont des équations qui visent à respectivement mettre en évidence la valeur maximale et la valeur minimale de l'entrée, causant, par conséquent, une certaine tendance à avoir moins de poids maximisés par les MN. Finalement, notamment par rapport à ReLU ou *hard tanh*, *softplus* possède une gestion des valeurs négatives pour les cas de $\lambda_k \approx 0$ qui peuvent notamment arriver si le *weight decay* des MN est trop élevé.

Aussi, tel qu'il est notamment utilisé dans le BN et le SN, des calculs de *exponential moving average* (EMA) sont effectués durant l'apprentissage pour les calculs des maximums utilisant la *batch dimension* (N). Durant les phases de test et de validation, ces moyennes des maximums sont alors utilisées pour éviter de les calculer à l'aide des entrées reçues. Tel que proposé par les auteurs du SN (Luo *et al.*, 2018a,1), l'utilisation de *simple moving average* (SMA), *cumulative moving average* (CMA) et EMA sur une époque d'apprentissage où seules ces statistiques sont calculées avec les modèles entraînés, a également été expérimentée à des fins de comparaison par rapport au EMA calculé au cours de l'apprentissage. Nos expérimentations ont montré que les calculs de CMA et SMA donnaient toujours les mêmes performances, mais que ces trois calculs de *moving average* sur une époque d'apprentissage ne donnent pas nécessairement de meilleures performances par rapport au EMA calculé durant l'apprentissage. On note également que les valeurs de précisions obtenues par ceux-ci restent relativement très similaires pour les cas de CIFAR, mais qu'ils étaient généralement inférieurs au EMA de l'apprentissage pour

les deux cas de DTD ainsi que sur KDEF. De manière similaire aux recherches sur le SN, pour la section des expérimentations qui sera présentée plus loin, les mesures de performances présentées correspondent aux meilleures performances mesurées parmi ces différents calculs de *moving average* qui ne concerne que les trois MN de la SFSC.

2.4.2 La couche sur le choix du moyennage effectué (AC)

Pour ce qui est du choix du moyennage, nous avons développé une couche *Average Choice* (AC) dont le calcul est inspiré du cas du SN avec un *hard ratio* (Luo *et al.*, 2019a) où seule l'opération relative au poids maximal est utilisée. Dans le cas de l'AC, cette contrainte est reliée au fait que son calcul de moyenne vise à réduire le nombre de dimensions de l'entrée, et non pour l'utiliser pour effectuer un autre calcul qui sera appliqué avec l'entrée, comme le font notamment les couches de normalisation spatiale standard avec leur *Z-score normalization*. De ce fait même, l'AC ne peut pas combiner ses différents poids, mais doit tout de même effectuer un apprentissage où l'ensemble de ses poids peut être utilisé.

Les différents calculs de moyennage de l'AC consistent à effectuer une moyenne selon les dimensions mises à 1 dans la colonne *Output shape* du tableau 2.2, ce qui se traduit par les équations suivantes :

$$\mu_0 = \frac{1}{N}\sum_{n=1}^{N} h_{n,c,u,v} \qquad \mu_1 = \frac{1}{C}\sum_{c=1}^{C} h_{n,c,u,v} \qquad \mu_2 = \frac{1}{NC}\sum_{n=1}^{N}\sum_{c=1}^{C} h_{n,c,u,v} \tag{2.11}$$

où μ est une liste des équations pour les différentes moyennes du tableau 2.2 et h correspond à l'entrée reçue. L'idée de départ des calculs de l'AC est de diviser son vecteur de poids par la valeur du poids maximal afin de mettre celui-ci à 1 (100 %), puis de le multiplier avec le résultat relié à son calcul de moyenne. Cette multiplication permet alors de les associer ensemble pour les calculs du gradient lors de la phase d'apprentissage. En intégrant l'utilisation d'une fonction d'activation sur le vecteur de poids, on obtient les équations suivantes :

$$w_k = 0,5\frac{\lambda_k}{1 + |\lambda_k|} + 0,5 \tag{2.12}$$

$$q = \frac{w_k}{\max w_k} \qquad i = \arg\max q \tag{2.13}$$

$$\hat{h}_{n,c,u,v} = q_i \mu_i \tag{2.14}$$

où λ et w représentent les vecteurs de poids avant et après la fonction d'activation *softsign* avec une mise à l'échelle à [0,1], q correspond au vecteur de poids après la division par la valeur du poids maximale avec i qui représente l'indice du poids maximal et $\hat{h}$ représente le résultat final de l'AC.

Cependant, en ce qui concerne le gradient, il n'y a dans les faits aucune perturbation qui pousse l'AC à remettre en question son choix après la première itération de l'apprentissage. L'utilisation de perturbations extérieures telles que la présence d'un *weight decay* ou d'un bruit appliqué sur son vecteur de poids devient alors nécessaire pour résoudre ce problème. Pour ce qui est du bruit, il est notamment possible d'effectuer une multiplication de q avec un bruit gaussien de moyenne égale à 1, et dont l'écart-type est égal à celui-ci du vecteur de poids λ. Par le fait que $\lambda = \vec{1}$ à l'initialisation, l'intensité du bruit gaussien s'adapte en fonction de l'écart entre les poids qui sera nul au départ, mais qui va varier au cours de l'apprentissage. L'utilisation d'une valeur absolue après l'application du bruit est également préférable pour éviter d'avoir des magnitudes qui peuvent être négatives si le bruit devient très intense. L'utilisation d'une autre division n'est cependant pas nécessaire puisque cela sera géré par Norm1, tel que représenté dans la figure 2.1. La présence du bruit g devient également inutile durant les phases de test et de validation étant donné qu'on souhaite alors obtenir le choix final appris par l'AC. En intégrant la présence du bruit, l'équation 2.13 devient légèrement différente durant la phase d'apprentissage :

$$g = \mathcal{N}(1,\, \sigma(\lambda)^2) \qquad \sigma(\lambda)^2 = \frac{1}{K-1}\sum_{k=1}^{K}(\lambda_k - \lambda')^2 \qquad \lambda' = \frac{1}{K}\sum_{k=1}^{K}\lambda_k \tag{2.15}$$

$$q_k = \left| g\frac{w_k}{\max w_k} \right| \qquad i = \arg\max q \tag{2.16}$$

où g correspond au vecteur de bruit gaussien et K indique le nombre de poids utilisé ou la longueur du vecteur λ.

Durant les phases de test et de validation, outre le fait que le bruit g soit désactivé, l'AC ne peut pas effectuer de calcul de *moving average* pour ces calculs de moyennes utilisant la dimension de lot (*batch dimension*) (N), étant donné que cela reviendrait à remplacer l'entrée par une valeur statistique. Par conséquent, durant les phases de test et de validation, les poids qui sont reliés à des calculs de moyenne sur la *batch dimension* (N) deviennent équivalents à ceux ayant cette dimension à 1 dans la colonne *Output shape* dans le tableau 2.2. Ceci se traduit par le fait que le poids #0 devient équivalent au poids #-1 et le poids #2 devient équivalent au poids #1. De manière peu surprenante, les expérimentations ont montré qu'il est préférable d'utiliser le même mode de fonctionnement durant les phases d'apprentissage, de validation et de test. Par conséquent, l'AC utilise dans les faits que les cas #-1 et #1 du tableau 2.2, mais nos expérimentations de diverses configurations pour les MN et AC ont montré que la présence de quatre poids, avec deux poids par option, était légèrement préférable par rapport à uniquement utiliser deux poids. Ceci vient notamment du fait que l'écart-type n'est pas invariant à la duplication des valeurs et, par conséquent, en présence de seulement deux ou quatre valeurs, l'utilisation de quatre poids permet d'avoir un écart-type plus faible par rapport à uniquement utiliser deux poids.

Pour ce qui est de la fonction d'activation *softsign*, bien que ce choix vienne surtout de diverses expérimentations de configuration, le fait que celle-ci soit originalement bornée entre les valeurs de -1 et 1 convient avec la présence d'un choix binaire entre deux calculs de moyennes. Cette plage est cependant mise entre [0,1], durant nos expérimentations, afin que comme pour les MN, les poids après la fonction d'activation puissent être exprimés en %.

Le chapitre suivant porte sur nos deux autres méthodes qui ont été développées au cours de cette recherche et qui servent également comme point de comparaison par rapport à la SFSC.

CHAPITRE 3

LES FILTRES FRÉQUENTIELS PASSE-HAUT EN LOI DE PUISSANCE

Le concept de notre seconde et troisième nouvelles méthodes consiste à l'utilisation des deux équations suivantes en remplacement de la magnitude originale M des images d'entrées :

$$\bar{M}_{n,c,u,v}(1 - \widehat{M}_{n,c,u,v})\omega^{\gamma}_{u,v} = (\bar{M}_{n,c,u,v} - \bar{M}_{n,c,u,v}\widehat{M}_{n,c,u,v})\omega^{\gamma}_{u,v} \tag{3.1}$$

$$M_{n,c,u,v}\omega^{\gamma}_{u,v} \tag{3.2}$$

où l'équation 3.1 consiste à l'équation 2.5 auquel on a ajouté le filtre fréquentiel en loi de puissance (*power-law*) ω^{γ} afin qu'il soit appliqué sur le résultat de la SFSC, alors que l'équation 3.2 consiste à uniquement appliquer ce filtre sur la magnitude originale des entrées des CNN. Le filtre fréquentiel ω^{γ} à la même taille spatiale que M et correspond à la γ-ième dérivée dans le domaine de Fourier si $\gamma \in \mathbb{N}^*$, mais on peut aussi utiliser $\gamma \in \mathbb{R}^+$ afin d'avoir davantage de liberté sur la forme du filtre. Le filtre ω^{γ} peut être exprimé dans le domaine de Fourier par les équations suivantes :

$$\begin{aligned} D_{u,v} &= \sqrt{u^2 + v^2} \\ \omega^{\gamma}_{u,v} &= \frac{\alpha D^{\gamma}_{u,v}}{\max \alpha D^{\gamma}_{u,v}} = \left(\frac{D_{u,v}}{\max D_{u,v}}\right)^{\gamma} \in [0, 1] \end{aligned} \tag{3.3}$$

où $D_{u,v}$ est la distance euclidienne par rapport au centre ou la fréquence 0 (DC), $\alpha \in \mathbb{R}_{>0}$ et γ peuvent tous les trois être en principe, des hyperparamètres ou des paramètres, tout comme pour la normalisation effectuée qui peut être avec ou sans apprentissage, avec l'utilisation d'un MN.

Dans le cas de α, la présence d'une normalisation pour mettre les valeurs du filtre entre [0,1] a pour effet de le rendre inutile, autant sur le plan mathématique que sur le plan expérimental. De ce fait, l'utilisation de $\alpha = 1$ est préférable et suffisante. Pour $D_{u,v}$, son utilisation en tant que paramètre après l'avoir initialisé selon la distance euclidienne par rapport à DC permet, dans certains cas, d'aider à stabiliser l'apprentissage du filtre et aider le modèle à y faire certains

ajustements. Pour γ où l'utilisation de valeur positive permet l'utilisation de filtres passe-haut dont la forme est globalement inversée à celle de M, son utilisation en tant que paramètre permet au modèle de librement l'ajuster selon ses besoins. Malgré tout, l'application d'un γ fixe et bien ajusté peut tout à fait donner de bonnes performances aux modèles. Finalement, la normalisation par valeur maximale peut se rapporter aux cas du MN selon les poids #2 ou #0 d'une AC qui serait utilisée sur son entrée (tableau 2.2), selon si l'on se sert de respectivement un ou C filtres pour les C canaux de l'entrée reçue. Son application sert à notamment s'assurer que l'échelle des valeurs du filtre soit toujours la même, tout en permettant d'avoir d'autres paramètres et contraintes pour ajuster la forme du filtre utilisé.

De manière similaire à ce qui a notamment été fait par Ma *et al.* (2020) et De Silva *et al.* (2020), le filtre fréquentiel ω^γ possède une forme prédéterminée ayant un apprentissage sur certains paramètres qui permettent surtout d'ajuster la forme du filtre tout au long de l'apprentissage. Cette façon de faire se distingue notamment des chercheurs ayant opté pour des filtres fixes, des filtres où seule l'initialisation est selon une forme précise, ou d'autres ayant modulé leur filtre appris avec des filtres fixes. Cependant, contrairement à plusieurs de ces recherches, le filtre fréquentiel ω^γ s'applique par une simple multiplication et ne peut, en aucun cas, changer le nombre de canaux de l'entrée, contrairement à une couche de convolution spatiale ou mise dans le domaine de Fourier. De plus, l'utilisation de valeurs aléatoires pour l'initialisation des différents paramètres est remplacée par l'utilisation de valeurs prédéterminées comme donnant de bonnes performances, selon des tests préliminaires avec des filtres fixes. Les modèles commencent donc leur apprentissage des filtres de bonne qualité et les performances mesurées sont potentiellement moins affectées par l'aléatoire. Finalement, ce filtre fréquentiel est directement créé dans le domaine de Fourier au lieu d'être, par exemple, un filtre spatial qui a été converti dans le domaine de Fourier et qui possède alors également une phase qui doit, en principe, être prise en compte et qui va modifier celles des images.

Afin d'aider à la compréhension du filtre fréquentiel ω^γ et de son effet sur les images d'entrées des CNN, la prochaine section illustre différentes figures sur ces deux principaux aspects.

3.1 Illustration sur l'effet de γ et du MN sur les filtres fréquentiels en loi de puissance

Les deux principaux aspects qui distinguent le filtre fréquentiel ω^γ des autres recherches ayant utilisé des filtres définis ou ayant une définition dans le domaine de Fourier, sont surtout sa forme en loi de puissance ainsi que l'utilisation d'un MN fixe ou appris afin de modifier sa forme, et non le résultat obtenu par le filtrage de son entrée. Afin d'aider à la compréhension de l'influence de ces aspects sur la forme du filtre utilisé, cette section présente différentes figures sur ces deux aspects.

En premier lieu, la figure 3.1 présente les formes globales des filtres fréquentiels ω^γ pour différentes valeurs de γ. Logiquement, puisque ce filtre correspond à la γ-ième dérivée dans le domaine de Fourier pour $\gamma \in \mathbb{N}^*$, plus la valeur de γ est élevée et plus les basses fréquences sont atténuées afin de ne conserver que les hautes fréquences. De son côté, l'utilisation de la normalisation entre [0,1] fait en sorte que γ ne change que la courbure de la ligne de ce filtre et permet d'éviter des augmentations massives des magnitudes des moyennes et hautes fréquences pour les valeurs de γ élevées. Les formes 2D de ces filtres aux différentes valeurs de γ correspondent alors à la rotation de ces lignes autour de l'axe central à l'origine telle la forme d'un cône, comme il sera présenté dans la figure 3.3.

Par la suite, la figure 3.2 présente les résultats obtenus par l'application de filtres fréquentiels fixes de forme ω^γ pour différentes valeurs de γ avec, d'un côté, les résultats obtenus dans le domaine spatial et de l'autre côté la magnitude de ces images mises sous une forme 1D. Ces formes de magnitude 1D sont obtenues en effectuant la moyenne des magnitudes selon leurs distances par rapport au centre (DC). Les images spatiales résultantes de l'application de ces filtres et qui étaient peu discernables tout en étant quasiment identiques à leur version en nuance de gris, ont été mises sous cette forme avec l'application d'une palette de couleurs, afin de plus facilement les voir. Finalement, les images ont été mises selon leur taille utilisée avec nos modèles soit en $32 \times 32 \times 3$ pixels pour l'image de CIFAR (Krizhevsky *et al.*, 2009) et $224 \times 224 \times 3$ pour celles de DTD (Cimpoi *et al.*, 2014) et de KDEF (Lundqvist *et al.*, 1998).

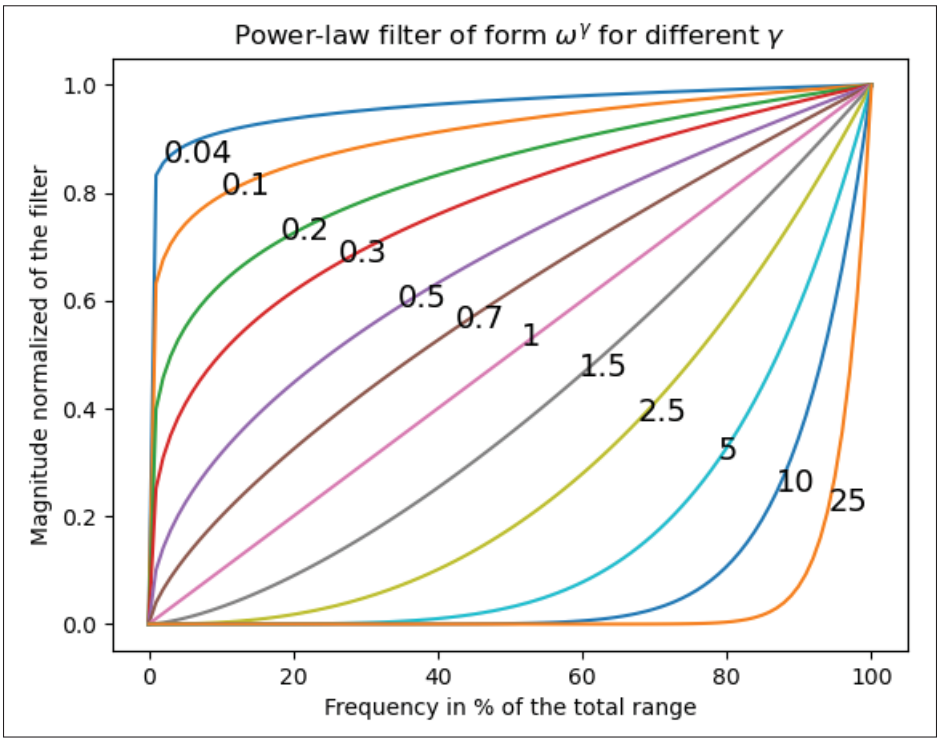
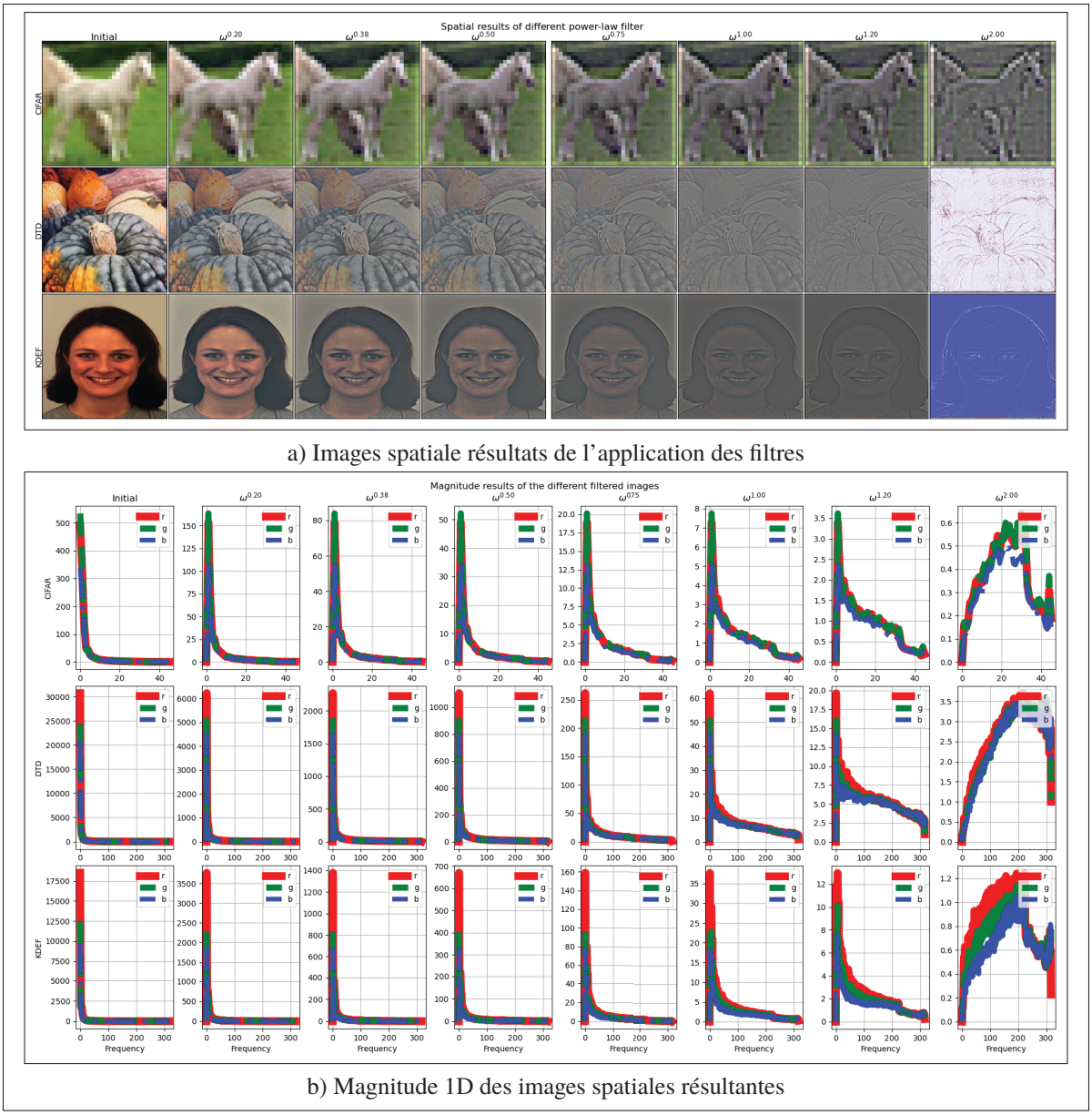


Figure 3.1 La forme globale des filtres fréquentiels de forme ω^γ

Tel qu'illustré dans ces figures, l'augmentation graduelle de la valeur de γ a pour effet de réduire de plus en plus les basses fréquences qui sont liées aux couleurs et aux variations d'intensité, ce qui a pour effet de mettre en évidence les hautes fréquences qui sont liées aux arêtes et aux détails dans les images. Durant ce processus, les images spatiales tendent alors vers une version en nuance de gris avec les trois canaux qui deviennent de plus en plus similaires. Ceci a alors pour effet de fortement diminuer le niveau de saturation des couleurs et à tendre vers un niveau de luminosité commun pour n'avoir qu'un seul niveau de gris global dans lesquelles les arêtes sont celles qui ressortent le plus de l'image. De l'autre côté, les formes globales des magnitudes résultantes restent similaires à celles des images originales jusqu'à un certain point où la tendance s'inverse, tel qu'illustré par les figures de $\gamma = 2$ qui correspondent aux dérivées secondes. On peut également constater qu'avec l'utilisation de la définition de la dérivée première ($\gamma = 1$) et seconde ($\gamma = 2$) dans le domaine de Fourier, ces dérivées des intensités des images ne sont, dans les faits, pas réellement bruitées, contrairement à ce que peuvent le laisser prétendre les images résultantes obtenues avec certains filtres spatiaux des dérivées. Pour ces

derniers, bien que cet effet de bruit soit notamment lié à la réduction des basses fréquences, une certaine part est également attribuable aux valeurs que les chercheurs utilisent dans le domaine spatial.



a) Images spatiale résultats de l'application des filtres

b) Magnitude 1D des images spatiales résultantes

Figure 3.2 Exemple de résultats obtenues par l'application de différents filtre de forme ω^{γ} pour différentes valeurs de γ

Par la suite, la figure 3.3 présente les effets relatifs aux différents MN appris et fixes sur les filtres fréquentiels fixes de forme $\omega^{0,5}$ et ω^{1}. Pour les MN appris, les poids sélectionnés correspondent aux deux formes de distribution observées ayant les écarts de poids les plus élevés parmi nos expérimentations. Juste après, la figure 3.4 présente, quant à elle, les images résultantes de l'application de ces filtres sur les mêmes images que celles utilisées pour la figure 3.2.

On peut notamment constater que l'apprentissage du MN permet d'ajuster la forme du filtre de manière à ce que celui-ci puisse correspondre à des valeurs de γ plus faibles que celle initiale. Cependant, les deux formes de distribution maximales observées donnent, malgré tout, des filtres et des images résultantes relativement similaires. Par conséquent, bien que le choix de la valeur d'initialisation de γ soit le facteur qui influence le plus la forme du filtre, l'utilisation d'un MN appris peut permettre des ajustements relativement précis sur la forme exacte utilisée. De l'autre côté, dans le cas du MN fixe selon le poids #3 du tableau 2.1, celui-ci n'élimine que DC et conserve donc le reste du spectre fréquentiel constant, en raison de sa définition et de l'utilisation de filtres fixes, pour cet exemple. Bien qu'il soit inutile avec des filtres fixes, celui-ci peut avoir un certain intérêt si les différents canaux ne sont pas identiques. De la même manière, les MN fixes selon les poids #0 et #1 du tableau 2.1 sont identiques dans cet exemple, mais ils donnent des résultats différents pour des filtres appris étant donné que leur niveau de contrainte sur la forme des filtres utilisés n'est pas le même.

À l'aide de l'ensemble de ces éléments relatifs aux différentes configurations possibles sur ces filtres fréquentiels, la prochaine section présente les configurations finales qui ont été sélectionnées.

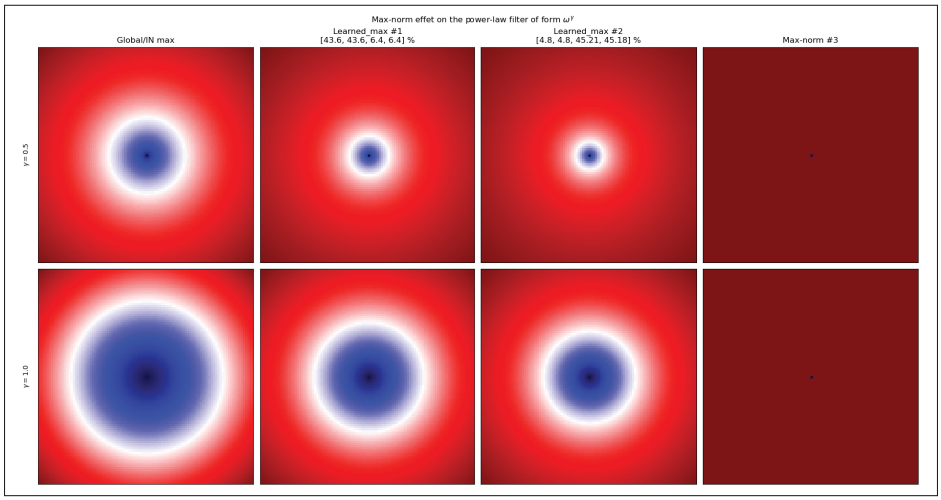
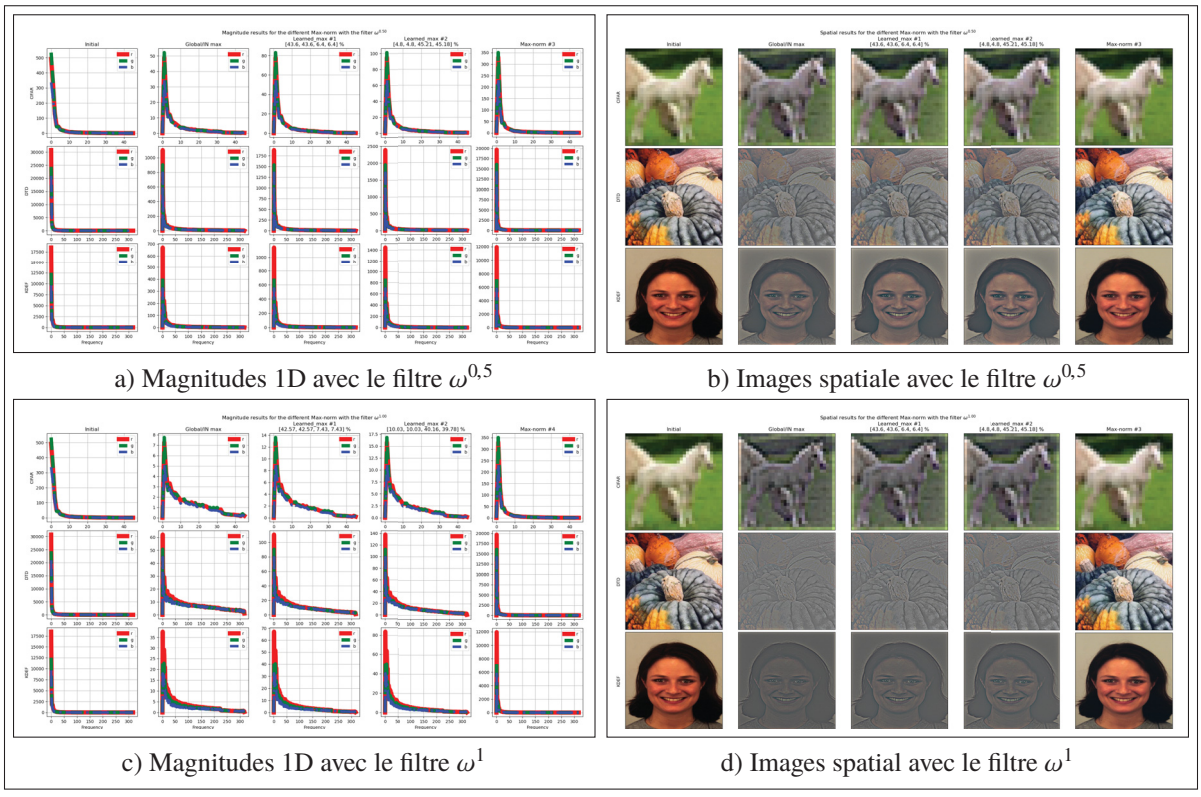


Figure 3.3 Comparaison des effets relatifs a l'utilisation des MN sur les filtres fréquentiels de forme ω^γ



a) Magnitudes 1D avec le filtre $\omega^{0,5}$

b) Images spatiale avec le filtre $\omega^{0,5}$

c) Magnitudes 1D avec le filtre ω^1

d) Images spatial avec le filtre ω^1

Figure 3.4 Exemple de résultats obtenues par les différents filtres de $\omega^{0,5}$ et ω^1 avec l'utilisation des différents MN de la figure 3.3

3.2 Présentation des configurations finales des filtres fréquentiels

Au cours de cette recherche, diverses expérimentations ont été effectuées sur les configurations des hyperparamètres ou paramètres D, γ et du MN des filtres fréquentiels ω^{γ} de l'équation 3.3. D'autre part, diverses formes sur l'utilisation de ces filtres avec la SFSC ont également été expérimentées avant d'arriver à son utilisation définie par l'équation 3.1.

Les premières expérimentations avec ces filtres ont été effectuées sur CIFAR-10 avec des filtres fixes afin de s'assurer que leurs formes initiales puissent améliorer les performances des modèles avant d'y ajouter diverses formes d'apprentissages. Ces expérimentations ont notamment montré que les meilleures valeurs de γ se situent dans la plage approximative de $0,5 \pm 0,25$ ou [0,25, 0,75]. Par conséquent, les expérimentations subséquentes avec des formes d'apprentissages ont été effectuées avec des valeurs d'initialisation du paramètre γ de 0,5 et de 1,0, où 0,5 correspond à la valeur centrale de la plage et 1,0 correspond à la première dérivée et la forme de base de ce filtre.

Pour ce qui est de l'utilisation de ce filtre avec la SFSC, diverses formes ont été expérimentées dont notamment l'utilisation de deux filtres fréquentiels avec l'un sur $\widehat{M}$ et l'autre sur le résultat final de la SFSC, ainsi que l'utilisation d'un filtrage uniquement effectué sur $\widehat{M}$, dans l'équation 2.5. Cependant, les diverses équations ayant un filtrage sur $\widehat{M}$ ont notamment pour défaut d'avoir un impact important sur les poids appris par les deux MN et l'AC qui sont liés à $\widehat{M}$. De plus, les paramètres appris par celui-ci sont également plus instables et sensibles aux gradients. Par conséquent, l'utilisation d'un filtre ω^{γ} avec la SFSC, telle que définie par l'équation 3.1, a semblé préférable.

De manière similaire à la problématique des paramètres de la méthode de De Silva *et al.* (2020), le paramètre γ de ω^{γ} est notamment fortement dépendant du choix des hyperparamètres de l'algorithme de gradient et possède une certaine sensibilité à la valeur d'initialisation utilisée. Pour tenter d'y remédier, certaines expérimentations avec l'utilisation d'une fonction d'activation sur le paramètre γ ont été effectuées dans le but de réduire la pente de son gradient. Au final, bien que les diverses fonctions d'activation expérimentées ont toutes obtenu des performances très

similaires, c'est la fonction d'activation *softsign* (comme pour l'AC) avec des valeurs mises entre [0,2] (au lieu de [0,1] pour l'AC) qui semblait préférable. Les expérimentations avec les divers modèles ont également montré qu'il est globalement préférable de l'utiliser sur CIFAR et KDEF, mais pas sur DTD. L'utilisation ou non de cette fonction d'activation a été considérée comme étant un hyperparamètre de base de données. Par conséquent, tous les modèles utilisés sur une base de données doivent employer la même configuration, mais le choix sur son utilisation peut changer entre les bases de données.

À la suite de l'ensemble des expérimentations effectuées avec différents modèles, les configurations finales sélectionnées pour l'utilisation des filtres ω^γ avec la SFSC (équation 3.1) ou les images d'entrées des CNN (équation 3.2) sont présentées dans le tableau 3.1 où quatre configurations finales ont été sélectionnées. Pour les cas avec des MN appris, seuls les quatre poids du cas #0 du tableau 2.2 de l'AC sont utilisés étant donné que cela inclut également le poids manquant qui sert de pondération sur l'entrée pour le calcul du dénominateur du MN. Les tailles (C, H, W) avec $C = 3$ font référence à la forme de la magnitude M des images d'entrées ou de la sortie de la SFSC.

Tableau 3.1 Configurations finales des filtres fréquentiels de forme ω^γ

No	**Used with the SFSC**	**learn γ**	**learn D**	**initial γ**	**Nb filters**	**MN used**	**Nb param added**
1	1	1	1	0,5	1	Global max (#0)	$HW + 1$
2	1	0	0	0,5	3	MN learned	4
3	0	0	1	1	3	IN max (#1)	$3HW$
4	0	0	1	1	3	MN learned	$3HW + 4$

Le prochain chapitre présente l'ensemble des résultats expérimentaux obtenus par nos modèles avec et sans l'utilisation de nos trois nouvelles méthodes.

CHAPITRE 4

RÉSULTATS EXPÉRIMENTAUX

Pour rappel, dans le cadre de cette recherche, l'objectif principal est de développer des nouvelles méthodes polyvalentes et aptes à améliorer les taux de précisions de différents CNN utilisés sur différents types de bases de données d'images. Pour ce faire, nos expérimentations ont été effectuées sur la base de données d'images naturelles CIFAR, la base de données de textures DTD ainsi que la base de données de reconnaissance d'émotions du visage KDEF. Dans l'ensemble, un total de 14 architectures de modèles de CNN ont été expérimentées afin de viser ces objectifs. Nos expérimentations sur CIFAR ont porté sur CIFAR-10 et CIFAR-100 avec et sans augmentation de données, tandis que celles sur DTD ont été faites avec des modèles pré-entraînés sur ImageNet et des modèles sans pré-entrainement (*from scratch*). Dans le cas de KDEF, les expérimentations se sont limitées au cas sans pré-entraînement (*from scratch*). Toutes les expérimentations sur DTD et KDEF étaient avec de l'augmentation de données.

Dans le cadre de cette recherche, nos méthodes ont été développées de manière à ce qu'elles puissent s'ajouter à des CNN existant sans avoir à modifier de quelconques parties de ces CNN, que cela touche à leurs architectures ou aux algorithmes de gradients utilisés. Conséquemment, lorsque nos méthodes sont intégrées à un CNN, la partie relative au CNN originale reste toujours inchangée, mais les aspects relatifs aux caractéristiques d'apprentissages propres à nos méthodes peuvent cependant être ajustés aux besoins. Ceci vient notamment du fait que nos méthodes sont placées au tout début des CNN et, en conséquence, ceux-ci peuvent notamment recevoir des gradients relativement faibles en raison de la profondeur du CNN, ce qui empêche nos méthodes d'avoir des poids bien ajustés. Ainsi, certaines optimisations ont été effectuées sur les hyperparamètres relatifs à nos méthodes, mais ceux relatifs à la partie des CNN restent toujours identiques aux CNN originaux sans nos méthodes.

Finalement, il est important de souligner que pour toutes nos expérimentations (avec et sans nos méthodes), les images sont transformées dans le domaine de Fourier selon la procédure décrite dans la section 2.1, avant d'être ensuite remises dans le domaine spatial et de recevoir

une *Z-score normalization* non pondérée à l'aide d'un BN sans poids. Cette procédure a pour objectif de s'assurer que les seules différences entre nos modèles avec et sans nos méthodes soient la modification ou non de la magnitude des entrées et que tout le reste soit parfaitement identique. L'utilisation d'un BN sans poids vise à s'assurer que les statistiques des entrées soient toujours selon le même format et ce, peu importe les modifications effectuées sur les entrées dans le domaine de Fourier. D'autres recherches ont également utilisé des BN sans poids sur leurs entrées modifiées, mais ceci est le plus souvent uniquement précisé dans leur code lorsque celui-ci est disponible (edouardoyallon, 2017 ; Oyallon *et al.*, 2017).

4.1 Détails sur les conditions d'expérimentations

Toutes nos expérimentations ont été faites sur PyTorch à l'aide de trois ordinateurs. Les expérimentations sur CIFAR ont été effectuées sur deux ordinateurs ayant, pour le premier, une carte graphique (GPU) NVIDIA GTX 1080 ayant 8 Go de mémoire vive (RAM), et pour l'autre, une carte graphique NVIDIA GTX 1660 ayant 6 Go de mémoire vive. Les résultats reliés à la base de données DTD et KDEF ont, quant à eux, été obtenus sur un autre ordinateur ayant une carte graphique TITAN RTX possédant 24 Go de mémoire vive. Toutes nos expérimentations ont été effectuées avec l'utilisation d'un seul GPU par modèle, mais plusieurs modèles ont parfois été exécutés sur le même ordinateur afin de réduire les temps d'exécution globaux des expérimentations et cela n'avait aucun impact sur les taux de précision mesurés.

4.1.1 Détails d'implémentations : CIFAR

La base de données CIFAR (*Canadian Institute For Advanced Research*) (Krizhevsky *et al.*, 2009) est une base de données d'images naturelles en couleur de 50k images d'apprentissage et 10k images de tests en 32×32 pixels. La base de données CIFAR-10 est divisé en 10 classes alors que CIFAR-100 est divisée en 100 classes avec, dans les deux cas, une distribution uniforme d'images entre les classes pour les images d'apprentissage et de test. Dans le cadre de cette recherche, tous les modèles ont été entraînés avec l'ensemble de la base de données d'apprentissage et de tests. Les cas de CIFAR-10+ et CIFAR-100+ correspondent aux cas avec l'utilisation de

l'augmentation de données standard de la littérature (He, Zhang, Ren & Sun, 2016a ; He *et al.*, 2016b ; Huang *et al.*, 2017 ; Zagoruyko & Komodakis, 2016). Cette augmentation de données consiste à d'abord appliquer une complétion de zéro (*zero padding*) de 4 pixels par côté sur les images pour obtenir une taille de 40×40 pixels, puis d'effectuer un découpage aléatoire en 32×32 pour ensuite y appliquer un *horizontal flip* avec une probabilité de 50 % d'être effectué. Une *Z-score normalization* est finalement appliquée afin d'avoir une moyenne et un écart-type globaux de respectivement 0 et 1.

Dans le cas des ResNet (He *et al.*, 2016b), Wide ResNet (WRN) (Zagoruyko & Komodakis, 2016) et DenseNet (Huang *et al.*, 2017), les architectures et détails d'implémentations correspondent à celles décrites dans leur article respectif par rapport à l'utilisation de CIFAR. Dans le cas des VGG (Simonyan & Zisserman, 2014), la méthode employée est basée sur Fu (2019). Tous utilisent l'algorithme de descente de gradient stochastique (SGD) avec un momentum (ResNet et VGG) ou un *nesterov momentum* (Wide ResNet et DenseNet) de 0,9, avec un *weight decay* de 1×10^{-4} (ResNet) ou 5×10^{-4}, un *batch size* de 128 ou 64 (DenseNet), sur 200 (ResNet et Wide ResNet) ou 300 époques (VGG et DenseNet). Pour le ResNet et le DenseNet, le taux d'apprentissage initial est de 0,1 et est divisé par 10 à 50 % et 75 % du nombre total d'époques. Pour le ResNet-110, un *warmup* avec un taux d'apprentissage de 0,01 est utilisé durant les 400 premières itérations pour ensuite retourner à son taux d'apprentissage normal de 0,1. Dans le cas des Wide ResNet, le taux d'apprentissage initial est de 0,1 et est divisé par 5 à 30 %, 60 % et 80 % du nombre total d'époques. Pour les VGG, le taux d'apprentissage initial est de 0,05 (0,1/2) et est ensuite divisé par 2 à tous les 10 % du nombre total d'époques. Étant donné que les entrées de la première couche entièrement connectée (FC) des VGG est de taille spatiale unitaire et possède 512 canaux, leurs deux premières FC utilisent 512 sorties au lieu de 4092. Pour l'initialisation des poids, tous utilisent celle décrite dans leur article respectif.

Pour ce qui est des CNN avec nos méthodes, tel que mentionné précédemment, quelques expérimentations ont été effectuées afin de déterminer certaines modifications légères concernant uniquement les hyperparamètres de SGD qui sont utilisés pour les paramètres de nos méthodes. Aussi, il est important de noter que, de manière générale, sans ces optimisations supplémentaires

sur CIFAR, les écarts de performance peuvent être légèrement négatifs par rapport aux modèles originaux. Cependant, de simples légères corrections sur les hyperparamètres d'apprentissages de SGD pour les paramètres de nos méthodes ont permis d'aisément y remédier pour la plupart des modèles qui ont été utilisés.

De manière générale pour la SFSC, celle-ci semble mieux fonctionner avec l'utilisation d'un *weight decay* de 1×10^{-4} (étant donné que 5×10^{-4} diminue trop les poids) et l'utilisation d'un momentum normal (pas un *nesterov momentum*). Pour le ResNet-110, le *warmup* n'est pas appliqué sur les couches de la SFSC, tandis que pour le VGG-16 avec BN, le taux d'apprentissage initial est de 0,1 (au lieu de 0,05). Finalement, pour le WRN, le taux d'apprentissage est divisé par 10 (au lieu d'être divisé par 5) à 30 %, 60 % et 80 % du nombre total d'époques.

Étant donné que les filtres fréquentiels de forme ω^γ sont à la même profondeur dans les modèles que la SFSC, leurs tests d'optimisations ont surtout porté sur l'utilisation ou non d'une division par deux du taux d'apprentissage initial de la SFSC. Ces modifications concernent uniquement les paramètres D et γ de l'équation 3.3 et exclut le MN appris pour les filtres #2 et #4 du tableau 3.1. Une autre règle qui a été utilisée lors des expérimentations est que les configurations des hyperparamètres de SGD doivent au moins rester les mêmes entre les deux cas avec et sans augmentation de données (CIFAR-10/100 et CIFAR-10+/100+) ou entre les deux cas de CIFAR-10 et de CIFAR-100 (CIFAR-10/10+ et CIFAR-100/100+). Le tableau 4.1 présente un résumé des taux d'apprentissage utilisés pour les filtres fréquentiels sur CIFAR. Dans le cas du VGG-11+BN, tous ont employé le taux d'apprentissage de 0,05 (comme pour le CNN) tandis que pour le filtre #2, son MN appris utilise les mêmes hyperparamètres que pour la SFSC et il est donc exclu de ce tableau. À noter également que pour le filtre #4 avec le ResNet-110, celui-ci n'utilise pas de *warmup* alors que les trois autres filtres en ont.

Tableau 4.1 Taux d'apprentissage initiaux pour les filtres fréquentiels sur CIFAR avec, en gras, les valeurs qui diffèrent de celles utilisées par les CNN originaux

No	VGG-16+BN		ResNet-56	ResNet-110		DenseNet {L = 100, K = 12}				WRN 16-8
	CIFAR-10/10+	CIFAR-100/100+		CIFAR-10/100	CIFAR-10+/100+	CIFAR-10	CIFAR-10+	CIFAR-100	CIFAR-100+	
1	0,1		**0,05**	**0,05**	0,1	0,1	**0,05**	0,1	**0,05**	**0,05**
3	**0,05**		0,1	**0,05**		**0,05**	0,1	**0,05**	0,1	0,1
4	0,1	**0,05**	**0,05**	**0,05**		**0,05**				**0,05**

Les premières expérimentations ont également été effectuées à l'aide d'un très petit CNN qui a servi de modèle de référence et dont son architecture est représentée dans la figure 4.1. Comme pour tous les autres CNN utilisés pour de la classification d'images, la couche entièrement connectée (FC) (sans biais) a un nombre de neurones égal au nombre de classes de la base de données utilisée et est suivie de l'utilisation de la fonction d'activation *softmax* et du calcul de l'erreur à minimiser selon l'entropie croisée (*cross-entropy*) pour le calcul du gradient. Toutes les couches de convolution (conv) ont 32 filtres ainsi qu'une complétion de zéro (*zero-padding*) de un pixel par côté pour les filtres 3×3 et de deux pixels pour les filtres 5×5. Les couches de convolution sont suivies d'une couche de BN et de ReLU selon l'ordre (conv, ReLU, BN) (au lieu de l'ordre (conv, BN, ReLU)) (ducha aiki, 2016). L'initialisation des poids des couches de convolutions est effectuée à l'aide de l'algorithme Kaiming (He, Zhang, Ren & Sun, 2015) alors que, pour la couche entièrement connectée, les poids sont initialisés selon une distribution normale de moyenne 0 et d'écart-type 1×10^{-2} (Simonyan & Zisserman, 2014). Pour les couches de BN, les poids sont initialisés à 1 et tous les biais du CNN sont initialisés à 0.

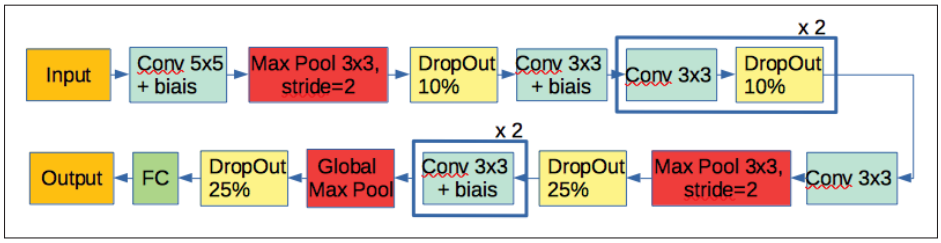


Figure 4.1 Architecture de notre CNN de référence

Dans le cas de notre CNN de référence, nous avons utilisé l'algorithme de descente de gradient stochastique adaptatif Adam avec AMSGrad avec les valeurs par défaut d'Adam pour ses hyperparamètres (Kingma & Ba, 2014 ; Reddi, Kale & Kumar, 2019), sauf pour le *weight decay* qui était de 1×10^{-4}. Aussi, les taux d'apprentissage des biais étaient multipliés par 2 par rapport à celles des autres couches (Cheng et al., 2017).

Il est également à noter que l'ensemble des résultats présentés sur CIFAR est basé sur cinq exécutions, avec différentes initialisations aléatoires des poids, tels qu'il est parfois présenté

dans la littérature pour démontrer la reproductibilité des expériences (He *et al.*, 2016a,1 ; Huang *et al.*, 2017 ; Zagoruyko & Komodakis, 2016). Cependant, ce sont plutôt 10 exécutions qui ont été effectuées dans le cas de notre CNN de référence étant donné sa rapidité d'exécution qui permet de plus facilement faire davantage d'expérimentations que pour les autres CNN.

4.1.2 Détails d'implémentations : DTD

La base de données DTD (*Describable Textures Dataset*) (Cimpoi *et al.*, 2014) est une base de données de texture considérée comme étant parmi les plus difficiles pour cette catégorie d'images (Cimpoi *et al.*, 2014) et qui possède 47 classes ayant chacune 120 images prises dans des conditions non contrôlées. Cette base de données possède 10 mélanges aléatoires prédéfinis (*split*) ayant 40 images d'apprentissage, de validation et de test, par classe. Étant donné que les bases de données de validation n'ont pas été utilisées au cours de nos expérimentations, celles-ci ont été mises avec celles d'apprentissage comme d'autres l'ont également fait dans la littérature (Fujieda *et al.*, 2017,1 ; Lin, 2020 ; Lin, RoyChowdhury & Maji, 2017 ; Oraibi, 2019). Les résultats sont habituellement rapportés par la moyenne et l'écart type de la précision obtenue sur les 10 *split*, mais nous avons également ajouté la médiane ainsi qu'une autre mesure sur la distribution des résultats obtenus.

Pour cette base de données, étant donné que certains modèles pré-entraînés sur ImageNet (Deng *et al.*, 2009) ont été employés, les images finales utilisées par nos différents CNN sont mises en 224×224 pixels comme sur ImageNet et comme l'ont également fait d'autres chercheurs ayant utilisé des CNN avec des couches entièrement connectées (Fujieda *et al.*, 2017,1 ; Gao, Beijbom, Zhang & Darrell, 2016 ; Lin, 2020 ; Lin & Maji, 2016 ; Peeples, Xu & Zare, 2020 ; Xue, Zhang & Dana, 2018). Tous les apprentissages ont été effectués avec de l'augmentation de données où les images sont d'abord mises en 256×256 pixels, avant d'appliquer un *random resized crop* avec *random size* de 0,8 à 1,0 et un *random aspect ratio* de 3/4 à 4/3, par rapport à la taille de l'image en 256×256. Un découpage aléatoire pour obtenir une portion de 224×224 est ensuite effectué et est suivi d'un *horizontal flip* avec 50 % de probabilité d'être effectué.

Finalement, une *Z-score normalization* est appliquée. Cette méthode d'augmentation de données correspond à celle notamment utilisée par Xue *et al.* (2018) et Peeples *et al.* (2020).

Pour ce qui est de l'algorithme de gradient, différentes expérimentations ont été effectuées et deux façons de faire principales ont été conservées. La première est l'utilisation d'Adam avec AMSGrad (telle qu'employée sur CIFAR avec notre CNN de référence) tandis que la seconde consiste à l'utilisation de SGD avec le même *scheduler* que les VGG sur CIFAR. Dans le cas de l'utilisation de SGD, le taux d'apprentissage initial est de 0,005 (0,01/2) (Peeples *et al.*, 2020 ; Sharma et al., 2018 ; Xue *et al.*, 2018 ; Zhang, Xue & Dana, 2017), avec un momentum de 0,9 et un *weight decay* de 1×10^{-4}. Le taux d'apprentissage est ensuite divisé par 2 à tous les 10 % du nombre total d'époques qui était de 300 époques pour les CNN non pré-entraînés (*from scratch*) et de 100 époques pour ceux pré-entraînés sur ImageNet (Gao *et al.*, 2016 ; Lin, 2020 ; Sharma *et al.*, 2018). Dans le cas des CNN pré-entraînés, le taux d'apprentissage initial est divisé par 10 pour les couches pré-entraînées. Finalement, toutes les expérimentations sur DTD ont été faites avec des tailles de lots (*batch size*) de 64 (Peeples *et al.*, 2020) pour limiter la consommation de mémoire.

Pour les CNN pré-entraînés sur ImageNet, tous les modèles, à l'exception du BNInception (Ioffe & Szegedy, 2015b), utilisent SGD, où ce dernier a obtenu de meilleures performances avec Adam avec AMSGrad. Comme modification concernant uniquement les paramètres de la SFSC, les ResNet utilisent un taux d'apprentissage initial de 0,5 qui est ensuite divisé par 10 pour le ResNet-18 et le ResNet34, tandis qu'il est divisé par 2 pour le ResNet-50. Dans le cas du VGG-11+BN, les meilleurs résultats ont été obtenus avec un taux d'apprentissage initial de 0,4 qui est ensuite multiplié par 0,75 à chaque réduction du taux d'apprentissage. Dans le cas du BNInception, les paramètres de la SFSC utilisent SGD (au lieu de Adam avec AMSGrad comme pour le reste du CNN) avec les mêmes configurations que pour le VGG-11+BN, soit un taux d'apprentissage initial de 0,4 qui est ensuite multiplié par 0,75 à chaque 10 % du nombre total d'époques. L'utilisation de taux d'apprentissage élevé pour ces modèles a surtout permis de compenser les gradients relativement faibles envoyés à la SFSC en raison des poids pré-entraînés

pour ces modèles qui ont besoin de taux d'apprentissage très faibles et le fait que les taux d'apprentissage sont réduits à chaque 10 époques.

Dans les cas des CNN non pré-entraînés (*from scratch*), les ResNet et le BNInception étaient meilleurs avec l'utilisation de Adam avec AMSGrad, alors que AlexNet (Krizhevsky, Sutskever & Hinton, 2012) avec BN, VGG-M (Chatfield, Simonyan, Vedaldi & Zisserman, 2014) avec BN et VGG-11 avec BN étaient mieux avec SGD. Comme modification qui s'applique uniquement aux paramètres de la SFSC, dans le cas de SGD, le taux d'apprentissage initial est 0,1 ($0,01 \times 10$). Aucune modification n'est faite pour la SFSC dans le cas de l'utilisation de Adam avec AMSGrad.

En ce qui concerne les filtres fréquentiels de forme ω^γ, ceux-ci ont obtenu de très mauvaises performances pour les cas de DTD avec des CNN pré-entraînés et ont donc été exclus des résultats qui seront présentés par la suite. Dans le cas des CNN non pré-entraînés (*from scratch*) utilisant SGD pour la partie de leur CNN, AlexNet avec BN et VGG-M avec BN utilisaient un taux d'apprentissage initial de 0,05 pour les filtres #1 et #3 et de 0,1 pour le filtre #4. Dans le cas du VGG-11 avec BN, le filtre #1 utilisait un taux d'apprentissage initial de 0,1 alors qu'il était de 0,05 pour les filtres #3 et #4. Comme avant, les MN appris des filtres #2 et #4 restent, de leur côté, selon les configurations utilisées pour la SFSC. Il est également à noter que, pour les filtres fréquentiels #1, #3 et #4, les taux d'apprentissage initiaux de 0,05 et de 0,1 pour SGD obtenaient des taux de précisions relativement similaires et avec des écarts de performances positifs par rapport aux modèles originaux. Comme pour la SFSC, aucune modification particulière n'a été faite pour les modèles utilisant Adam avec AMSGrad.

Dans le cas du CNN de références, nous avons employé la même architecture que sur CIFAR ainsi que l'utilisation de Adam avec AMSGrad pour l'ensemble du modèle.

Dans le cas du AlexNet et VGG-M sans pré-entraînement (*from scrach*), des BN ont été ajoutés à la suite de leur convolution selon l'ordre (conv, BN, ReLU) tel qu'il est utilisé dans les VGG avec BN (avec ou sans pré-entraînement) ainsi que sur le BNInception et le ResNet original (ayant le format *post-activation*). Dans le cas du VGG-M avec des BN, les LRN ont été enlevés

au profit des BN (Ioffe & Szegedy, 2015b). Aussi, tel que c'est le cas dans les VGG avec BN, aucune couche de BN n'a été ajoutée entre les couches entièrement connectées (FC).

Il est également à noter qu'étant donné qu'il a été expérimentalement démontré sur les deux cas de DTD qu'il était nécessaire d'avoir des BN pour la SFSC, AlexNet et VGG-M pré-entraînés (sans BN) ont été exclus des résultats présentés. Ce besoin d'avoir des BN a également été vérifié sur les quatre cas de CIFAR avec des VGG-11 où celui avec un pré-entraînement sur ImageNet a obtenu de bonnes performances contrairement à celui sans pré-entraînement. L'utilisation de BN dans le CNN est donc préférable pour son utilisation avec la SFSC, mais étant donné que l'utilisation de BN dans les CNN de classification d'images est devenue un standard après l'arrivée du ResNet (He *et al.*, 2016a,1 ; Hu *et al.*, 2018 ; Huang *et al.*, 2017 ; Xie *et al.*, 2017 ; Zagoruyko & Komodakis, 2016), ceci cause peu de tort à la SFSC. Étant donné que la sensibilité aux gradients des filtres fréquentiels est plus élevée que pour la SFSC, on peut supposer que ce constat s'applique également à ceux-ci, bien qu'aucune expérimentation de ce genre n'a été effectuée avec eux au cours de cette recherche.

4.1.3 Détails d'implémentations : KDEF

La base de données KDEF (*Karolinska Directed Emotional Faces*) (Lundqvist *et al.*, 1998) est une base de données de reconnaissances des expressions faciales des visages ayant un total de 4900 images de 70 personnes (35 hommes et 35 femmes) prises en photos selon sept expressions faciales (neutre, heureux, en colère, effrayé, dégoûté, triste, surpris), cinq prises de vue (-90, -45, 0, 45, 90 degrés) et deux séances de photographies. La base de données nécessite cependant quelques corrections qui ont dû être effectuées manuellement (Williams, 2017), telle que des images manquantes ou corrompues qui ont nécessité de faire des miroitages (*horizontal flip*) afin d'ensuite les ajouter. De plus, certaines images n'avaient pas la taille définie par les auteurs de cette base de données (762×542) et ont donc été mises à cette taille. Étant donné que KDEF n'a pas de bases de données d'apprentissage et de test prédéfinis, nous avons opté pour une procédure similaire à celle de DTD. Ainsi, 10 mélanges aléatoires prédéfinis (*split*) ayant chacun 80 % des images pour l'apprentissage et 20 % pour les tests ont été utilisés, avec un

échantillonnage qui s'assure que l'ensemble des images relatives à chaque individu de chacune des séances ne soit qu'en apprentissage ou en test, mais pas les deux à la fois. Comme sur DTD, les images finales envoyées aux CNN sont en 224×224 pixels. Afin d'aider à l'apprentissage des modèles, de l'augmentation de données consistantes à effectuer un changement aléatoire du niveau de luminosité des images de ± 25 % combiné à un *horizontal flip* ayant 50 % de probabilité d'être effectué a été utilisée.

Les modèles utilisés sur KDEF sont les mêmes que ceux non pré-entraînés (*from scratch*) qui ont été employé sur DTD et nous avons expérimenté les deux mêmes méthodes de gradients. Les ResNet ainsi que le BNInception ont mieux fonctionné avec l'utilisation de Adam avec AMSGrad tandis que AlexNet avec BN, VGG-M avec BN et VGG-11 avec BN ont mieux fonctionné avec SGD. Cependant, nos expérimentations avec la SFSC ont montré que celle-ci donnait des pertes de performances d'environ 2 à 5 % lorsque la partie du CNN utilisait SGD, ce qui pourrait être lié à une convergence trop hâtive relative à l'ajout de la SFSC. La source de ce problème a été confirmée avec le BNInception original qui a obtenu des performances quasi identiques pour l'utilisation de Adam et de SGD, mais seul celui avec Adam a pu être amélioré par la SFSC. Par conséquent, AlexNet avec BN, VGG-M avec BN et VGG-11 avec BN qui avaient eu de très mauvaises performances avec Adam, ont été exclus des expérimentations avec la SFSC et les filtres fréquentiels.

Pour la SFSC, celle-ci fonctionne mieux avec l'utilisation de SGD et les tests d'optimisation effectués se sont limités aux meilleures configurations identifiées sur DTD avec les modèles pré-entraînés sur ImageNet, soit des taux d'apprentissage initiaux de 0,4 ou 0,5 qui sont ensuite multipliés par 0,1, 0,5 ou 0,75 à tous les 10 % du nombre total d'époques qui était de 300 époques. Ainsi, le ResNet-18 et le ResNet-34 utilisent un taux d'apprentissage de 0,5 qui est ensuite multiplié par 0,1 pour le ResNet-18 et par 0,5 pour le ResNet-34. Pour le BNInception et le ResNet-50, le taux d'apprentissage est de 0,4 et est multiplié par 0,75 à tous les 10 % du nombre total d'époques.

Dans le cas des filtres fréquentiels où leurs paramètres sont utilisés avec SGD comme pour la SFSC, comme pour avant, leurs expérimentations se sont limitées à l'utilisation ou non de taux d'apprentissage initiaux divisés par deux par rapport à ceux de la SFSC. Les cas qui en ont utilisé sont le ResNet-18 avec le filtre #1, le ResNet-50 et le BNInception avec le filtre #3 ainsi que tous les modèles à l'exception du ResNet-50 pour le filtre #4. Aussi, tel que précisé précédemment, le filtre #1 utilise sa fonction d'activation sur son paramètre γ pour cette base de données comme c'est également le cas sur CIFAR (section 3.2).

4.2 Résultats expérimentaux de la SFSC

Cette section présente les résultats obtenus par nos différents modèles avec et sans la SFSC, les moyennes des poids appris par les différents MN et les choix finaux des AC, ainsi que certaines figures qui illustrent les résultats intermédiaires et finaux obtenus par la SFSC.

4.2.1 Présentation des poids appris par la SFSC

Afin de mieux comprendre les interactions entre les paramètres de la SFSC et son interaction avec les modèles et les bases de données utilisées, il est intéressant d'analyser les poids appris par la SFSC lors des différentes expérimentations effectuées. Les tableaux 4.2, 4.3 et 4.4 présentent les moyennes des poids appris par les MN et les choix finaux des AC des SFSC utilisés par nos modèles sur respectivement CIFAR, DTD et KDEF. Pour ces tableaux, des échelles de couleurs ont été mises pour chacun des MN, de chacun des CNN, afin de mettre en évidence quels sont les poids les plus élevés et les plus faibles. En ce qui concerne les colonnes des écarts entre les valeurs maximales et minimales de chaque MN, des échelles de couleurs globales à l'ensemble des cas représentés dans les tableaux ont été ajoutées pour mettre en évidence les modèles qui ont eu les poids les plus écartés et donc les mieux appris. À noter également que le poids #3 a été supprimé des deux Norm0 en raison du fait que son utilisation seule donnait de loin les pires performances sur notre CNN de référence sur CIFAR-10. Il a donc été décidé de l'enlever pour les expérimentations suivantes. Dans les faits, son utilisation avec la combinaison des poids ne devrait causer aucun problème et il sera probablement parmi les poids les plus faibles, ce qui va

donc contribuer à diminuer les autres poids plus importants. Son intégration ne devrait toutefois pas causer de variations importantes sur les performances mesurées.

Tableau 4.2 Tableau de la moyenne des poids appris sur les 5 exécutions aux poids aléatoires différents avec les modèles sur CIFAR (10 exécutions pour notre CNN de référence). Les poids des MN et les écarts entre leur poids maximal et minimal sont en %

Architecture	Dataset	AC	Norm0 #1 (pour le calcul de $\widehat{M}$)							Norm0 #2 (pour le calcul de $\widehat{M}$)							Norm1				Écart Max-Min		
			#0	#1	#2	#3	#4	#5	#6	#0	#1	#2	#3	#4	#5	#6	(0 or 4)	(1 or 2)	(5 or 6)	(3)	Norm0 #1	Norm0 #2	Norm1
ResNet-56	CIFAR-10	1	19,44	15,27	12,62	_	18,80	17,07	16,80	16,43	5,30	6,75	_	7,81	26,53	37,18	34,05	18,22	27,05	20,68	6,82	31,87	15,82
	CIFAR-10+	1	32,33	11,51	10,34	_	29,50	8,28	8,03	28,33	4,12	5,97	_	8,64	20,76	32,18	63,94	13,95	11,49	10,61	24,30	28,07	53,33
	CIFAR-100	1	27,94	14,07	11,41	_	25,31	10,64	10,63	18,73	2,84	4,23	_	5,21	18,39	50,59	56,13	13,96	17,78	12,14	17,31	47,74	43,99
	CIFAR-100+	1	43,34	7,40	6,92	_	34,31	4,13	3,90	37,69	2,05	3,85	_	5,53	13,65	37,22	81,60	7,66	5,74	5,00	39,43	35,64	76,59
ResNet-110	CIFAR-10	1	21,07	15,40	13,05	_	20,08	15,35	15,04	18,69	5,87	7,27	_	9,61	25,03	33,53	37,91	19,26	23,60	19,23	8,03	27,66	18,68
	CIFAR-10+	1	31,36	12,00	10,88	_	29,07	8,47	8,21	27,48	4,42	6,41	_	8,82	21,14	31,73	62,48	14,71	11,79	11,02	23,15	27,31	51,46
	CIFAR-100	1	28,65	13,65	11,85	_	25,65	10,08	10,12	20,92	3,17	4,70	_	5,64	19,16	46,42	56,42	14,41	17,12	12,05	18,57	43,26	44,38
	CIFAR-100+	1	42,41	7,96	7,46	_	33,20	4,60	4,37	34,42	2,37	4,15	_	5,92	14,82	38,32	78,93	8,71	6,53	5,83	38,04	35,96	73,10
VGG-11+BN	CIFAR-10	1	15,02	17,30	12,90	_	14,56	20,14	20,08	4,52	2,81	2,59	_	4,12	37,84	48,12	22,42	16,91	30,75	29,92	7,24	45,52	13,83
	CIFAR-10+	1	27,08	14,80	12,00	_	25,22	10,64	10,27	3,70	1,08	1,24	_	2,84	33,53	57,61	54,01	15,94	15,98	14,08	16,81	56,53	39,93
	CIFAR-100	1	14,89	12,77	9,72	_	14,36	23,73	24,53	4,26	1,73	1,77	_	3,31	27,63	61,30	25,09	10,12	38,07	26,73	14,81	59,57	27,95
	CIFAR-100+	1	33,15	12,65	10,43	_	29,05	7,64	7,08	4,40	0,58	0,79	_	2,08	20,18	71,97	67,49	11,71	11,67	9,12	26,07	71,39	58,37
VGG-16+BN	CIFAR-10	1	16,84	15,04	12,60	_	16,40	19,61	19,52	7,45	4,39	4,70	_	6,27	31,84	45,35	25,58	18,02	31,55	24,85	7,02	40,96	13,53
	CIFAR-10+	1	28,54	12,87	11,40	_	26,42	10,55	10,22	10,72	2,70	3,43	_	5,46	26,25	51,43	56,10	15,40	15,54	12,96	18,32	48,74	43,14
	CIFAR-100	1	18,11	12,73	11,24	_	17,46	19,89	20,56	8,97	2,91	3,35	_	5,48	23,91	55,38	33,84	13,92	33,73	18,52	9,32	52,47	19,92
	CIFAR-100+	1	36,30	10,32	9,41	_	30,96	6,70	6,31	16,78	1,48	2,34	_	3,67	16,59	59,14	71,58	10,81	9,73	7,89	29,99	57,66	63,69
DenseNet {L = 100, K = 12}	CIFAR-10	1	23,63	15,56	13,25	_	21,89	13,14	12,54	21,91	5,87	9,01	_	7,25	21,83	34,14	42,15	19,43	19,41	19,01	11,09	28,27	23,14
	CIFAR-10+	1	31,42	12,52	11,35	_	28,24	8,40	8,06	34,71	6,05	8,52	_	10,21	17,13	23,37	61,12	15,85	11,59	11,44	23,36	28,67	49,69
	CIFAR-100	1	29,79	15,84	12,54	_	26,15	7,98	7,70	27,05	3,15	6,16	_	4,49	17,50	41,64	59,56	17,22	11,47	11,74	22,09	38,49	48,09
	CIFAR-100+	1	40,49	9,14	8,46	_	32,32	4,96	4,63	41,84	3,50	6,02	_	5,07	13,22	30,34	76,91	9,95	6,73	6,40	35,86	38,34	70,50
WRN 16-8	CIFAR-10	1	10,51	17,81	8,16	_	10,43	26,96	26,13	16,72	4,44	6,60	_	7,32	25,93	39,00	34,80	14,73	21,16	29,31	18,80	34,56	20,07
	CIFAR-10+	1	28,42	15,94	11,56	_	26,91	8,81	8,36	30,22	3,28	5,05	_	7,89	20,88	32,67	55,74	17,09	11,38	15,79	20,06	29,39	44,36
	CIFAR-100	1	12,94	48,25	7,15	_	12,03	9,87	9,76	34,52	3,38	6,70	_	7,75	15,98	31,67	47,79	12,54	14,88	24,80	41,10	31,14	35,25
	CIFAR-100+	1	33,76	16,78	10,52	_	29,52	4,94	4,48	43,39	2,14	4,58	_	6,21	14,00	29,68	68,90	15,72	6,47	8,91	29,28	41,26	62,42
CNN de référence	CIFAR-10	1	36,09	8,57	7,21	_	29,99	9,32	8,82	15,90	1,55	2,10	_	3,45	23,88	53,12	68,44	9,66	13,50	8,40	28,88	51,57	60,04
	CIFAR-10+	1	39,41	8,19	7,00	_	33,17	6,23	6,01	22,29	1,54	2,34	_	4,49	22,21	47,13	75,01	9,29	8,69	7,01	33,40	45,59	68,00
	CIFAR-100	1	41,36	6,70	6,78	_	34,17	5,51	5,47	19,78	1,03	1,35	_	3,21	19,94	54,70	78,91	7,17	8,06	5,85	35,90	53,66	73,06
	CIFAR-100+	1	44,04	7,12	7,43	_	33,93	3,86	3,62	24,92	1,02	1,42	_	3,19	16,78	52,66	83,64	6,60	5,40	4,35	40,42	51,64	79,29

Tableau 4.3 Tableau de la moyenne des poids appris sur les 10 *splits* pour les 2 cas de DTD. Les poids des MN et les écarts entre leur poids maximal et minimal sont en %

Dataset	Architecture	AC	Norm0 #1 (pour le calcul de $\widehat{M}$)							Norm0 #2 (pour le calcul de $\widehat{M}$)							Norm1				Écart Max-Min		
			#0	#1	#2	#3	#4	#5	#6	#0	#1	#2	#3	#4	#5	#6	(0 or 4)	(1 or 2)	(5 or 6)	(3)	Norm0 #1	Norm0 #2	Norm1
DTD from scratch	CNN de référence	-1	28,94	21,08	10,29	_	28,21	6,04	5,45	7,43	1,11	1,24	_	4,53	40,80	44,88	44,72	20,24	11,99	23,06	23,49	43,76	32,73
	ResNet-18	-1	17,50	27,23	12,84	_	16,44	12,88	13,12	8,73	2,52	2,92	_	5,08	39,68	41,08	42,10	16,52	14,12	27,27	14,39	38,56	27,98
	ResNet-34	-1	17,50	24,03	14,60	_	16,95	13,46	13,46	9,91	2,95	3,32	_	5,81	38,67	39,34	40,60	17,50	15,63	26,27	10,57	36,39	24,97
	ResNet-50	-1	20,27	22,89	13,06	_	18,72	12,37	12,69	11,73	4,95	5,66	_	7,21	34,51	35,94	41,23	19,21	15,43	24,12	10,52	30,99	25,80
	BN-Inception	-1	18,97	23,60	15,70	_	17,97	11,90	11,86	11,63	3,27	3,02	_	6,80	37,18	38,10	42,99	18,59	14,48	23,94	11,74	35,08	28,51
	VGG-11+BN	-1	11,19	35,57	14,71	_	12,10	13,52	12,90	5,11	0,59	0,59	_	2,35	43,45	47,90	42,41	19,58	13,80	24,21	24,38	47,32	28,61
	VGG-M+BN	-1	12,08	39,32	13,85	_	13,49	11,02	10,25	7,64	0,61	0,63	_	2,77	41,18	47,16	47,70	21,92	12,14	18,24	29,07	46,54	35,55
	AlexNet+BN	-1	13,55	38,38	14,09	_	15,17	9,80	9,00	7,88	0,50	0,49	_	2,29	41,45	47,39	46,69	21,74	11,12	20,45	29,37	46,90	35,57
DTD pré-entraîné sur ImageNet	ResNet-18	1	28,43	12,17	12,81	_	26,51	10,09	9,99	42,97	7,53	12,37	_	12,14	12,64	12,36	58,68	16,42	14,75	10,16	18,44	35,44	48,52
	ResNet_34	1	28,26	12,44	12,98	_	26,26	10,07	9,98	43,03	5,94	11,82	_	11,33	13,98	13,91	59,32	16,07	15,03	9,58	18,28	37,08	49,74
	ResNet-50	1	28,87	12,36	12,47	_	26,71	9,86	9,73	44,72	8,50	13,71	_	8,58	12,21	12,28	61,06	15,12	14,58	9,23	19,13	36,21	51,83
	BN-Inception	1	20,67	16,85	14,53	_	19,87	13,99	14,09	31,61	9,38	8,88	_	8,52	20,39	21,23	40,72	21,70	18,32	19,26	6,68	23,09	22,41
	VGG-11+BN	1	26,83	12,87	12,41	_	25,35	11,34	11,20	40,96	7,81	12,26	_	5,35	17,03	16,60	57,31	15,72	15,96	11,01	15,64	35,61	46,31

Tableau 4.4 Tableau de la moyenne des poids appris sur les 10 *splits* pour KDEF. Les poids des MN et les écarts entre leur poids maximal et minimal sont en %

Architecture	AC	Norm0 #1 (pour le calcul de $\widehat{M}$)							Norm0 #2 (pour le calcul de $\widehat{M}$)							Norm1				Écart Max-Min		
		#0	#1	#2	#3	#4	#5	#6	#0	#1	#2	#3	#4	#5	#6	(0 or 4)	(1 or 2)	(5 or 6)	(3)	Norm0 #1	Norm0 #2	Norm1
CNN de référence	1	20,56	16,07	14,72	_	18,04	15,71	14,91	19,80	7,58	11,72	_	5,13	24,77	31,00	33,57	21,45	31,02	13,96	5,84	25,88	19,60
ResNet-18	1	13,84	17,35	17,02	_	15,26	18,41	18,12	1,90	2,03	2,58	_	1,56	35,30	56,63	25,65	25,12	28,96	20,27	4,57	55,07	8,68
ResNet-34	1	16,90	14,82	14,28	_	18,87	17,86	17,26	1,22	1,38	1,84	_	1,06	34,55	59,95	35,69	20,27	28,37	15,67	4,59	58,88	20,03
ResNet-50	-1	10,73	17,10	16,67	_	11,66	21,97	21,87	1,85	1,99	2,65	_	1,52	34,28	57,72	22,17	23,51	34,14	20,18	11,24	56,19	13,97
BNInception	-1	13,74	17,87	17,65	_	14,87	18,09	17,77	1,33	1,64	1,98	_	0,99	35,35	58,71	26,76	25,66	26,39	21,19	4,35	57,72	5,57

Dans l'ensemble, on peut notamment observer que les différents modèles entraînés dans les mêmes conditions globales (CIFAR, DTD *from scratch*, DTD pré-entraîné et KDEF) ont

généralement tous visé les mêmes poids à maximiser et à minimiser. La SFSC est donc relativement reproductive et fiable par rapport à la distribution globale des poids apprise pour une tâche spécifique, et ce, malgré l'utilisation de méthodes de descente de gradient (*optimizer & scheduler*) et d'architectures différentes entre les CNN. Malgré tout, certaines exceptions sont également présentes sur notamment CIFAR-10/100 (sans augmentation de données) avec les VGG (sur Norm0 #1 et Norm1) et le WRN 16-8 (sur les deux Norm0) ainsi que sur certains cas de KDEF (sur Norm0 #1 et Norm1). Pour la plupart d'entre eux, ces exceptions sont surtout reliées au fait que les écarts des poids de leurs MN concernés sont relativement petits en raison des faibles gradients reçus. Ceci va notamment en concordance avec le fait que les écarts des poids et les performances des modèles sont généralement plus faibles sans augmentation de données sur CIFAR et le fait que KDEF reste une base de données assez facile. Le fait que leurs gradients aient été plus faibles semble avoir causé des distributions des poids différentes des autres modèles, mais qui restaient néanmoins les mêmes entre les différentes exécutions. Aussi, tel qu'il sera présenté plus loin, ceci n'a pas empêché la plupart de ces cas d'exception d'obtenir de meilleures performances à l'aide de la SFSC.

Ces tableaux démontrent également que l'utilisation de deux Norm0 (au lieu d'un seul) est préférable étant donné que les poids les plus élevés pour $\bar{M}$ et $\widehat{M}$ sont généralement assez différents. Expérimentalement, l'utilisation de $\bar{M}$ en remplacement de M ainsi que l'utilisation de deux Norm0 au lieu d'un seul pour les deux chemins dans la figure 2.1, avaient notamment permis de largement augmenter la reproductibilité et la fiabilité des résultats vis-à-vis du choix des fonctions d'activations utilisées par chacune des couches de la SFSC.

Par rapport à l'AC, on peut notamment constater que tous les modèles sur CIFAR et sur DTD avec des CNN pré-entraînés ont visé l'utilisation du poids #1 (moyenne des couleurs). De l'autre côté, ceux sur DTD *from scratch* ont plutôt visé le poids #-1 (ne pas faire de moyennage). Dans le cas de KDEF, les quatre premiers modèles du tableau 4.4 ont visé le poids #1 tandis que les deux derniers ont plutôt visé le poids #-1. Cependant, cette déviation pour ces deux derniers cas semble surtout reliée au choix de la valeur de la réduction du taux d'apprentissage qui est assez

faible et a donc mené à des réductions plus importantes des poids de l'AC en raison du *weight decay*.

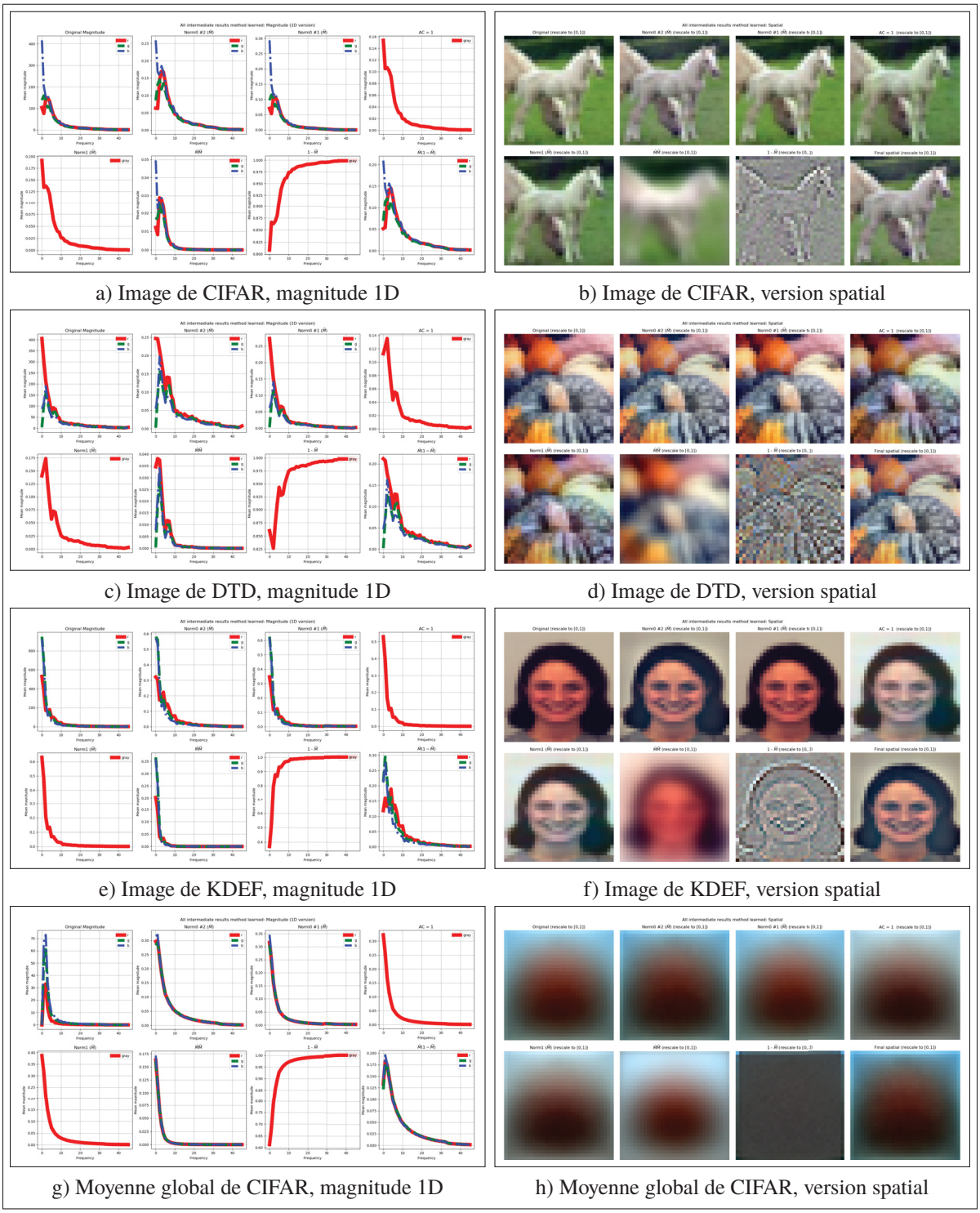
Le fait que les modèles entraînés sur CIFAR ainsi que ceux pré-entraînés sur ImageNet aient visé le même choix pour l'AC montre que le moyennage des canaux de couleurs des entrées est préférable pour la SFSC sur des bases de données d'images naturelles, en raison de la forte similarité entre les couleurs. De l'autre côté, il semble préférable de garder les couleurs séparées pour l'analyse des textures sur DTD. Étant donné que la SFSC s'adapte aux besoins et aux spécificités des modèles, ce sont les propriétés relatives aux images naturelles d'ImageNet qui prennent le dessus pour les modèles pré-entraînés utilisés sur DTD. Malgré ces similitudes, Norm0 #2 est de son côté différent entre les modèles de CIFAR et ceux pré-entraînés qui sont utilisés sur DTD, en ayant notamment un ordre de poids légèrement différents entre les trois poids les plus élevés. Dans le cas de KDEF, les tendances globales des poids appris par l'AC et Norm0 #2 sont notamment similaires à ceux sur CIFAR et sur DTD avec les modèles pré-entraînés, indiquant ainsi certaines similitudes sur les caractéristiques les importantes entre ces types d'images et ces tâches de classification.

En somme, ces différents tableaux sur les poids de la SFSC montrent notamment que ses poids suivent le plus souvent la même forme de distribution entre les modèles entraînés selon les mêmes conditions, et que l'utilisation de deux Norm0 et d'une AC permet de mieux s'adapter aux besoins des CNN. De l'autre côté, l'analyse de ces poids peut permettre d'y voir certaines similarités sur les besoins des modèles entre les bases de données et peut également fournir certaines informations sur les caractéristiques les plus importantes pour ces CNN. Ceci pourrait, par exemple, servir à effectuer certaines opérations qui exploiteraient davantage ces dimensions dans le domaine de Fourier ou dans d'autres domaines pour de futures recherches.

La prochaine section présente une analyse plus approfondie des effets de la SFSC à l'aide d'exemples sur les résultats intermédiaires et finaux obtenus par son utilisation.

4.2.2 Présentation d'exemples et analyse des effets de la SFSC sur les images

Afin de mieux comprendre les effets de la SFSC sur les images d'entrées des CNN, les figures 4.2, 4.3, 4.4 et 4.5 présentent des exemples des résultats intermédiaires et finaux obtenus par la SFSC avec les modèles ayant eu les écarts de poids des MN les plus élevés sur chacune des bases de données utilisées, soit respectivement CIFAR, DTD *from scratch*, DTD pré-entraîné et KDEF. Les quatre exemples utilisés sont une image de chacune des bases de données ainsi qu'une moyenne des résultats obtenus sur une époque d'apprentissage. Les images reçoivent les traitements associés à ces modèles, ce qui inclut un ajustement de leur taille selon celle qu'ils utilisent et une *Z-score normalization* qui va notamment mettre à zéro les DC de l'image moyenne de leur base de données, mais pas nécessairement les DC des images individuelles. De la même façon, les DC des images moyennes des bases de données d'apprentissage de DTD et KDEF ne sont pas exactement à zéro en raison d'un calcul effectué sur l'ensemble de la base de données au lieu d'être spécifique à chacun des *splits*. Ces traitements propres à chacune des bases données peuvent également donner des images originales qui sont visuellement différentes entre les modèles. Pour chacun des exemples, les sous-figures présentées sont dans l'ordre : l'image originale, le résultat de Norm0 #2 ($\bar{M}$), Norm0 #1 ($\widehat{M}$), l'AC, Norm1 (si l'AC = 1), $\bar{M}\widehat{M}$, $1 - \widehat{M}$ et le résultat final. Les figures de $\bar{M}\widehat{M}$ et $1 - \widehat{M}$ représentent alors respectivement ce qui est soustrait ou multiplié avec la sortie de Norm0 #2 ($\bar{M}$) afin d'obtenir le résultat final de la SFSC qui est $\bar{M}(1 - \widehat{M})$ (équation 2.5). Toutes les images spatiales sont mises avec des valeurs entre [0,1] pour mieux afficher leurs couleurs, tandis que les magnitudes 1D sont obtenues en effectuant la moyenne des magnitudes selon leur distance par rapport au centre (DC). Aucune échelle logarithmique n'a été utilisée pour les magnitudes étant donné que cela est plutôt inutile sur les magnitudes des étapes intermédiaires et finales.



a) Image de CIFAR, magnitude 1D

b) Image de CIFAR, version spatial

c) Image de DTD, magnitude 1D

d) Image de DTD, version spatial

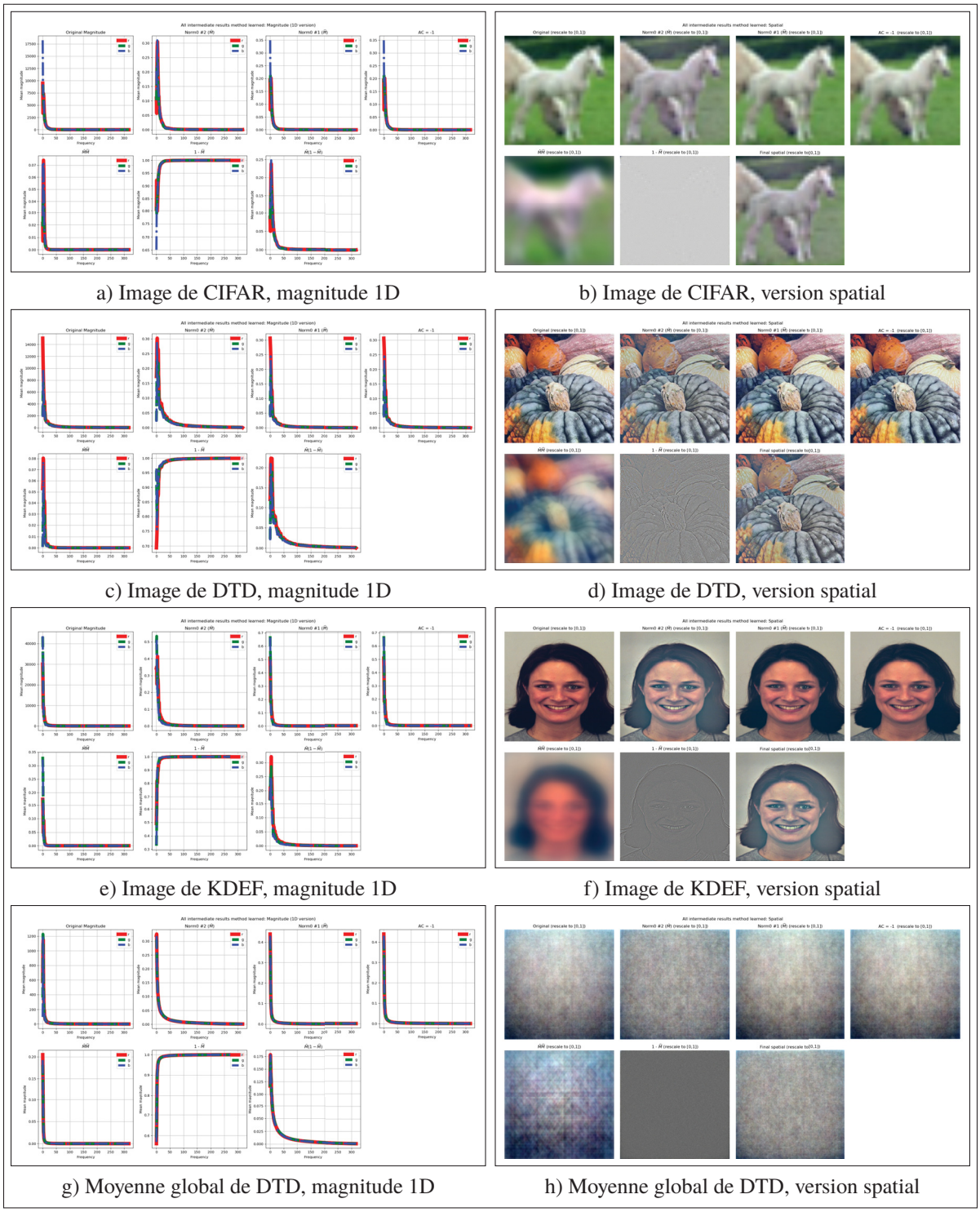
e) Image de KDEF, magnitude 1D

f) Image de KDEF, version spatial

g) Moyenne global de CIFAR, magnitude 1D

h) Moyenne global de CIFAR, version spatial

Figure 4.2 Les résultats intermédiaires obtenus par notre CNN de références sur CIFAR-100+ avec différentes images ainsi que la moyenne globale des résultats obtenus sur l'ensemble de la base de données d'apprentissage de CIFAR-100+



a) Image de CIFAR, magnitude 1D

b) Image de CIFAR, version spatial

c) Image de DTD, magnitude 1D

d) Image de DTD, version spatial

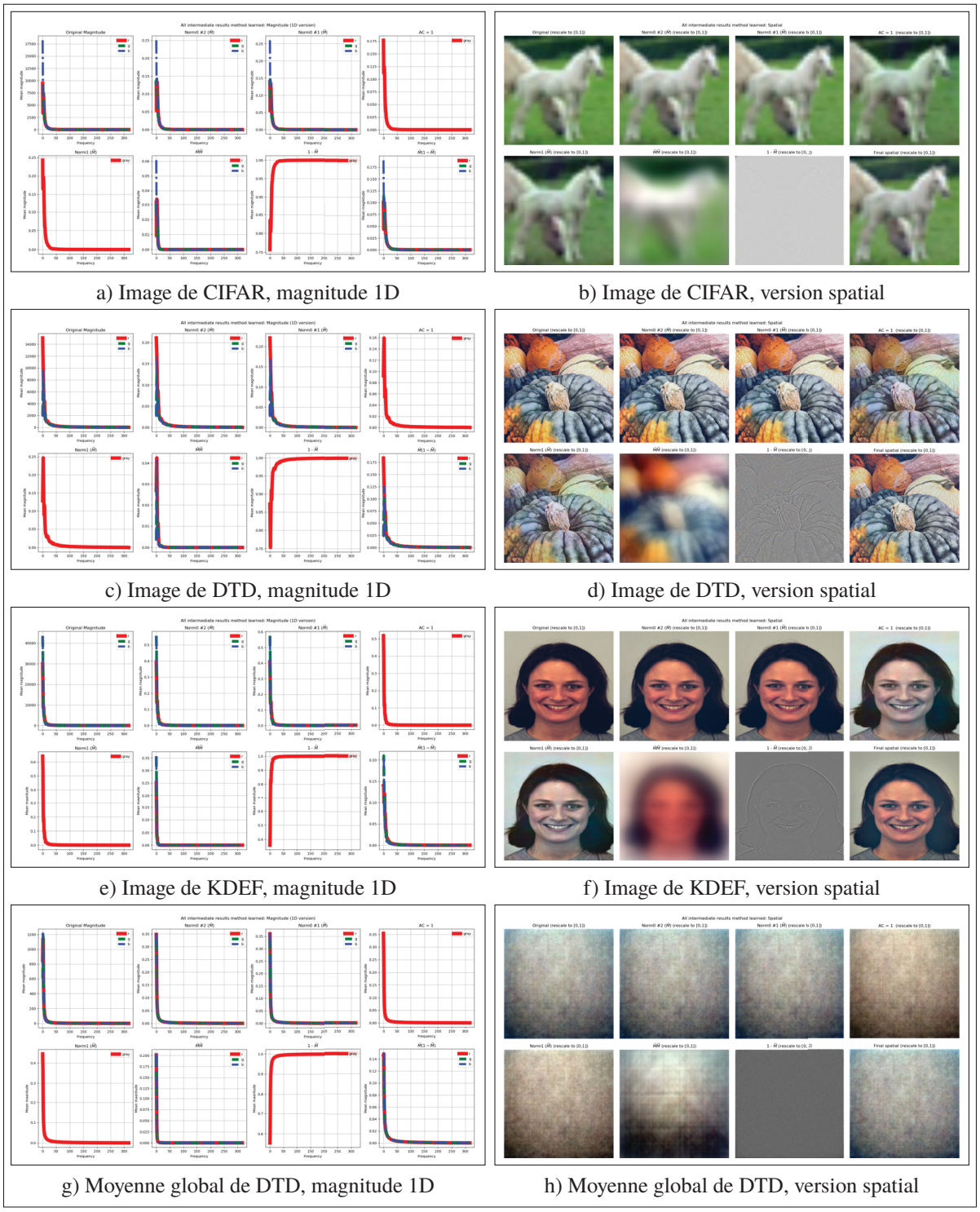
e) Image de KDEF, magnitude 1D

f) Image de KDEF, version spatial

g) Moyenne global de DTD, magnitude 1D

h) Moyenne global de DTD, version spatial

Figure 4.3 Les résultats intermédiaires obtenus par le AlexNet+BN sur DTD *from scratch* avec différentes images ainsi que la moyenne globale des résultats obtenus sur l'ensemble de la base de données d'apprentissage de DTD, *split* #7



a) Image de CIFAR, magnitude 1D

b) Image de CIFAR, version spatial

c) Image de DTD, magnitude 1D

d) Image de DTD, version spatial

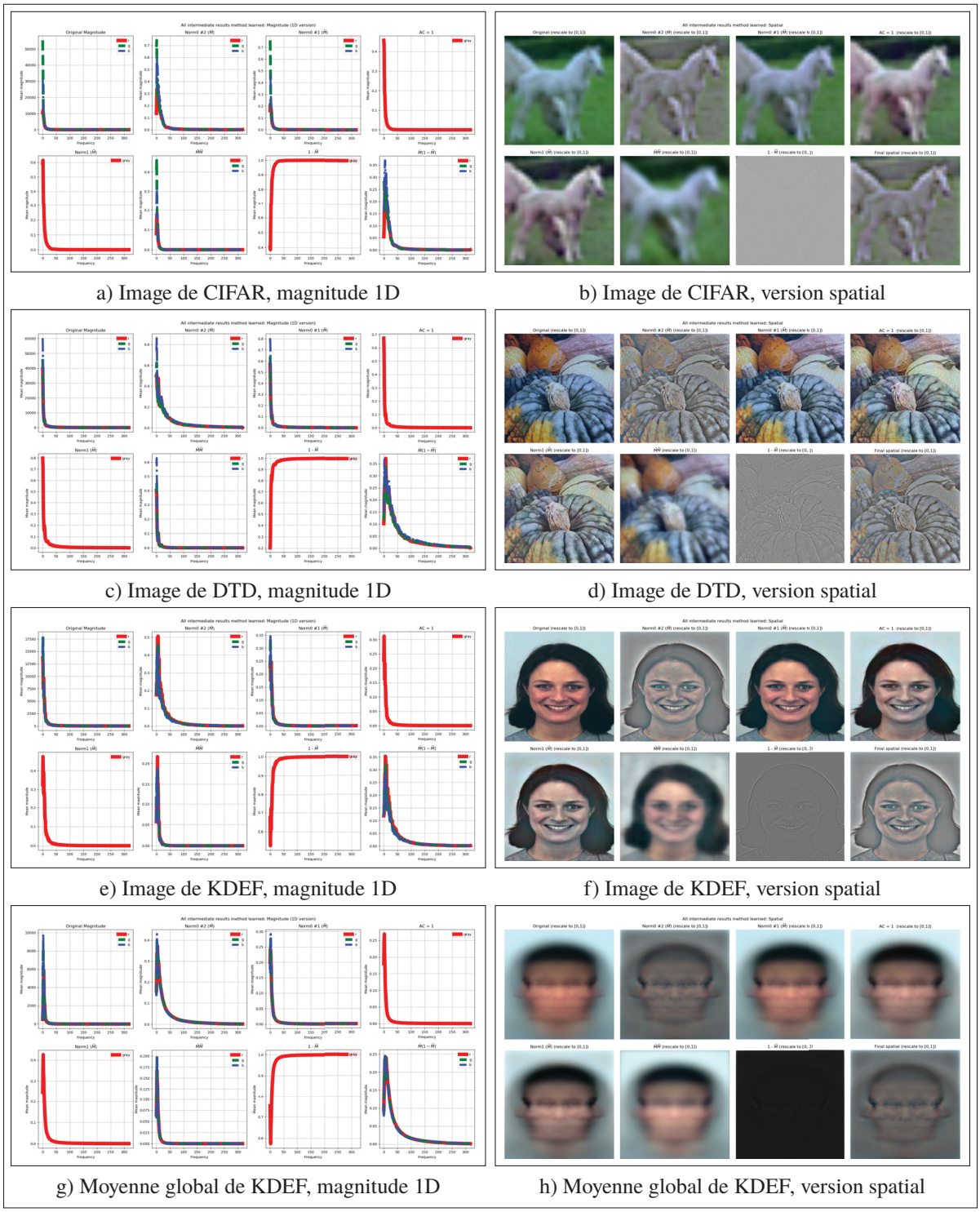
e) Image de KDEF, magnitude 1D

f) Image de KDEF, version spatial

g) Moyenne global de DTD, magnitude 1D

h) Moyenne global de DTD, version spatial

Figure 4.4 Les résultats intermédiaires obtenus par le ResNet-50 pré-entraîné (sur ImageNet) sur DTD avec différentes images ainsi que la moyenne globale des résultats obtenus sur l'ensemble de la base de données d'apprentissage de DTD, *split* #3



a) Image de CIFAR, magnitude 1D

b) Image de CIFAR, version spatial

c) Image de DTD, magnitude 1D

d) Image de DTD, version spatial

e) Image de KDEF, magnitude 1D

f) Image de KDEF, version spatial

g) Moyenne global de KDEF, magnitude 1D

h) Moyenne global de KDEF, version spatial

Figure 4.5 Les résultats intermédiaires obtenus par le ResNet-34 sur KDEF avec différentes images ainsi que la moyenne globale des résultats obtenus sur l'ensemble de la base de données d'apprentissage de KDEF, *split* #4

Comme le montrent notamment très bien les magnitudes 1D de la figure 4.2 où les images sont mises en 32×32 pixels, la SFSC a notamment pour effet de fortement diminuer les magnitudes très élevées des basses fréquences qui sont autour de DC, mais sans pour autant les éliminer, contrairement à plusieurs filtres passe-haut et passe-bande de la littérature. La forme globale des magnitudes reste similaire à celles originales, mais contrairement aux exemples de filtres fréquentiels fixes de la section 3.1, les proportions entre les magnitudes des canaux peuvent être modifiées d'une manière propre à chacune des fréquences de l'entrée. Cette modification influe notamment sur l'effet de grisâtrement des images qui a été souligné dans la section 3.1 et qui amène notamment une forte réduction de la saturation des couleurs et un niveau de luminosité plus neutre en raison d'écart moins prononcé entre les magnitudes des différents canaux de couleurs. Les arêtes et les détails des images deviennent alors plus visibles en ayant moins d'information sur les variations d'intensité globales, ce qui peut aider à accélérer la convergence des modèles en ayant plus rapidement accès à ces informations qui sont normalement apprises plus tard au cours de l'apprentissage (Rahaman *et al.*, 2019 ; Xu *et al.*, 2019 ; Xu, 2018). Cependant, étant donné que l'ensemble du spectre reste important (Tsuzuku & Sato, 2019 ; Yin *et al.*, 2019a) (voir Annexe I), le fait d'avoir malgré tout conservé les basses fréquences, au lieu de les éliminer ou de très fortement les atténuer, permet aux modèles d'aller chercher certaines informations sur les basses fréquences.

Selon les modèles, cet effet de grisâtrement qui est notamment présent sur les images finales de la SFSC semble surtout venir de Norm0 #2 ($\bar{M}$) (pour les modèles de CIFAR et DTD *from scratch*), de l'AC (pour le modèle de DTD pré-entraîné) ou les deux (pour le modèle de KDEF). Ceci s'explique cependant à l'aide des tableaux des poids de la section 4.2.1 et de l'équation 2.5 de la SFSC. Le niveau de proportionnalité entre les magnitudes des canaux de couleurs est conservé si l'on applique la même modification aux trois canaux de chaque image, comme le font notamment les poids #2 (*LN max*) et #5 (*Gray max image*) du tableau 2.1. Cependant, d'autres poids peuvent approximativement la conserver selon les propriétés des images (comme le poids #6 (*RGB max image*)) ou selon la combinaison des poids maximaux utilisés. Étant donné que $1 - \widehat{M}$ est un filtre passe-haut d'atténuation des basses fréquences qui est appliqué sur la

sortie de Norm0 #2 ($\bar{M}$), Norm0 #2 est celui qui va normalement le plus influencer la forme des magnitudes finales obtenues. Cependant, la partie $1 - \widehat{M}$ apporte une certaine contribution qui ressort notamment davantage lorsque celle de Norm0 #2 ($\bar{M}$) n'est pas assez présente, comme pour le modèle de DTD pré-entraîné où cette modification des images semble surtout venir de l'AC. Pour ce modèle ainsi que celui de KDEF qui ont l'AC = 1, bien que $\widehat{M}$ sera en nuance de gris, Norm0 #1 peut y effectuer certaines modifications en amont qui seront ensuite davantage mises en avant par l'AC. Norm1 peut également effectuer certains ajustements finaux pour le calcul de $1 - \widehat{M}$. Les deux parties de l'équation 2.5 sont donc importantes pour la SFSC étant donné qu'elles ont des contributions différentes sur les images finales de la SFSC. $\bar{M}$ semble surtout influencer la forme globale du résultat final tandis que $1 - \widehat{M}$ va plutôt apporter certaines corrections plus légères qui vont surtout affecter les basses ainsi que les moyennes fréquences si les images sont de taille assez petite comme sur CIFAR.

Au bout du compte, la SFSC a donc un certain effet d'épuration des couleurs similaire à celui qu'on peut notamment observer avec les filtres fréquentiels passe-haut, mais avec la différence notoire d'avoir uniquement atténué les basses fréquences au lieu de les éliminer. La SFSC effectue également des variations sur les proportionnalités des magnitudes des canaux de couleurs qui varient en fonction des fréquences et qui sont uniques à chaque image puisqu'elles sont utilisées pour s'auto-filtrer. Les images obtenues permettent de plus facilement accéder aux informations des moyennes et hautes fréquences, ce qui devrait aider les modèles à pouvoir converger un peu plus rapidement et potentiellement atteindre des performances plus élevées. Finalement, les deux parties de l'équation 2.5 de la SFSC ($\bar{M}$ et $1 - \widehat{M}$) apportent toutes les deux des contributions distinctes sur les aspects visuels des résultats finaux et semblent donc toutes les deux importants.

La prochaine section présente les résultats obtenus par les modèles originaux (sans nos méthodes) et ceux avec la SFSC qui devrait en principe obtenir de meilleures performances uniquement en raison d'une modification de leurs images d'entrées.

4.2.3 Présentation des résultats pour les modèles originaux et ceux avec la SFSC

Les tableaux 4.5, 4.6 et 4.7 présentent les résultats obtenus par les modèles originaux (sans nos méthodes) et ceux avec la SFSC sur respectivement CIFAR, DTD et KDEF. Pour chacun des tableaux, les performances obtenues sont présentées selon les moyennes, écarts-types et médianes ainsi qu'une description de la distribution globale des gains et des pertes de performances obtenues par la SFSC par rapport aux modèles originaux, lors de chacune des exécutions. Cette description sur la distribution des écarts de performances a notamment été ajoutée afin d'avoir un aperçu plus détaillé sur la reproductibilité et la fiabilité des résultats obtenus puisqu'ils peuvent parfois être remis en question dans la littérature. En effet, un gain de performance global peut être lié à de la chance avec, par exemple, uniquement 50 % ou 60 % des cas réussis, ou des écarts-types plus élevés que les gains globaux mesurés. L'ajout de cette description permet donc de savoir ce qu'il en est réellement, que les différences de performances globales des moyennes ou médianes soient positives ou négatives. Comme dit dans la section 4.1, tous les modèles sur CIFAR ont eu cinq exécutions sauf pour le CNN de référence qui en a eu 10, comme ceux sur DTD et KDEF où les statistiques sont selon leurs 10 *splits*. Pour les quatre cas de CIFAR, les "+" indiquent les cas avec l'utilisation de l'augmentation de données standards et toutes les expérimentations sur DTD et KDEF ont de l'augmentation de données. Étant donné que la SFSC ajoute uniquement 20 poids aux CNN, le nombre de paramètres présenté pour chacun des modèles inclut l'utilisation de la SFSC.

Tableau 4.5 Résultats des CNN sur CIFAR avec 5 exécutions aux poids aléatoires différents (10 exécutions pour notre CNN de référence). Les "+" indiquent avec l'augmentation de données et les chiffres entre parenthèses indiquent les différences par rapport aux modèles originaux

Architecture	Dataset	Params	Original		Avec la SFSC		Nb différence positive		Nb différence négative	
			Mean ± Std	Median	Mean ± Std	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %
ResNet-56	CIFAR-10	855,79k	86,33 ± 1,05	86,67	**87,99 ± 0,24 (1,66)**	**87,99 (1,32)**	4	1	0	0
	CIFAR-10+		93,33 ± 0,09	93,34	**93,65 ± 0,23 (0,33)**	**93,67 (0,33)**	0	5	0	0
	CIFAR-100	861,64k	56,78 ± 1,15	56,09	**59,69 ± 0,52 (2,91)**	**59,82 (3,73)**	5	0	0	0
	CIFAR-100+		70,43 ± 0,28	70,51	**70,88 ± 0,33 (0,45)**	**71,08 (0,57)**	0	5	0	0
ResNet-110	CIFAR-10	1,73M	87,16 ± 0,26	87,14	**88,18 ± 0,38 (1,02)**	**88,26 (1,12)**	3	2	0	0
	CIFAR-10+		93,88 ± 0,14	93,81	**94,13 ± 0,06 (0,26)**	**94,12 (0,31)**	0	5	0	0
	CIFAR-100	1,74M	59,68 ± 0,53	59,74	**60,93 ± 0,46 (1,25)**	**61 (1,26)**	3	2	0	0
	CIFAR-100+		72,2 ± 0,42	72,35	**72,55 ± 0,27 (0,35)**	**72,63 (0,28)**	0	3	2	0
VGG-11+BN	CIFAR-10	9,76M	87,05 ± 0,35	87,11	**88,51 ± 0,29 (1,46)**	**88,59 (1,48)**	4	1	0	0
	CIFAR-10+		92,41 ± 0,14	92,39	**92,82 ± 0,2 (0,41)**	**92,86 (0,47)**	0	5	0	0
	CIFAR-100	9,8M	60,27 ± 0,29	60,37	**62,31 ± 0,23 (2,04)**	**62,29 (1,92)**	5	0	0	0
	CIFAR-100+		70,57 ± 0,18	70,47	**71,77 ± 0,13 (1,2)**	**71,72 (1,25)**	4	1	0	0
VGG-16+BN	CIFAR-10	15,25M	89,11 ± 0,19	89,18	**89,92 ± 0,24 (0,82)**	**90,03 (0,85)**	2	3	0	0
	CIFAR-10+		93,9 ± 0,22	93,9	**94,11 ± 0,09 (0,21)**	**94,11 (0,21)**	0	5	0	0
	CIFAR-100	15,3M	63,68 ± 0,51	63,79	**64,7 ± 0,38 (1,01)**	**64,68 (0,89)**	2	3	0	0
	CIFAR-100+		73,87 ± 0,18	73,93	**74,14 ± 0,18 (0,27)**	**74,08 (0,15)**	0	5	0	0
DenseNet {L = 100, K =12}	CIFAR-10	769,18k	**93,78 ± 0,19**	**93,87**	93,7 ± 0,13 (-0,08)	93,64 (-0,23)	0	2	3	0
	CIFAR-10+		95,14 ± 0,28	**95,21**	**95,18 ± 0,17 (0,04)**	95,12 (-0,09)	0	3	2	0
	CIFAR-100	800,05k	**75,02 ± 0,41**	**74,82**	74,35 ± 0,25 (-0,68)	74,38 (-0,44)	0	0	4	1
	CIFAR-100+		**76,81 ± 0,47**	**77,15**	76,42 ± 0,19 (-0,39)	76,39 (-0,76)	0	2	3	0
WRN 16-8	CIFAR-10	10,97M	**91 ± 0,61**	**90,92**	90,68 ± 0,5 (-0,32)	90,41 (-0,51)	0	1	4	0
	CIFAR-10+		**95,49 ± 0,15**	**95,5**	95,37 ± 0,12 (-0,12)	95,38 (-0,12)	0	2	3	0
	CIFAR-100	11,01M	**66,96 ± 0,61**	**66,78**	66,24 ± 0,54 (-0,71)	66,43 (-0,35)	0	1	2	2
	CIFAR-100+		**79,29 ± 0,2**	**79,36**	78,54 ± 0,16 (-0,75)	78,53 (-0,83)	0	0	4	1
CNN de référence	CIFAR-10	58,61k	83,13 ± 0,32	83,09	**83,49 ± 0,27 (0,35)**	**83,46 (0,37)**	0	7	3	0
	CIFAR-10+		84,77 ± 0,26	84,8	**84,93 ± 0,37 (0,15)**	**84,88 (0,08)**	0	6	4	0
	CIFAR-100	61,49k	45,76 ± 0,4	45,74	**46,36 ± 0,28 (0,6)**	**46,45 (0,71)**	2	8	0	0
	CIFAR-100+		46,07 ± 0,25	46,01	**46,6 ± 0,33 (0,53)**	**46,6 (0,59)**	1	9	0	0

Tableau 4.6 Résultats des CNN pré-entraînés et *from scratch* sur DTD avec les statistiques reliées à leur exécution sur les 10 *splits*. Les chiffres entre parenthèses indiquent les différences par rapport aux modèles originaux

Cas de DTD	Architecture	Params	Original		Avec la SFSC		Nb différence positive		Nb différence négative	
			Mean ± Std	Median	Mean ± Std	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %
DTD *from scratch*	CNN de référence	59,8k	39,18 ± 0,91	39,13	**39,56 ± 0,79 (0,38)**	**39,5 (0,37)**	4	1	4	1
	ResNet-18	11,2M	43,42 ± 1,29	43,14	**45,26 ± 1,54 (1,84)**	**45,3 (2,16)**	5	4	1	0
	ResNet-34	21,31M	44,99 ± 2,38	45,13	**47,81 ± 2,38 (2,82)**	**48,01 (2,88)**	8	1	1	0
	ResNet-50	23,6M	45,81 ± 1,69	46,31	**48,77 ± 1,16 (2,96)**	**48,62 (2,31)**	9	1	0	0
	BN-Inception	10,32M	48,84 ± 0,94	48,91	**49,93 ± 1,34 (1,09)**	**49,6 (0,69)**	7	2	0	1
	VGG-11+BN	128,96M	40,9 ± 1,74	40,48	**44,25 ± 1,69 (3,35)**	**44,04 (3,57)**	10	0	0	0
	VGG-M+BN	99,01M	43,13 ± 0,88	42,9	**45,2 ± 1,19 (2,07)**	**45,14 (2,24)**	9	1	0	0
	AlexNet+BN	57,2M	41,46 ± 1,12	41,22	**43,49 ± 1,6 (2,03)**	**43,41 (2,19)**	8	2	0	0
DTD pré-entraîné sur ImageNet	ResNet-18	11,2M	67,4 ± 0,99	67,5	**67,77 ± 0,64 (0,37)**	**67,8 (0,29)**	1	7	2	0
	ResNet_34	21,31M	69,51 ± 0,65	69,29	**69,85 ± 1,2 (0,34)**	**69,53 (0,24)**	1	7	1	1
	ResNet-50	23,6M	71,31 ± 1,11	71,49	**72,13 ± 0,64 (0,82)**	**72,1 (0,61)**	7	2	0	1
	BN-Inception	10,32M	66,86 ± 1,11	66,68	**67,01 ± 1,26 (0,15)**	**66,94 (0,26)**	1	6	3	0
	VGG-11+BN	128,96M	70,18 ± 0,67	70,06	**70,4 ± 0,83 (0,22)**	**70,13 (0,08)**	0	8	2	0

Tableau 4.7 Résultats des CNN sur KDEF avec les statistiques reliées à leur exécution sur les 10 *splits*. Les chiffres entre parenthèses indiquent les différences par rapport aux modèles originaux

Architecture	Params	Original		Avec la SFSC		Nb différence positive		Nb différence négative	
		Mean ± Std	Median	Mean ± Std	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %
CNN de référence	58,52k	83,52 ± 2,21	83,83	**85,02 ± 1,4 (1,5)**	**84,44 (0,61)**	6	2	1	1
ResNet-18	11,18M	91,4 ± 1,4	91,73	**92 ± 1,14 (0,6)**	**92,04 (0,31)**	2	6	2	0
ResNet-34	21,29M	91,8 ± 1,05	91,79	**92,14 ± 1,11 (0,35)**	**92,5 (0,72)**	2	5	2	1
ResNet-50	23,52M	90,5 ± 1,78	90,66	**91,31 ± 1,71 (0,81)**	**91,23 (0,56)**	5	4	1	0
BNInception	10,28M	91,67 ± 1,18	91,89	**92,42 ± 0,78 (0,75)**	**92,35 (0,46)**	3	6	1	0

Pour les expérimentations sur les quatre cas de CIFAR qui sont présentés dans le tableau 4.5, tous les résultats obtenus avec le ResNet-56, ResNet-110, VGG-11+BN, VGG-16+BN et le CNN de référence présentent des améliorations de performances globales relativement importantes. Parmi ceux-ci, seul le ResNet-110 sur CIFAR-100+ et le CNN de référence sur CIFAR-10/10+ n'ont pas eu l'ensemble de leurs exécutions amélioré par la SFSC. Les gains de performances obtenues par la SFSC sont généralement plus élevés sur les cas sans augmentation de données où la simplification des images combinée au léger effet d'augmentation de données de l'AC semble avoir contribué à réduire le surapprentissage des modèles pour les aider à atteindre de meilleures performances. Malgré tout, la SFSC améliore également les performances de ces modèles pour les cas avec de l'augmentation de données où ils présentent alors moins de surapprentissage et des performances plus élevées.

Néanmoins, les deux meilleurs modèles expérimentés sur CIFAR qui sont le WRN 16-8 et le DenseNet (L = 100, K = 12) présentent de leur côté des baisses de performances légères qui sont le plus souvent de moins de 1 %. Dans le cas du WRN 16-8, cette baisse des performances est relativement légère par rapport à d'autres méthodes de la littérature comme la *scattering transform* d'ordre deux (-1,41 à -2,60 % sur CIFAR-10+ et -7,3 % sur CIFAR-100+) (Oyallon *et al.*, 2017 ; Ren *et al.*, 2019 ; Rodriguez *et al.*, 2020) ou celle avec des ondelettes apprises de Rodriguez *et al.* (2020) (-1,97 % sur CIFAR-10+ et -4,69 % sur CIFAR-100+) et elles ajoutent toutes les deux environ 34,5M de paramètres au modèle original. Cependant, d'autres comme la méthode des filtres de Gabor de Ma *et al.* (2020) présentent de légers gains de performances sur CIFAR-10+/100+ avec le WRN 16-8. Dans le cas de la SFSC, ces baisses de performances pour ces deux modèles peuvent notamment être liées à leur architecture particulière qui leur

permet de potentiellement utiliser plus aisément certaines informations des images qui ont été supprimées et qui seraient nécessaires pour atteindre des performances aussi élevées (principe de *data processing inequality*). En effet, le DenseNet possède des connexions denses qui permettent aux couches de convolution d'avoir accès à l'ensemble des sorties des couches précédentes de son bloc et le WRN possède beaucoup de filtres sur peu de profondeur. De l'autre côté, étant donné que les différences de performances sont relativement faibles, cela peut également être relié à d'autres facteurs plus mineurs qui sont propres à ces deux modèles. Il y a notamment l'utilisation de la forme *pre-activation* (BN, ReLU, conv) pour les couches de convolution et l'utilisation d'un *nesterov momentum* pour la partie du CNN alors que la SFSC fonctionne mieux avec un *momentum* normal. Cependant, il est peu probable qu'un changement de ces deux facteurs mineurs soit suffisant pour obtenir une tendance globale positive pour ces deux modèles.

Dans le cas des expérimentations sur DTD qui sont présentées dans le tableau 4.6, tous ont eu des améliorations des performances considérables où la majorité ou l'ensemble des exécutions effectuées avec les SFSC ont eu des améliorations de performances par rapport aux modèles originaux. Dans le cas de DTD *from scratch* où le surapprentissage est particulièrement très élevé pour les modèles ayant plusieurs millions de paramètres, la SFSC a permis des améliorations de performances relativement élevées en aidant les modèles à converger mieux et plus rapidement. Bien que le surapprentissage était moins élevé pour le CNN de référence, la SFSC a quand même eu une tendance globale d'amélioration de performances pour ce très petit modèle. Dans le cas de DTD avec des CNN pré-entraînés sur ImageNet, où la plupart des recherches de la littérature ne modifient que les sorties de leurs modèles, la SFSC a, de son côté, permis des améliorations de performances légères, mais fiables et reproductives, en ne modifiant que leurs entrées. Tel que présenté dans les sections 4.2.1 et 4.2.2, la SFSC s'est, pour ce faire, adaptée au fait que le CNN avait été initialement entraîné sur des images naturelles afin d'obtenir des poids et des rendus d'images similaires à ceux des modèles de CIFAR. Les images obtenues présentent alors des corrections plus mineures que ceux des modèles de DTD *from scratch*, mais permettent des améliorations de performances sur ces modèles qui n'ont eu que très peu de modifications

sur leurs couches de convolution en raison du taux d'apprentissage très faible utilisé pour les couches pré-entraînées.

Dans le cas des expérimentations sur KDEF qui sont présentées dans le tableau 4.7, tous les modèles ont eu des améliorations des performances lors de la majorité de leur exécution avec la SFSC. Bien que les performances maximales globales des modèles originaux soient autour de 90 à 91 %, la SFSC a pu les aider à aller un peu plus loin sur cette base de données relativement facile avec une difficulté similaire à CIFAR-10+, mais qui présente beaucoup de surapprentissage et des risques de problèmes liés à des convergences trop rapides pour certains modèles (voir la section 4.1).

En somme, les résultats obtenus sur CIFAR, DTD et KDEF montrent que la SFSC est capable d'améliorer les performances de divers CNN aux types d'architectures, profondeurs et aux nombres de paramètres variés et ce, en changeant uniquement leurs images d'entrée. Elle est adaptée aux apprentissages *from scratch* ainsi qu'aux modèles pré-entraînés et elle ne nécessite aucune modification de l'architecture ou de la méthodologie d'apprentissage relatif à ce qui concerne la partie du CNN. Malgré tout, l'un de ses principaux défauts est le fait qu'il est parfois nécessaire de faire certains ajustements sur les hyperparamètres d'apprentissage globaux de la SFSC pour obtenir des améliorations de performances globales. Cependant, les meilleures configurations d'une base de données sont parfois également les mêmes pour une autre situation qui présente des problématiques de gradients similaires, comme cela a été le cas pour DTD avec des modèles pré-entraînés et KDEF (section 4.1). De l'autre côté, comme dans le cas de CIFAR, la SFSC possède certains besoins spécifiques qui seront le plus souvent les mêmes pour les différents modèles et avec, au mieux, quelques ajustements mineurs pour optimiser ces performances. Étant donné que les poids des MN et de l'AC sont des vecteurs 1D, l'analyse de l'évolution de leurs poids au cours de l'apprentissage est également facilitée par rapport à des poids ayant davantage de dimensions comme ceux des couches de convolution ou d'autres méthodes de la littérature.

La prochaine section présente les résultats expérimentaux et les poids appris par les filtres fréquentiels du chapitre 3.

4.3 Résultats expérimentaux pour les filtres fréquentiels en loi de puissance

De manière similaire à ce qui a été fait dans la section 4.2, cette section présente les poids appris et les résultats obtenus par les quatre configurations des filtres fréquentiels du tableau 3.1. Étant donné que les sections 3.1 et 4.2.2 exposent respectivement des exemples sur l'application de filtres fréquentiels fixes et de la SFSC sur des images, cette section n'inclut pas d'exemple relié à l'application des filtres fréquentiels appris. Il y aurait autrement beaucoup trop de figures en raison de l'utilisation de quatre configurations de filtres fréquentiels et d'un grand nombre de sous-configurations de poids appris sur les différentes bases de données, comme il sera souligné dans la prochaine section.

4.3.1 Présentation des poids appris par les filtres fréquentiels

Comme pour la SFSC, l'analyse des poids appris par les filtres fréquentiels permet de mieux comprendre comment ceux-ci se sont adaptés aux différents modèles et bases de données qui ont été utilisés. Pour ce faire, les tableaux 4.8, 4.9 et 4.10 présentent les moyennes des poids appris par les filtres fréquentiels du tableau 3.1 sur, respectivement, CIFAR, DTD *from scratch* et KDEF. Tel que spécifié dans le tableau 3.1, les filtres #1 et #2 s'appliquent sur la sortie d'une SFSC qui est apprise en même temps que les filtres fréquentiels tandis que les filtres #3 et #4 s'appliquent directement sur les images d'entrées des CNN. Dans le cas des SFSC des filtres #1 et #2, leur distribution des poids est généralement similaire à celle des SFSC originaux (de la section 4.2.1) avec, au mieux, un ordre légèrement différent entre les deux ou trois poids maximaux des MN et les différences peuvent être plus accentuées si les MN de leur SFSC ont de faibles écarts de poids. Par conséquent, les distributions détaillées des SFSC des filtres #1 et #2 ont été omises et seuls les écarts entre les poids maximaux et minimaux des MN et les différences entre ces écarts de poids et ceux des SFSC originaux sont présentés. Dans le cas du filtre #3 qui apprend uniquement ω, celui-ci est omis des tableaux puisqu'aucune des statistiques

n'est présentée par rapport à ω. Comme pour les tableaux de la section 4.2.1, des échelles de couleurs propres à chacun des MN appris des filtres #2 et #4 mettent en évidence les poids les plus élevés et les plus faibles. En ce qui concerne les colonnes montrant les écarts entre les valeurs maximales et minimales de chaque MN ainsi que les médianes des γ appris par le filtre #1, des échelles de couleurs globales spécifiques à chacune de ces colonnes mettent en évidence les différences entre les modèles.

Tableau 4.8 Tableau des moyennes des poids appris par les filtres fréquentiels sur les 5 exécutions aux poids aléatoires avec les différents modèles sur CIFAR (10 exécutions pour notre CNN de référence). Les poids des MN et les écarts entre leur poids maximal et minimal sont en %

Architecture	Dataset	No	γ (si appris)		Norm2 (pour le filtre fréquentiel)				AC	Écart Max-Min				Diff écart vs SFSC		
			Mean ± Std	Median	(0 or 2)	(1 or 4)	(3 or 6)	(5)		Norm2	Norm0 #1	Norm0 #2	Norm1	Norm0 #1	Norm0 #2	Norm1
ResNet-56	CIFAR-10	1	0,73 ± 0,04	0,72					1		8,18	27,88	19,94	1,36	-3,99	4,12
		2	0,5		33,04	33,04	16,96	16,96	1	16,07	7,18	25,99	17,29	0,36	-5,88	1,46
		4	1		31,87	31,87	18,19	18,07		13,80						
	CIFAR-10+	1	0,57 ± 0,05	0,58					1		22,67	40,25	52,40	-1,63	12,18	-0,93
		2	0,5		29,25	29,25	20,75	20,75	1	8,51	22,69	41,69	53,08	-1,61	13,62	-0,25
		4	1		24,49	24,49	25,61	25,40		1,12						
	CIFAR-100	1	0,78 ± 0,05	0,76					1		12,89	41,83	38,07	-4,42	-5,92	-5,92
		2	0,5		38,37	38,37	11,63	11,63	1	26,75	15,49	37,59	41,96	-1,82	-10,16	-2,03
		4	1		34,34	34,34	15,76	15,55		18,79						
	CIFAR-100+	1	0,54 ± 0,06	0,51					1		36,81	57,43	73,98	-2,62	21,79	-2,62
		2	0,5		30,11	30,11	19,89	19,89	1	10,22	37,83	58,47	75,39	-1,60	22,82	-1,20
		4	1		21,97	21,97	29,12	26,94		7,15						
ResNet-110	CIFAR-10	1	0,73 ± 0,09	0,7					1		7,70	27,02	19,08	-0,32	-0,65	0,40
		2	0,5		32,33	32,33	17,67	17,67	1	14,66	7,34	25,67	18,75	-0,69	-1,99	0,07
		4	1		30,04	30,04	20,02	19,90		10,14						
	CIFAR-10+	1	0,53 ± 0,06	0,56					1		22,60	38,79	51,83	-0,55	11,48	0,37
		2	0,5		25,46	25,46	24,54	24,54	1	0,91	22,43	42,08	52,55	-0,72	14,78	1,09
		4	1		22,67	22,67	27,37	27,29		4,69						
	CIFAR-100	1	0,72 ± 0,06	0,69					1		13,80	40,88	39,45	-4,77	-2,38	-4,92
		2	0,5		35,38	35,38	14,62	14,62	1	20,75	15,94	38,90	42,12	-2,63	-4,36	-2,26
		4	1		32,65	32,65	17,44	17,26		15,40						
	CIFAR-100+	1	0,53 ± 0,04	0,51					1		35,75	53,39	71,19	-2,29	17,43	-1,91
		2	0,5		28,48	28,48	21,52	21,52	1	6,96	36,13	56,01	72,37	-1,92	20,05	-0,73
		4	1		19,05	19,05	32,15	29,75		13,10						
VGG-11+BN	CIFAR-10	1	0,88 ± 0,01	0,88					1		7,02	20,63	13,21	-0,22	-24,89	-0,62
		2	0,5		37,45	37,45	12,55	12,55	1	24,91	9,68	32,65	13,28	2,44	-12,87	-0,55
		4	1		37,90	37,89	12,11	12,09		25,81						
	CIFAR-10+	1	0,71 ± 0	0,71					1		14,96	29,50	36,84	-1,85	-27,03	-3,09
		2	0,5		38,92	38,92	11,08	11,08	1	27,84	14,81	39,83	36,49	-1,99	-16,70	-3,43
		4	1		36,53	36,53	13,48	13,46		23,08						
	CIFAR-100	1	0,87 ± 0,01	0,87					1		11,93	27,70	23,26	-2,88	-31,87	-4,68
		2	0,5		40,20	40,20	9,80	9,80	1	30,41	18,08	42,71	29,78	3,27	-16,85	1,83
		4	1		40,23	40,22	9,78	9,78		30,45						
	CIFAR-100+	1	0,72 ± 0	0,72					1		23,61	36,57	56,00	-2,46	-34,82	-2,37
		2	0,5		42,57	42,57	7,43	7,43	1	35,14	23,21	49,83	56,27	-2,86	-21,55	-2,10
		4	1		37,45	37,44	12,60	12,51		24,94						
VGG-16+BN	CIFAR-10	1	1,16 ± 0,04	1,14					1		5,23	22,42	13,42	-1,79	-18,54	-0,11
		2	0,5		35,45	35,45	14,55	14,55	1	20,89	8,90	27,08	14,11	1,88	-13,88	0,58
		4	1		35,56	35,54	14,46	14,44		21,12						
	CIFAR-10+	1	0,81 ± 0,01	0,8					1		18,23	33,49	44,20	-0,09	-15,25	1,06
		2	0,5		37,73	37,73	12,27	12,27	1	25,46	17,06	25,77	42,38	-1,26	-22,96	-0,75
		4	1		35,70	35,68	14,30	14,32		21,40						
	CIFAR-100	1	1,25 ± 0,01	1,24					1		8,60	33,82	25,99	-0,72	-18,65	6,07
		2	0,5		38,62	38,62	11,38	11,38	1	27,24	12,33	35,48	22,09	3,01	-16,98	2,17
		4	1		38,19	38,19	11,82	11,80		26,40						
	CIFAR-100+	1	0,8 ± 0,01	0,8					1		28,83	44,03	63,10	-1,16	-13,62	-0,59
		2	0,5		41,07	41,07	8,93	8,93	1	32,15	29,07	36,70	64,17	-0,92	-20,95	0,48
		4	1		36,79	36,79	13,25	13,17		23,62						
DenseNet {L = 100, K = 12}	CIFAR-10	1	0,61 ± 0,03	0,61					1		11,57	32,01	27,15	0,48	3,74	4,01
		2	0,5		29,65	29,65	20,35	20,35	1	9,29	12,43	32,54	28,95	1,35	4,27	5,81
		4	1		28,65	28,64	21,66	21,05		7,59						
	CIFAR-10+	1	0,35 ± 0,03	0,36					1		22,34	39,01	49,25	-1,02	10,34	-0,43
		2	0,5		17,75	17,75	32,25	32,25	1	14,50	23,11	39,27	50,86	-0,25	10,60	1,17
		4	1		10,63	10,63	39,71	39,02		29,08						
	CIFAR-100	1	0,68 ± 0,02	0,68					1		18,75	50,78	45,35	-3,34	12,28	-2,74
		2	0,5		31,71	31,71	18,29	18,29	1	13,41	23,69	50,88	53,70	1,60	12,38	5,61
		4	1		32,07	32,06	18,39	17,48		14,59						
	CIFAR-100+	1	0,38 ± 0,06	0,39					1		34,94	56,60	69,05	-0,92	18,26	-1,45
		2	0,5		13,44	13,44	36,56	36,56	1	23,11	35,58	55,61	70,59	-0,28	17,27	0,09
		4	1		10,03	10,03	40,16	39,78		30,13						

Tableau 4.8 Tableau des moyennes des poids appris par les filtres fréquentiels sur CIFAR (suite)

Architecture	Dataset	No	γ (si appris)		Norm2 (pour le filtre fréquentiel)				AC	Écart Max-Min				Diff écart vs SFSC		
			Mean ± Std	Median	(0 or 2)	(1 or 4)	(3 or 6)	(5)		Norm2	Norm0 #1	Norm0 #2	Norm1	Norm0 #1	Norm0 #2	Norm1
WRN 16-8	CIFAR-10	1	0,65 ± 0,08	0,63					1		18,28%	22,82%	29,21%	-0,52%	-11,74%	9,13%
		2	0,5		33,89%	33,89%	16,11%	16,11%	1	17,78%	24,69%	30,46%	25,75%	5,89%	-4,10%	5,68%
		4	1		39,94%	39,92%	10,09%	10,05%		29,89%						
	CIFAR-10+	1	0,55 ± 0,05	0,53					1		21,38%	38,91%	52,22%	1,32%	9,52%	7,86%
		2	0,5		28,65%	28,65%	21,35%	21,35%	1	7,31%	22,02%	36,17%	53,22%	1,97%	6,78%	8,86%
		4	1		33,54%	33,53%	16,51%	16,41%		17,13%						
	CIFAR-100	1	0,58 ± 0,05	0,57					1		28,24%	34,98%	31,97%	-12,86%	3,84%	-3,28%
		2	0,5		41,74%	41,74%	8,26%	8,26%	1	33,48%	30,68%	31,81%	36,03%	-10,42%	0,67%	0,78%
		4	1		42,14%	42,12%	7,89%	7,85%		34,30%						
	CIFAR-100+	1	0,37 ± 0,01	0,37					1		31,09%	39,23%	67,41%	1,82%	-2,03%	4,98%
		2	0,5		21,74%	21,74%	28,26%	28,26%	1	6,53%	31,28%	39,14%	67,76%	2,00%	-2,11%	5,34%
		4	1		34,71%	34,70%	15,42%	15,17%		19,54%						
CNN de référence	CIFAR-10	1	0,57 ± 0,03	0,57					1		29,56%	36,07%	62,63%	0,68%	-15,50%	2,59%
		2	0,5		31,32%	31,32%	18,68%	18,68%	1	12,63%	29,03%	42,72%	61,33%	0,15%	-8,85%	1,29%
		4	1		36,13%	36,05%	17,92%	9,90%		26,23%						
	CIFAR-10+	1	0,48 ± 0,02	0,48					1		33,00%	44,11%	68,86%	-0,40%	-1,48%	0,87%
		2	0,5		24,03%	24,03%	25,97%	25,97%	1	1,94%	31,98%	51,05%	68,08%	-1,42%	5,46%	0,09%
		4	1		24,31%	24,40%	30,58%	20,71%		9,87%						
	CIFAR-100	1	0,62 ± 0,03	0,62					1		35,03%	45,84%	73,25%	-0,87%	-7,82%	0,19%
		2	0,5		36,98%	36,98%	13,02%	13,02%	1	23,97%	35,39%	60,10%	74,26%	-0,51%	6,43%	1,20%
		4	1		36,27%	36,12%	18,56%	9,05%		27,22%						
	CIFAR-100+	1	0,57 ± 0,02	0,57					1		39,45%	56,56%	78,92%	-0,97%	4,91%	-0,38%
		2	0,5		32,02%	32,02%	17,98%	17,98%	1	14,05%	40,11%	68,30%	79,25%	-0,32%	16,66%	-0,04%
		4	1		24,91%	24,91%	29,22%	20,96%		8,26%						

Tableau 4.9 Tableau des moyennes des poids appris par les filtres fréquentiels sur les 10 *splits* de DTD *from scratch*. Les poids des MN et les écarts entre leur poids maximal et minimal sont en %

Architecture	No	γ (si appris)		Norm2 pour le filtre fréquentiel)				AC	Écart Max-Min				Diff écart vs SFSC		
		Mean ± Std	Median	(0 or 2)	(1 or 4)	(3 or 6)	(5)		Norm2	Norm0 #1	Norm0 #2	Norm1	Norm0 #1	Norm0 #2	Norm1
ResNet-18	1	0,62 ± 0,02	0,62					-1		8,16%	29,40%	28,66%	-6,23%	-9,16%	0,68%
	2	0,5		37,10%	37,10%	12,90%	12,90%	-1	24,20%	7,51%	23,03%	26,93%	-6,88%	-15,53%	-1,05%
	4	1		22,12%	22,12%	27,95%	27,80%		5,83%						
ResNet-34	1	0,63 ± 0,02	0,63					-1		8,89%	27,43%	27,88%	-1,68%	-8,96%	2,91%
	2	0,5		36,34%	36,34%	13,66%	13,66%	-1	22,68%	7,47%	21,96%	25,39%	-3,10%	-14,43%	0,42%
	4	1		23,37%	23,37%	26,70%	26,55%		3,33%						
ResNet-50	1	0,65 ± 0,03	0,64					-1		10,99%	27,51%	26,44%	0,47%	-3,48%	0,64%
	2	0,5		37,03%	37,03%	12,97%	12,97%	-1	24,06%	10,67%	18,63%	25,35%	0,15%	-12,36%	-0,45%
	4	1		22,88%	22,88%	27,19%	27,05%		4,31%						
BNInception	1	0,61 ± 0,02	0,61					-1		8,64%	29,90%	29,89%	-3,10%	-5,18%	1,38%
	2	0,5		36,20%	36,20%	13,80%	13,80%	-1	22,40%	8,04%	21,07%	29,07%	-3,70%	-14,01%	0,56%
	4	1		17,33%	17,33%	32,77%	32,58%		15,44%						
AlexNet+BN	1	0,35 ± 0,02	0,35					-1		3,13%	36,34%	31,69%	-26,24%	-10,56%	-3,88%
	2	0,5		26,56%	26,56%	23,44%	23,44%	-1	3,11%	7,73%	27,67%	33,89%	-21,64%	-19,23%	-1,68%
	4	1		4,80%	4,80%	45,21%	45,18%		40,41%						
VGG-M+BN	1	0,37 ± 0,02	0,38					-1		6,19%	28,39%	30,90%	-22,88%	-18,15%	-4,65%
	2	0,5		25,09%	25,09%	24,91%	24,91%	-1	0,18%	6,84%	29,16%	33,85%	-22,23%	-17,38%	-1,70%
	4	1		5,76%	5,76%	44,25%	44,23%		38,49%						
VGG-11+BN	1	0,5 ± 0,03	0,51					-1		6,96%	26,19%	24,29%	-17,42%	-21,13%	-4,32%
	2	0,5		30,06%	30,06%	19,94%	19,94%	-1	10,13%	7,94%	37,23%	25,39%	-16,44%	-10,09%	-3,22%
	4	1		10,17%	10,17%	39,86%	39,81%		29,69%						
CNN de référence	1	0,57 ± 0,04	0,57					1		26,72%	29,59%	53,17%	3,23%	-14,17%	20,44%
	2	0,5		38,30%	38,30%	11,70%	11,70%	-1	26,61%	25,20%	27,83%	36,56%	1,71%	-15,93%	3,83%
	4	1		15,59%	15,59%	34,70%	34,12%		19,12%						

Tableau 4.10 Tableau des moyennes des poids appris par les filtres fréquentiels sur les 10 *splits* de KDEF. Les poids des MN et les écarts entre leur poids maximal et minimal sont en %

Architecture	No	γ (si appris)		Norm2 pour le filtre fréquentiel)				AC	Écart Max-Min				Diff écart vs SFSC		
		Mean ± Std	Median	(0 or 2)	(1 or 4)	(3 or 6)	(5)		Norm2	Norm0 #1	Norm0 #2	Norm1	Norm0 #1	Norm0 #2	Norm1
ResNet-18	1	0,95 ± 0,03	0,95					1		3,71	20,64	18,80	-0,86	-34,43	10,12
	2	0,5		40,47	40,47	9,53	9,53	1	30,93	5,71	15,58	11,83	1,15	-39,49	3,15
	4	1		34,54	34,54	15,48	15,43		19,11						
ResNet-34	1	0,89 ± 0,03	0,89					1		1,71	20,10	18,56	-2,88	-38,78	-1,46
	2	0,5		43,60	43,60	6,40	6,40	1	37,21	5,90	25,11	17,77	1,31	-33,78	-2,25
	4	1		36,43	36,43	13,59	13,55		22,88						
ResNet-50	1	0,85 ± 0,05	0,84					1		2,43	16,01	12,53	-8,80	-40,18	-1,44
	2	0,5		40,66	40,66	9,34	9,34	1	31,32	13,26	26,80	8,73	2,03	-29,40	-5,24
	4	1		37,36	37,36	12,66	12,61		24,75						
BNInception	1	0,97 ± 0,07	0,95					1		2,50	16,48	15,03	-1,86	-41,23	9,46
	2	0,5		41,22	41,22	8,78	8,78	-1	32,44	5,92	33,66	11,40	1,57	-24,06	5,83
	4	1		38,15	38,15	11,87	11,84		26,31						
CNN de référence	1	0,66 ± 0,04	0,67					1		15,27	21,03	36,20	9,42	-4,85	16,59
	2	0,5		35,58	35,58	14,42	14,42	1	21,16	6,83	18,98	24,87	0,99	-6,89	5,27
	4	1		25,90	25,90	27,84	20,36		7,48						

Dans le cas du paramètre γ qui est appris uniquement par le filtre fréquentiel #1, les expérimentations ont surtout montré une forte sensibilité de ce paramètre par rapport au gradient, ce qui peut lui causer de fortes variations de sa valeur au cours de l'apprentissage. C'est également pour cette raison que des expérimentations avec des taux d'apprentissage réduits ou l'utilisation d'une fonction d'activation sur γ ont été effectuées avec les filtres fréquentiels (voir la section 4.1). Malgré tout, les valeurs de γ présentées dans les tableaux sont parfois très différentes entre les modèles d'une même base de données, bien qu'ils y aient des similitudes pour les modèles ayant des configurations similaires pour leurs algorithmes de gradients. Bien que ces correspondances soient normales et également applicables à d'autres paramètres comme ceux de la SFSC, il est relativement étrange de voir des variations aussi importantes sur la valeur de γ et donc, sur les effets recherchés et utilisés sur les images avec des modèles qui sont pourtant entraînés sur la même base de données. En comparaison, dans le cas de la SFSC, malgré que ces écarts de poids changent selon les modèles, la forme de sa distribution des poids et donc, les effets de son application sur les images, est généralement la même pour les différents modèles utilisés sur une même base de données. Le fait d'avoir autant de variations sur la valeur de γ entre les modèles peut indiquer un risque d'avoir des valeurs sous-optimales et donc, une limitation sur les performances maximales atteignables pour les modèles qui l'utilisent. Ce problème pourrait notamment être lié à la simplicité de l'implémentation utilisée pour l'apprentissage de γ qui laisse peut-être trop de liberté à celui-ci, le rendant alors davantage dépendant de la configuration

de l'algorithme de gradient que des besoins réels des modèles. Ceci aurait également du sens avec le fait que les trois autres configurations de filtres fréquentiels qui ont été sélectionnés ont plutôt des valeurs de γ fixes où deux d'entre elles utilisent notamment un MN appris pour ajuster la forme de leur filtre fréquentiel.

Dans le cas des MN appris des filtres fréquentiels #2 et #4, ceux-ci ont seulement deux formes de distributions pour leurs quatre poids et les effets de ces distributions sont illustrés dans la section 3.1. Basés sur les similitudes des définitions de ces poids présentées dans le tableau 2.1, les deux premiers poids visent la proportionnalité globale relative à chaque canal ($[1, 1, 1]$ et $[1, C, 1]$) tandis que les deux derniers sont plutôt selon la proportionnalité entre les canaux propres à chacune des fréquences ($[1, 1, S]$ et $[1, C, S]$). Cependant, ceci est surtout valable en raison de l'implémentation des filtres fréquentiels où ils sont initialement identiques entre les différents canaux. Ceci cause donc des définitions initialement identiques pour les deux poids de chaque groupe. Même si cela peut changer au cours de l'apprentissage, notamment avec le paramètre ω pour les filtres #1, #3 et #4, les expérimentations ont montré que celui-ci ne reçoit que de légères corrections de ses valeurs et que la forme globale du filtre reste donc visuellement similaire à celle originale. Bien que l'apprentissage de ω peut alors sembler plutôt inutile, son utilisation en tant que paramètre semble amener un effet de stabilisation pour l'apprentissage de ces filtres fréquentiels, mais avec un coût élevé sur le nombre de paramètres supplémentaires.

Pour ce qui est des SFSC qui sont utilisés avec les filtres fréquentiels #1 et #2, comme mentionné précédemment, les distributions des poids de leur MN sont généralement similaires à celles originales avec, au mieux, un ordre légèrement différent entre les deux ou trois poids maximisés. Cependant, la majorité des modèles des trois bases de données présentent des baisses plus ou moins importantes des écarts de poids par rapport à ceux originaux pour leurs deux Norm0, sauf pour les Norm0 #2 des SFSC du filtre #2 sur KDEF où ceux-ci ont tous été améliorés. Dans le cas de Norm1, il y a eu des améliorations pour environ 60 % des modèles avec le filtre #1 sur DTD *from scratch* et KDEF ainsi que ceux avec le filtre #2 sur CIFAR et KDEF. Il reste cependant que, dans un sens global, les deux premiers MN de la SFSC sont le plus souvent moins bien appris lorsque ceux-ci sont combinés à ces filtres fréquentiels. Ceci peut avoir pour

conséquence d'utiliser des SFSC sous-optimales par rapport à celles originales, mais ceci peut également être lié à d'autres facteurs. Le fait que les filtres fréquentiels et la SFSC ont des effets similaires par rapport au grisâtrement des images et la réduction des basses fréquences peut entraîner une réduction des besoins de corrections des poids de la SFSC qui n'aurait donc pas besoin d'avoir des écarts de poids aussi élevés. De l'autre côté, les filtres fréquentiels peuvent amener une convergence initiale plus rapide causant des gradients plus faibles et donc, des poids moins bien appris pour la SFSC. Dans le cas de l'AC, la tendance globale de son choix final est, le plus souvent, la même que celle de la SFSC originale et seuls deux modèles, qui sont mis en évidence dans les tableaux, ont eu des divergences à ce sujet.

En somme, ces tableaux des poids permettent de mieux comprendre les quatre filtres fréquentiels du tableau 3.1 en mettant notamment en évidence de possibles faiblesses pour le filtre #1 ayant le paramètre γ ainsi que pour les poids de la SFSC des filtres #1 et #2. Dans le cas des MN des filtres #1 et #2, les distributions des poids seraient probablement différentes si ω avait des valeurs aléatoires ou si les γ étaient différents entre les trois canaux.

La prochaine section présente les résultats obtenus par les quatre configurations de filtres fréquentiels et compare leurs performances par rapport à celles des modèles originaux et celles avec la SFSC qui ont été présentées dans la section 4.2.3.

4.3.2 Présentation des résultats obtenus par les filtres fréquentiels

Les tableaux 4.11, 4.12 et 4.13 présentent les performances obtenues par les quatre filtres fréquentiels du tableau 3.1 sur respectivement CIFAR, DTD *from scratch* et KDEF. Pour chacun des tableaux, les performances obtenues sont présentées selon les moyennes, écarts-types et médianes ainsi que les écarts de performances par rapport aux modèles originaux et ceux avec la SFSC originale qui ont été exposés dans la section 4.2.3. Ces écarts de performances sont selon les différences de moyennes et de médianes ainsi que selon les distributions globales des gains et des pertes de performances mesurés, lors de chacune des exécutions. Comme dit dans la section 4.1, tous les modèles sur CIFAR ont eu cinq exécutions sauf pour le CNN de référence qui en

a eu 10, comme ceux sur DTD et KDEF, où les statistiques sont par rapport à leurs 10 *splits*. Pour les quatre cas de CIFAR, les "+" indiquent les cas avec l'utilisation de l'augmentation de données standards et toutes les expérimentations sur DTD et KDEF ont de l'augmentation de données.

Tableau 4.11 Résultats des CNN avec les filtres fréquentiels sur CIFAR avec 5 exécutions aux poids aléatoires différents (10 exécutions pour notre CNN de référence). Les "+" indiquent l'utilisation de l'augmentation de données

Architecture	Dataset	No	Filtre fréquentiel		Différence de performance par rapport à la SFSC						Différence de performance par rapport a la référence					
					Diff acc		Nb différence positive		Nb différence négative		Diff acc		Nb différence positive		Nb différence négative	
			Mean ± Std	Median	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %
ResNet-56	CIFAR-10	1	87,8 ± 0,33	87,8	-0,19	-0,19	0	2	3	0	**1,47**	**1,13**	3	2	0	0
		2	88,03 ± 0,21	88,01	**0,04**	**0,02**	0	2	3	0	**1,7**	**1,34**	3	2	0	0
		3	88,05 ± 0,43	88,05	**0,06**	**0,06**	0	3	2	0	**1,72**	**1,38**	4	1	0	0
		4	87,92 ± 0,23	87,99	-0,07	0	0	2	3	0	**1,59**	**1,32**	4	1	0	0
	CIFAR-10+	1	93,59 ± 0,15	93,63	-0,06	-0,04	0	3	2	0	**0,27**	**0,29**	0	5	0	0
		2	93,5 ± 0,15	93,45	-0,16	-0,22	0	3	2	0	**0,17**	**0,11**	0	4	1	0
		3	93,44 ± 0,19	93,31	-0,21	-0,36	0	1	4	0	**0,11**	-0,03	0	3	2	0
		4	93,55 ± 0,14	93,55	-0,1	-0,12	0	2	3	0	**0,22**	**0,21**	0	4	1	0
	CIFAR-100	1	59,64 ± 0,66	59,46	-0,04	-0,36	1	1	2	1	**2,87**	**3,37**	4	1	0	0
		2	60 ± 0,53	59,98	**0,31**	**0,16**	1	1	3	0	**3,22**	**3,89**	5	0	0	0
		3	59,3 ± 0,52	59,36	-0,38	-0,46	0	1	4	0	**2,53**	**3,27**	5	0	0	0
		4	59,11 ± 0,65	59,3	-0,58	-0,52	0	1	3	1	**2,33**	**3,21**	4	1	0	0
	CIFAR-100+	1	70,74 ± 0,38	70,9	-0,13	-0,18	0	1	4	0	**0,32**	**0,39**	0	5	0	0
		2	70,73 ± 0,18	70,71	-0,14	-0,37	0	1	4	0	**0,3**	**0,2**	0	4	1	0
		3	70,82 ± 0,36	70,94	-0,06	-0,14	0	1	4	0	**0,39**	**0,43**	1	3	1	0
		4	70,63 ± 0,26	70,76	-0,24	-0,32	0	0	5	0	**0,2**	**0,25**	0	5	0	0
ResNet-110	CIFAR-10	1	88,28 ± 0,43	88,32	**0,1**	**0,06**	0	3	2	0	**1,12**	**1,18**	4	1	0	0
		2	87,79 ± 0,83	87,73	-0,39	-0,53	0	2	2	1	**0,63**	**0,59**	2	2	1	0
		3	87,92 ± 0,28	87,83	-0,26	-0,43	0	2	3	0	**0,76**	**0,69**	0	5	0	0
		4	88,06 ± 0,19	88,06	-0,12	-0,2	0	3	2	0	**0,9**	**0,92**	2	3	0	0
	CIFAR-10+	1	93,96 ± 0,23	94,02	-0,17	-0,1	0	0	5	0	**0,08**	**0,21**	0	3	2	0
		2	94,03 ± 0,18	93,94	-0,11	-0,18	0	2	3	0	**0,15**	**0,13**	0	4	1	0
		3	93,84 ± 0,14	93,91	-0,29	-0,21	0	0	5	0	-0,04	**0,1**	0	2	3	0
		4	94,05 ± 0,18	94,11	-0,08	-0,01	0	2	3	0	**0,17**	**0,3**	0	5	0	0
	CIFAR-100	1	60,42 ± 0,19	60,44	-0,51	-0,56	0	1	3	1	**0,74**	**0,7**	1	3	1	0
		2	60,11 ± 0,69	59,86	-0,82	-1,14	0	1	1	3	**0,43**	**0,12**	0	4	1	0
		3	59,88 ± 0,7	59,74	-1,06	-1,26	0	1	1	3	**0,2**	0	0	3	2	0
		4	60,04 ± 0,51	60,09	-0,89	-0,91	0	1	1	3	**0,36**	**0,35**	1	3	1	0
	CIFAR-100+	1	72,52 ± 0,28	72,43	-0,03	-0,2	0	2	3	0	**0,32**	**0,08**	0	3	2	0
		2	72,38 ± 0,28	72,43	-0,17	-0,2	0	2	3	0	**0,18**	**0,08**	1	1	3	0
		3	72,44 ± 0,26	72,43	-0,11	-0,2	0	2	3	0	**0,24**	**0,08**	1	3	1	0
		4	72,32 ± 0,37	72,33	-0,23	-0,3	0	2	3	0	**0,12**	-0,02	1	2	2	0
VGG-11+BN	CIFAR-10	1	88,58 ± 0,13	88,54	**0,08**	-0,05	0	2	3	0	**1,54**	**1,43**	5	0	0	0
		2	88,51 ± 0,21	88,63	0	**0,04**	0	2	3	0	**1,46**	**1,52**	4	1	0	0
		3	88,75 ± 0,29	88,8	**0,24**	**0,21**	0	4	1	0	**1,7**	**1,69**	5	0	0	0
		4	88,5 ± 0,24	88,51	-0,01	-0,08	0	2	3	0	**1,45**	**1,4**	5	0	0	0
	CIFAR-10+	1	92,99 ± 0,15	93,05	**0,17**	**0,19**	0	3	2	0	**0,58**	**0,66**	0	5	0	0
		2	93,02 ± 0,11	93,05	**0,2**	**0,19**	0	3	2	0	**0,61**	**0,66**	0	5	0	0
		3	93,01 ± 0,22	92,93	**0,18**	**0,07**	0	3	2	0	**0,6**	**0,54**	0	5	0	0
		4	92,89 ± 0,24	92,79	**0,07**	-0,07	0	2	3	0	**0,48**	**0,4**	0	4	1	0
	CIFAR-100	1	62,45 ± 0,03	62,44	**0,14**	**0,15**	0	4	1	0	**2,18**	**2,07**	5	0	0	0
		2	62,49 ± 0,42	62,41	**0,18**	**0,12**	0	4	1	0	**2,22**	**2,04**	5	0	0	0
		3	62,68 ± 0,35	62,83	**0,38**	**0,54**	0	4	1	0	**2,41**	**2,46**	5	0	0	0
		4	62,77 ± 0,58	63,12	**0,46**	**0,83**	0	4	1	0	**2,5**	**2,75**	5	0	0	0
	CIFAR-100+	1	71,9 ± 0,28	71,84	**0,13**	**0,12**	0	3	2	0	**1,33**	**1,37**	5	0	0	0
		2	71,99 ± 0,24	71,93	**0,22**	**0,21**	0	3	2	0	**1,42**	**1,46**	5	0	0	0
		3	71,77 ± 0,1	71,75	0	**0,03**	0	3	2	0	**1,2**	**1,28**	4	1	0	0
		4	71,73 ± 0,22	71,73	-0,04	0,01	0	2	3	0	**1,16**	**1,26**	3	2	0	0
VGG-16+BN	CIFAR-10	1	89,78 ± 0,16	89,82	-0,14	-0,21	0	2	3	0	**0,67**	**0,64**	0	5	0	0
		2	89,81 ± 0,16	89,76	-0,11	-0,27	0	1	4	0	**0,7**	**0,58**	0	5	0	0
		3	89,74 ± 0,23	89,76	-0,19	-0,27	0	1	4	0	**0,63**	**0,58**	1	4	0	0
		4	89,72 ± 0,19	89,68	-0,2	-0,35	0	1	4	0	**0,62**	**0,5**	0	5	0	0
	CIFAR-10+	1	94,08 ± 0,07	94,09	-0,03	-0,02	0	2	3	0	**0,17**	**0,19**	0	5	0	0
		2	94,03 ± 0,09	94,03	-0,08	-0,08	0	1	4	0	**0,12**	**0,13**	0	4	1	0
		3	94,03 ± 0,09	93,98	-0,08	-0,13	0	2	3	0	**0,12**	**0,08**	0	4	1	0
		4	93,97 ± 0,2	93,93	-0,14	-0,18	0	1	4	0	**0,07**	**0,03**	0	2	3	0
	CIFAR-100	1	64,43 ± 0,33	64,46	-0,27	-0,22	0	1	4	0	**0,75**	**0,67**	2	2	1	0
		2	64,39 ± 0,19	64,37	-0,31	-0,31	0	2	2	1	**0,71**	**0,58**	1	4	0	0
		3	64,39 ± 0,21	64,36	-0,31	-0,32	0	1	3	1	**0,7**	**0,57**	1	4	0	0
		4	64,71 ± 0,17	64,79	**0,01**	**0,11**	0	4	1	0	**1,03**	**1**	3	2	0	0
	CIFAR-100+	1	73,92 ± 0,21	73,86	-0,22	-0,22	0	1	4	0	**0,05**	-0,07	0	2	3	0
		2	73,98 ± 0,31	74,01	-0,16	-0,07	0	1	4	0	**0,12**	**0,08**	0	4	1	0
		3	74,07 ± 0,13	74,07	-0,07	-0,01	0	2	3	0	**0,21**	**0,14**	0	3	2	0
		4	74,02 ± 0,22	73,94	-0,12	-0,14	0	1	4	0	**0,15**	**0,01**	0	3	2	0

Tableau 4.11 Résultats des CNN avec les filtres fréquentiels sur CIFAR (suite)

Architecture	Dataset	No	Filtre fréquentiel		Différence de performance par rapport à la SFSC						Différence de performance par rapport a la référence					
					Diff acc		Nb différence positive		Nb différence négative		Diff acc		Nb différence positive		Nb différence négative	
			Mean ± Std	Median	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %
DenseNet {L = 100, K = 12}	CIFAR-10	1	93,56 ± 0,23	93,6	-0,15	-0,04	0	2	3	0	-0,23	-0,27	0	2	3	0
		2	93,7 ± 0,21	93,78	0	**0,14**	0	2	3	0	-0,08	-0,09	0	3	2	0
		3	93,62 ± 0,14	93,68	-0,08	**0,04**	0	2	3	0	-0,16	-0,19	0	2	3	0
		4	93,57 ± 0,11	93,6	-0,14	-0,04	0	1	4	0	-0,22	-0,27	0	1	4	0
	CIFAR-10+	1	95 ± 0,18	94,96	-0,18	-0,16	0	2	3	0	-0,14	-0,25	0	1	4	0
		2	94,96 ± 0,12	94,95	-0,21	-0,17	0	2	3	0	-0,17	-0,26	0	1	4	0
		3	95,08 ± 0,17	95,08	-0,1	-0,04	0	1	4	0	-0,06	-0,13	0	1	4	0
		4	95,03 ± 0,16	95,01	-0,15	-0,11	0	2	3	0	-0,11	-0,2	0	1	4	0
	CIFAR-100	1	73,5 ± 0,52	73,46	-0,84	-0,92	0	0	4	1	-1,52	-1,36	0	0	0	5
		2	73,86 ± 0,17	73,83	-0,49	-0,55	0	0	5	0	-1,16	-0,99	0	0	1	4
		3	73,76 ± 0,46	73,91	-0,59	-0,47	0	0	4	1	-1,26	-0,91	0	0	1	4
		4	73,81 ± 0,29	73,76	-0,54	-0,62	0	0	5	0	-1,21	-1,06	0	0	1	4
	CIFAR-100+	1	76,64 ± 0,28	76,56	**0,21**	**0,17**	0	3	2	0	-0,18	-0,59	0	1	4	0
		2	76,65 ± 0,25	76,61	**0,23**	**0,22**	0	4	1	0	-0,16	-0,54	0	2	3	0
		3	76,74 ± 0,38	76,78	**0,31**	**0,39**	0	4	1	0	-0,08	-0,37	0	2	3	0
		4	76,91 ± 0,31	76,79	**0,49**	**0,4**	0	5	0	0	**0,1**	-0,36	0	3	2	0
WRN 16-8	CIFAR-10	1	90,88 ± 0,57	90,76	**0,2**	**0,35**	0	3	2	0	-0,12	-0,16	1	0	4	0
		2	90,72 ± 0,22	90,72	**0,04**	**0,31**	0	3	2	0	-0,27	-0,2	0	1	4	0
		3	91,42 ± 0,42	91,48	**0,74**	**1,07**	2	2	1	0	**0,42**	**0,56**	0	4	1	0
		4	91,05 ± 0,51	90,79	**0,37**	**0,38**	0	5	0	0	**0,05**	-0,13	0	2	3	0
	CIFAR-10+	1	95,4 ± 0,06	95,43	**0,03**	**0,05**	0	4	1	0	-0,09	-0,07	0	1	4	0
		2	95,19 ± 0,18	95,14	-0,19	-0,24	0	1	4	0	-0,3	-0,36	0	0	5	0
		3	95,3 ± 0,17	95,3	-0,07	-0,08	0	2	3	0	-0,19	-0,2	0	0	5	0
		4	95,36 ± 0,08	95,39	-0,01	0,01	0	2	3	0	-0,13	-0,11	0	0	5	0
	CIFAR-100	1	65,78 ± 0,49	65,57	-0,46	-0,86	1	0	3	1	-1,18	-1,21	0	0	2	3
		2	65,63 ± 0,7	65,71	-0,61	-0,72	0	2	1	2	-1,33	-1,07	0	0	1	4
		3	67,23 ± 0,75	67,22	**0,99**	**0,79**	2	3	0	0	**0,27**	**0,44**	0	4	1	0
		4	67,27 ± 0,49	67,35	**1,03**	**0,92**	2	3	0	0	**0,32**	**0,57**	0	3	2	0
	CIFAR-100+	1	78,63 ± 0,27	78,78	**0,09**	**0,25**	0	4	1	0	-0,66	-0,58	0	0	5	0
		2	78,29 ± 0,35	78,32	-0,25	-0,21	0	2	3	0	-1	-1,04	0	0	2	3
		3	78,74 ± 0,21	78,77	**0,2**	**0,24**	0	5	0	0	-0,55	-0,59	0	0	5	0
		4	78,47 ± 0,21	78,52	-0,07	-0,01	0	1	4	0	-0,82	-0,84	0	0	5	0
CNN de référence	CIFAR-10	1	83,45 ± 0,26	83,37	-0,04	-0,1	0	4	6	0	**0,31**	**0,28**	0	7	3	0
		2	83,54 ± 0,25	83,53	**0,05**	**0,06**	0	6	4	0	**0,41**	**0,44**	0	8	2	0
		3	83,54 ± 0,28	83,45	**0,05**	-0,02	0	7	3	0	**0,41**	**0,35**	1	7	2	0
		4	83,57 ± 0,08	83,57	**0,08**	**0,1**	0	5	5	0	**0,43**	**0,47**	0	9	1	0
	CIFAR-10+	1	85,12 ± 0,19	85,19	**0,19**	**0,31**	0	6	4	0	**0,35**	**0,4**	0	9	1	0
		2	85,1 ± 0,28	85,04	**0,17**	**0,16**	0	6	4	0	**0,33**	**0,24**	2	5	3	0
		3	85,13 ± 0,16	85,13	**0,21**	**0,25**	0	7	3	0	**0,36**	**0,33**	0	10	0	0
		4	85,17 ± 0,22	85,26	**0,25**	**0,38**	0	8	2	0	**0,4**	**0,46**	0	10	0	0
	CIFAR-100	1	46,14 ± 0,3	46,15	-0,22	-0,31	0	4	6	0	**0,38**	**0,4**	0	9	1	0
		2	46,13 ± 0,39	45,99	-0,23	-0,47	0	3	7	0	**0,37**	**0,24**	1	6	3	0
		3	45,92 ± 0,43	45,88	-0,45	-0,57	0	1	6	3	**0,15**	**0,14**	1	4	5	0
		4	46,12 ± 0,36	46,01	-0,24	-0,44	0	2	8	0	**0,36**	**0,27**	1	7	2	0
	CIFAR-100+	1	46,77 ± 0,43	46,63	**0,16**	**0,03**	0	6	4	0	**0,7**	**0,62**	3	6	1	0
		2	46,34 ± 0,32	46,18	-0,26	-0,43	0	2	7	1	**0,27**	**0,16**	0	8	2	0
		3	46,75 ± 0,51	46,76	**0,14**	**0,16**	1	3	6	0	**0,68**	**0,74**	4	4	2	0
		4	46,65 ± 0,53	46,63	**0,05**	**0,03**	1	5	3	1	**0,59**	**0,62**	3	5	2	0

Pour les expérimentations sur les quatre cas de CIFAR qui sont présentées dans le tableau 4.11, les résultats obtenus indiquent que les filtres fréquentiels ont généralement très bien réussi par rapport aux modèles originaux pour le ResNet-56, ResNet-110, VGG-11+BN, VGG-16+BN et le CNN de référence. Cependant, seuls les filtres fréquentiels utilisés avec le VGG-11+BN et le CNN de référence ont eu la majorité ou toutes les exécutions sur CIFAR, avec de meilleures performances globales par rapport à la SFSC. Dans le cas du DenseNet et du WRN 16-8 où la SFSC n'a pas très bien réussi par rapport aux modèles originaux, les filtres fréquentiels ont dépassé la SFSC sur certains cas de CIFAR. Ceci inclut ceux de CIFAR-100+ avec le DenseNet et ceux de CIFAR-10 avec le WRN 16-8 où les quatre filtres fréquentiels ont eu de meilleures performances globales. Dans le cas du WRN 16-8, les filtres #1, #3 et #4 ont également mieux réussi que la SFSC sur trois des quatre cas de CIFAR. Néanmoins, ces améliorations par rapport

Tableau 4.12 Résultats des CNN avec les filtres fréquentiels sur DTD *from scratch* avec les statistiques reliées à leurs exécutions sur les 10 *splits*

Architecture	No	Filtre fréquentiel		Différence de performance par rapport à la SFSC						Différence de performance par rapport a la référence					
				Diff acc		Nb différence positive		Nb différence négative		Diff acc		Nb différence positive		Nb différence négative	
		Mean ± Std	Median	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %
ResNet-18	1	44,4 ± 1,91	44,52	-0,85	-0,78	2	2	1	5	**0,94**	**1,38**	6	1	0	3
	2	44,64 ± 1,95	44,47	-0,62	-0,83	1	4	1	4	**1,19**	**1,33**	3	4	3	0
	3	43,81 ± 1,54	43,73	-1,45	-1,57	2	1	1	6	**0,39**	**0,59**	4	2	2	2
	4	44,33 ± 2,09	44,21	-0,93	-1,09	2	2	1	5	**0,91**	**1,07**	6	0	2	2
ResNet-34	1	45,4 ± 2,72	45,13	-2,41	-2,88	2	0	1	7	**0,41**	0	5	1	1	3
	2	45,61 ± 1,31	45,96	-2,2	-2,05	2	0	1	7	**0,61**	**0,83**	6	0	0	4
	3	44,04 ± 1,39	44,13	-3,77	-3,88	2	0	0	8	-0,95	-1,01	3	1	1	5
	4	45,07 ± 1,82	45,08	-2,74	-2,93	2	1	0	7	**0,08**	-0,05	5	0	0	5
ResNet-50	1	46,27 ± 1,74	45,67	-2,5	-2,95	0	1	1	8	**0,46**	-0,64	3	3	3	1
	2	47,21 ± 1,46	47,5	-1,56	-1,12	1	1	2	6	**1,3**	**1,2**	5	3	2	0
	3	45,03 ± 1,67	44,31	-3,74	-4,31	0	0	1	9	-0,78	-2	1	3	2	4
	4	46,36 ± 1,75	46,28	-2,41	-2,34	1	0	0	9	**0,55**	-0,03	4	2	2	2
BNInception	1	40,38 ± 3,09	39,5	-9,55	-10,11	0	0	0	10	-8,46	-9,41	0	0	0	10
	2	46,38 ± 3,35	46,38	-3,55	-3,22	2	0	0	8	-2,47	-2,53	1	2	1	6
	3	46,8 ± 0,82	47	-3,13	-2,61	0	0	1	9	-2,04	-1,91	0	1	0	9
	4	47,03 ± 1,8	47,29	-2,9	-2,31	0	0	1	9	-1,81	-1,62	0	1	3	6
AlexNet+BN	1	42,83 ± 1,56	42,66	-0,66	-0,75	0	3	3	4	**1,37**	**1,44**	6	3	1	0
	2	42,87 ± 1,36	43,17	-0,62	-0,24	0	2	5	3	**1,41**	**1,95**	5	4	1	0
	3	38,62 ± 1,4	38,94	-4,87	-4,47	0	0	0	10	-2,84	-2,29	0	0	2	8
	4	41,87 ± 1,49	41,97	-1,63	-1,44	0	0	1	9	**0,4**	**0,75**	3	5	0	2
VGG-M+BN	1	44,63 ± 1,19	44,28	-0,57	-0,85	1	2	3	4	**1,49**	**1,38**	7	3	0	0
	2	44,3 ± 1,42	43,65	-0,9	-1,49	0	3	3	4	**1,16**	**0,74**	6	3	1	0
	3	41,12 ± 1,43	41,36	-4,08	-3,78	0	0	0	10	-2,01	-1,54	0	0	2	8
	4	44,18 ± 1,67	43,8	-1,02	-1,34	1	1	2	6	**1,05**	**0,9**	4	5	1	0
VGG-11+BN	1	46 ± 1,52	45,61	**1,74**	**1,57**	7	2	1	0	**5,1**	**5,14**	10	0	0	0
	2	45,81 ± 1,64	45,46	**1,56**	**1,41**	6	3	1	0	**4,91**	**4,98**	10	0	0	0
	3	44,56 ± 1,42	44,31	**0,31**	**0,27**	1	7	1	1	**3,66**	**3,83**	10	0	0	0
	4	45,46 ± 1,96	44,87	**1,2**	**0,83**	6	2	2	0	**4,55**	**4,39**	10	0	0	0
CNN de référence	1	38,57 ± 1,23	38,78	-0,99	-0,72	0	2	3	5	-0,61	-0,34	1	4	0	5
	2	39,13 ± 1,48	38,99	-0,43	-0,51	2	3	2	3	-0,05	-0,13	3	3	0	4
	3	38,36 ± 1,15	38,49	-1,21	-1,01	0	2	2	6	-0,82	-0,64	1	2	4	3
	4	38,56 ± 1,26	38,62	-1,01	-0,88	1	2	0	7	-0,63	-0,51	1	2	3	4

Tableau 4.13 Résultats des CNN avec les filtres fréquentiels sur KDEF avec les statistiques reliées à leurs exécutions sur les 10 *splits*

Architecture	No	Filtre fréquentiel		Différence de performance par rapport à la SFSC						Différence de performance par rapport a la référence					
				Diff acc		Nb différence positive		Nb différence négative		Diff acc		Nb différence positive		Nb différence négative	
		Mean ± Std	Median	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %
ResNet-18	1	92,02 ± 1,11	92,25	**0,02**	**0,2**	1	4	5	0	**0,62**	**0,52**	4	4	1	1
	2	91,9 ± 1,13	92,14	-0,1	**0,1**	0	6	4	0	**0,5**	**0,41**	1	7	1	1
	3	90,5 ± 1,37	90,51	-1,5	-1,53	0	0	1	9	-0,9	-1,22	0	1	5	4
	4	91,23 ± 1,25	91,23	-0,77	-0,82	0	2	4	4	-0,16	-0,51	0	5	3	2
ResNet-34	1	91,62 ± 1,47	92,25	-0,52	-0,25	1	2	5	2	-0,17	**0,46**	2	4	2	2
	2	92,21 ± 1,38	92,66	**0,07**	**0,16**	1	6	3	0	**0,42**	**0,87**	2	6	2	0
	3	90,34 ± 1,39	90,31	-1,81	-2,19	0	0	3	7	-1,46	-1,48	0	1	3	6
	4	90,88 ± 1,3	91,23	-1,27	-1,28	0	1	1	8	-0,92	-0,56	0	2	3	5
ResNet-50	1	90,27 ± 1,86	90,77	-1,04	-0,46	0	1	4	5	-0,23	**0,11**	1	3	4	2
	2	91,24 ± 1,33	91,63	-0,07	**0,41**	1	5	2	2	**0,74**	**0,97**	4	4	1	1
	3	89,1 ± 1,78	88,98	-2,2	-2,24	0	0	1	9	-1,4	-1,68	0	2	0	8
	4	90,08 ± 1,79	90,31	-1,22	-0,91	0	0	5	5	-0,42	-0,35	0	5	2	3
BNInception	1	89,35 ± 1,97	89,7	-3,07	-2,66	0	0	0	10	-2,33	-2,2	0	0	1	9
	2	91,83 ± 1,31	91,94	-0,59	-0,41	0	3	3	4	**0,15**	**0,05**	1	3	6	0
	3	89,57 ± 1,1	89,6	-2,85	-2,76	0	0	0	10	-2,1	-2,3	0	1	0	9
	4	90,11 ± 0,88	89,8	-2,31	-2,55	0	0	0	10	-1,56	-2,09	0	1	2	7
CNN de référence	1	84,77 ± 1,7	84,64	-0,25	**0,2**	2	3	2	3	**1,26**	**0,81**	5	3	1	1
	2	85,25 ± 1,59	84,85	**0,23**	**0,41**	3	2	3	2	**1,74**	**1,03**	5	3	2	0
	3	84,49 ± 1,52	83,88	-0,53	-0,56	2	1	4	3	**0,97**	**0,05**	4	2	3	1
	4	84,81 ± 1,6	84,95	-0,21	**0,51**	3	3	1	3	**1,29**	**1,13**	4	3	2	1

à la SFSC sont généralement insuffisantes pour dépasser les modèles originaux où seuls les filtres #3 et #4 du WRN 16-8 le dépassent sur CIFAR-10 et CIFAR-100. Selon un plan global par rapport à CIFAR, les filtres fréquentiels ont, dans la majorité des cas, dépassé les performances de la SFSC dans le cas du VGG-11+BN (pour les quatre filtres), du CNN de référence (filtres

#3 et #4) ainsi que du WRN 16-8 (filtres #1, #3 et #4). Cependant, la SFSC reste globalement meilleure pour l'ensemble des autres modèles et ce sont surtout les cas du VGG-11+BN et du CNN de référence qui ont pu dépasser à la fois la SFSC et les modèles originaux, sur la majorité des quatre cas de CIFAR.

Dans le cas des expérimentations sur DTD *from scratch* qui sont présentées dans le tableau 4.12, les filtres fréquentiels ont eu de meilleures performances que la majorité des modèles originaux, à l'exception du BNInception et du CNN de référence. Dans le cas du BNInception, ces pertes de performances relativement élevées ne semblent pas liées aux poids appris puisque leurs statistiques sont similaires aux autres modèles (tableau 4.9). Cependant, elles pourraient être liées à l'architecture particulière de ce modèle où sa forme pré-entraînée avait nécessité une configuration particulière pour les gradients de la SFSC (voir la section 4.1). Dans le cas du CNN de référence, celui-ci est plus difficile à dépasser notamment en raison de son faible niveau de surapprentissage et du fait que la SFSC a apporté des gains de performances légers par rapport à ceux obtenus avec les autres modèles. De l'autre côté, de manière similaire aux résultats obtenus sur CIFAR, le VGG-11+BN est le seul modèle où les filtres fréquentiels lui ont permis d'atteindre des performances plus élevées que la SFSC et le modèle original. Pour tous les autres, la SFSC a globalement permis de bien meilleures améliorations de performances qui dépassent largement les meilleurs résultats obtenus par les filtres fréquentiels. Par conséquent, ceux-ci semblent avoir permis de combler certaines lacunes des modèles de DTD *from scratch*, mais leurs effets de réduction des magnitudes semblent être trop élevés par rapport à ceux de la SFSC originale. Il est également possible que cela soit lié à la forme de ces filtres pouvant être sous-optimaux pour l'analyse des textures ou qui pourrait manquer de flexibilité. À ce sujet, le filtre #3, soit celui qui garde le mieux sa forme initiale, a globalement le moins bien fonctionner sur DTD *from scratch* alors qu'il a obtenu les meilleures performances dans l'ensemble sur CIFAR. Pour DTD *from scratch*, ce sont plutôt dans l'ordre les filtres #2 et #1 qui ont globalement les mieux réussi par rapport aux modèles originaux, bien qu'ils aient généralement nui aux performances de leur SFSC.

Dans le cas des expérimentations sur KDEF qui sont présentées dans le tableau 4.13, les filtres fréquentiels #1 et #2 ont permis d'obtenir de meilleures performances que la plupart des modèles originaux, alors que les filtres #3 et #4 n'ont pu améliorer que le CNN de référence. Cependant, seul le filtre #2 a amélioré les performances de sa SFSC sur tous les modèles à l'exception du BNInception qui fait figure à part comme sur DTD *from scratch*, tout en ayant également le même constat sur les poids appris par les filtres (tableau 4.10). Les bonnes performances du filtre #2 peuvent être notamment liées à son utilisation combinée avec la SFSC qui est absente des filtres #3 et #4 qui n'ont pas très bien fonctionné. Par ailleurs, le fait d'avoir également utilisé un $\gamma = 0,5$ (au lieu de 1 pour les filtres #3 et #4) ainsi qu'un MN appris, a permis au filtre #2 de mieux conserver les basses fréquences. À l'opposé, le fait que le filtre #1 a appris des γ proches de 1, pour la plupart des modèles, lui a probablement causé certaines limitations sur ses performances maximales. Il semble donc que les améliorations des performances plus élevées du filtre #2 soient surtout liées à sa meilleure conservation des basses fréquences où son atténuation a été globalement bénéfique pour sa SFSC sur presque tous les modèles.

Dans l'ensemble, les filtres fréquentiels du tableau 3.1 ont le plus souvent permis d'obtenir de meilleures performances que les modèles originaux en incluant mêmes certains cas du WRN 16-8 et du DenseNet sur CIFAR où la SFSC n'a pas réussi à les améliorer. Néanmoins, d'un point de vue global, ces filtres n'ont pas réussi à dépasser les performances de la SFSC sur la plupart des modèles. Ceux-ci entraînent également des coûts supplémentaires en calcul et en mémoire relativement élevés, surtout pour les filtres #1, #3 et #4 qui apprennent ω. On note également que ces filtres fréquentiels ont obtenu de très mauvaises performances avec les modèles pré-entraînés utilisés sur DTD, probablement en raison de modifications trop importantes sur les images par rapport à celles auxquelles étaient habitués les modèles (voire la section 4.1). Malgré tout, le fait que certains de ces filtres aient parfois permis de dépasser les performances de la SFSC pourrait révéler certains indicateurs sur la manière de l'améliorer, en y incluant une partie des effets de ces filtres sur la forme de la magnitude finale des images obtenues. Ceci pourrait se faire avec, par exemple, des ajustements sur la manière de fonctionner de la SFSC ou en l'intégrant avec d'autres formes de filtres qui pourraient effectuer des ajustements plus précis ou localisés dans

le domaine fréquentiel, par rapport à ceux effectués par les filtres fréquentiels du tableau 3.1. Pour ceux-ci, le filtre #3 est celui qui a globalement le mieux fonctionné sur CIFAR, mais qui a également été le plus mauvais sur DTD *from scratch* et KDEF où ce sont, dans l'ordre, les filtres #2 et #1 qui ont donné les meilleurs résultats. Cependant, le filtre #2 est également celui qui a le moins bien fonctionné sur CIFAR, alors que le filtre #1 a été le deuxième meilleur filtre sur CIFAR. Cependant, celui-ci se démarque peu des autres et se fait le plus souvent devancer par la SFSC originale. L'utilisation seule de la SFSC semble donc, au final, rester la meilleure approche parmi celles expérimentées au cours de cette recherche.

Afin d'aller un peu plus loin, la prochaine section présente des expérimentations supplémentaires qui ont été réalisées afin de répondre à certaines questions qui pourrait être soulevée quant aux caractéristiques de la SFSC et des filtres fréquentiels qui ont été utilisés au cours de cette recherche.

4.4 Expérimentations supplémentaires sur la SFSC et les filtres fréquentiels

À la suite de l'ensemble des résultats présentés jusqu'à maintenant, divers aspects relatifs aux calculs de la SFSC ainsi qu'aux filtres fréquentiels peuvent être remis en question ou être intéressants d'approfondir pour mieux comprendre ces nouvelles méthodes. À cette fin, cette section a pour objectif de présenter des éléments de réponses pour tenter de répondre à certains questionnements au travers des interrogations suivantes :

Dans le cas de la SFSC :

- basé sur les tableaux des poids appris présentés dans la section 4.2.1, est-il vraiment utile d'apprendre Norm1 et l'AC ? En effet, Norm1 vise le plus souvent à maximiser le même poids (son poids #0) alors que l'AC vise globalement le même choix final pour la plupart des modèles entraînés sur une même base de données. De plus, bien que les deux Norm0 visent essentiellement des combinaisons de poids, qu'en est-il s'il n'y avait pas d'apprentissage et que tous les poids de la SFSC étaient fixes ?
- bien que des arguments aient été présentés dans la section 4.2.2 en faveur de l'importance des chemins liés aux deux Norm0, est-ce qu'utiliser uniquement Norm0 #2 ($\bar{M}$) pourrait être suffisant, voire meilleurs, dans certains cas ? En effet, éliminer la partie $(1 - \widehat{M})$ de l'équation 2.5 de la SFSC a pour effet d'effectuer uniquement une normalisation fréquentielle des images d'entrées (suivie d'un BN sans poids dans le domaine spatial), n'amenant ainsi aucune perte d'information contrairement à l'équation 2.5.

Pour les filtres fréquentiels en loi de puissance de forme ω^γ :

- en sachant que les filtres fréquentiels sélectionnés sont à la base des filtres fixes auxquels certains apprentissages ont été ajoutés pour leur permettre de mieux s'adapter aux modèles, comment se comparent ceux-ci par rapport à leurs formes fixes ? Ceci est d'autant plus intriguant quand on sait que le filtre #3 garde une forme globale quasi identique à celle de son initialisation en raison de son MN fixé sur le poids #1 (*IN max*) (tableau 2.1) et le fait que son paramètre ω semble visuellement identique à celui original. Or, ce filtre est également celui qui a globalement eu les meilleures performances sur CIFAR.

Afin de répondre à ces questionnements, des expérimentations ont été effectuées avec le CNN de référence sur les quatre cas de CIFAR selon les mêmes 10 exécutions et configurations d'apprentissage que celles utilisées jusqu'à maintenant. Ces expérimentations se sont limitées à ce modèle sur CIFAR notamment en raison du grand nombre de configurations à évaluer et du fait que ces configurations ne présentent généralement pas d'amélioration majeure des performances par rapport à celles qui ont été présentées précédemment. Malgré tout, ces expérimentations permettent d'obtenir certains éléments de réponses à ces interrogations. De futures recherches pourront se baser sur ces résultats expérimentaux pour aller plus loin avec les différentes façons de faire présentées au cours de cette recherche ou se comparer par rapport à celles-ci. Certaines pistes pour de futures recherches sont également proposées au travers de cette section.

4.4.1 La SFSC : Apprentissage partiel et formes sans apprentissage

La première expérimentation supplémentaire vise à savoir s'il est vrai de croire que toutes les couches de la SFSC devraient être apprises ou si des formes entièrement fixes ou partiellement apprises peuvent être suffisantes, voire même légèrement meilleures, dans certains cas. Les formes partiellement apprises consistent à fixer l'AC ou Norm1 selon un seul de leur poids, étant donné que ceux-ci visent habituellement les mêmes poids pour l'ensemble des modèles entraînés sur une même base de données (section 4.2.1). Étant donné qu'à l'opposé les deux Norm0 maximisent toujours plusieurs poids à la fois, ceux-ci sont maintenus avec des apprentissages pour les formes partiellement apprises qui ont été expérimentées. Dans le cas des formes entièrement fixes, des expérimentations ont d'abord été effectuées sur les 245 configurations valides et possibles sur CIFAR-10 sur 100 époques avec le CNN de référence. Les quatre meilleures configurations ont par la suite été utilisées pour effectuer les expérimentations complètes habituelles des cas de CIFAR.

Les résultats obtenus pour ces expérimentations sont présentés dans le tableau 4.14 selon le même format de présentation que dans la section 4.3.2 avec des comparaisons par rapport à la SFSC entièrement apprise ainsi que par rapport au modèle original. Pour les colonnes des

configurations des couches de la SFSC, les *L* indiquent que ces couches sont avec apprentissage tandis que les autres indiquent le numéro de leur poids unique et fixe utilisé.

Tableau 4.14 Résultats du CNN de référence avec les différentes formes de la SFSC partiellement apprises et sans apprentissage (poids fixes) sur CIFAR avec 10 exécutions par configurations. Les "+" indiquent les cas avec de l'augmentation de données

Dataset	Configuration				Résultats		Différence de performance par rapport à la SFSC						Différence de performance par rapport a la référence					
							Diff acc		Nb différence positive		Nb différence négative		Diff acc		Nb différence positive		Nb différence négative	
	AC	Norm0 #1	Norm0 #2	Norm1	Mean ± Std	Median	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %
CIFAR-10	L	L	L	#0	83,6 ± 0,35	83,52	**0,11**	**0,06**	0	5	5	0	**0,46**	**0,43**	1	8	1	0
	#1	L	L	#0	83,68 ± 0,17	83,66	**0,19**	**0,19**	0	6	4	0	**0,55**	**0,56**	0	9	1	0
	#1	L	L	L	83,46 ± 0,17	83,48	-0,02	**0,01**	0	5	5	0	**0,33**	**0,38**	0	9	1	0
	#-1	L	L	_	83,6 ± 0,21	83,64	**0,11**	**0,18**	0	8	2	0	**0,47**	**0,55**	2	6	2	0
CIFAR-10+	L	L	L	#0	84,85 ± 0,12	84,88	-0,07	0	0	5	5	0	**0,08**	**0,08**	0	7	3	0
	#1	L	L	#0	85,03 ± 0,26	85,04	**0,11**	**0,17**	0	5	5	0	**0,26**	**0,25**	0	9	1	0
	#1	L	L	L	84,9 ± 0,24	84,89	-0,03	**0,01**	0	3	7	0	**0,13**	**0,09**	1	6	3	0
	#-1	L	L	_	84,85 ± 0,33	84,91	-0,08	**0,03**	0	3	7	0	**0,08**	**0,12**	0	6	4	0
CIFAR-100	L	L	L	#0	46,05 ± 0,29	46,11	-0,31	-0,35	0	1	8	1	**0,29**	**0,37**	0	8	2	0
	#1	L	L	#0	46,27 ± 0,49	46,18	-0,09	-0,28	1	3	6	0	**0,51**	**0,43**	2	6	2	0
	#1	L	L	L	46,27 ± 0,28	46,22	-0,09	-0,23	0	3	7	0	**0,51**	**0,48**	2	6	2	0
	#-1	L	L	_	46,28 ± 0,33	46,2	-0,09	-0,26	0	5	4	1	**0,51**	**0,45**	2	8	0	0
CIFAR-100+	L	L	L	#0	46,14 ± 0,29	46,13	-0,46	-0,48	0	1	7	2	**0,07**	**0,11**	2	5	3	0
	#1	L	L	#0	46,76 ± 0,29	46,86	**0,15**	**0,26**	0	7	3	0	**0,69**	**0,85**	4	6	0	0
	#1	L	L	L	46,67 ± 0,33	46,74	**0,07**	**0,14**	0	6	4	0	**0,6**	**0,73**	5	4	1	0
	#-1	L	L	_	46,43 ± 0,5	46,54	-0,17	-0,07	0	3	6	1	**0,36**	**0,52**	3	4	3	0
CIFAR-10	#-1	#4	#5	_	83,18 ± 0,22	83,18	-0,31	-0,29	0	3	6	1	**0,04**	**0,08**	0	6	4	0
	#1	#0	#5	#0	83,15 ± 0,18	83,22	-0,34	-0,24	0	2	8	0	**0,01**	**0,13**	0	4	6	0
	#1	#1	#0	#0	83,09 ± 0,3	83,12	-0,4	-0,34	0	1	9	0	-0,04	**0,03**	0	5	5	0
	#1	#4	#5	#0	83,22 ± 0,23	83,2	-0,27	-0,26	0	1	9	0	**0,08**	**0,11**	0	6	4	0
CIFAR-10+	#-1	#4	#5	_	84,26 ± 0,3	84,27	-0,67	-0,61	0	1	7	2	-0,51	-0,53	0	1	8	1
	#1	#0	#5	#0	84,41 ± 0,26	84,4	-0,52	-0,48	0	1	7	2	-0,36	-0,4	0	1	8	1
	#1	#1	#0	#0	84,72 ± 0,27	84,73	-0,2	-0,14	0	3	7	0	-0,05	-0,06	0	4	6	0
	#1	#4	#5	#0	84,53 ± 0,24	84,5	-0,4	-0,38	0	1	9	0	-0,24	-0,3	0	3	7	0
CIFAR-100	#-1	#4	#5	_	45,34 ± 0,51	45,42	-1,02	-1,04	0	0	5	5	-0,42	-0,32	0	3	7	0
	#1	#0	#5	#0	45,45 ± 0,5	45,48	-0,92	-0,98	0	0	6	4	-0,32	-0,27	0	3	5	2
	#1	#1	#0	#0	45,42 ± 0,27	45,34	-0,94	-1,12	0	0	5	5	-0,34	-0,41	0	2	8	0
	#1	#4	#5	#0	45,82 ± 0,49	45,83	-0,55	-0,63	0	2	5	3	**0,05**	**0,09**	1	6	1	2
CIFAR-100+	#-1	#4	#5	_	45,63 ± 0,42	45,76	-0,97	-0,84	0	0	7	3	-0,44	-0,25	0	1	8	1
	#1	#0	#5	#0	45,47 ± 0,31	45,47	-1,14	-1,13	0	0	3	7	-0,6	-0,54	0	0	8	2
	#1	#1	#0	#0	45,75 ± 0,39	45,78	-0,85	-0,82	0	1	4	5	-0,32	-0,23	0	3	6	1
	#1	#4	#5	#0	45,74 ± 0,32	45,83	-0,86	-0,77	0	0	7	3	-0,33	-0,18	0	2	7	1

Dans le cas de formes entièrement fixes sélectionnées, celles-ci ont légèrement amélioré les performances du modèle original sur CIFAR-10, ainsi que sur CIFAR-100 dans le cas de la quatrième configuration fixe. Cependant, elles ont surtout occasionné de légères baisses de performances sur les autres cas plus difficiles de CIFAR. Bien que ces formes fixes aient utilisé des poids pour leurs MN qui correspondent plutôt bien aux poids maximisés par la SFSC entièrement apprise (section 4.2.1), le fait de n'utiliser aucune combinaison de poids limite le potentiel maximal de la SFSC. Malgré tout, ces résultats indiquent que des formes fixes de la SFSC peuvent améliorer les modèles dans les cas simples comme sur CIFAR-10. Les bonnes performances de la SFSC ne viennent donc pas uniquement des apprentissages effectués par ses couches, mais également du calcul effectué par l'équation 2.5.

Dans le cas des formes partiellement apprises, celles-ci ont globalement amélioré le modèle original, mais seuls les deux cas ayant utilisé l'AC = 1 ont pu également légèrement améliorer la SFSC, à l'exception des cas sur CIFAR-100. Le fait que la meilleure configuration partiellement

apprise soit celle ayant fixé l'AC et Norm1 est cependant relativement peu surprenant par rapport à ce qui a été présenté au sujet des modèles sur CIFAR avec la SFSC dans la section 4.2.2. En effet, tel qu'il avait été présenté, les modèles sur CIFAR et DTD *from scratch* ont des distributions des poids pour leurs deux Norm0 qui rendent le rôle de l'AC et de Norm1 relativement faible sur le rendu final. Par conséquent, le fait d'avoir fixé l'AC et optionnellement Norm1 a permis de simplifier la SFSC de manière à légèrement améliorer ses performances sur trois des quatre cas de CIFAR, en perdant un peu de sa capacité d'adaptation. En effet, il est probable que ces formes ayant fixé l'AC = 1 ne conviennent pas aussi bien pour les cas de DTD *from scratch* qui visent l'AC = −1. Il est de même pour les modèles où l'AC et Norm1 ont eu davantage d'importance sur le rendu final, comme sur KDEF et les modèles pré-entraînés utilisés sur DTD.

En fin de compte, les formes partiellement apprises de la SFSC peuvent parfois obtenir de légers gains de performances par rapport à la SFSC entièrement apprise en ayant une forme plus spécialisée selon les besoins du modèle. Cependant, ceci se fait au détriment d'une perte sur sa capacité d'adaptation pour son utilisation avec d'autres bases de données et potentiellement aussi avec d'autres modèles. Étant donné que les gains sont généralement assez légers, utiliser la SFSC entièrement apprise reste préférable dans un cas général. Malgré tout, il est possible que les formes partiellement apprises ou entièrement fixes se démarquent davantage dans des cas où les modèles ont beaucoup de difficulté à apprendre leurs poids. Dans le cas des formes entièrement fixes, celles-ci ont notamment pu démontrer que les performances de la SFSC ne viennent pas seulement de ses apprentissages, mais également de l'équation 2.5. Cependant, ceci est également vérifié par les résultats de la section suivante où l'implication plutôt faible de $\widehat{M}$ pour les modèles de CIFAR a été complètement supprimée de la SFSC afin d'uniquement normaliser en fréquences les entrées des CNN.

4.4.2 La SFSC : Uniquement normaliser en fréquences les entrées ou uniquement utiliser l'AC

La deuxième expérimentation supplémentaire vise à notamment comparer la SFSC par rapport à une forme de celle-ci où la partie $(1 - \widehat{M})$ a été supprimée de l'équation 2.5 afin d'uniquement

utiliser Norm0 #2 ($\bar{M}$). Dans un tel cas, l'entrée ne reçoit qu'une normalisation fréquentielle et ne subit aucune perte d'information. En principe, ceci pourrait donner de meilleures performances pour des cas comme ceux de CIFAR ou DTD *from scratch* où la partie de $\widehat{M}$ semble avoir relativement peu d'impact sur le résultat visuel des images obtenues par la SFSC (section 4.2.2). Dans le même sens, il est également possible d'utiliser des formes qui n'utiliseraient que l'AC et Norm1 afin d'avoir une comparaison plus équitable par rapport à la SFSC puisque ces deux formes auront le léger effet d'augmentation de données lié à l'aléatoire de l'AC.

En prenant en compte les principes des formes ayant des mixtes de poids fixes et appris, diverses formes ont été expérimentées en utilisant Norm0 #2 combiné à une AC et Norm1, ainsi qu'uniquement utiliser l'AC avec son Norm1. Dans le cas de Norm0 #2, les formes avec et sans l'AC sont équivalentes à l'utilisation seule de chacun des deux chemins du schéma de la SFSC dans la figure 2.1.

Les résultats liés à ces expérimentations sont présentés dans le tableau 4.15 avec des comparaisons par rapport à la SFSC et le modèle original. Pour les colonnes des configurations, les L indiquent que ces couches sont avec apprentissage tandis que les autres indiquent le numéro de leur poids unique et fixe utilisé.

Dans les cas avec uniquement l'AC et Norm1, les résultats obtenus indiquent très clairement que ceux ayant fixé l'AC $= 1$ ont beaucoup mieux fonctionné que ceux qui l'apprennent. Lorsque l'AC est utilisée seule, son effet d'augmentation de données obtient de moins bonnes performances par rapport à toujours mettre en nuance de gris les magnitudes des images. Cependant, comme illustré dans la section 4.2.2, ceci ne met pas pour autant les images en nuance de gris dans le domaine spatial en raison des canaux de couleurs qui sont également présents dans les phases des images. Malgré ces piètres performances pour la plupart d'entre eux, le dernier cas avec l'AC et Norm1 fixé a été respectivement légèrement meilleur et ex aequo sur CIFAR-10 et CIFAR-100, par rapport au modèle original. Il reste cependant que, dans un sens global, il semble préférable de ne pas utiliser d'AC directement sur les magnitudes des images d'entrées.

Tableau 4.15 Résultats du CNN de référence pour les différents cas avec uniquement Norm0 #2 ($\bar{M}$) et uniquement l'AC sur les différents cas de CIFAR avec 10 exécutions. Les "+" indiquent les cas avec de l'augmentation de données

Dataset	Configuration			Résultats		Différence de performance par rapport à la SFSC						Différence de performance par rapport a la référence					
						Diff acc		Nb différence positive		Nb différence négative		Diff acc		Nb différence positive		Nb différence négative	
	Norm0 #2	AC	Norm1	Mean ± Std	Median	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %
CIFAR-10	L	#-1	_	83,57 ± 0,25	83,62	**0,08**	**0,16**	0	6	4	0	**0,44**	**0,53**	0	10	0	0
	L	L	L	83,1 ± 0,32	83,23	-0,39	-0,24	0	3	7	0	**-0,03**	**0,13**	0	3	7	0
	L	L	#0	83,46 ± 0,16	83,47	-0,03	**0,01**	0	4	6	0	**0,32**	**0,38**	0	8	2	0
	L	#1	L	82,14 ± 0,39	82,11	-1,35	-1,35	0	0	1	9	-1	-0,98	0	1	4	5
	L	#1	#0	82,1 ± 0,26	82,13	-1,39	-1,33	0	0	1	9	-1,03	-0,96	0	0	5	5
CIFAR-10+	L	#-1	_	84,93 ± 0,2	85,01	0	**0,13**	0	4	6	0	**0,16**	**0,21**	0	8	2	0
	L	L	L	84,49 ± 0,25	84,55	-0,44	-0,33	0	1	8	1	-0,28	-0,25	0	3	7	0
	L	L	#0	84,78 ± 0,19	84,76	-0,15	-0,12	0	3	7	0	0	-0,04	0	6	4	0
	L	#1	L	83,52 ± 0,27	83,46	-1,4	-1,42	0	0	2	8	-1,25	-1,34	0	0	2	8
	L	#1	#0	83,35 ± 0,31	83,4	-1,57	-1,48	0	0	0	10	-1,42	-1,4	0	0	3	7
CIFAR-100	L	#-1	_	45,96 ± 0,31	46,04	-0,4	-0,42	0	3	6	1	**0,2**	**0,29**	1	3	6	0
	L	L	L	45,22 ± 0,56	45,46	-1,14	-0,99	0	0	3	7	-0,54	-0,28	0	1	7	2
	L	L	#0	45,81 ± 0,53	45,89	-0,55	-0,57	0	3	5	2	**0,05**	**0,14**	1	5	3	1
	L	#1	L	44,09 ± 0,35	43,98	-2,28	-2,47	0	0	0	10	-1,68	-1,76	0	0	1	9
	L	#1	#0	44,26 ± 0,24	44,19	-2,1	-2,27	0	0	0	10	-1,5	-1,56	0	0	1	9
CIFAR-100+	L	#-1	_	46,56 ± 0,36	46,6	-0,04	0	0	6	3	1	**0,49**	**0,59**	0	8	2	0
	L	L	L	46,19 ± 0,54	46,13	-0,41	-0,48	0	2	7	1	**0,12**	**0,12**	0	5	5	0
	L	L	#0	46,3 ± 0,58	46,26	-0,3	-0,34	0	3	6	1	**0,24**	**0,25**	1	6	3	0
	L	#1	L	44,48 ± 0,36	44,34	-2,13	-2,26	0	0	0	10	-1,59	-1,67	0	0	1	9
	L	#1	#0	44,35 ± 0,53	44,32	-2,26	-2,29	0	0	1	9	-1,72	-1,7	0	0	2	8
CIFAR-10	_	L	L	17,27 ± 22,72	10	-66,22	-73,46	0	0	0	10	-65,86	-73,09	0	0	1	9
	_	L	#0	82,15 ± 0,4	82,07	-1,34	-1,4	0	0	2	8	-0,98	-1,03	0	1	5	4
	_	#1	L	82,13 ± 0,35	82,02	-1,35	-1,44	0	0	4	6	-1	-1,07	0	0	3	7
	_	#1	#0	83,41 ± 0,17	83,41	-0,07	-0,05	0	4	6	0	**0,28**	**0,32**	0	8	2	0
CIFAR-10+	_	L	L	60,98 ± 35,21	82,8	-23,94	-2,07	0	0	2	8	-23,79	-1,99	0	0	4	6
	_	L	#0	84,06 ± 0,7	84,34	-0,87	-0,54	0	1	6	3	-0,71	-0,46	0	1	6	3
	_	#1	L	83,54 ± 0,29	83,52	-1,39	-1,36	0	0	2	8	-1,23	-1,28	0	0	5	5
	_	#1	#0	84,75 ± 0,2	84,72	-0,17	-0,16	0	3	7	0	-0,02	-0,08	0	6	4	0
CIFAR-100	_	L	L	8,7 ± 16,07	1,1	-37,66	-45,35	0	0	0	10	-37,06	-44,64	0	0	0	10
	_	L	#0	45,3 ± 0,69	45,47	-1,06	-0,98	0	1	4	5	-0,46	-0,27	0	2	7	1
	_	#1	L	44,04 ± 0,37	44,07	-2,32	-2,39	0	0	0	10	-1,72	-1,68	0	0	1	9
	_	#1	#0	45,74 ± 0,51	45,76	-0,62	-0,69	0	3	5	2	-0,02	**0,02**	1	4	4	1
CIFAR-100+	_	L	L	9,57 ± 17,96	1,02	-37,03	-45,59	0	0	0	10	-36,49	-45	0	0	0	10
	_	L	#0	45,74 ± 0,94	45,93	-0,86	-0,68	0	2	4	4	-0,33	-0,09	1	3	3	3
	_	#1	L	44,37 ± 0,29	44,27	-2,24	-2,33	0	0	0	10	-1,7	-1,74	0	0	0	10
	_	#1	#0	45,91 ± 0,62	46	-0,69	-0,6	0	1	8	1	-0,16	-0,01	0	4	5	1

Dans le cas des formes avec uniquement Norm0 #2, les deux meilleurs sont, dans l'ordre, l'utilisation seule de Norm0 #2 ainsi que le cas ayant également appris l'AC mais fixé Norm1. Ces formes ont réussi à améliorer le modèle original sur l'ensemble ou la majorité des cas de CIFAR. Cependant, elles restent globalement légèrement inférieures à la SFSC où seule la première configuration a obtenu des médianes légèrement meilleures sur CIFAR-10/10+. Par conséquent, de manière similaire à ce qui a été présenté dans la section 4.2.2, les deux parties de la SFSC ($\bar{M}$ et $1 - \widehat{M}$) seraient importantes même lorsque l'une d'entre elles semble avoir une contribution moindre que l'autre sur les aspects visuels des images obtenues. En sachant que ces configurations ayant uniquement utilisé Norm0 #2 (ou Norm0 #1 selon la façon de voir les choses) obtiennent déjà de légères baisses de performances sur CIFAR, il est probable que leurs écarts négatifs soient plus élevés sur les cas où la partie de $\widehat{M}$ a eu davantage d'importance comme avec les modèles pré-entraînés sur DTD et ceux sur KDEF (section 4.2.2).

En somme, ces résultats avec différentes formes ayant uniquement l'un des deux chemins de la SFSC ou ayant uniquement l'AC ont surtout permis de confirmer à nouveau que les deux parties

de la SFSC sont importantes, et ce, même dans les cas où seul l'un des deux semble être vraiment important. De plus, étant donné que $\bar{M}$ et M fonctionnent mieux sans la présence d'une AC, il semble préférable de garder son utilisation uniquement pour $\widehat{M}$ et non pour les deux parties de l'équation 2.5. Malgré le fait que la SFSC entièrement apprise semble être celle qui fonctionne globalement le mieux pour traiter les images d'entrées des CNN, il est également possible que certaines de ces formes alternatives fonctionnent mieux dans d'autres situations. L'une d'entre elles serait de les utiliser sur les sorties de l'ensemble ou de certaines des couches de neurones des modèles. Étant donné que la normalisation fréquentielle des MN permet d'accéder à davantage de dimensions que les normalisations spatiales, comme celles couvertes par le SN (Luo *et al.*, 2018b,1), leur utilisation à l'intérieur des modèles pourrait donner des résultats intéressants. Cependant, il serait s'en doute préférable que ces méthodes basées sur la SFSC soient combinées à des BN ou SN spatiaux, ou à leur équivalent en fréquences (section 1.1.2.2). Ceci permettrait de s'assurer que le format des statistiques finales soit le même que celui des modèles originaux. Diverses formes sont possibles sur leur intégration dans des modèles pour divers types de tâches qui peuvent être différentes de la classification d'images, mais ceci serait à explorer dans de futures recherches.

4.4.3 Les filtres fréquentiels : Comparaison par rapport à leur forme sans apprentissage

La troisième et dernière expérimentation supplémentaire vise à vérifier si l'utilisation de paramètres pour les filtres fréquentiels du tableau 3.1 est nécessairement préférable par rapport à utiliser des filtres fixes sans apprentissage. Bien que les expérimentations préliminaires qui ont servi à déterminer les configurations finales des filtres fréquentiels prenaient également en compte ces filtres fixes (section 3.2), il reste quand même intéressant de présenter ces résultats dans le cas d'expérimentations sur CIFAR. En effet, le filtre fréquentiel #3 est celui qui a globalement le mieux fonctionné avec les modèles sur CIFAR (section 4.3.2), tout en étant celui ayant le plus de contraintes sur la forme de son filtre et qui va, au final, rester relativement similaire à celle de son initialisation. Bien que celui-ci a eu les pires performances sur DTD *from scratch* et KDEF, cette comparaison entre les filtres appris et leurs formes équivalentes

fixes permet, en même temps, de vérifier si leurs améliorations de performances avec la plupart des modèles originaux viennent surtout de leur forme en loi de puissance (ω^γ) ou de l'utilisation de certains paramètres.

Le tableau 4.16 présente les résultats obtenus avec le modèle de référence sur CIFAR avec les filtres fréquentiels appris et leurs formes équivalentes fixes avec des comparaisons de leurs performances par rapport à la SFSC et les modèles originaux. Étant donné que les filtres fréquentiels sont initialisés selon $\gamma = 0,5$ (filtre #1 et #2) et $\gamma = 1$ (filtre #3 et #4) et que les filtres #1 et #2 sont utilisés avec la SFSC, les quatre cas possibles de filtres fixes selon ces deux aspects ont été expérimentés dans cette section.

Tableau 4.16 Résultats du CNN de référence avec les filtres fréquentiels fixes ou appris sur CIFAR sur les 10 exécutions. Les "+" indiquent les cas avec de l'augmentation de données

Dataset	Used with the SFSC	No	Filtre fréquentiel		Différence de performance par rapport à la SFSC						Différence de performance par rapport a la référence					
					Diff acc		Nb différence positive		Nb différence négative		Diff acc		Nb différence positive		Nb différence négative	
			Mean ± Std	Median	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %	Mean	Median	Nb ≥ 1 %	Nb ≥ 0 %	Nb ≥ -1 %	Nb < -1 %
CIFAR-10	1	1	83,45 ± 0,26	83,37	-0,04	-0,1	0	4	6	0	**0,31**	**0,28**	0	7	3	0
		2	83,54 ± 0,25	83,53	**0,05**	**0,06**	0	6	4	0	**0,41**	**0,44**	0	8	2	0
	0	3	83,54 ± 0,28	83,45	**0,05**	-0,02	0	7	3	0	**0,41**	**0,35**	1	7	2	0
		4	83,57 ± 0,08	83,57	**0,08**	**0,1**	0	5	5	0	**0,43**	**0,47**	0	9	1	0
	1	Fixed 0,5	83,62 ± 0,25	83,62	**0,13**	**0,16**	0	6	4	0	**0,48**	**0,53**	1	9	0	0
		Fixed 1,0	83,35 ± 0,29	83,24	-0,14	-0,22	0	3	7	0	**0,22**	**0,15**	0	8	2	0
	0	Fixed 0,5	83,49 ± 0,3	83,44	0	-0,03	0	3	7	0	**0,35**	**0,34**	0	7	3	0
		Fixed 1,0	83,4 ± 0,29	83,32	-0,09	-0,14	0	3	7	0	**0,26**	**0,23**	1	7	2	0
CIFAR-10+	1	1	85,12 ± 0,19	85,19	**0,19**	**0,31**	0	6	4	0	**0,35**	**0,4**	0	9	1	0
		2	85,1 ± 0,28	85,04	**0,17**	**0,16**	0	6	4	0	**0,33**	**0,24**	2	5	3	0
	0	3	85,13 ± 0,16	85,13	**0,21**	**0,25**	0	7	3	0	**0,36**	**0,33**	0	10	0	0
		4	85,17 ± 0,22	85,26	**0,25**	**0,38**	0	8	2	0	**0,4**	**0,46**	0	10	0	0
	1	Fixed 0,5	84,92 ± 0,26	84,87	0	-0,01	0	5	5	0	**0,15**	**0,07**	0	7	3	0
		Fixed 1,0	84,91 ± 0,42	85,05	-0,02	**0,17**	0	4	6	0	**0,14**	**0,25**	0	6	4	0
	0	Fixed 0,5	85,2 ± 0,21	85,2	**0,27**	**0,33**	0	7	3	0	**0,43**	**0,41**	1	9	0	0
		Fixed 1,0	84,88 ± 0,33	84,97	-0,04	**0,09**	0	3	7	0	**0,11**	**0,18**	0	6	4	0
CIFAR-100	1	1	46,14 ± 0,3	46,15	-0,22	-0,31	0	4	6	0	**0,38**	**0,4**	0	9	1	0
		2	46,13 ± 0,39	45,99	-0,23	-0,47	0	3	7	0	**0,37**	**0,24**	1	6	3	0
	0	3	45,92 ± 0,43	45,88	-0,45	-0,57	0	1	6	3	**0,15**	**0,14**	1	4	5	0
		4	46,12 ± 0,36	46,01	-0,24	-0,44	0	2	8	0	**0,36**	**0,27**	1	7	2	0
	1	Fixed 0,5	46,39 ± 0,44	46,45	**0,03**	0	0	5	5	0	**0,63**	**0,71**	3	6	1	0
		Fixed 1,0	45,83 ± 0,27	45,88	-0,53	-0,57	0	0	9	1	**0,07**	**0,14**	0	6	4	0
	0	Fixed 0,5	46,32 ± 0,41	46,37	-0,04	-0,08	0	3	7	0	**0,56**	**0,63**	1	9	0	0
		Fixed 1,0	45,99 ± 0,31	46,06	-0,37	-0,39	0	1	9	0	**0,23**	**0,32**	0	8	2	0
CIFAR-100+	1	1	46,77 ± 0,43	46,63	**0,16**	**0,03**	0	6	4	0	**0,7**	**0,62**	3	6	1	0
		2	46,34 ± 0,32	46,18	-0,26	-0,43	0	2	7	1	**0,27**	**0,16**	0	8	2	0
	0	3	46,75 ± 0,51	46,76	**0,14**	**0,16**	1	3	6	0	**0,68**	**0,74**	4	4	2	0
		4	46,65 ± 0,53	46,63	**0,05**	**0,03**	1	5	3	1	**0,59**	**0,62**	3	5	2	0
	1	Fixed 0,5	46,62 ± 0,31	46,56	**0,2**	**0,03**	0	6	3	1	**0,62**	**0,56**	2	7	1	0
		Fixed 1,0	46,25 ± 0,43	46,13	-0,17	-0,4	0	4	5	1	**0,25**	**0,13**	0	5	5	0
	0	Fixed 0,5	46,82 ± 0,49	46,63	**0,4**	**0,11**	1	5	4	0	**0,82**	**0,63**	3	6	1	0
		Fixed 1,0	46,55 ± 0,33	46,6	**0,13**	**0,08**	0	4	6	0	**0,55**	**0,6**	0	9	1	0

Dans le cas des filtres fixes, ceux-ci ont tous amélioré les performances du modèle original comme l'ont également fait ceux appris. Ceci permet de confirmer à nouveau que les filtres fréquentiels en loi de puissance (ω^γ) conviennent relativement bien aux modèles sur CIFAR, ainsi qu'à ceux utilisés sur d'autres bases de données comme l'ont montré les filtres appris (section 4.3.2). Cependant, parmi les filtres fixes, ce sont ceux ayant utilisé $\gamma = 0,5$ qui ont le mieux fonctionné par rapport à la SFSC originale en ayant des performances globales légèrement

supérieures sur deux ou trois des quatre cas de CIFAR. Dans le cas des filtres appris, ce sont plutôt les filtres #3 et #4 ayant $\gamma = 1$ (fixe) qui ont le mieux fonctionné par rapport à la SFSC où CIFAR-100 est le seul cas d'exception. Les meilleures valeurs de γ semblent donc inversées entre ceux appris et ceux fixes, mais les deux formes semblent légèrement meilleures lorsque les filtres sont directement appliqués aux images au lieu de la sortie de la SFSC. Bien qu'il soit difficile d'expliquer cette inversion sur ce choix de la valeur de γ pour ce modèle, il est également vrai que les différences sont relativement légères et peuvent donc mener à des conclusions différentes pour d'autres modèles.

Sur un plan global pour ce modèle sur CIFAR, les filtres fixes ayant utilisé $\gamma = 0,5$ ont eu des performances globales légèrement plus élevées que les filtres #3 et #4, montrant ainsi que des filtres fixes en loi de puissance peuvent parfois être meilleurs que ceux ayant des paramètres. Cependant, lors d'une prise en compte de davantage de modèles et de bases de données, comme lors des expérimentations préliminaires effectuées pour les filtres fréquentiels (section 3.2), les quatre filtres fréquentiels sélectionnés avaient globalement mieux réussi que ceux sans apprentissage. De manière semblable, bien que le filtre #3 a été celui qui a le mieux fonctionné sur CIFAR, il a obtenu les moins bonnes performances sur DTD *from scratch* et KDEF en raison de ses contraintes sur sa forme. Les autres filtres ayant moins de contraintes ont alors mieux fonctionné pour ces autres situations en raison de leur meilleur capacité d'adaptation. L'utilisation de paramètres et d'une certaine flexibilité sur la forme des filtres permet de les rendre plus polyvalents en étant, par contre, possiblement un peu moins précis pour des cas particuliers.

CONCLUSION ET RECOMMANDATIONS

Au cours de cette recherche, trois nouvelles méthodes de prétraitement des entrées à l'aide du domaine du Fourier ont été présentées et utilisées avec diverses formes d'architectures de CNN pour de la classification d'images naturelles, de textures et des émotions du visage. Contrairement à plusieurs autres méthodes de la littérature utilisant le domaine spectral, celles-ci ne nécessitent aucune modification de l'architecture ou des configurations des algorithmes d'apprentissage pour les poids des modèles. De plus, plusieurs mesures sont prises avec chacune des configurations afin de s'assurer d'avoir des mesures fiables et reproductibles. Les expérimentations ont également tenté d'isoler les différents facteurs pouvant influencer les résultats afin que les différences de performances ne soient causées que par l'utilisation ou non de nos méthodes. Cependant, étant donné que celles-ci ajoutent certains poids, des formes sans apprentissage et partiellement apprises ont été expérimentées afin, en autres, d'isoler la contribution liée à l'apprentissage par rapport aux formes originales de ces méthodes. Au travers de nos expérimentations, les différents objectifs définis au départ de cette recherche ont donc été atteints.

La première méthode qui est la SFSC simplifie les images d'entrées en y soustrayant les informations liées à l'autocorrélation de leur magnitude à l'aide d'un filtre correspondant à leur magnitude inversée. Elle se sert, par conséquent, d'un filtre propre à chacune des images d'entrées, au lieu d'utiliser des filtres aux formes préétablies qui seraient les mêmes, peu importe le contexte. La SFSC a pour effet principal d'atténuer les basses fréquences afin que les moyennes et hautes fréquences soient plus accessibles aux modèles qui pourront donc les apprendre plus rapidement, tout en étant moins influencés par les variations des basses fréquences. Cette méthode effectue également des changements sur les proportionnalités des magnitudes au travers des différentes dimensions de l'entrée à l'aide de l'utilisation de trois MN et d'une AC. Ces couches permettent d'ajuster les effets de cette méthode en fonction des besoins des modèles et peuvent également fournir des informations sur les aspects qui sont les plus importants pour les modèles. Les expérimentations ont montré que la SFSC est apte à améliorer les performances

de divers types d'architectures utilisés sur plusieurs types de bases de données où seuls les cas sur CIFAR avec le WRN 16-8 et le DenseNet (L = 100, K = 12) n'ont globalement pas été améliorés par l'utilisation de la SFSC. Cependant, les baisses de performances globales pour ces deux modèles d'exception étaient de moins de 1 %. Cette méthode fonctionne également avec les CNN aux poids pré-entraînés sur ImageNet où, contrairement à ce qui est le plus souvent fait dans la littérature, celle-ci ne modifie que son entrée au lieu de sa sortie. Cependant, ces méthodes qui ne modifient que les sorties des modèles pourraient, en principe, être appliquées conjointement avec la SFSC, tout comme peuvent l'être d'autres méthodes reliées aux entrées ou aux sorties des couches de convolution.

La deuxième et la troisième méthode consistent à l'utilisation de filtres fréquentiels passe-haut en loi de puissance qui sont respectivement utilisées avec la sortie de la SFSC et les images d'entrées des CNN. Diverses formes avec l'utilisation de paramètres permettant une évolution de la forme de ces filtres au cours de l'apprentissage ont été expérimentées et quatre configurations finales ont été sélectionnées. Les expérimentations ont montré que ces filtres pouvaient améliorer les modèles originaux dans la plupart des cas, mais que la SFSC est celle qui donne globalement les meilleures performances. Malgré tout, ces filtres ont notamment pu dépasser la SFSC pour ses deux cas d'exception sur CIFAR, le VGG-11+BN sur DTD *from scratch* et tous les cas de KDEF avec le filtre #2. Basées sur cela, certaines améliorations pourraient être effectuées sur la SFSC en modifiant, par exemple, ses calculs ou en l'utilisant conjointement à d'autres filtres fréquentiels que ceux qui ont été expérimentés.

Par rapport à la SFSC, soit celle ayant globalement le mieux fonctionné parmi nos trois méthodes, différentes avenues sont possibles pour de futures recherches. D'une part, des améliorations pourraient être apportées pour corriger certaines de ses lacunes dont le fait de devoir parfois ajuster les hyperparamètres d'apprentissage quand elle se sert de SGD et qui peuvent différer de ceux utilisés par le CNN. Malgré que ceci puisse être corrigé au travers de quelques expérimentations

d'optimisations, une correction ou une amélioration de cette faiblesse permettrait de faciliter son utilisation en ayant moins besoin de tester différentes configurations d'apprentissage pour obtenir des gains de performances. D'autre part, comme mentionnées précédemment, certaines modifications liées aux calculs de la SFSC pourraient être effectuées afin qu'elle puisse également améliorer le WRN 16-8 et le DenseNet sur CIFAR, comme l'ont fait certains filtres fréquentiels. Idéalement, ceci ne doit pas se faire au détriment d'une perte de sa capacité d'adaptation pour son utilisation avec diverses architectures et bases de données. De l'autre côté, la SFSC ou certaines de ces variantes pourraient également être utilisées pour d'autres tâches que de la classification d'images ainsi qu'ailleurs dans les CNN, telles que leur utilisation sur les sorties de certaines couches de convolution. Étant donné que les MN et l'AC de la SFSC sont basés sur le SN (Luo *et al.*, 2018b,1) et qu'ils couvrent toutes les combinaisons possibles des dimensions de leur entrée dans le domaine de Fourier, les MN et l'AC sont, en principes, également adaptés à l'utilisation d'autres entrées que des images 2D, tels que des entrées 1D ou 3D, mais ceci serait à vérifier par de futures recherches.

ANNEXE I

REPRÉSENTATION FRÉQUENTIELLE DES COUCHES DE CONVOLUTION DES MODÈLES

Ce chapitre de l'annexe illustre les représentations fréquentielles des magnitudes des couches de convolution des différents modèles qui ont été utilisés au cours de cette recherche. Pour les figures qui sont présentées, les graines aléatoires (*random seed*) et *split* (DTD et KDEF) sont les mêmes pour tous les modèles liés aux mêmes bases de données, afin de permettre une comparaison adéquate entre les poids appris par les modèles selon les méthodes utilisées pour le traitement de leur entrée. Ces représentations viennent également appuyer certaines recherches de la littérature qui ont montré que les modèles utilisent l'ensemble du spectre fréquentiel et non uniquement les hautes ou les basses fréquences (Tsuzuku & Sato, 2019 ; Yin *et al.*, 2019a). Afin de limiter le nombre d'illustrations pour cette annexe, une seule des exécutions de chacun des modèles a été sélectionnée, mais ceci permet, malgré tout, d'avoir des exemples de ce que les modèles apprennent. Il est également à noter que toutes les magnitudes illustrées ont été mises selon des échelles logarithmiques et que les calculs de différence de magnitude par rapport aux modèles originaux sont effectués après avoir appliqué ce changement d'échelle. Étant donné que les magnitudes suivent des courbures d'exponentielles inversées en raison de la définition de la transformée de Fourier (équation 2.1), les hautes fréquences peuvent sembler relativement faibles en comparaison des basses fréquences dans les représentations globales des couches de convolution. Cependant, comme il est notamment illustré dans les figures de la section 3, les hautes fréquences sont malgré tout utilisées par les CNN et elles ont donc une certaine importance qui peut être cachée par les différences de niveau de magnitude entre les basses et les hautes fréquences.

La section 1 illustre les représentations liées aux moyennes globales des magnitudes de l'ensemble des filtres de la première couche de convolution des différents modèles. Celles-ci permettent de mettre en évidence les régions fréquentielles qu'elles couvrent le plus et comment les différentes méthodes de prétraitement des entrées des modèles les ont influencées. Ces comparaisons sont également facilitées par la présence de figures sur les différences de magnitude globale entre les

premières couches des modèles orignaux avec ceux ayant utilisé nos nouvelles méthodes. Bien que les conclusions sur les effets de chacune des méthodes sur les modèles soient propres aux architectures et bases de données utilisées (Tsuzuku & Sato, 2019), certains aspects globaux peuvent être observés. Premièrement, les couches de convolutions semblent davantage se concentrer sur les basses fréquences en ayant des magnitudes plus élevées à des fréquences plus faibles. Cependant, les magnitudes à ces régions peuvent être plus faibles par rapport au modèle original. Deuxièmement, notamment pour les modèles sur les deux cas de DTD, KDEF et certains modèles sur CIFAR, les moyennes et hautes fréquences semblent davantage couvertes en ayant soit des magnitudes à ces régions plus élevées que le modèle original, soit des écarts de magnitudes plus faibles par rapport aux autres fréquences. Dans les deux cas, les représentations globales des modèles utilisant nos méthodes semblent indiquer que la première couche de convolution couvre davantage les moyennes et hautes fréquences par rapport aux modèles originaux. Malgré tout, basée sur le fait que la SFSC a généralement mieux fonctionné que les filtres fréquentiels, une certaine modération sur ces influences sur les premières couches de convolution serait également à considérer.

La section 2 illustre les représentations fréquentielles globales de la première couche de convolution ainsi que celles des couches de convolution plus profondes des modèles. Pour ces dernières, les couches sélectionnées sont celles ayant des filtres spatiaux de taille supérieure à 1×1 (ResNet-50 et DenseNet) et qui sont situées juste avant les changements de résolution spatiaux des caractéristiques. Dans le cas du BNInception, les représentations sont obtenues en combinant les magnitudes des couches dont leurs sorties sont concaténées ensemble. Ces figures sur les représentations fréquentielles des couches plus profondes permettent de mieux comprendre l'influence de nos méthodes sur ces couches tout en aidant à mieux identifier ce que les modèles visent à obtenir. Dans le cas de leur dernière couche de convolution, leur représentation ressemble, le plus souvent, à celle d'un filtre moyenneur 5×5. Cependant, il est probable que cet effet soit lié à l'utilisation d'une couche de moyennage globale (*global average pooling*) après cette dernière couche de convolution, pour la plupart des modèles, à l'exception des VGG, AlexNet et du VGG-M. Ceci fait également du sens avec la méthode de Ryu *et al.*

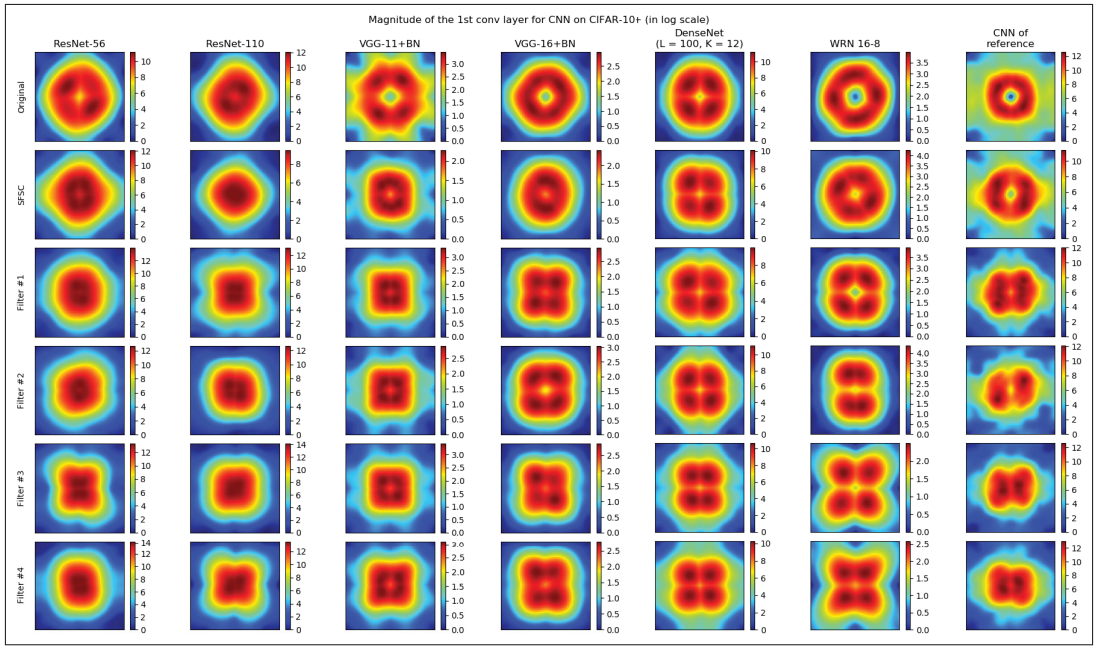
(2018) qui applique, dans certains cas, un filtre idéal passe-bas sur la magnitude de la dernière couche de convolution qui sera ensuite envoyée aux couches entièrement connectées (FC).

Finalement, les sections 3 et 4 illustrent respectivement les formes colorées des représentations fréquentielles et spatiales de chacun des filtres de la première couche de convolution avec et sans la présence de nos nouvelles méthodes. Deux modèles par base de données ont été sélectionnés avec le VGG-11 avec BN et le ResNet-50 qui sont utilisés sur trois des quatre bases de données, ainsi que le DenseNet (L = 100, K = 12) pour CIFAR-100+ et le BNInception pour KDEF. Il est également à noter que, contrairement aux figures des deux autres sous-sections de ce chapitre, les représentations de chacun des filtres sont très dépendantes de l'aléatoire. Malgré tout, ces figures permettent d'illustrer des exemples de ce que les modèles apprennent et elles mettent en évidence certains traits particuliers parfois relevés dans d'autres recherches. En effet, on peut notamment y observer des redondances sur les représentations fréquentielles de certains filtres qui sont très similaires. Ceci fait notamment du sens avec certaines recherches qui ont visé à réduire cette redondance comme celles sur le *pruning* (Molchanov, Tyree, Karras, Aila & Kautz, 2016). De l'autre côté, il semble y avoir certains liens entre le niveau des gradients et les performances des modèles avec la netteté des représentations fréquentielles des filtres. Ainsi, les filtres sur CIFAR et ImageNet (DTD avec des modèles pré-entraînés) ont le plus souvent des formes plutôt précises alors que ceux des modèles sur KDEF et DTD *from scratch* présentent des filtres pouvant avoir davantage de bruit en arrière-plan, soit des traces restantes de l'initialisation aléatoire.

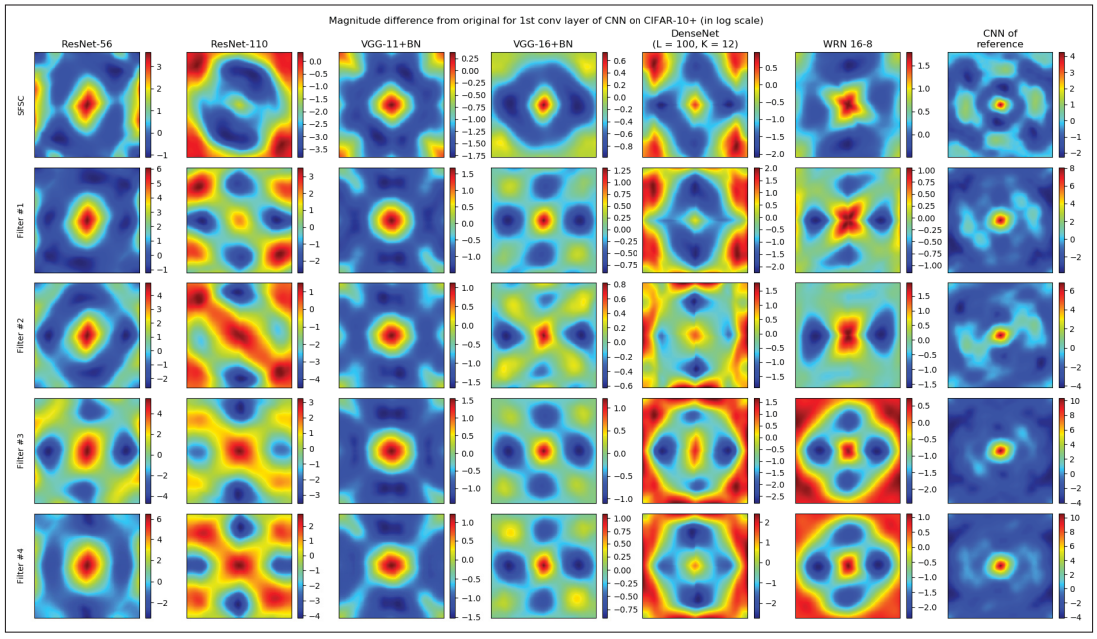
Bien que l'analyse approfondie des différences de représentations fréquentielles des couches des modèles peut sembler un peu superflue en raison du fait qu'elles sont automatiquement apprises par les modèles, certains chercheurs pourraient en tirer des informations utiles qui vont au-delà de celles énoncées précédemment. Ces informations peuvent notamment concerner les forces et faiblesses de nos méthodes, afin de, par exemple, mieux comprendre pourquoi certains filtres fréquentiels ont pu dépasser la SFSC dans certains cas ou pourquoi ceux-ci ont moins bien fonctionné que la SFSC, de manière générale. De l'autre côté, d'autres chercheurs pourraient développer leurs propres théories en prenant notamment en compte ces représentations

fréquentielles des couches apprises par nos modèles en prouvant, à nouveau, que les CNN utilisent l'ensemble du spectre fréquentiel.

1. Représentation fréquentielle de la première couche de convolution des CNN

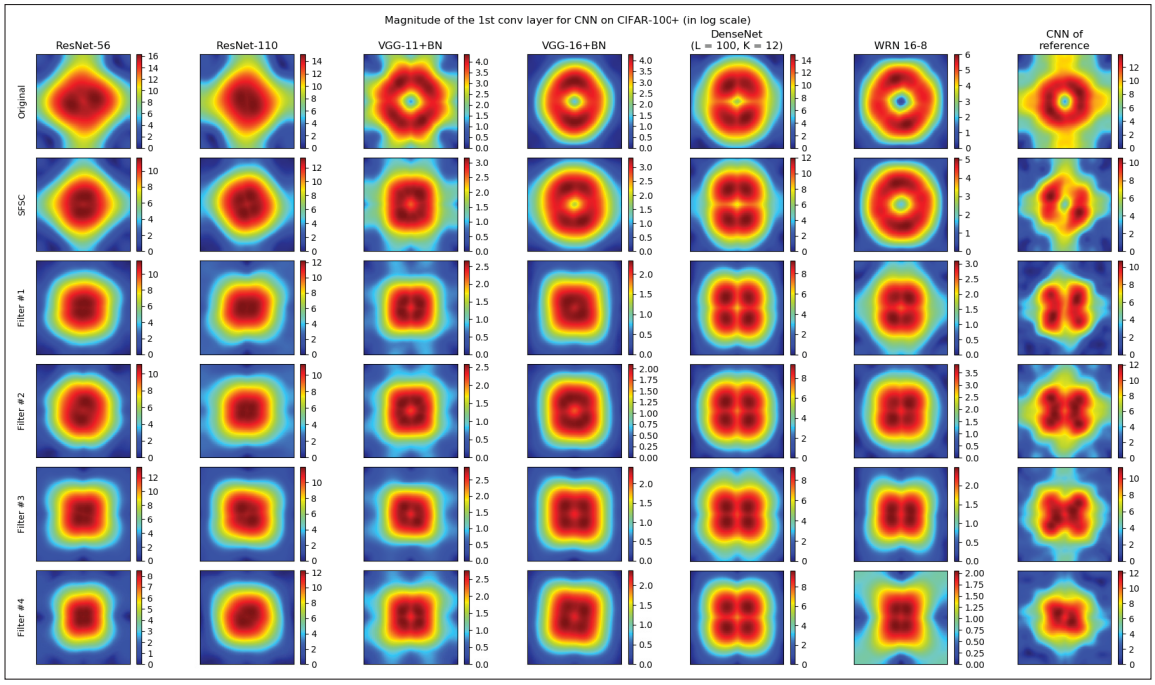


(a) Magnitude des premières couches de convolution des modèles sur CIFAR-10+

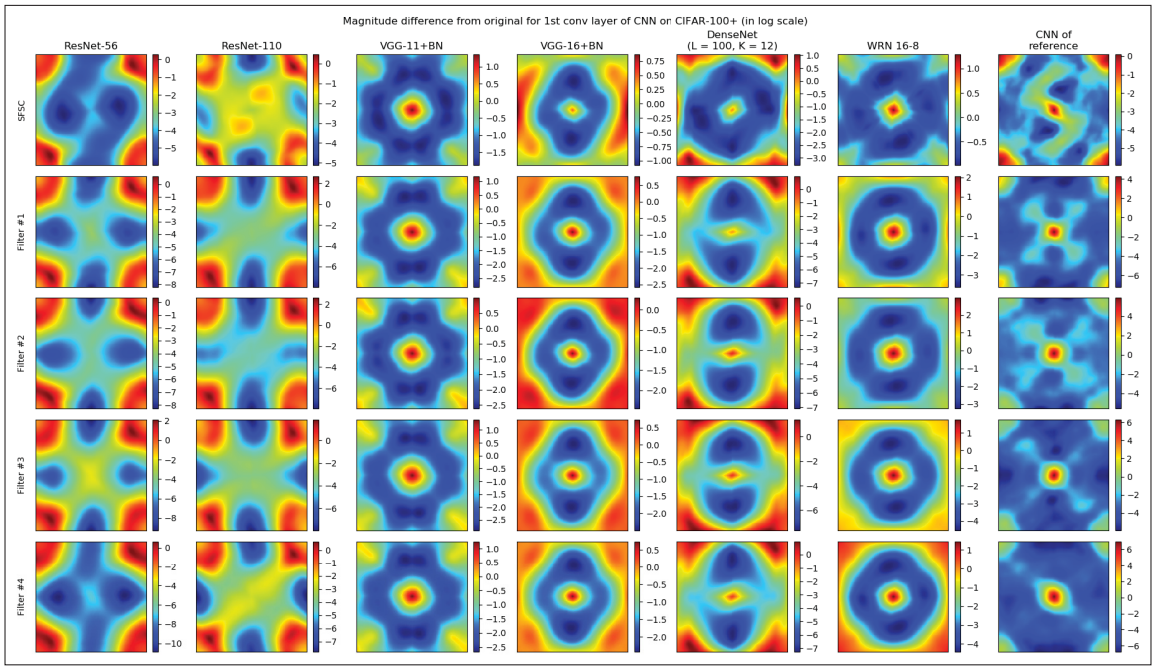


(b) Différence de magnitude par rapport aux modèles originaux (en utilisant des échelles logarithmiques)

Figure-A I-1 Magnitude des premières couches de convolution pour nos méthodes et différences de magnitude par rapport aux modèles originaux, sur CIFAR-10+

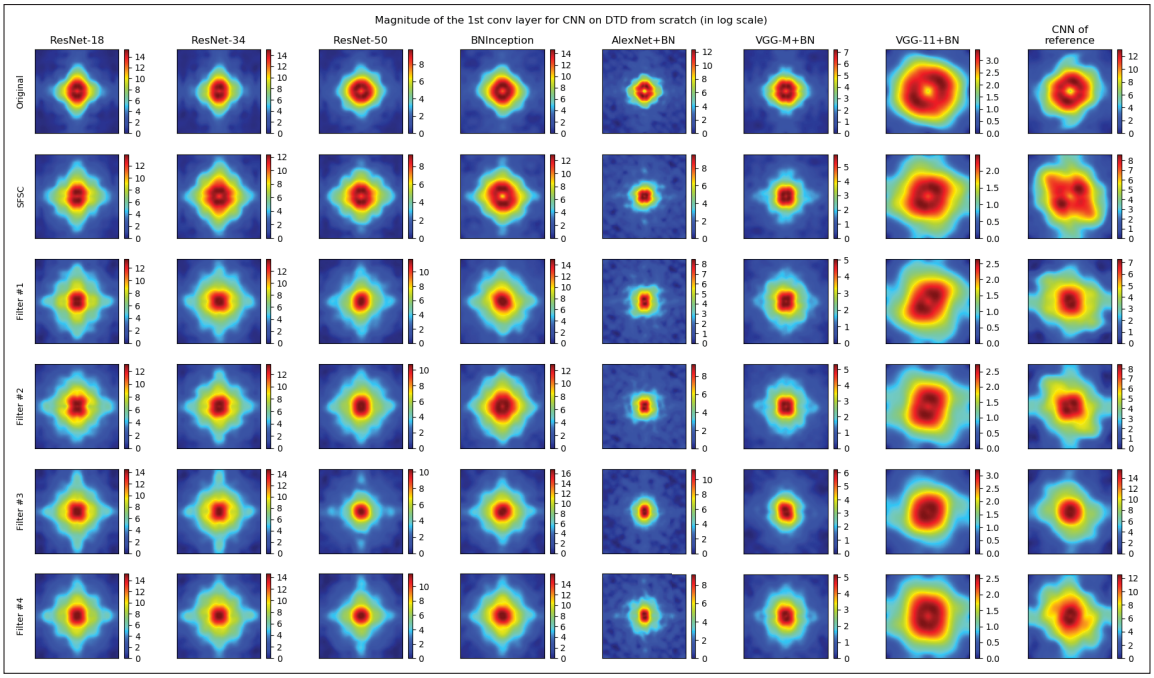


(a) Magnitude des premières couches de convolution des modèles sur CIFAR-100+

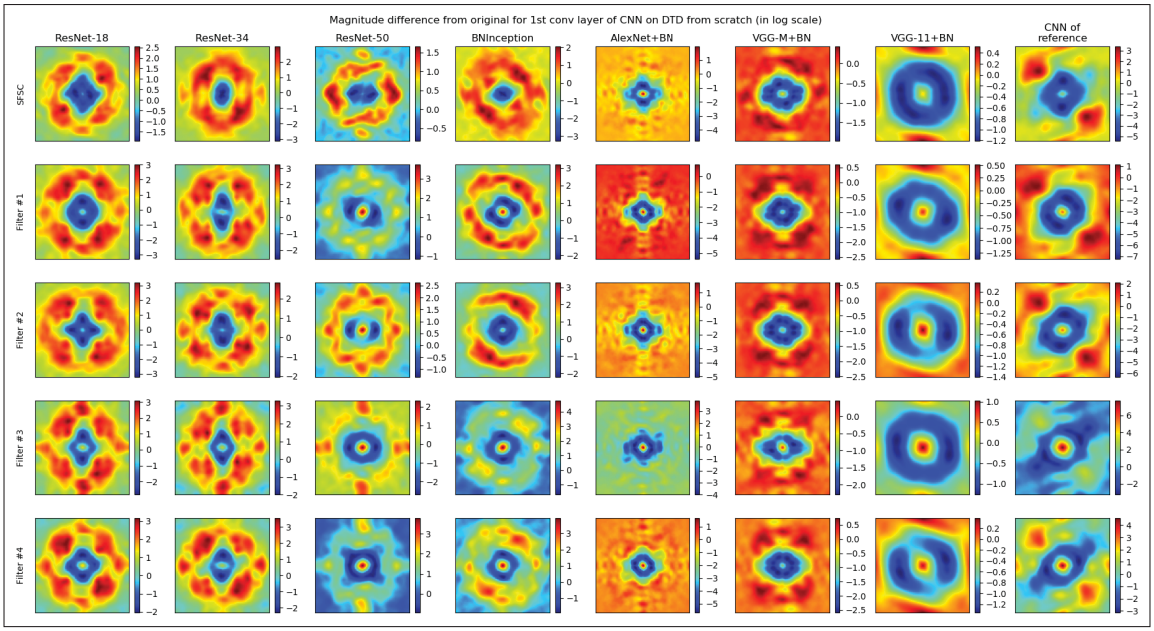


(b) Différence de magnitude par rapport aux modèles originaux (en utilisant des échelles logarithmiques)

Figure-A I-2 Magnitude des premières couches de convolution pour nos méthodes et différences de magnitude par rapport aux modèles originaux, sur CIFAR-100+



(a) Magnitude des premières couches de convolution des modèles sur DTD *from scratch*



(b) Différence de magnitude par rapport aux modèles originaux (en utilisant des échelles logarithmiques)

Figure-A I-3 Magnitude des premières couches de convolution pour nos méthodes et différences de magnitude par rapport aux modèles originaux, sur DTD *from scratch*

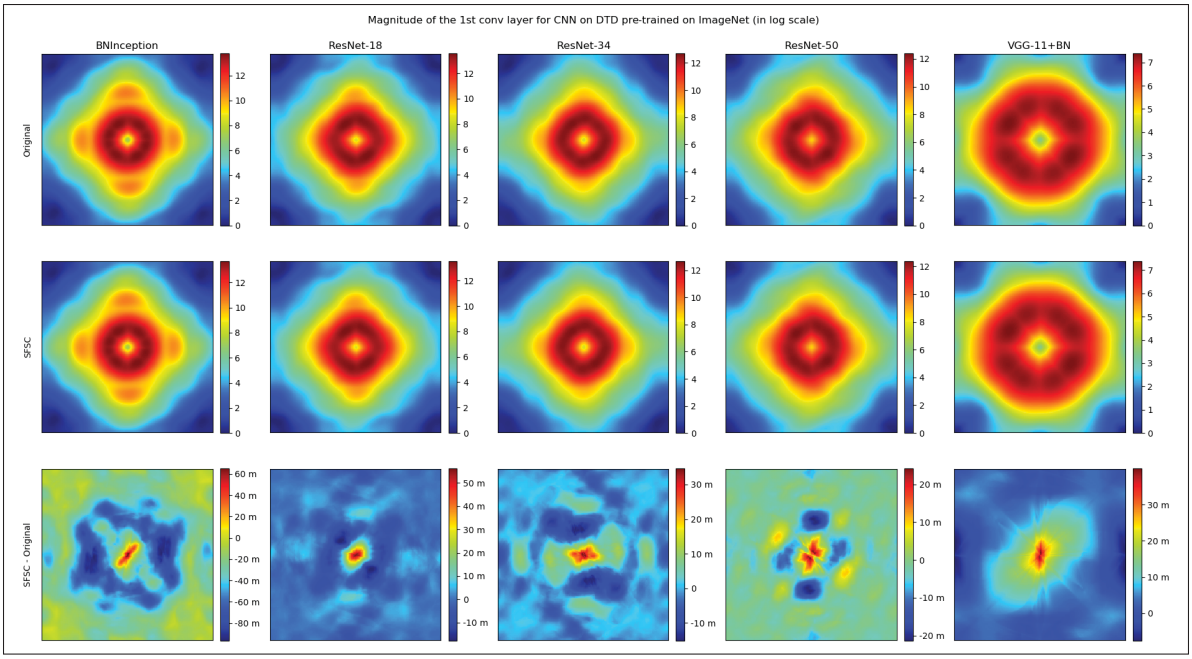
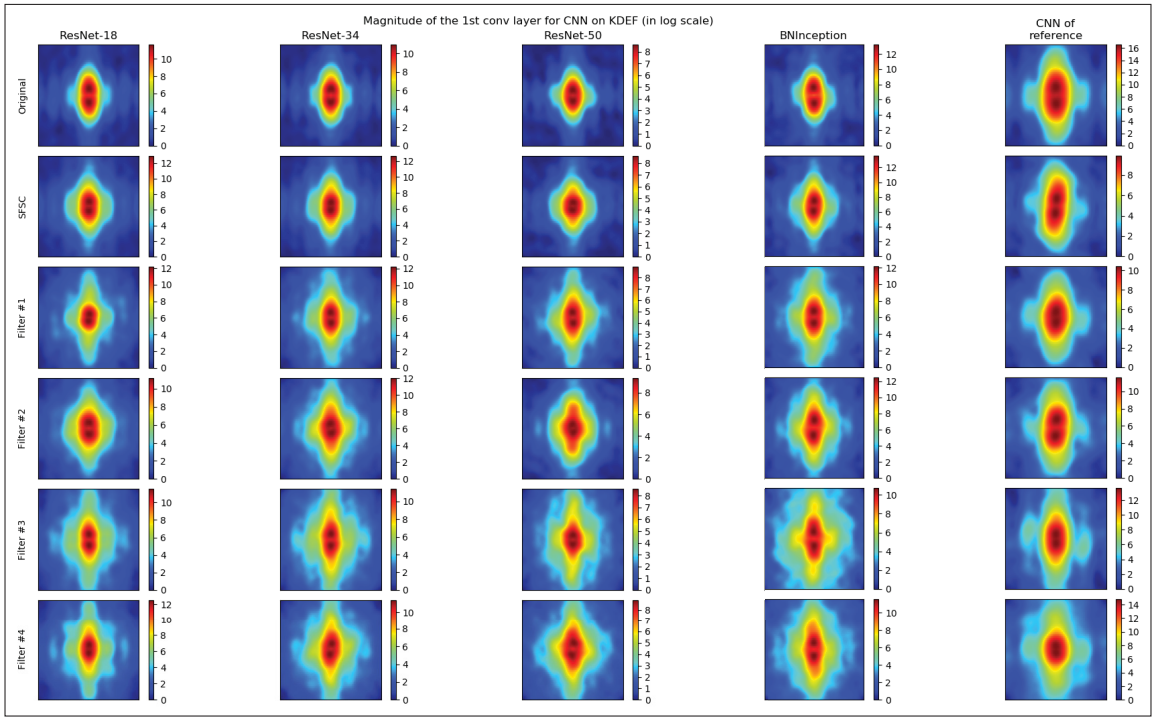
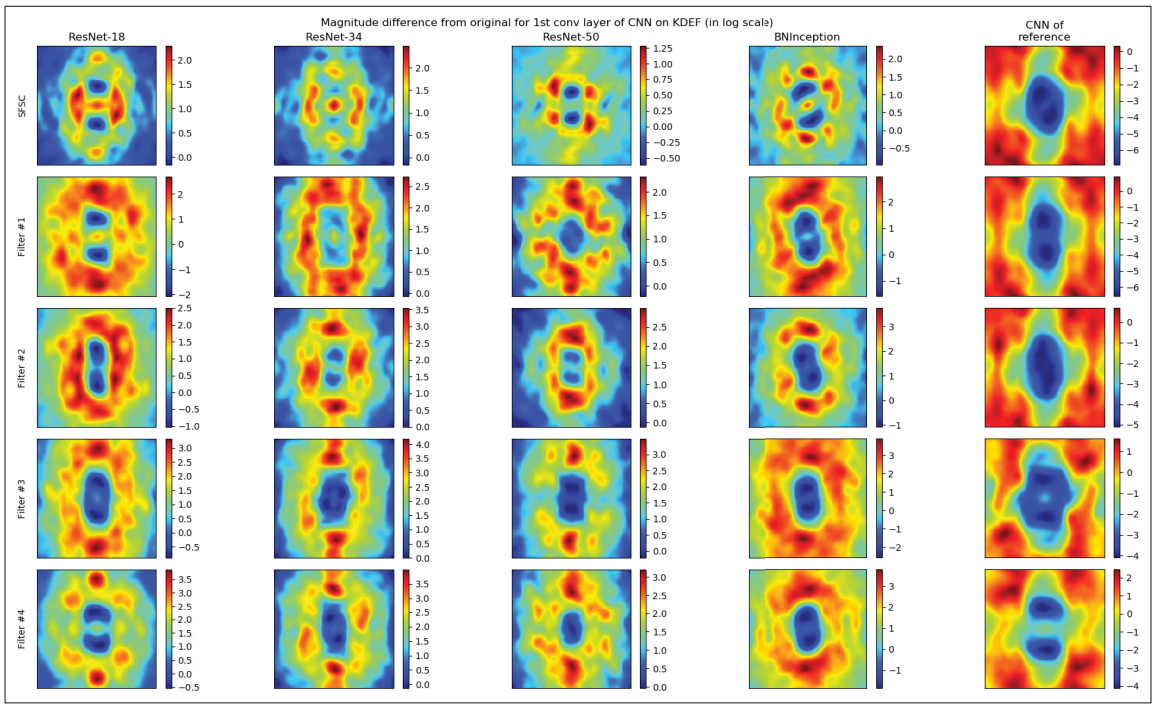


Figure-A I-4 Magnitude des premières couches de convolutions des modèles pré-entraînés sur ImageNet utilisé sur DTD avec et sans la SFSC et différence de magnitude entre ceux-ci



(a) Magnitude des premières couches de convolution des modèles sur KDEF

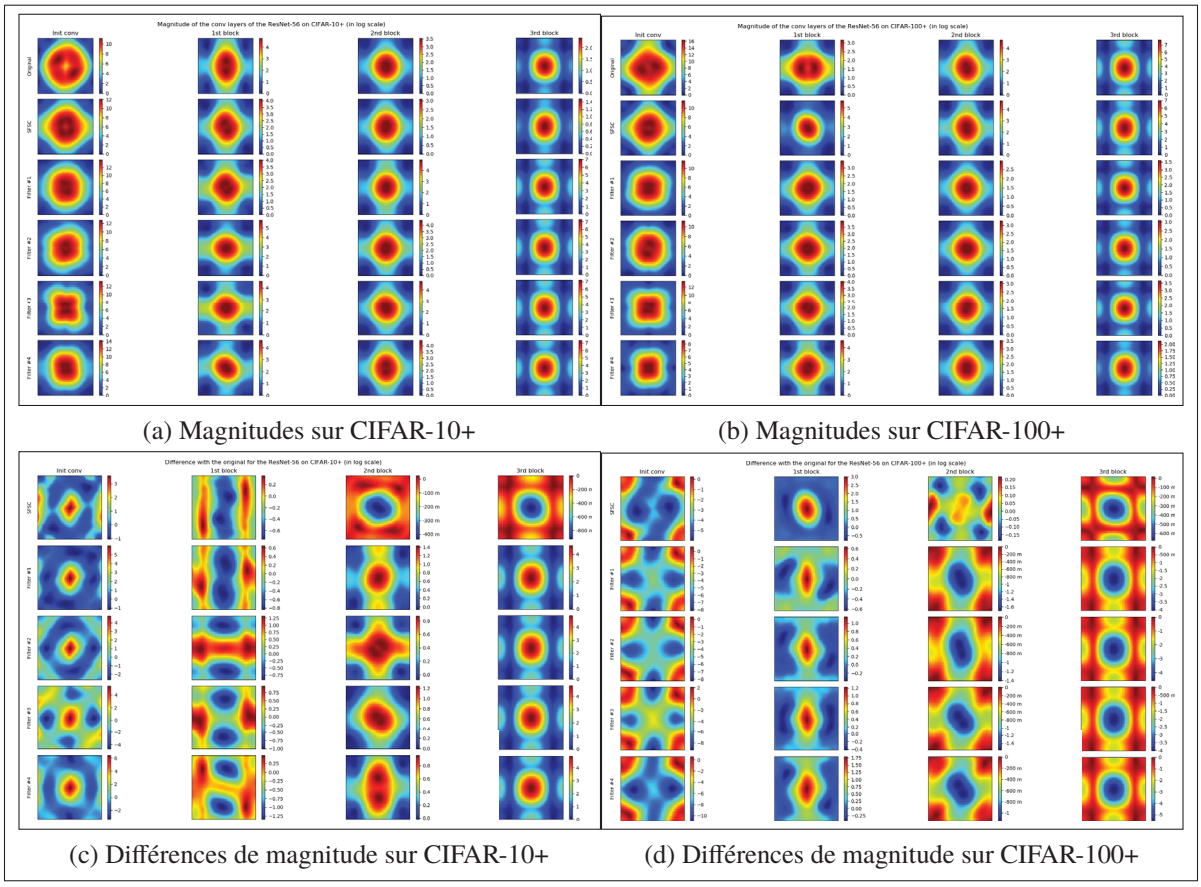


(b) Différence de magnitude par rapport aux modèles originaux (en utilisant des échelles logarithmiques)

Figure-A I-5 Magnitude des premières couches de convolution pour nos méthodes et différences de magnitude par rapport aux modèles originaux, sur KDEF

2. Représentation fréquentielle des couches profondes des CNN

2.1 CNN de CIFAR-10+ et CIFAR-100+



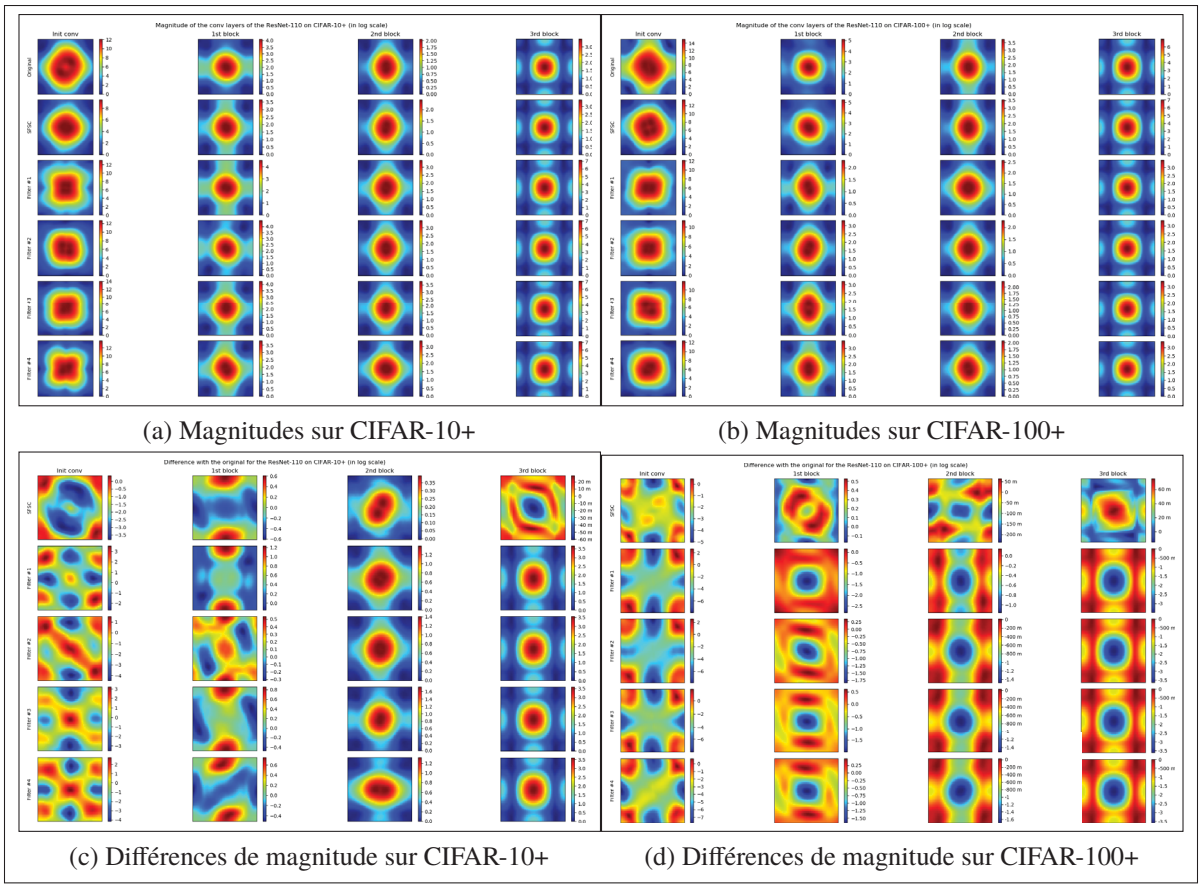
(a) Magnitudes sur CIFAR-10+

(b) Magnitudes sur CIFAR-100+

(c) Différences de magnitude sur CIFAR-10+

(d) Différences de magnitude sur CIFAR-100+

Figure-A I-6 Magnitude en échelle logarithmique des couches de convolution plus profondes du ResNet-56 sur CIFAR avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original



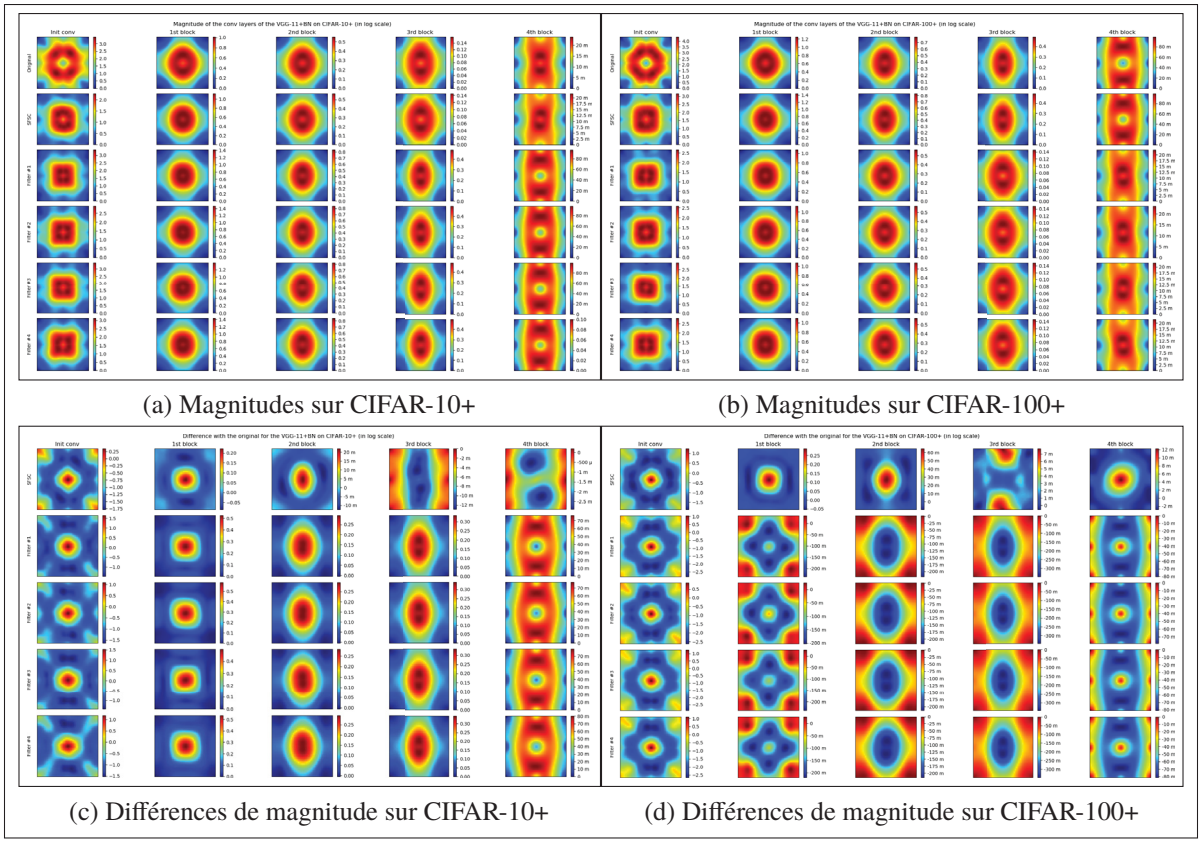
(a) Magnitudes sur CIFAR-10+

(b) Magnitudes sur CIFAR-100+

(c) Différences de magnitude sur CIFAR-10+

(d) Différences de magnitude sur CIFAR-100+

Figure-A I-7 Magnitude en échelle logarithmique des couches de convolution plus profondes du ResNet-110 sur CIFAR avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original



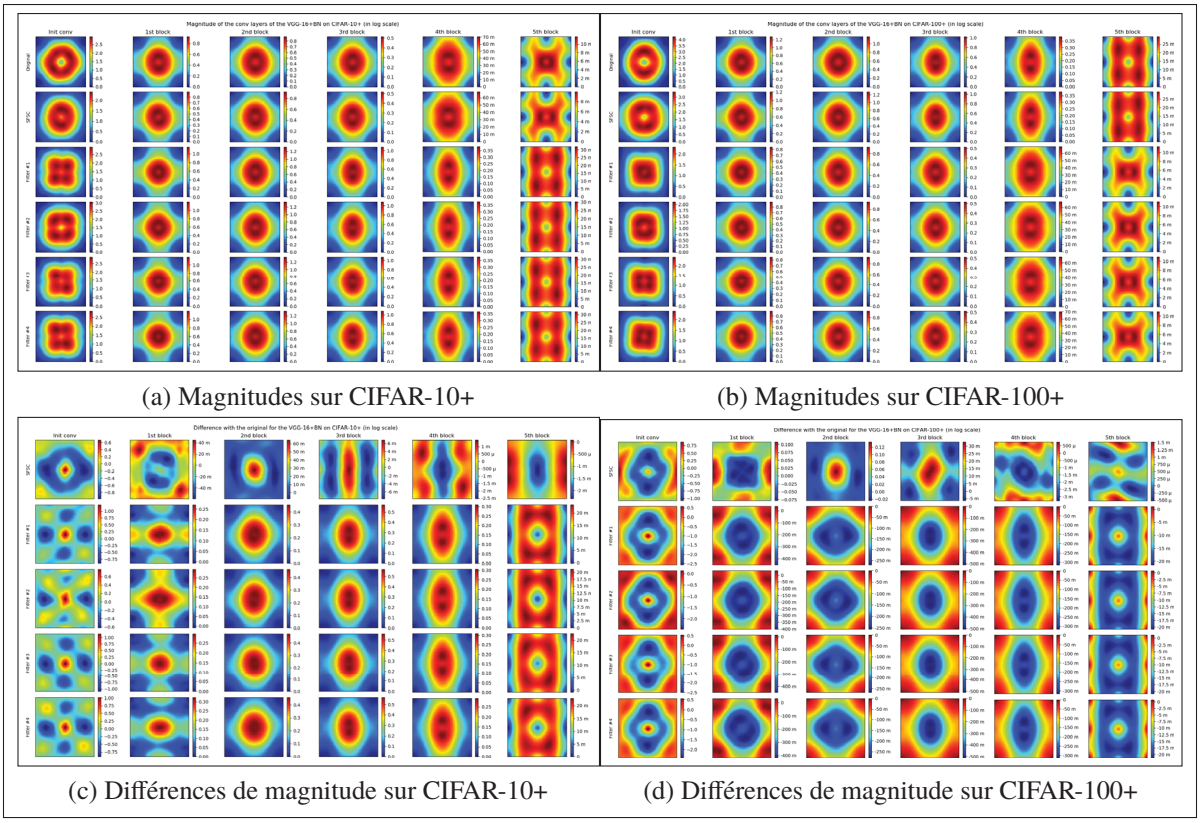
(a) Magnitudes sur CIFAR-10+

(b) Magnitudes sur CIFAR-100+

(c) Différences de magnitude sur CIFAR-10+

(d) Différences de magnitude sur CIFAR-100+

Figure-A I-8 Magnitude en échelle logarithmique des couches de convolution plus profondes du VGG-11 avec BN sur CIFAR avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original



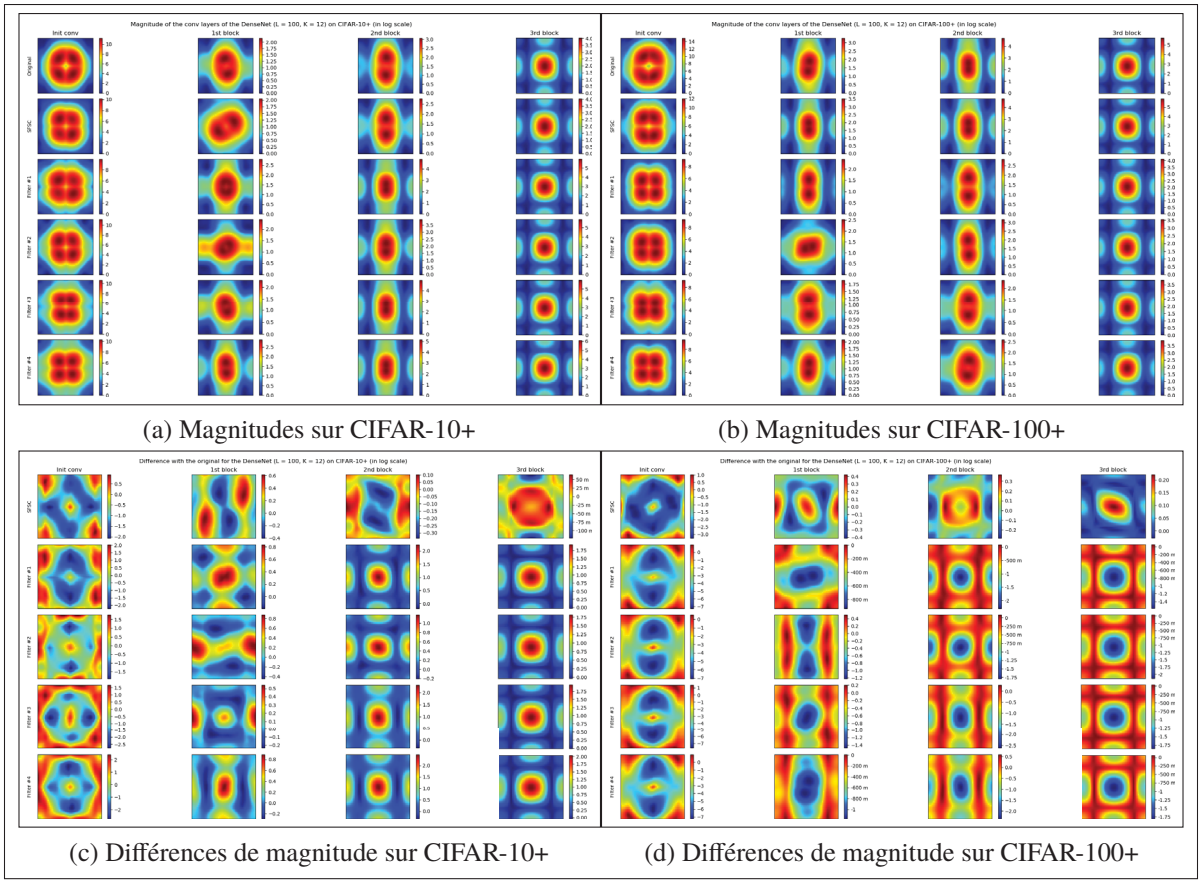
(a) Magnitudes sur CIFAR-10+

(b) Magnitudes sur CIFAR-100+

(c) Différences de magnitude sur CIFAR-10+

(d) Différences de magnitude sur CIFAR-100+

Figure-A I-9 Magnitude en échelle logarithmique des couches de convolution plus profondes du VGG-16 avec BN sur CIFAR avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original



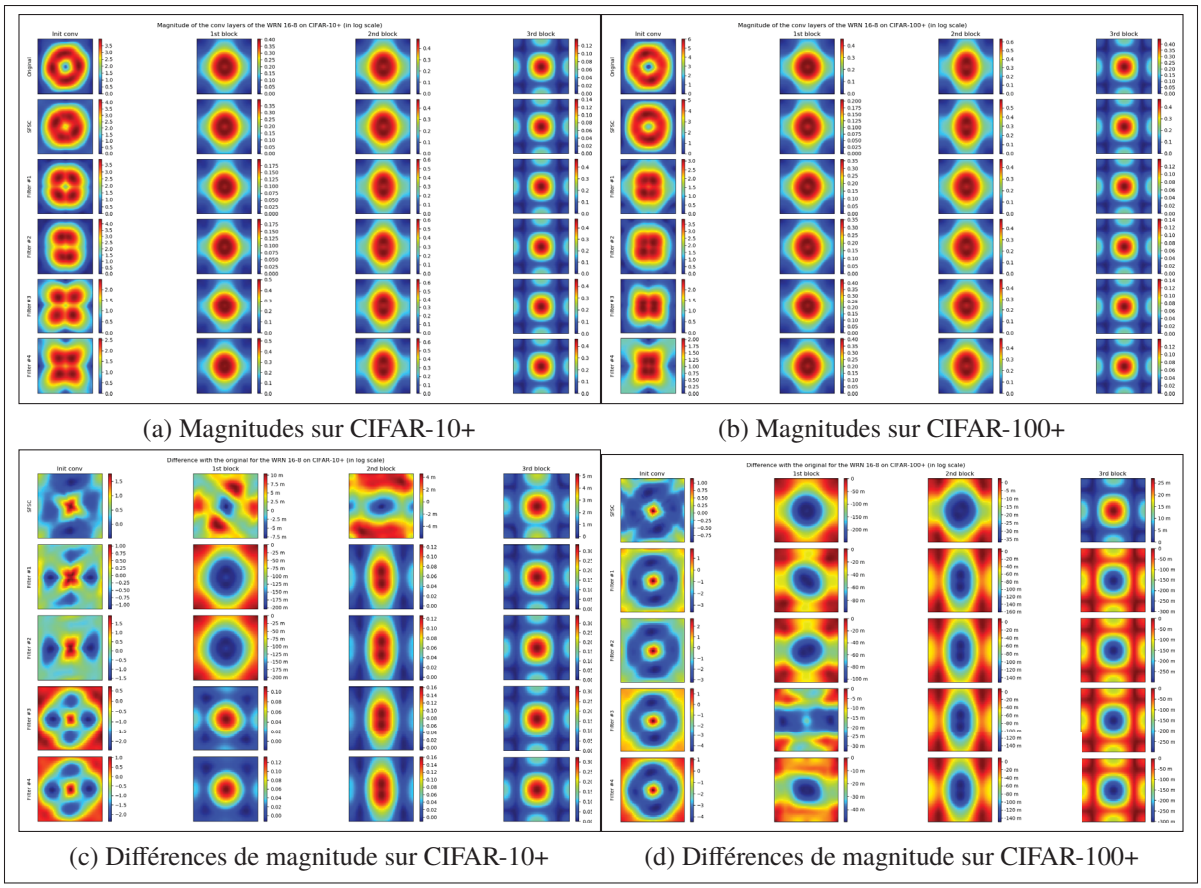
(a) Magnitudes sur CIFAR-10+

(b) Magnitudes sur CIFAR-100+

(c) Différences de magnitude sur CIFAR-10+

(d) Différences de magnitude sur CIFAR-100+

Figure-A I-10 Magnitude en échelle logarithmique des couches de convolution plus profondes du DenseNet (L = 100, K = 12) sur CIFAR avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original



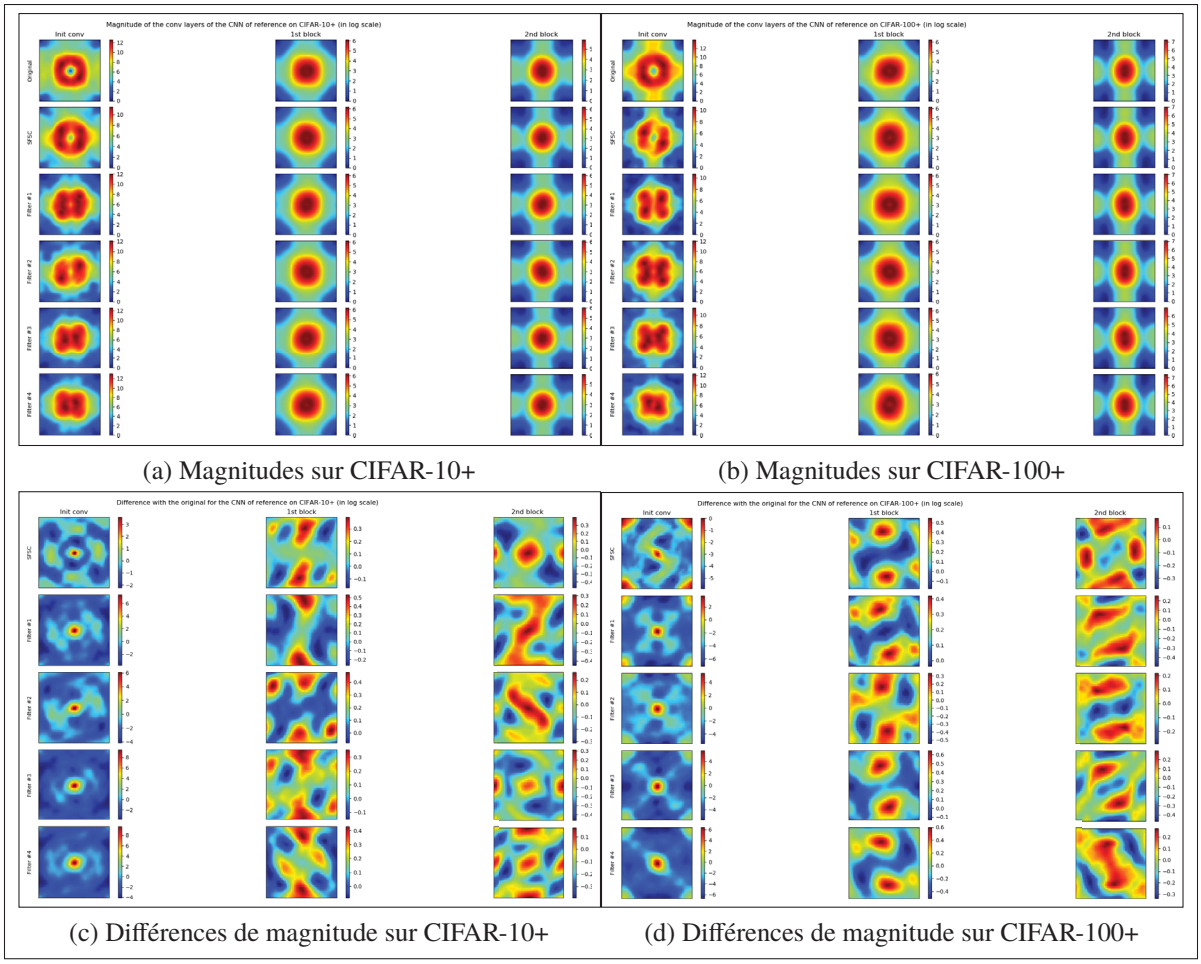
(a) Magnitudes sur CIFAR-10+

(b) Magnitudes sur CIFAR-100+

(c) Différences de magnitude sur CIFAR-10+

(d) Différences de magnitude sur CIFAR-100+

Figure-A I-11 Magnitude en échelle logarithmique des couches de convolution plus profondes du WRN 16-8 sur CIFAR avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original



(a) Magnitudes sur CIFAR-10+

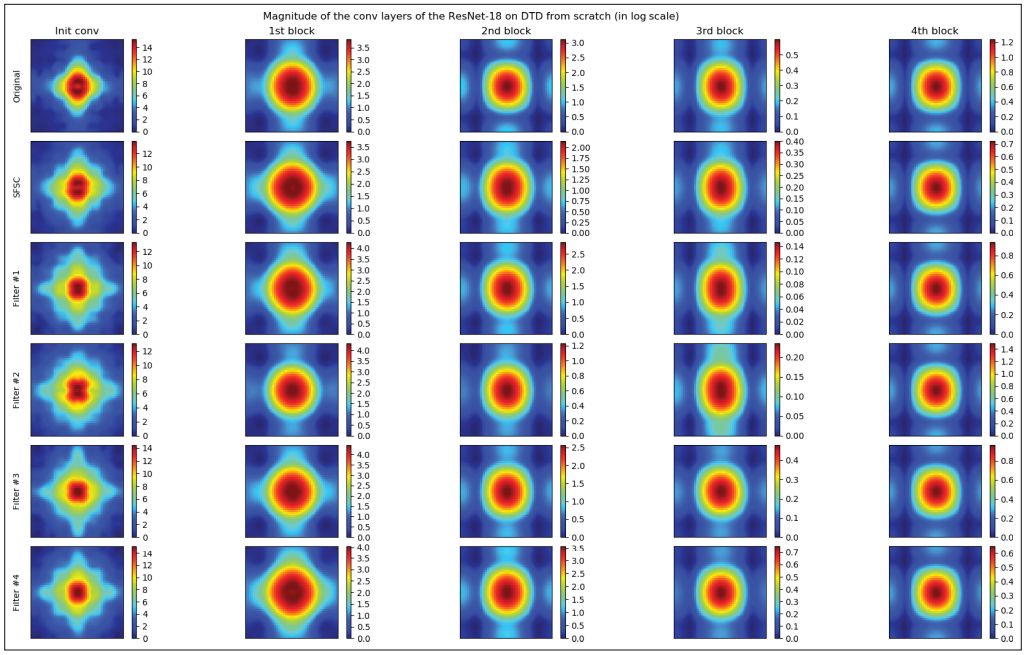
(b) Magnitudes sur CIFAR-100+

(c) Différences de magnitude sur CIFAR-10+

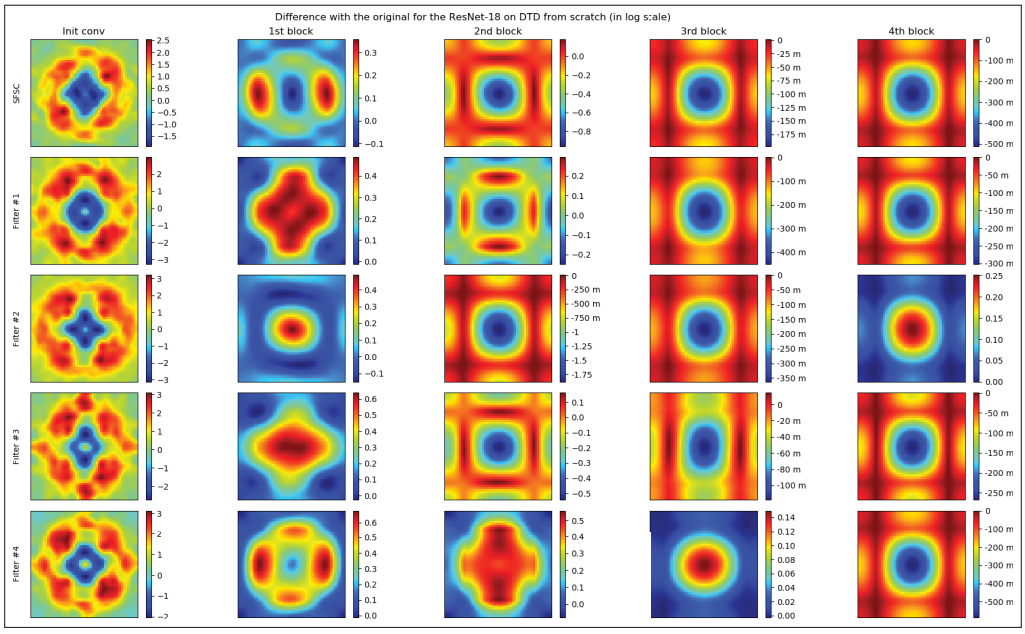
(d) Différences de magnitude sur CIFAR-100+

Figure-A I-12 Magnitude en échelle logarithmique des couches de convolution plus profondes du CNN de référence sur CIFAR avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original

2.2 CNN de DTD *from scratch*

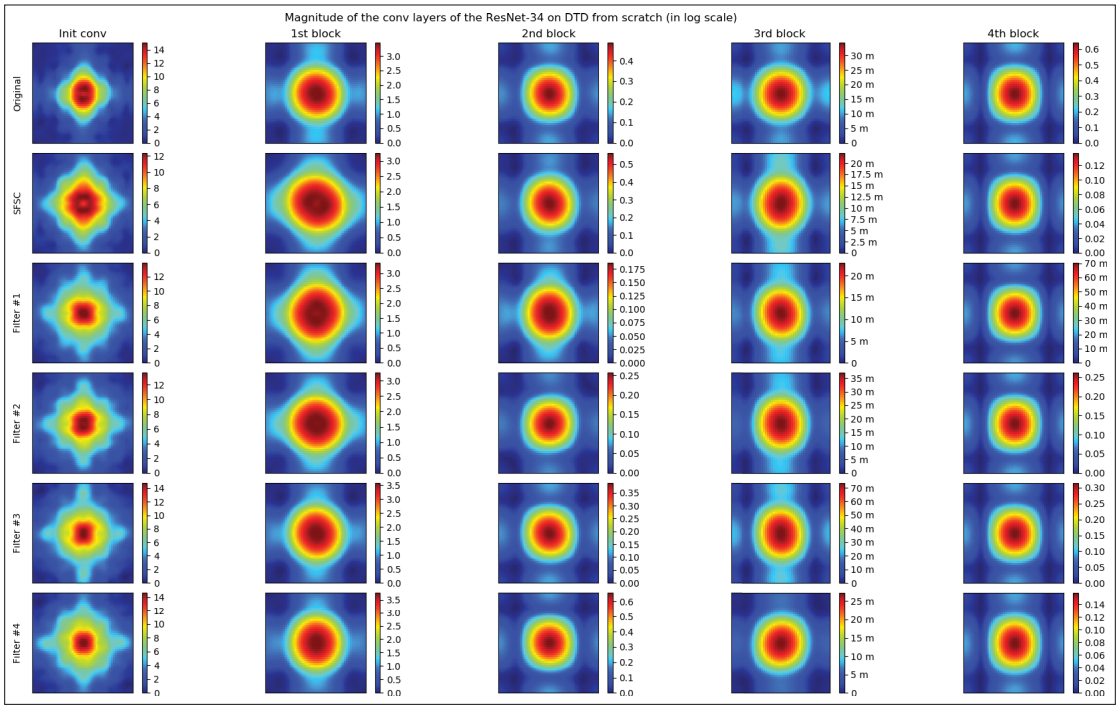


(a) Magnitudes sur DTD *from scratch*

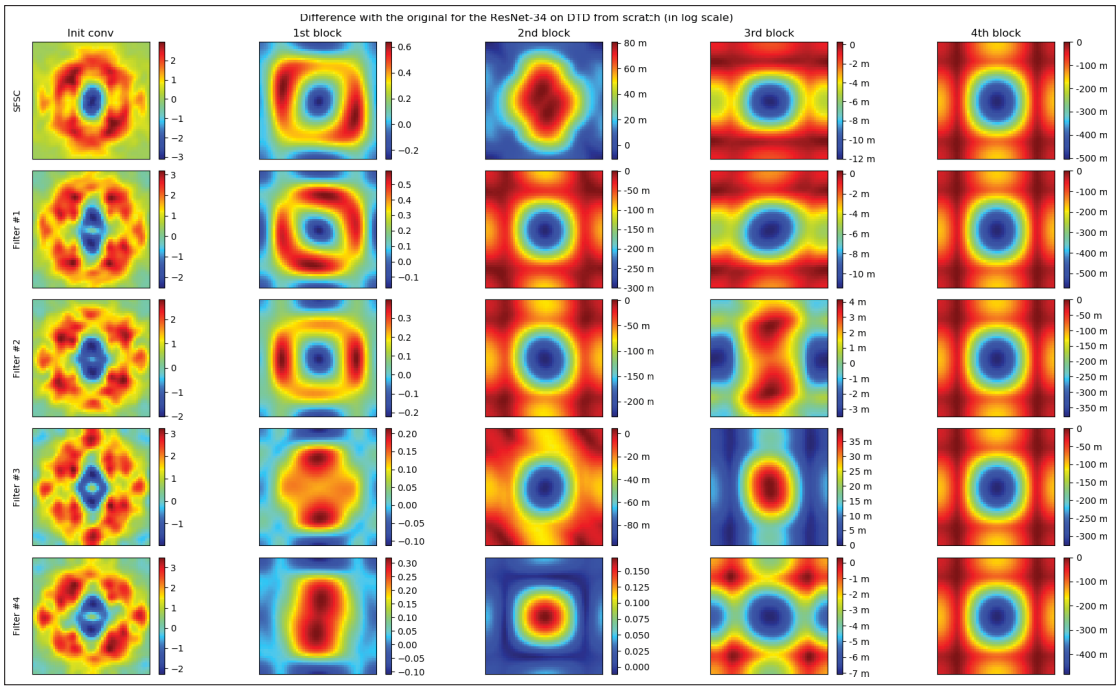


(b) Différences de magnitude sur DTD *from scratch*

Figure-A I-13 Magnitude en échelle logarithmique des couches de convolution plus profondes du ResNet-18 sur DTD *from scratch* avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original

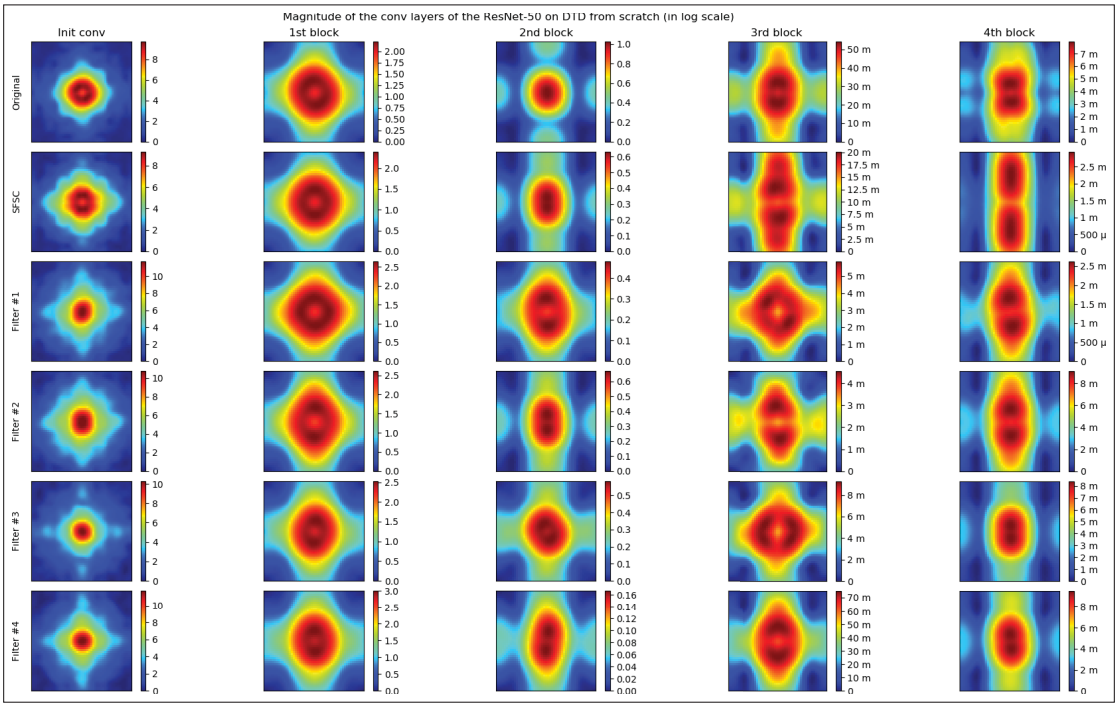


(a) Magnitudes sur DTD *from scratch*

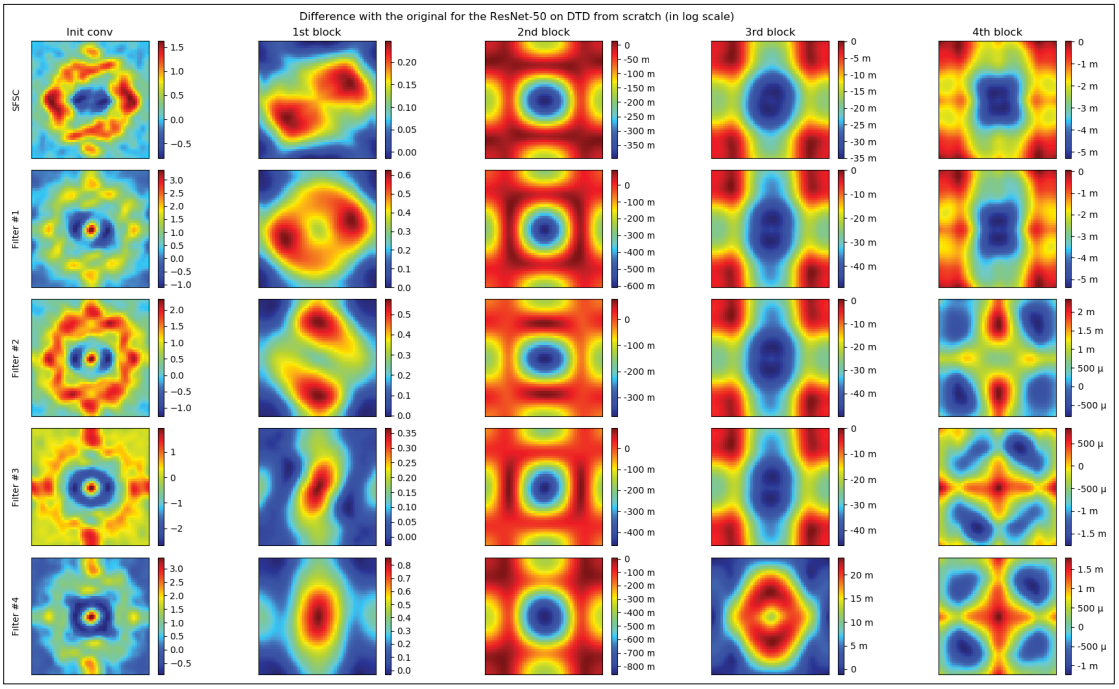


(b) Différences de magnitude sur DTD *from scratch*

Figure-A I-14 Magnitude en échelle logarithmique des couches de convolution plus profondes du ResNet-34 sur DTD *from scratch* avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original

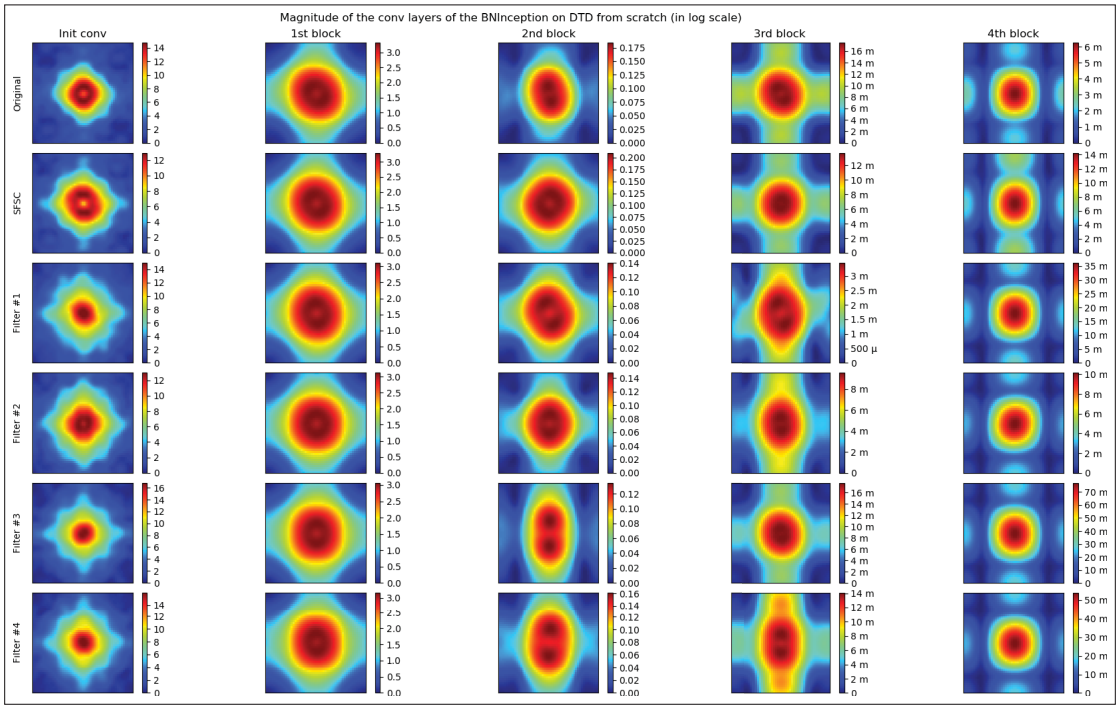


(a) Magnitudes sur DTD *from scratch*

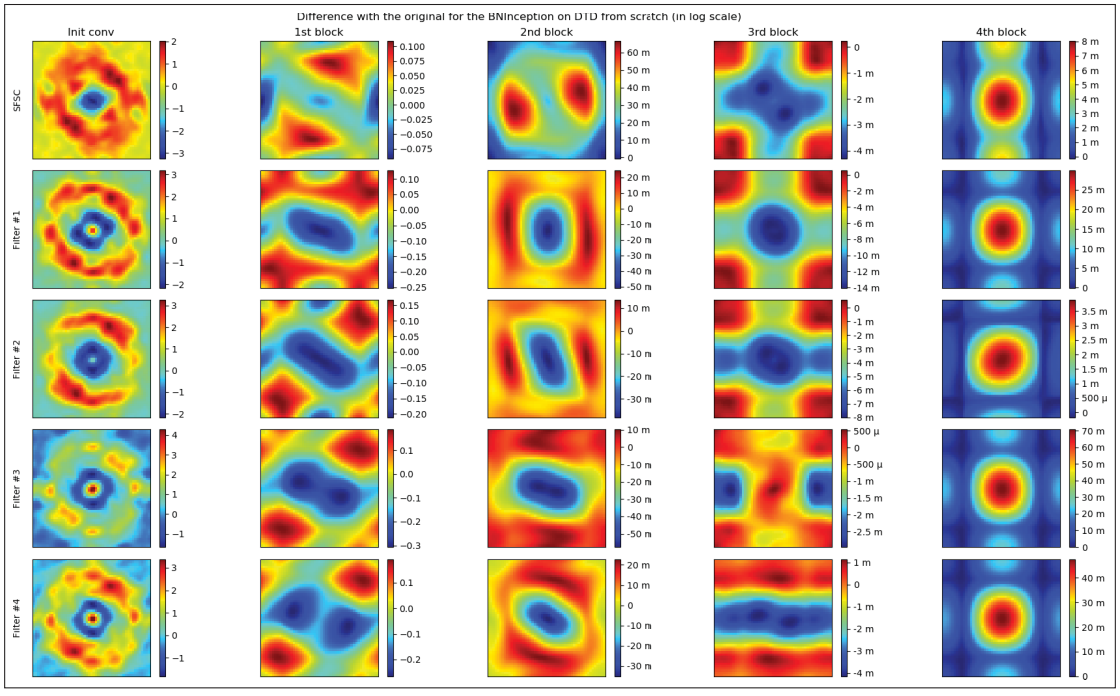


(b) Différences de magnitude sur DTD *from scratch*

Figure-A I-15 Magnitude en échelle logarithmique des couches de convolution plus profondes du ResNet-50 sur DTD *from scratch* avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original

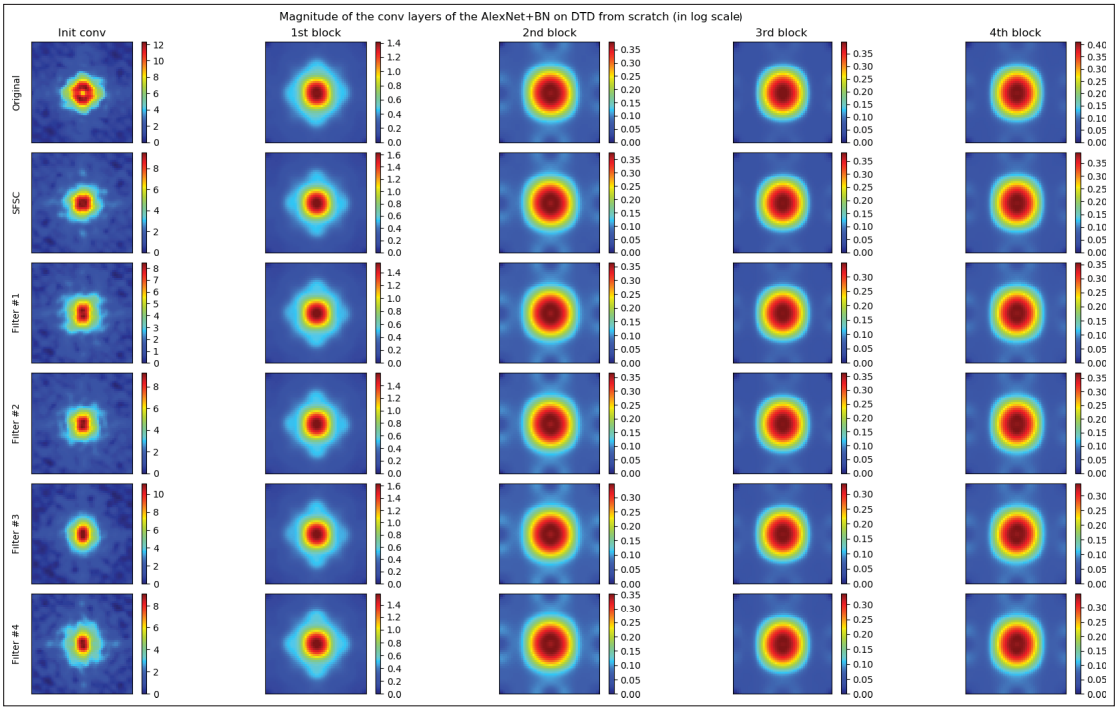


(a) Magnitudes sur DTD *from scratch*

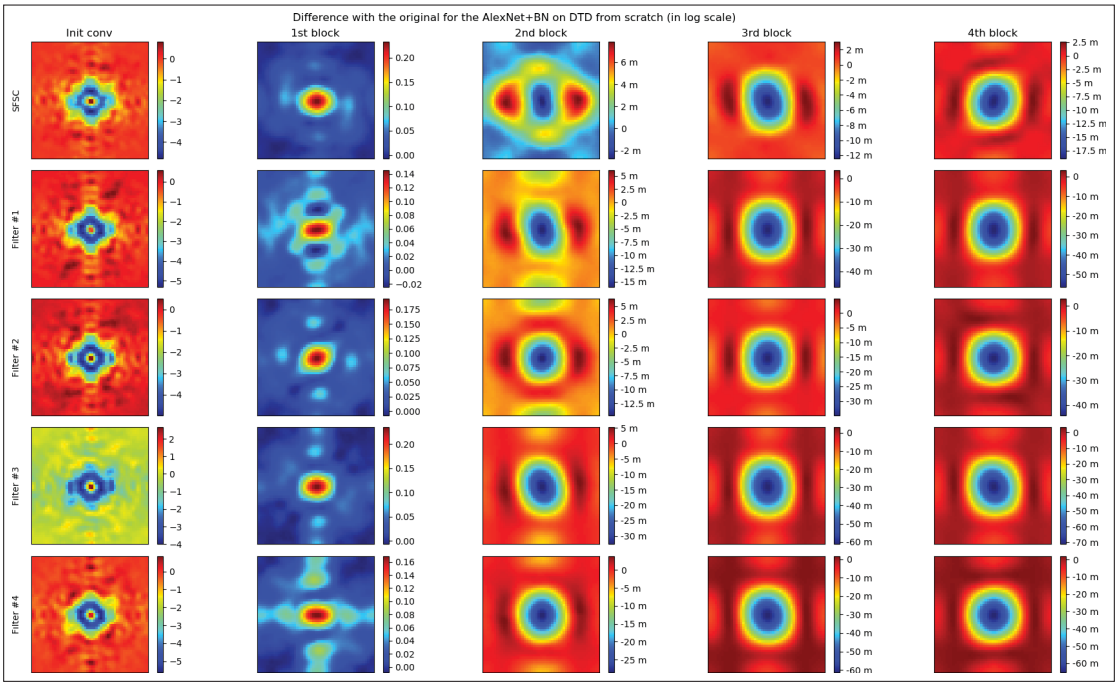


(b) Différences de magnitude sur DTD *from scratch*

Figure-A I-16 Magnitude en échelle logarithmique des couches de convolution plus profondes du BNInception sur DTD *from scratch* avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original

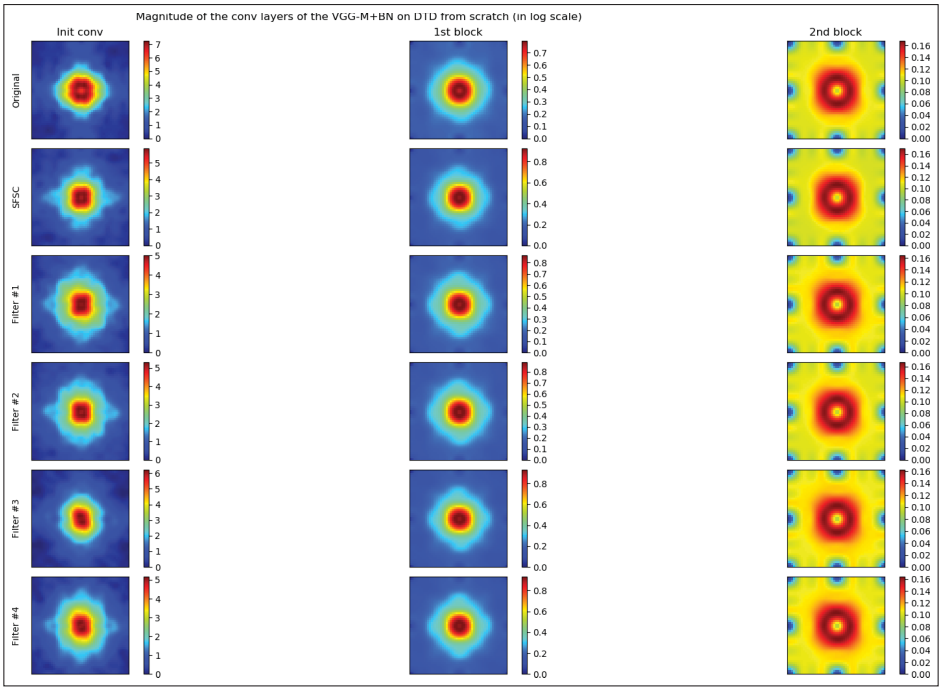


(a) Magnitudes sur DTD *from scratch*

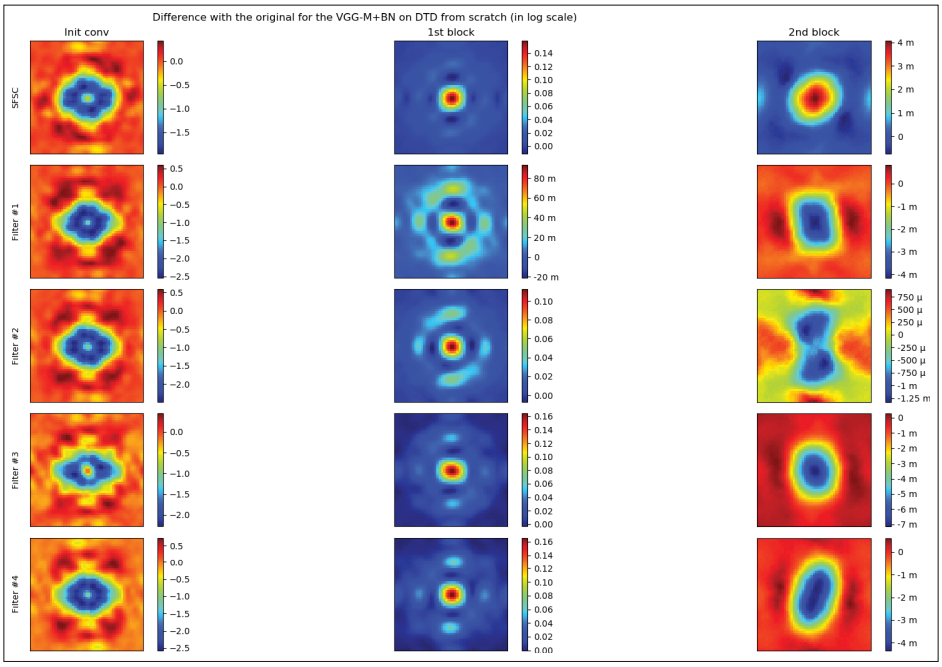


(b) Différences de magnitude sur DTD *from scratch*

Figure-A I-17 Magnitude en échelle logarithmique des couches de convolution plus profondes du AlexNet avec BN sur DTD *from scratch* avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original

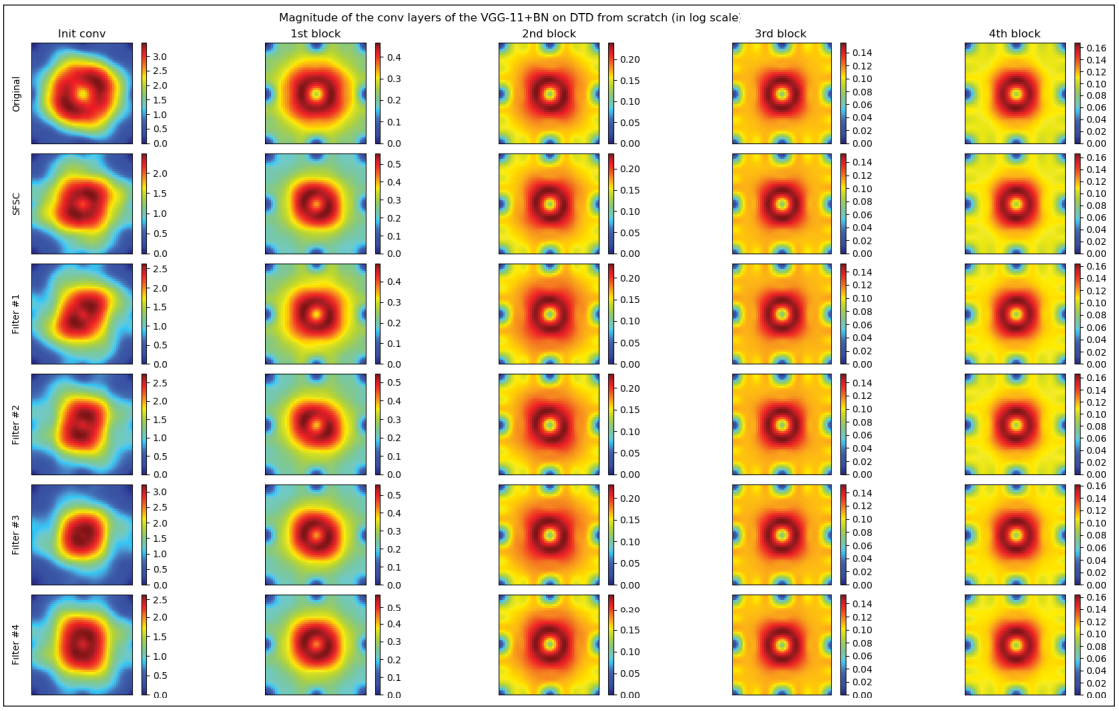


(a) Magnitudes sur DTD *from scratch*

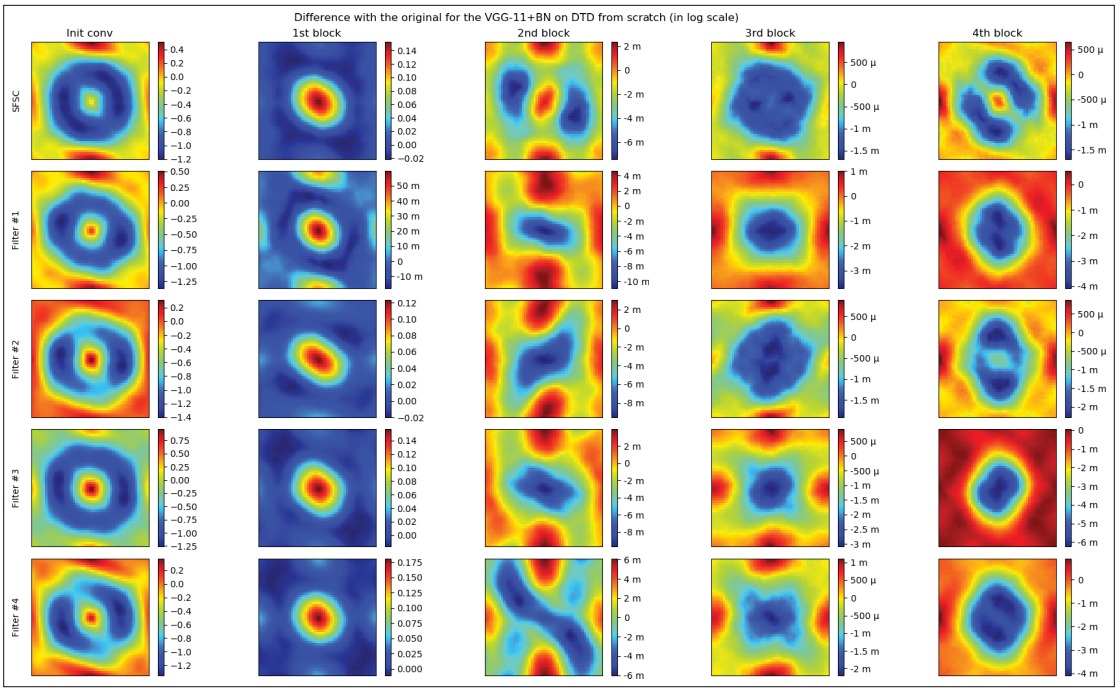


(b) Différences de magnitude sur DTD *from scratch*

Figure-A I-18 Magnitude en échelle logarithmique des couches de convolution plus profondes du VGG-M avec BN sur DTD *from scratch* avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original

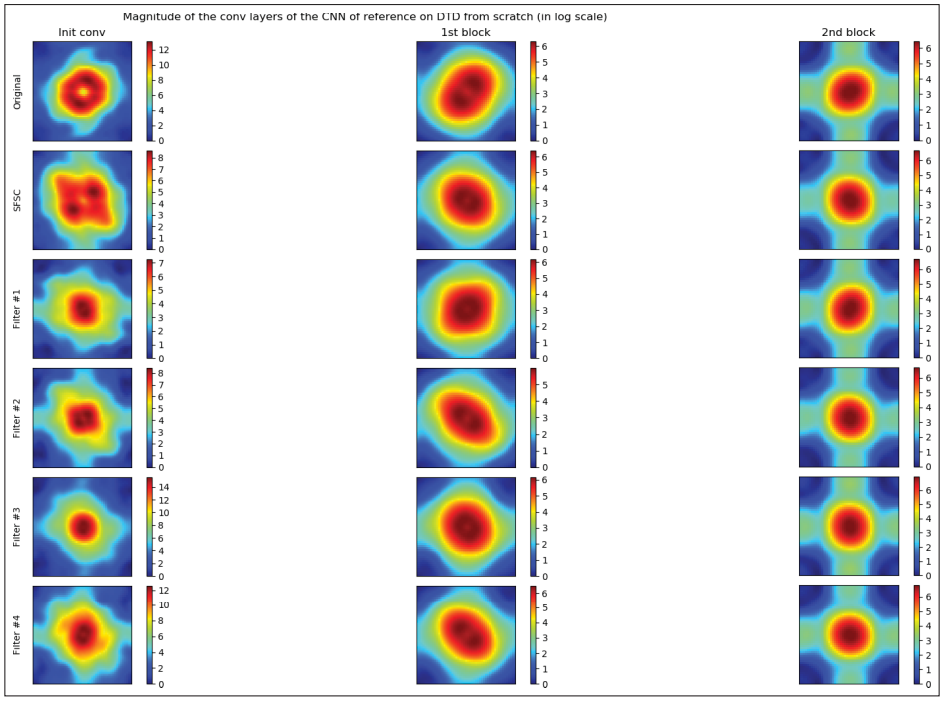


(a) Magnitudes sur DTD *from scratch*

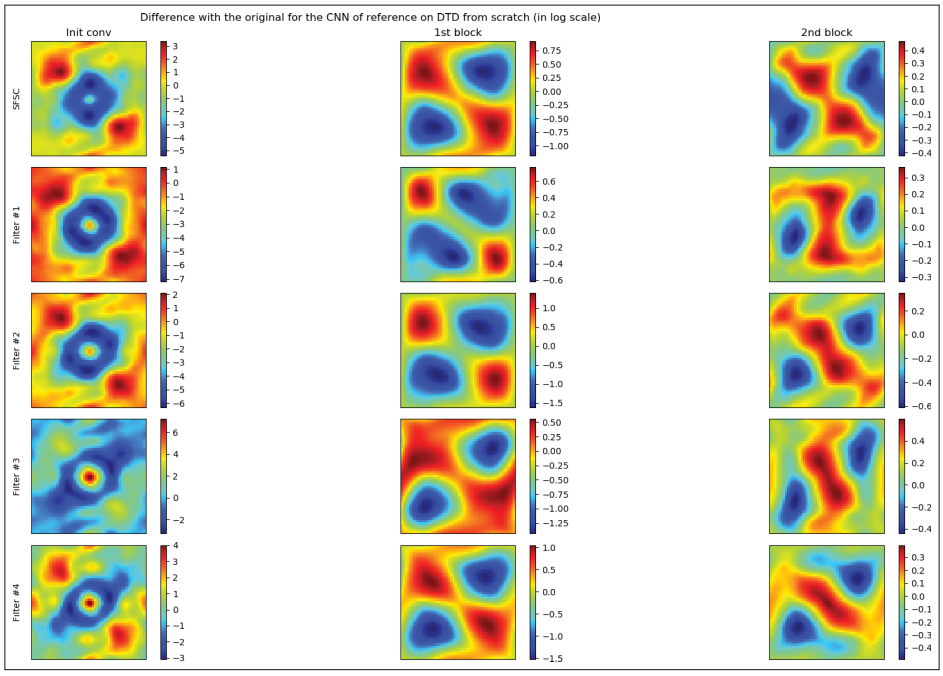


(b) Différences de magnitude sur DTD *from scratch*

Figure-A I-19 Magnitude en échelle logarithmique des couches de convolution plus profondes du VGG-11 avec BN sur DTD *from scratch* avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original



(a) Magnitudes sur DTD *from scratch*



(b) Différences de magnitude sur DTD *from scratch*

Figure-A I-20 Magnitude en échelle logarithmique des couches de convolution plus profondes du CNN de référence sur DTD *from scratch* avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original

2.3 CNN pré-entraînés sur ImageNet utilisés sur DTD

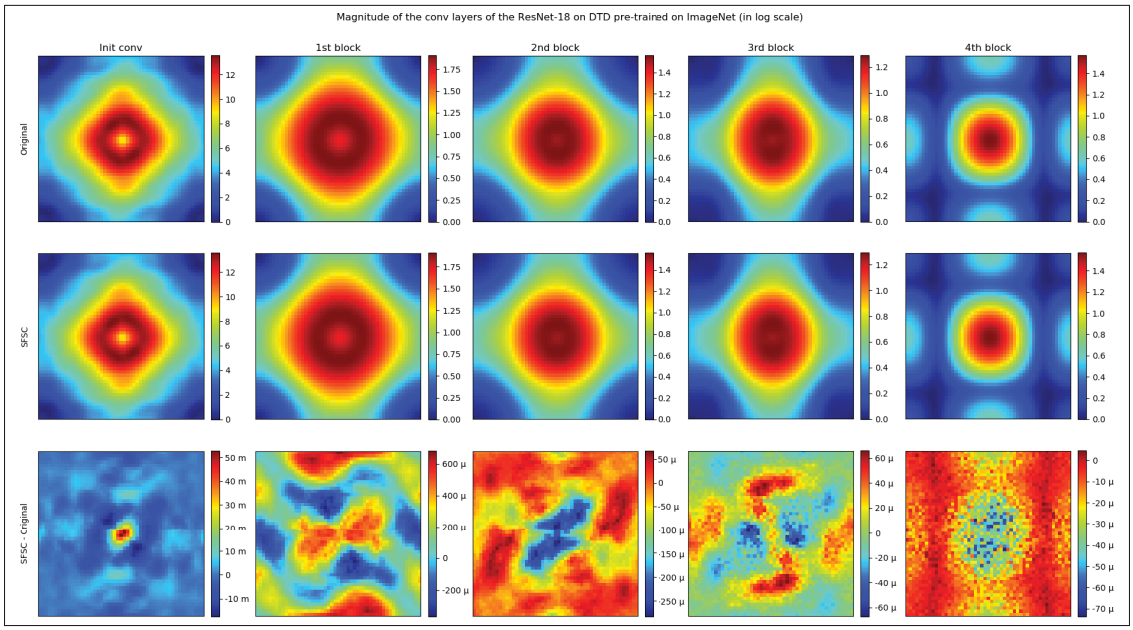


Figure-A I-21 Magnitude des couches de convolutions plus profondes du ResNet-18 pré-entraîné sur ImageNet utilisé sur DTD avec et sans la SFSC et différences de magnitude entre ceux-ci

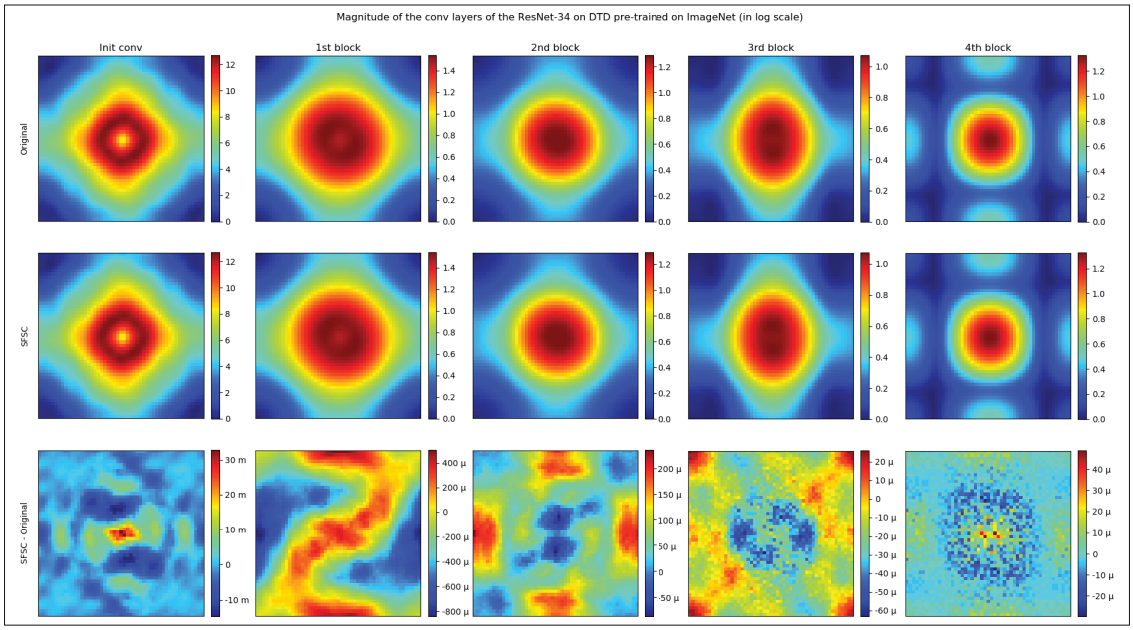


Figure-A I-22 Magnitude des couches de convolutions plus profondes du ResNet-34 pré-entraîné sur ImageNet utilisé sur DTD avec et sans la SFSC et différences de magnitude entre ceux-ci

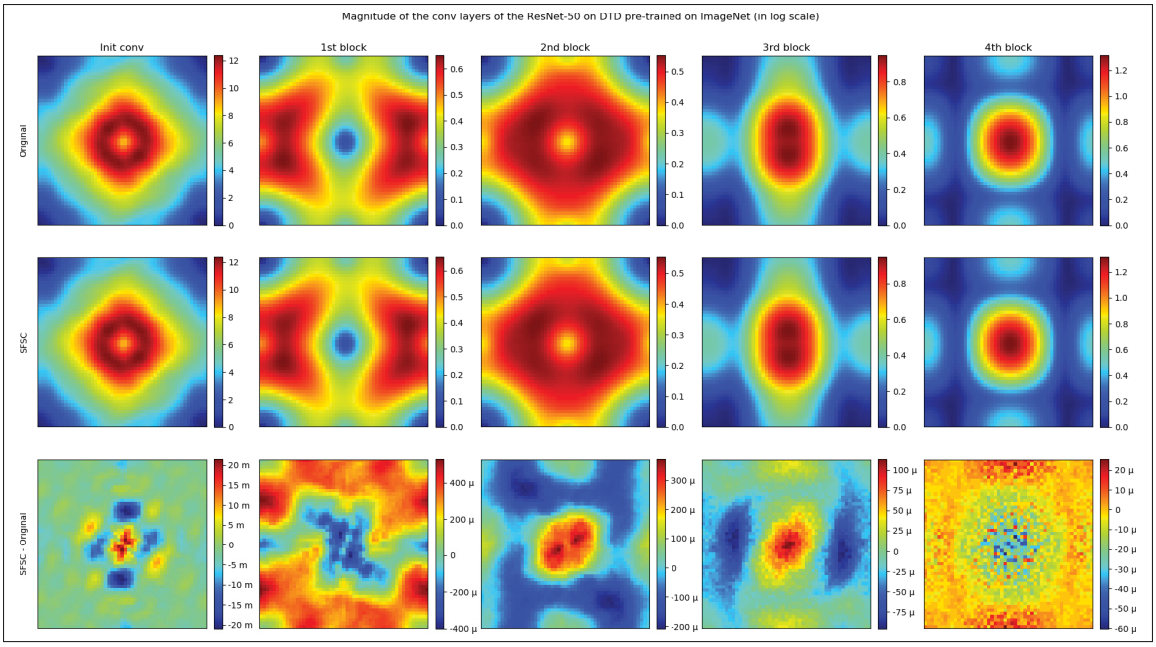


Figure-A I-23 Magnitude des couches de convolutions plus profondes du ResNet-50 pré-entraîné sur ImageNet utilisé sur DTD avec et sans la SFSC et différences de magnitude entre ceux-ci

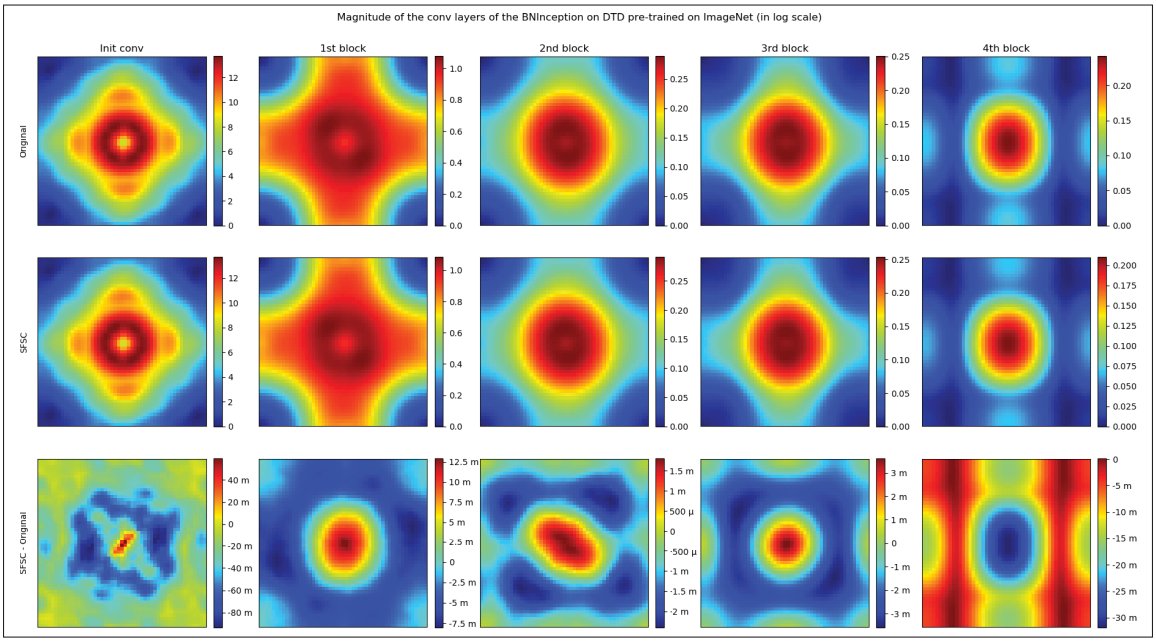


Figure-A I-24 Magnitude des couches de convolutions plus profondes du BNInception pré-entraîné sur ImageNet utilisé sur DTD avec et sans la SFSC et différences de magnitude entre ceux-ci

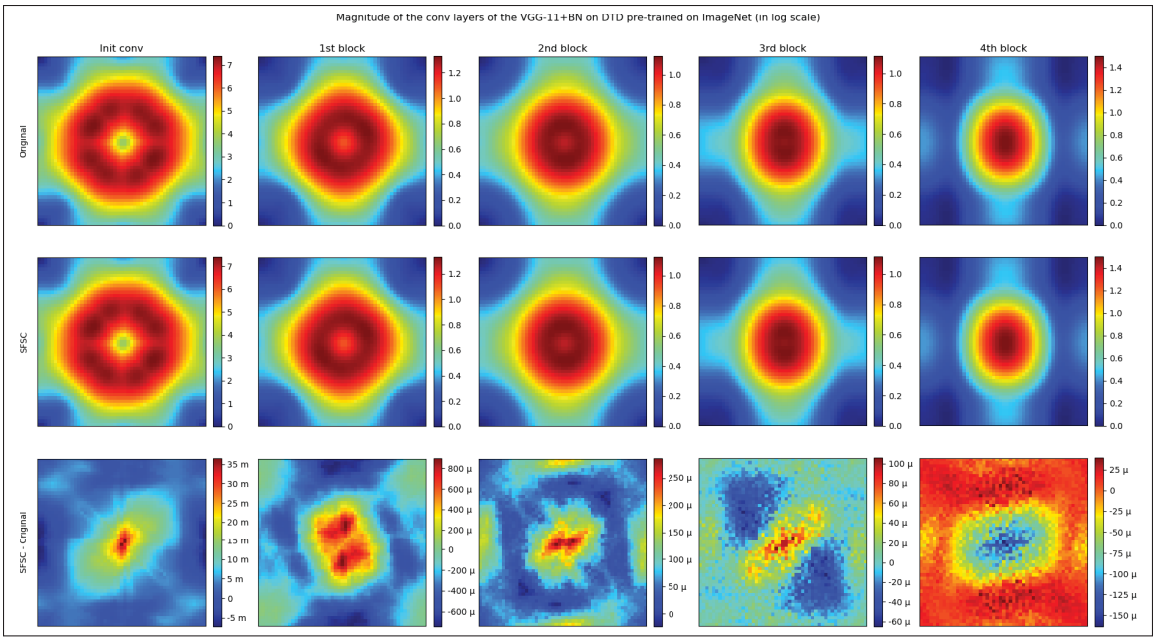
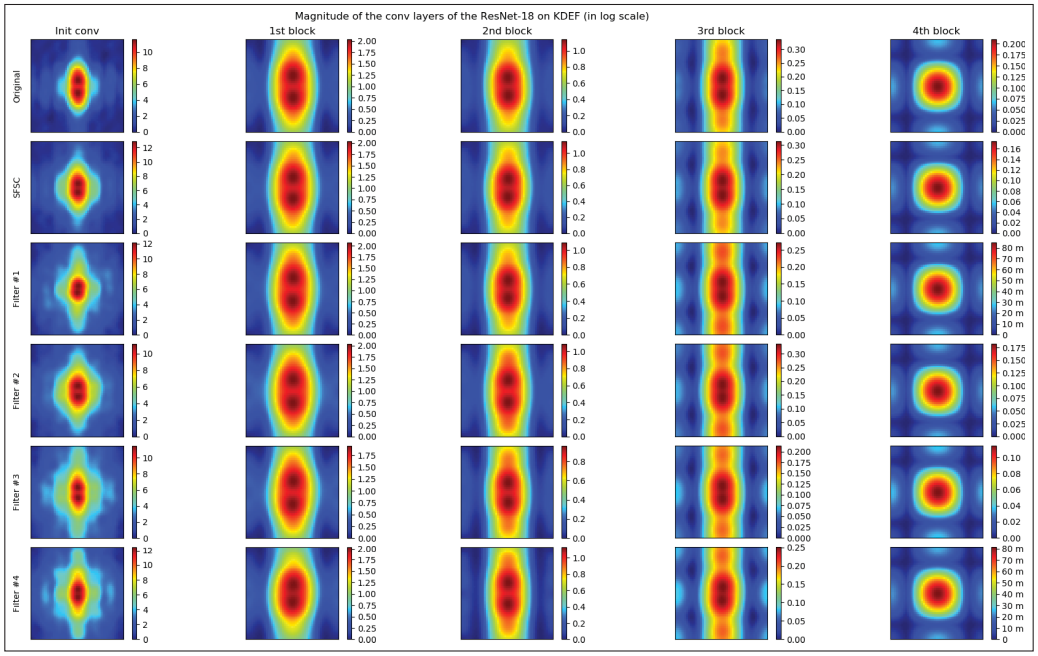
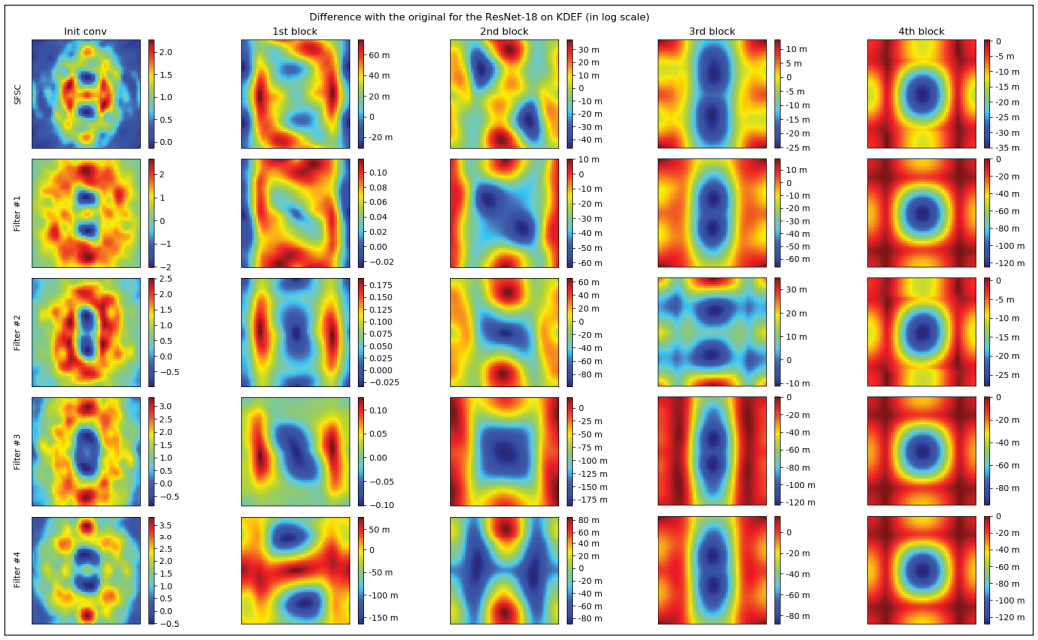


Figure-A I-25 Magnitude des couches de convolutions plus profondes du VGG-11 avec BN pré-entraîné sur ImageNet utilisé sur DTD avec et sans la SFSC et différences de magnitude entre ceux-ci

2.4 CNN de KDEF

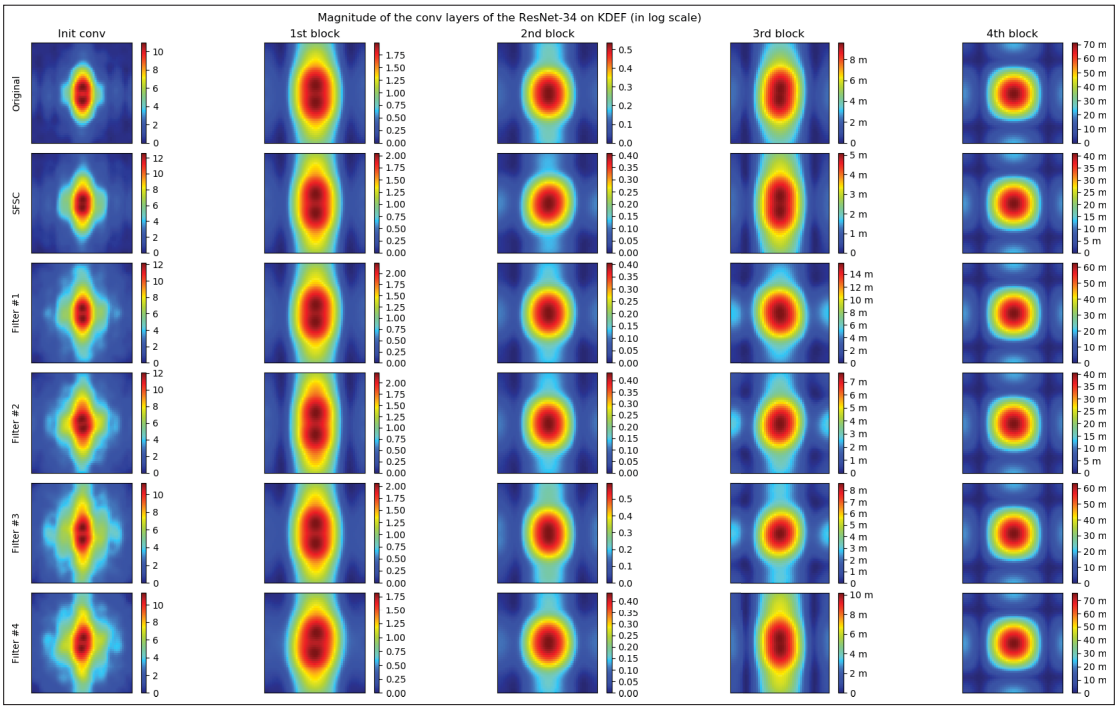


(a) Magnitudes sur KDEF

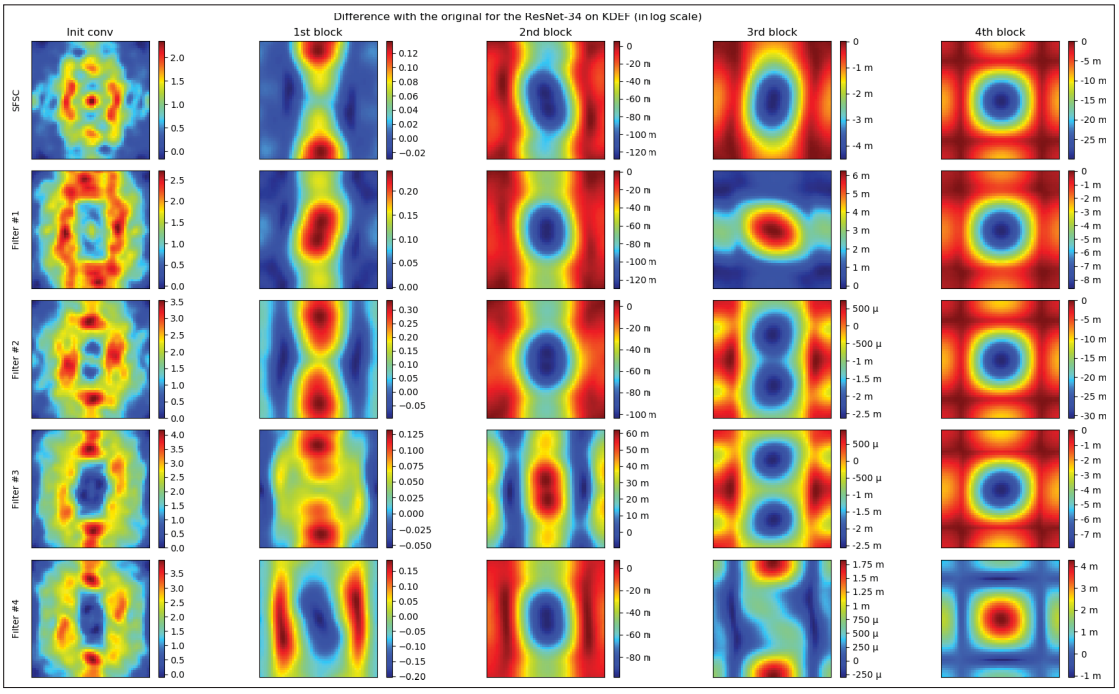


(b) Différences de magnitude sur KDEF

Figure-A I-26 Magnitude en échelle logarithmique des couches de convolution plus profondes du ResNet-18 sur KDEF avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original

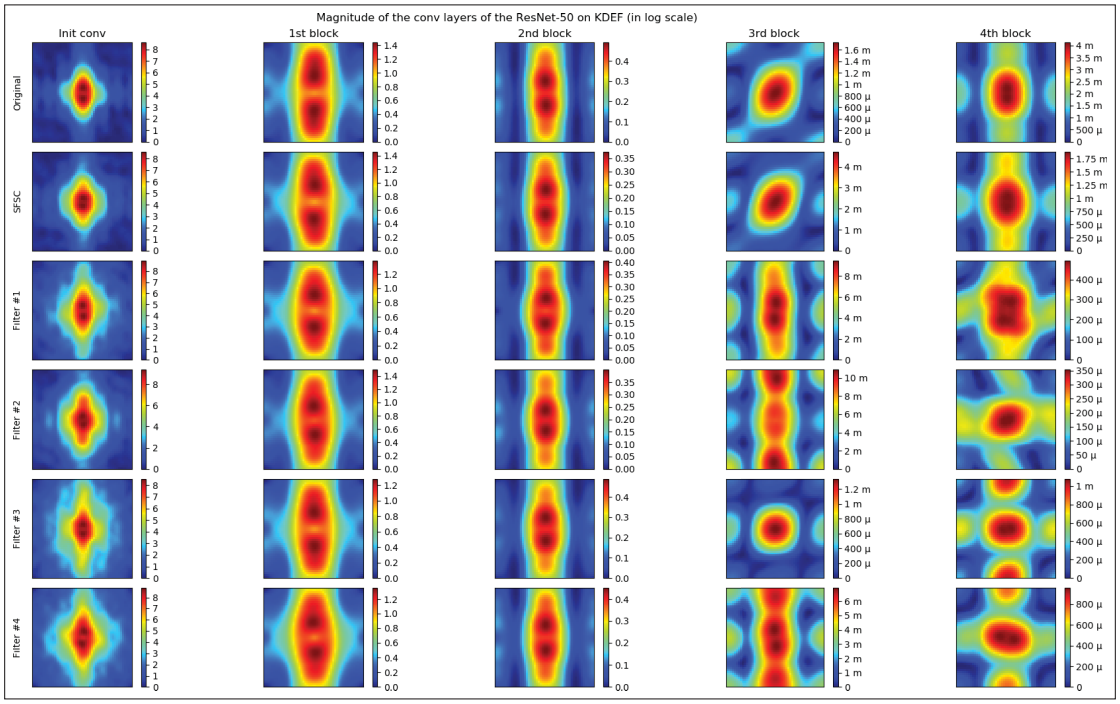


(a) Magnitudes sur KDEF

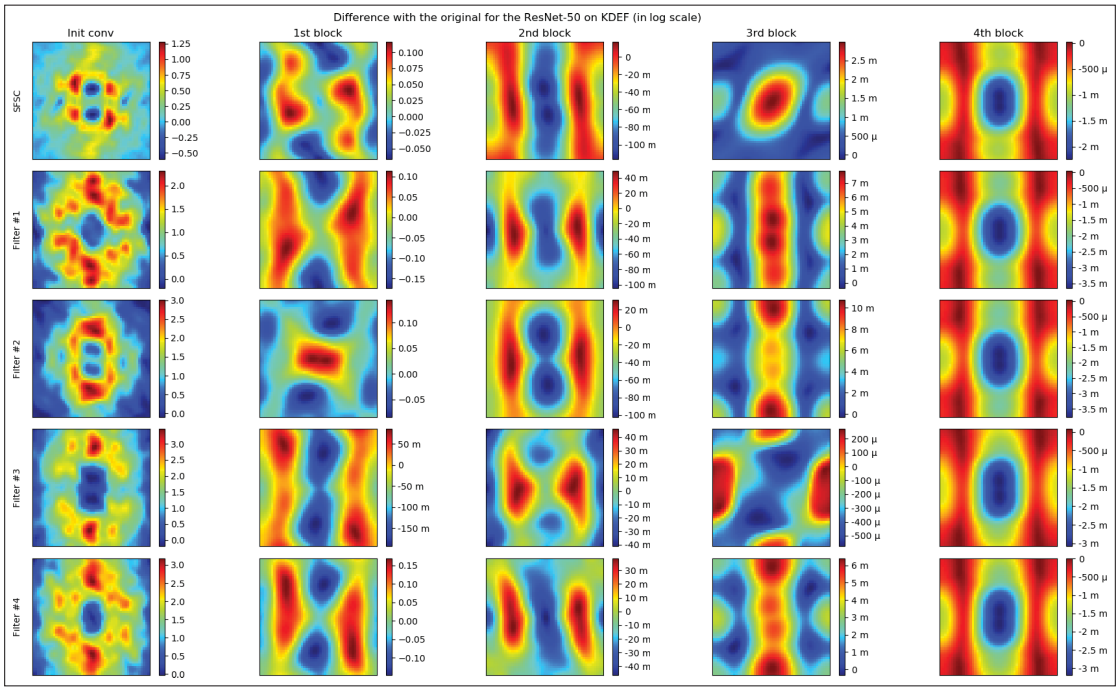


(b) Différences de magnitude sur KDEF

Figure-A I-27 Magnitude en échelle logarithmique des couches de convolution plus profondes du ResNet-34 sur KDEF avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original

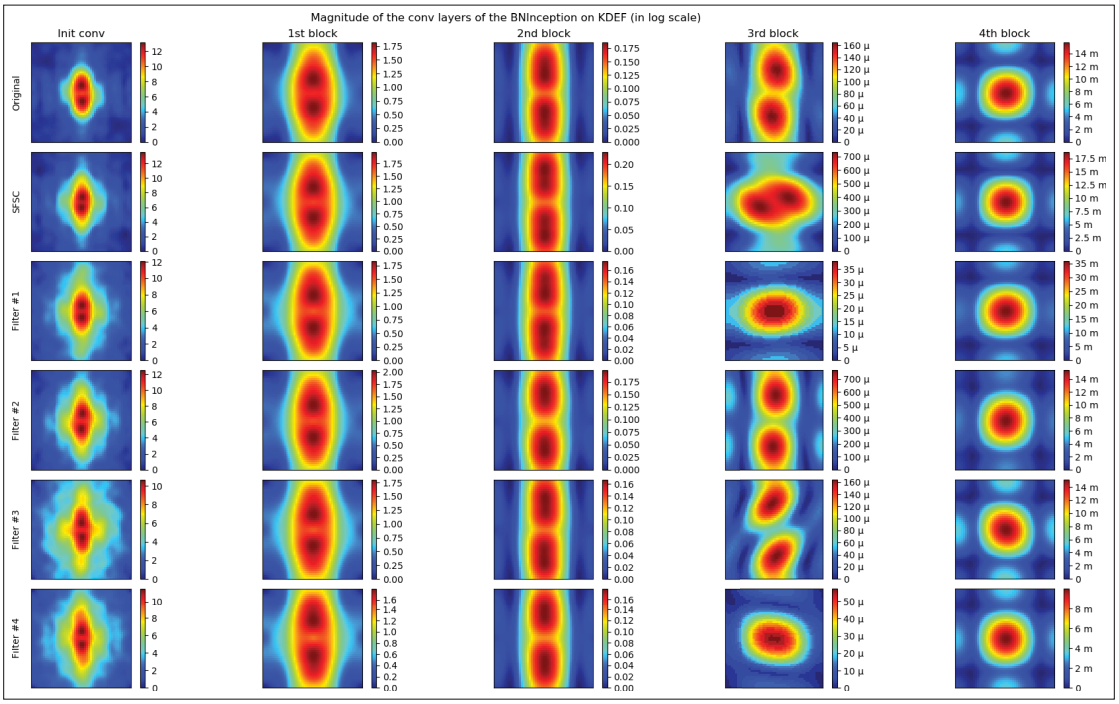


(a) Magnitudes sur KDEF

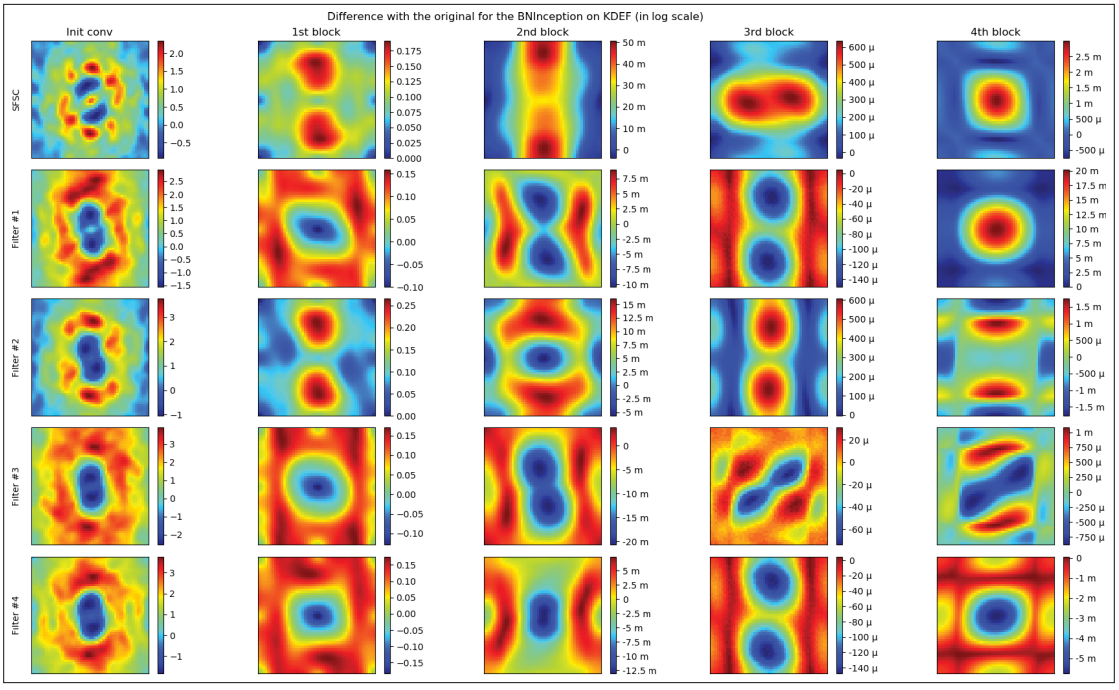


(b) Différences de magnitude sur KDEF

Figure-A I-28 Magnitude en échelle logarithmique des couches de convolution plus profondes du ResNet-50 sur KDEF avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original

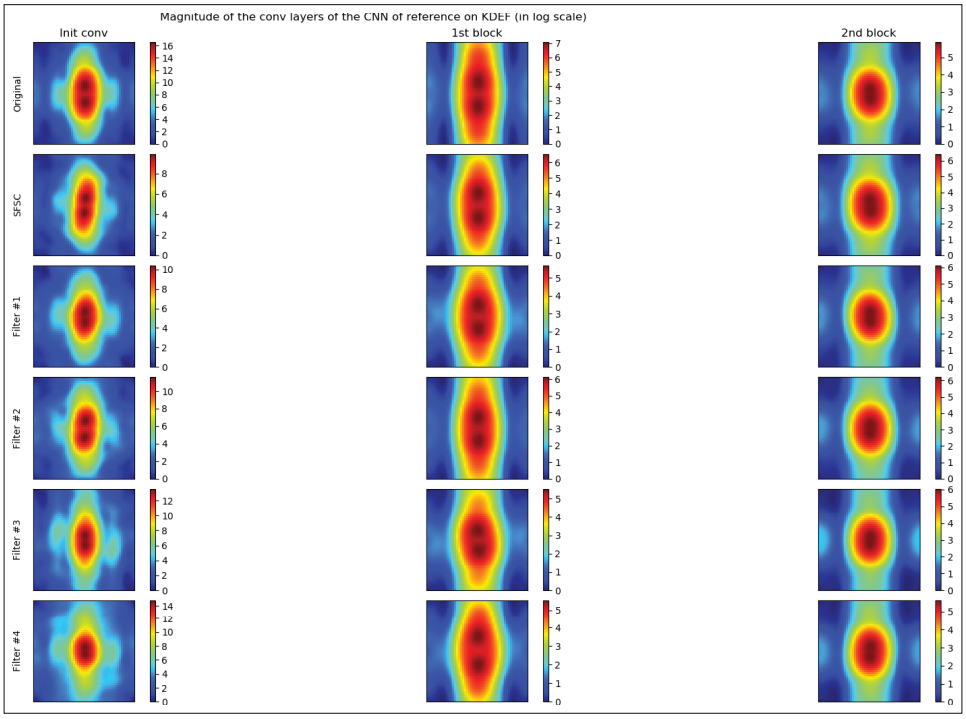


(a) Magnitudes sur KDEF

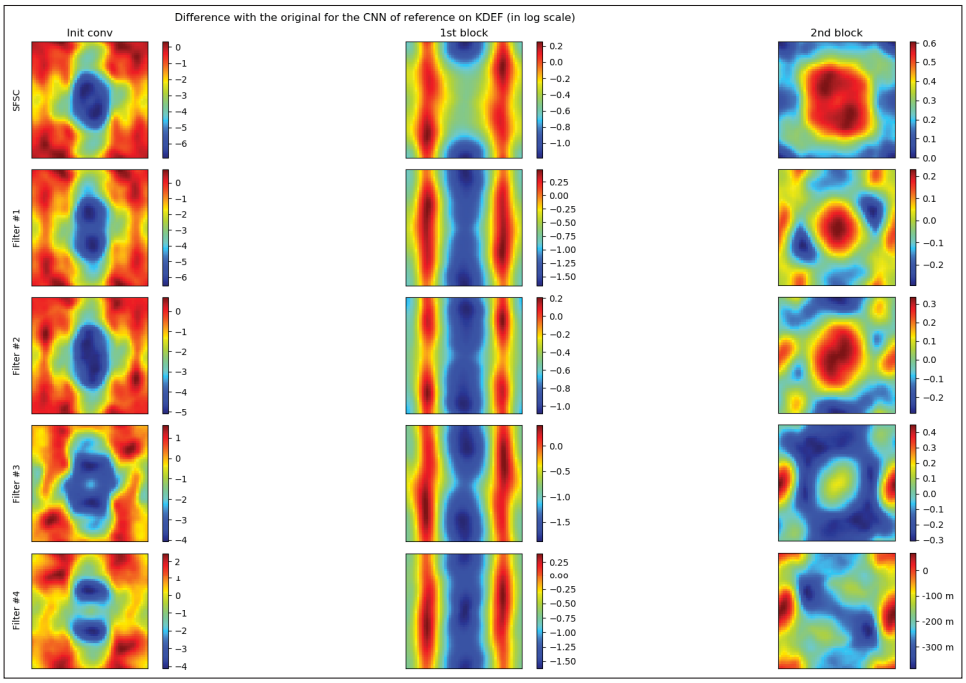


(b) Différences de magnitude sur KDEF

Figure-A I-29 Magnitude en échelle logarithmique des couches de convolution plus profondes du BNInception sur KDEF avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original



(a) Magnitudes sur KDEF



(b) Différences de magnitude sur KDEF

Figure-A I-30 Magnitude en échelle logarithmique des couches de convolution plus profondes du CNN de référence sur KDEF avec et sans nos méthodes et différences de magnitude entre nos méthodes et le modèle original

3. Représentation fréquentielle détaillée des premières couches de convolution

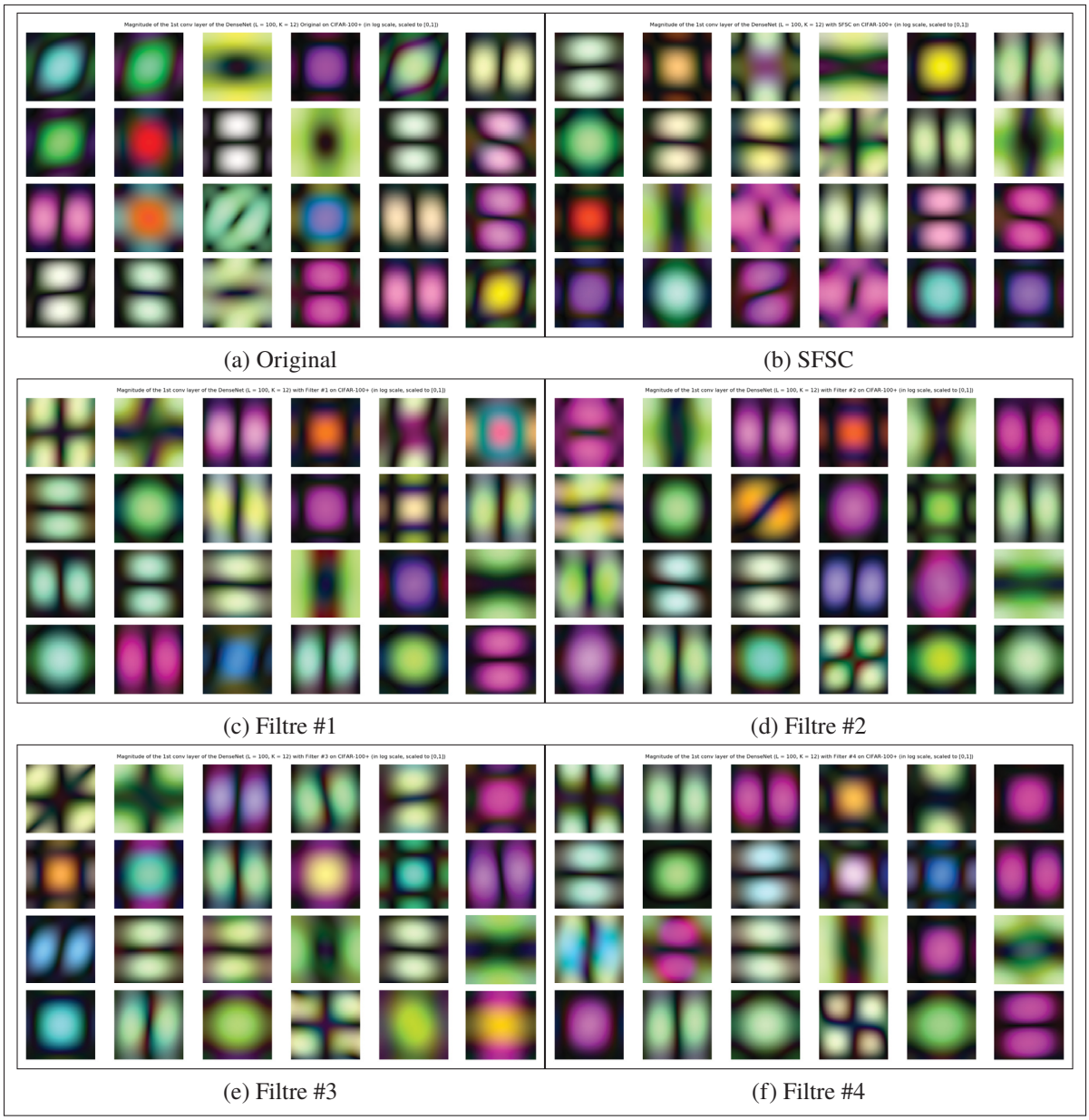
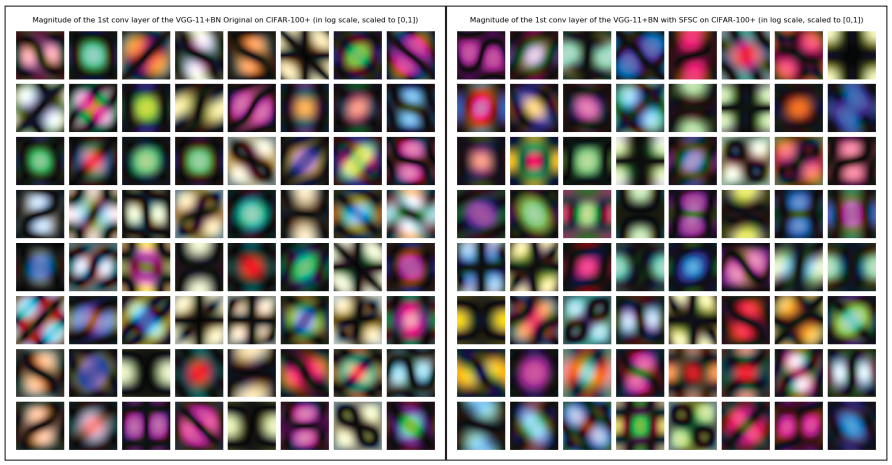
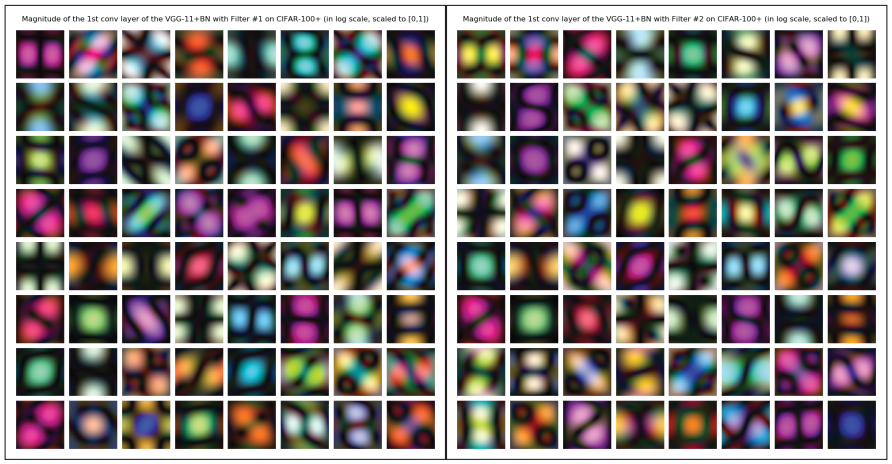


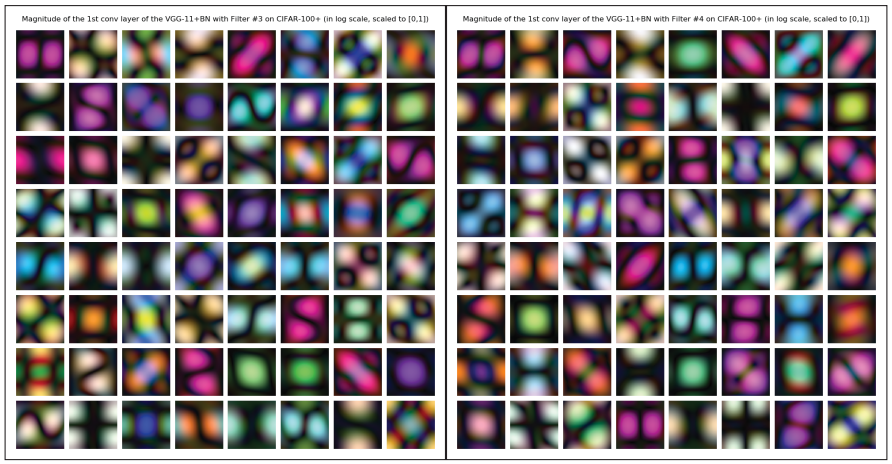
Figure-A I-31 Magnitude en échelle logarithmique de chacun des filtres de la première couche de convolution du DenseNet (L= 100, K = 12) sur CIFAR-100+ selon les différentes méthodes



(a) Original (b) SFSC

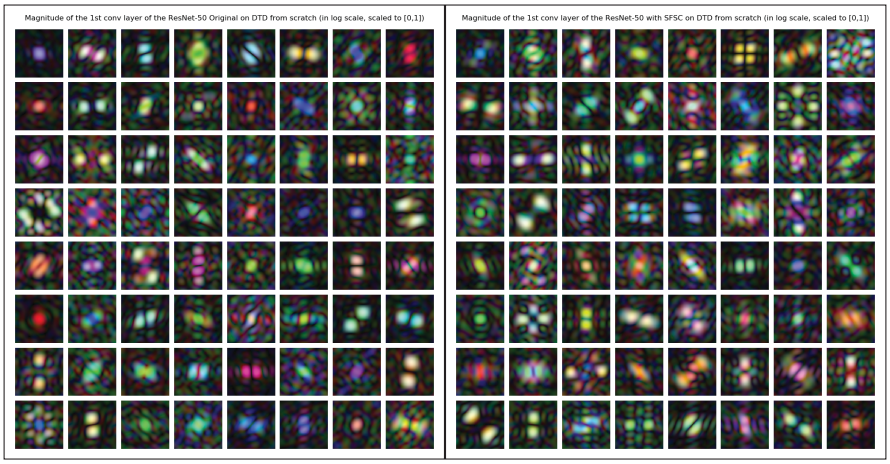


(c) Filtre #1 (d) Filtre #2

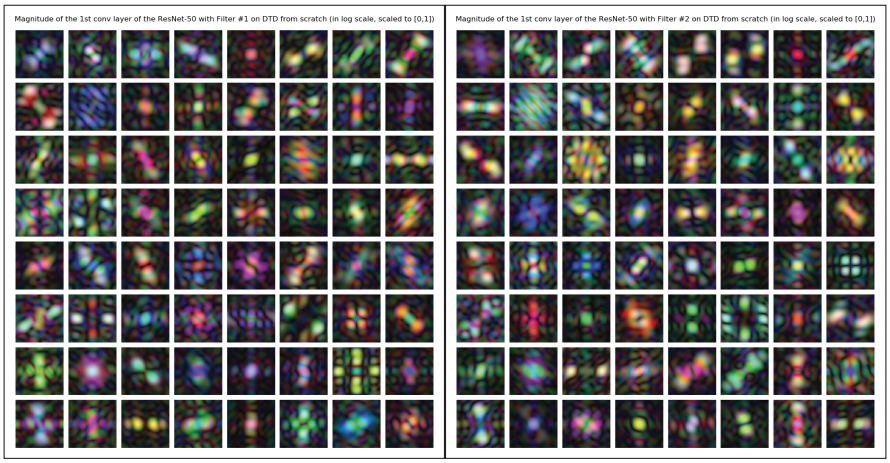


(e) Filtre #3 (f) Filtre #4

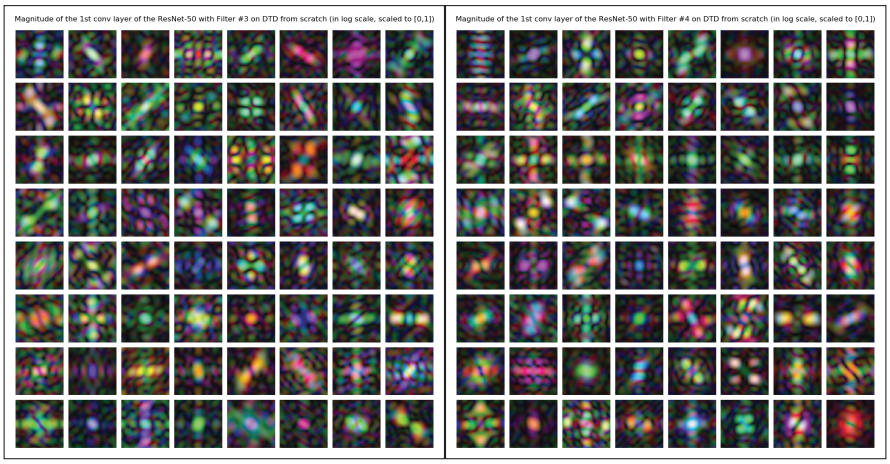
Figure-A I-32 Magnitude en échelle logarithmique de chacun des filtres de la première couche de convolution du VGG-11 avec BN sur CIFAR-100+ selon les différentes méthodes



(a) Original (b) SFSC

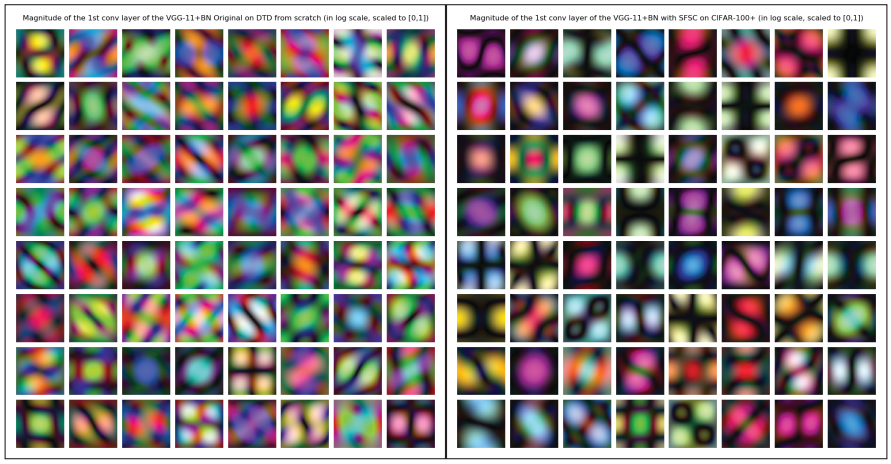


(c) Filtre #1 (d) Filtre #2

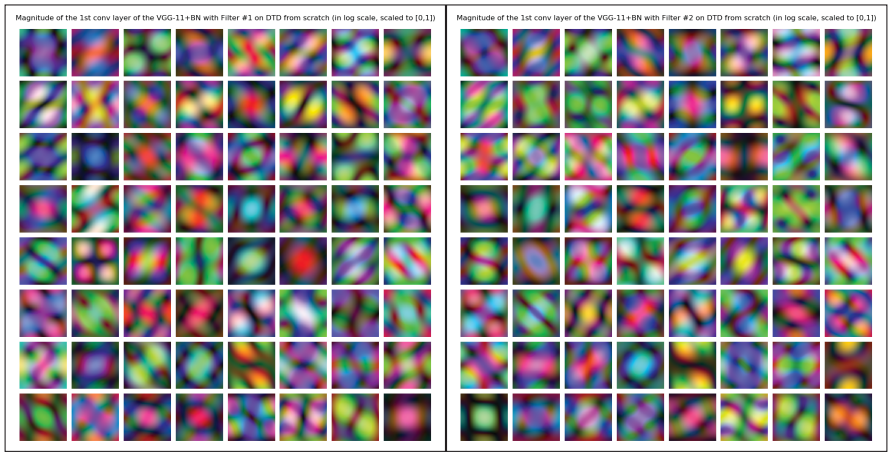


(e) Filtre #3 (f) Filtre #4

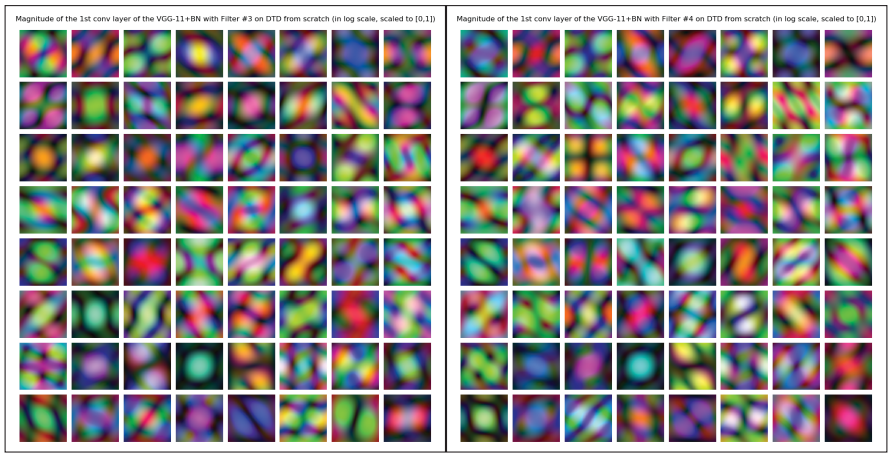
Figure-A I-33 Magnitude en échelle logarithmique de chacun des filtres de la première couche de convolution du ResNet-50 sur DTD *from scratch* selon les différentes méthodes



(a) Original (b) SFSC

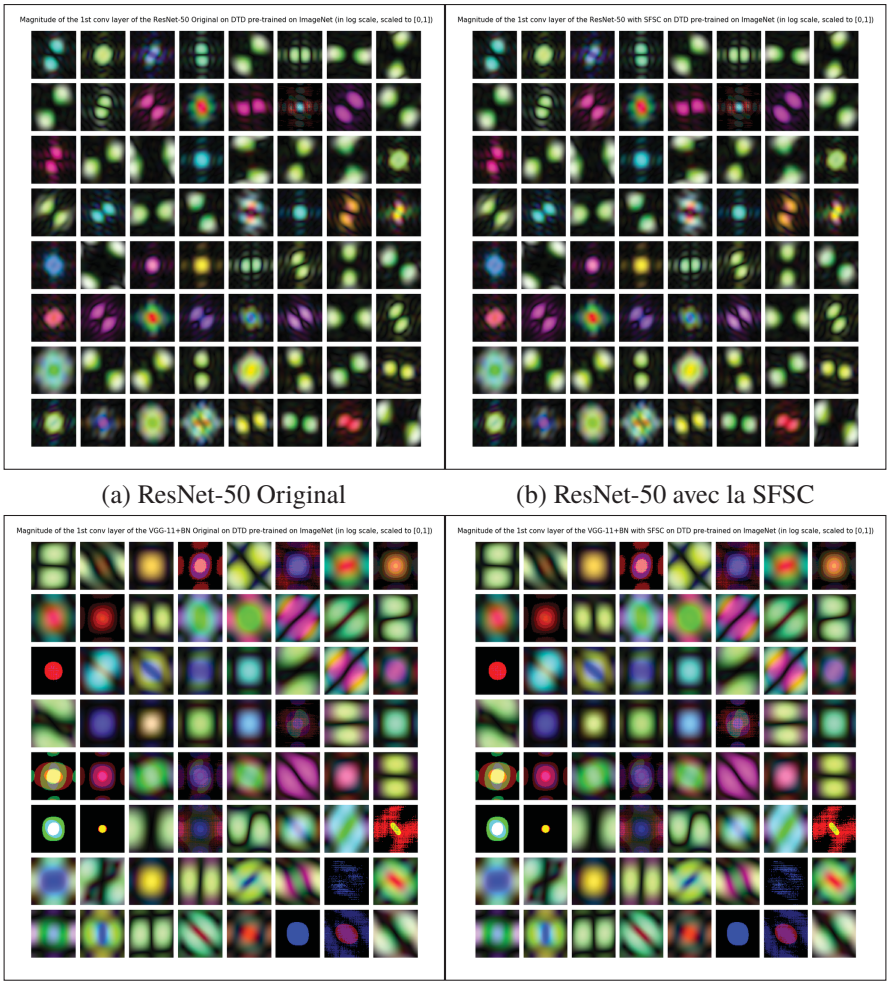


(c) Filtre #1 (d) Filtre #2



(e) Filtre #3 (f) Filtre #4

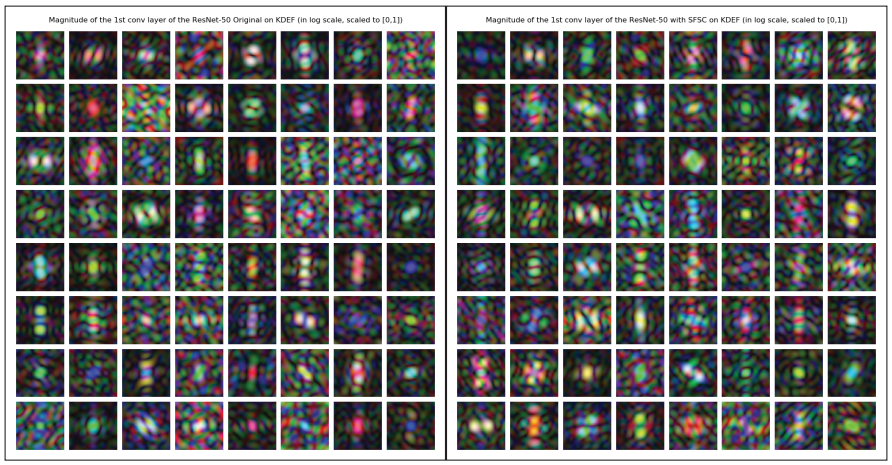
Figure-A I-34 Magnitude en échelle logarithmique de chacun des filtres de la première couche de convolution du VGG-11 avec BN sur DTD *from scratch* selon les différentes méthodes



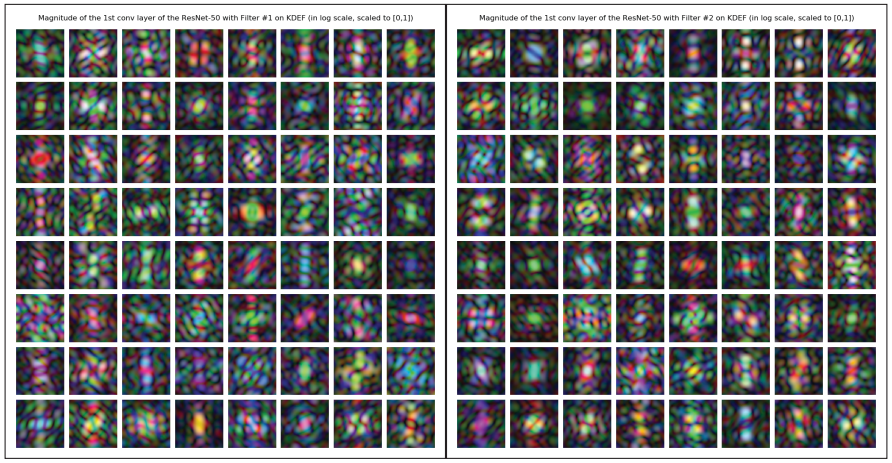
(a) ResNet-50 Original (b) ResNet-50 avec la SFSC

(c) VGG-11 avec BN original (d) VGG-11 avec BN avec la SFSC

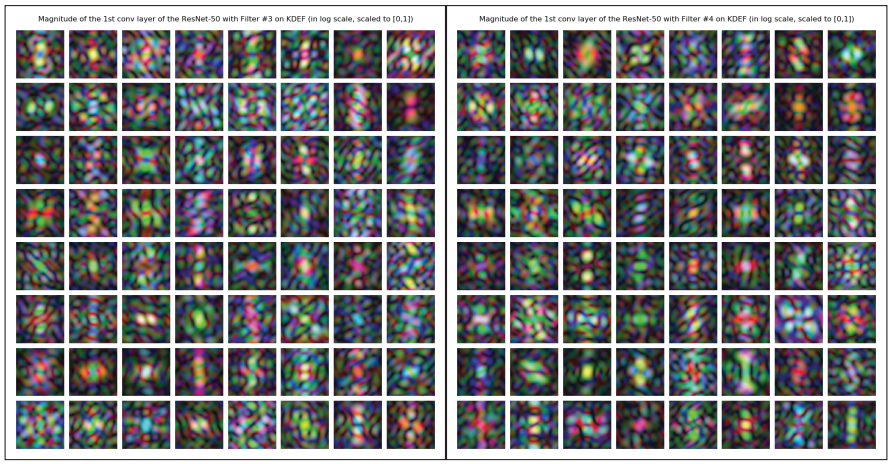
Figure-A I-35 Magnitude en échelle logarithmique de chacun des filtres de la première couche de convolution du ResNet-50 et du VGG-11 avec BN pré-entraînés utilisés sur DTD avec et sans la SFSC



(a) Original (b) SFSC

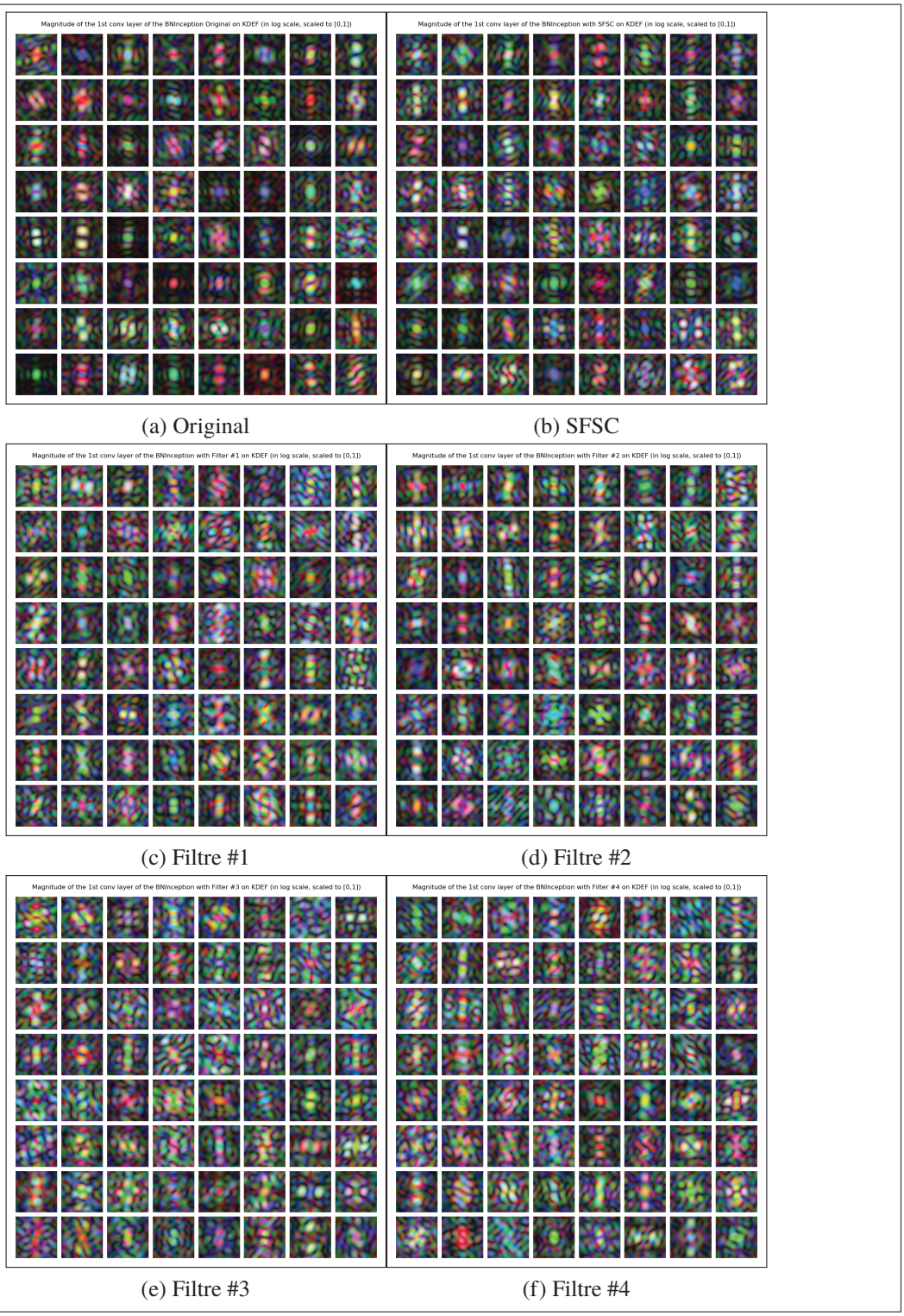


(c) Filtre #1 (d) Filtre #2



(e) Filtre #3 (f) Filtre #4

Figure-A I-36 Magnitude en échelle logarithmique de chacun des filtres de la première couche de convolution du ResNet-50 sur KDEF selon les différentes méthodes



(a) Original (b) SFSC

(c) Filtre #1 (d) Filtre #2

(e) Filtre #3 (f) Filtre #4

Figure-A I-37 Magnitude en échelle logarithmique de chacun des filtres de la première couche de convolution du BNInception sur KDEF selon les différentes méthodes

4. Représentation spatial détaillée des premières couches de convolution



(a) Original (b) SFSC

(c) Filtre #1 (d) Filtre #2

(e) Filtre #3 (f) Filtre #4

Figure-A I-38 Représentation spatiale de chacun des filtres de la première couche de convolution du DenseNet (L= 100, K = 12) sur CIFAR-100+ selon les différentes méthodes

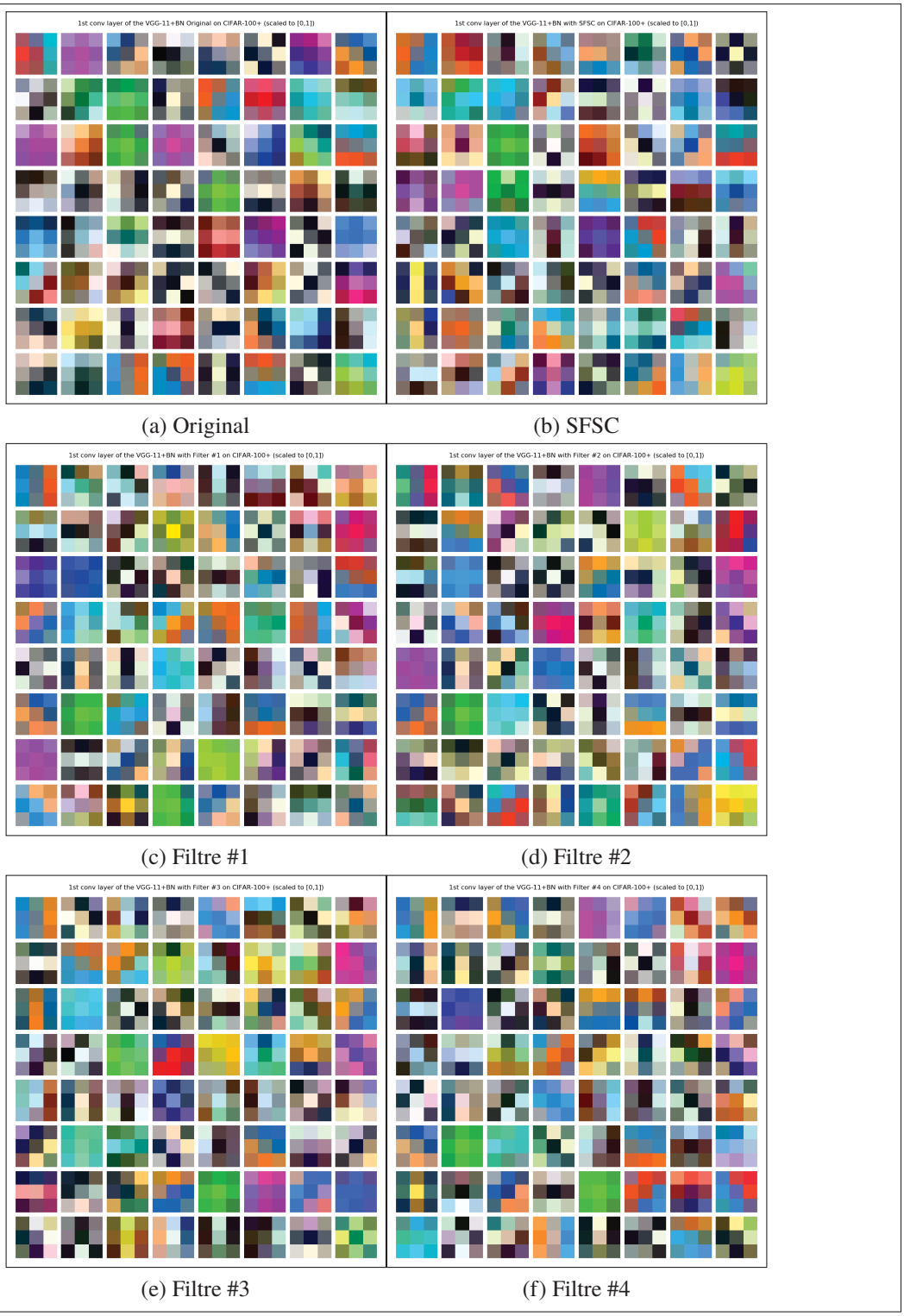
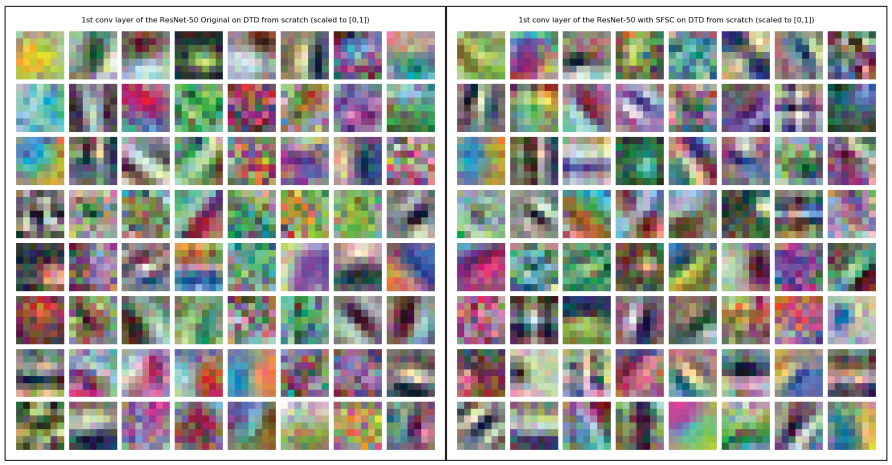
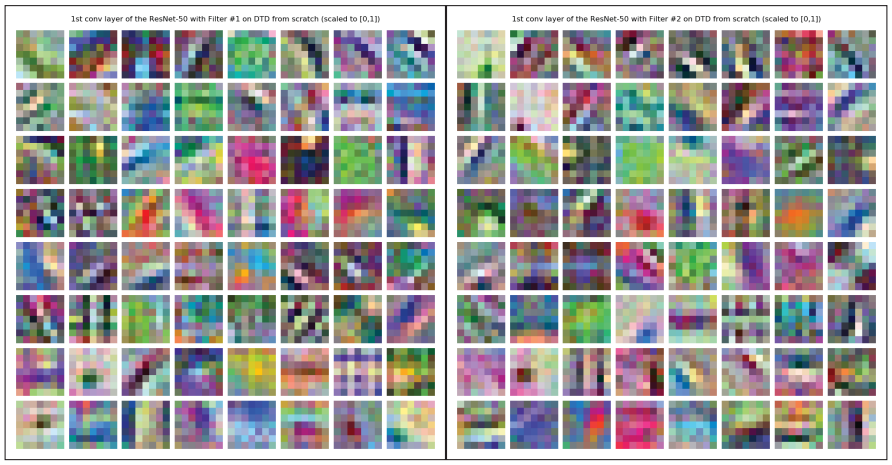


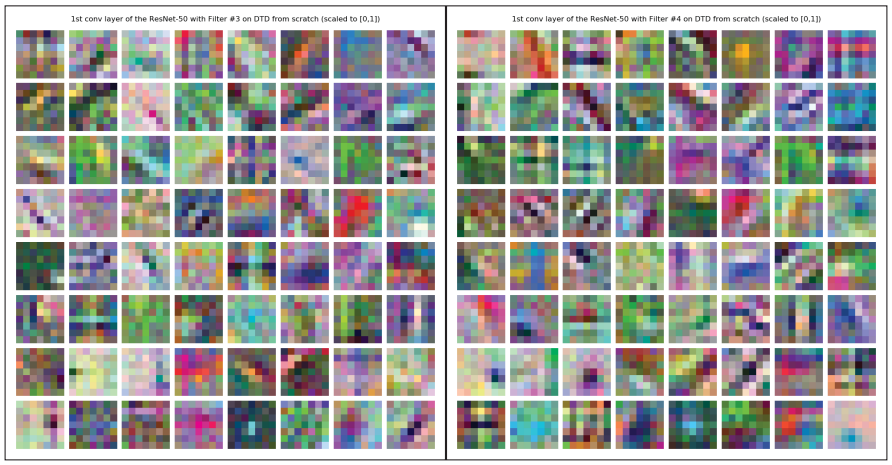
Figure-A I-39 Représentation spatiale de chacun des filtres de la première couche de convolution du VGG-11 avec BN sur CIFAR-100+ selon les différentes méthodes



(a) Original (b) SFSC

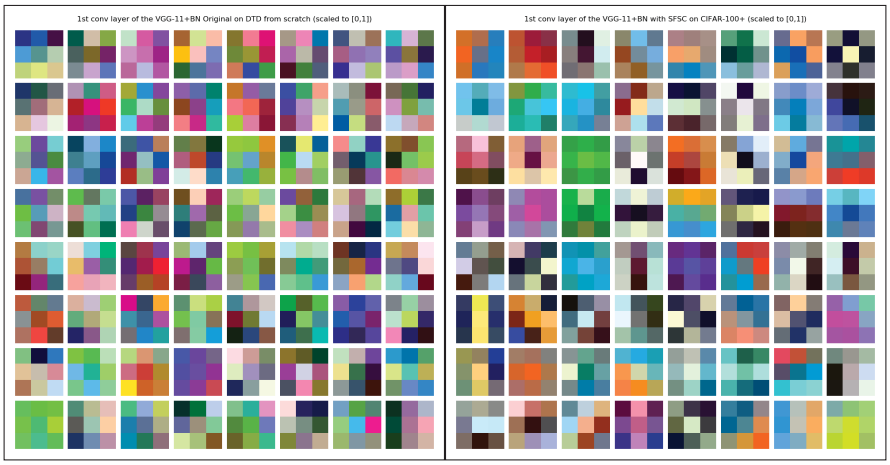


(c) Filtre #1 (d) Filtre #2

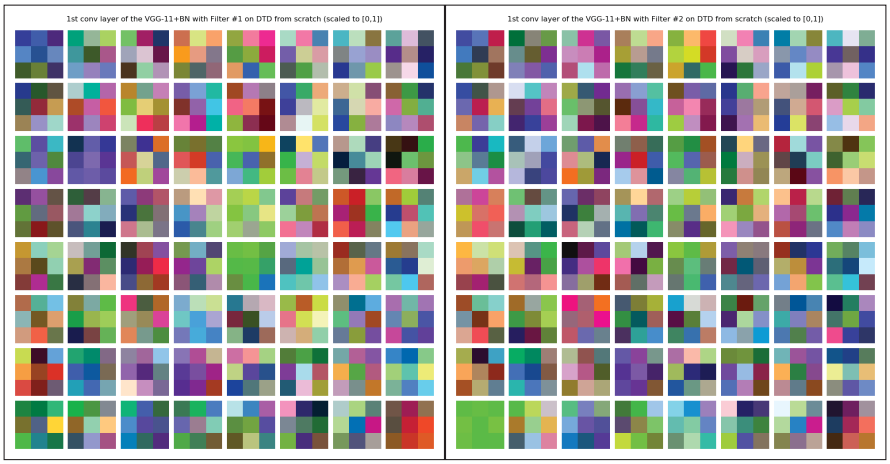


(e) Filtre #3 (f) Filtre #4

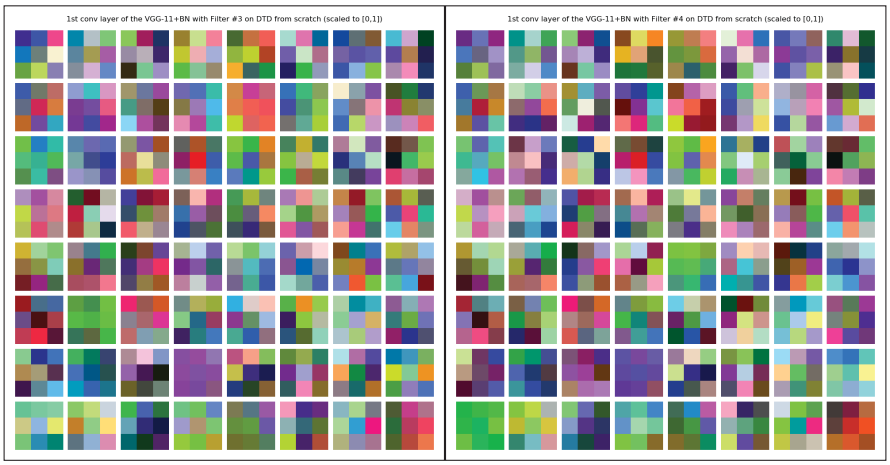
Figure-A I-40 Représentation spatiale de chacun des filtres de la première couche de convolution du ResNet-50 sur DTD *from scratch* selon les différentes méthodes



(a) Original (b) SFSC

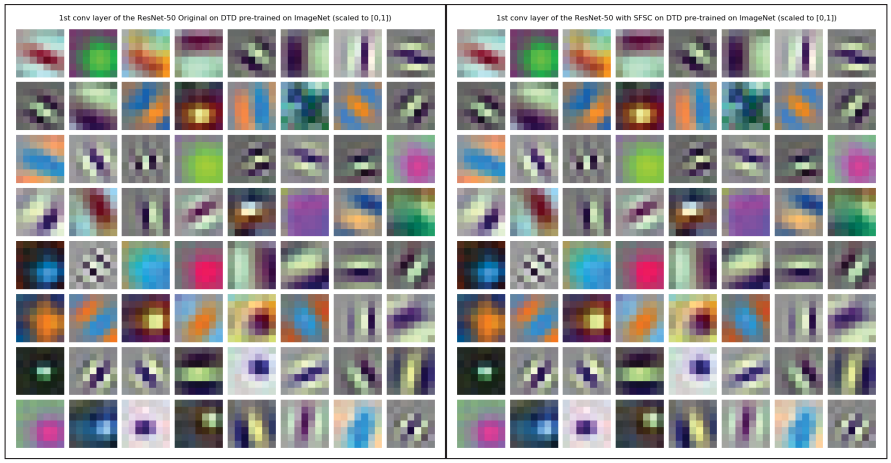


(c) Filtre #1 (d) Filtre #2

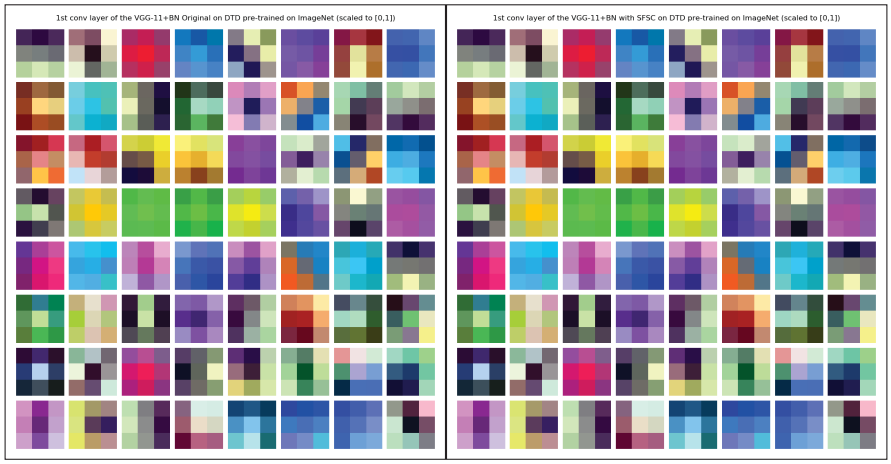


(e) Filtre #3 (f) Filtre #4

Figure-A I-41 Représentation spatiale de chacun des filtres de la première couche de convolution du VGG-11 avec BN sur DTD *from scratch* selon les différentes méthodes

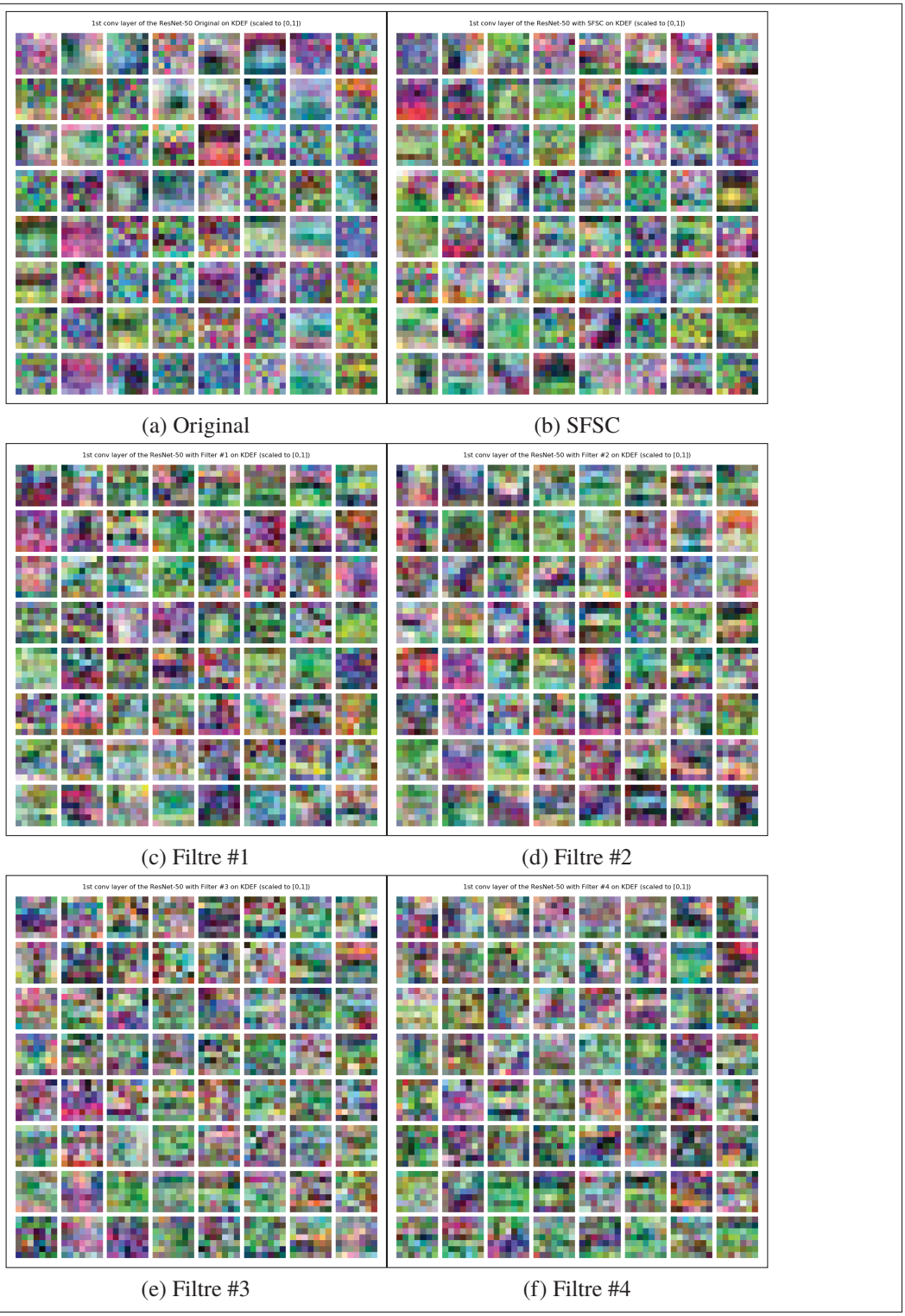


(a) ResNet-50 Original (b) ResNet-50 avec la SFSC



(c) VGG-11 avec BN original (d) VGG-11 avec BN avec la SFSC

Figure-A I-42 Représentation spatiale de chacun des filtres de la première couche de convolution du ResNet-50 et du VGG-11 avec BN pré-entraînés utilisés sur DTD avec et sans la SFSC

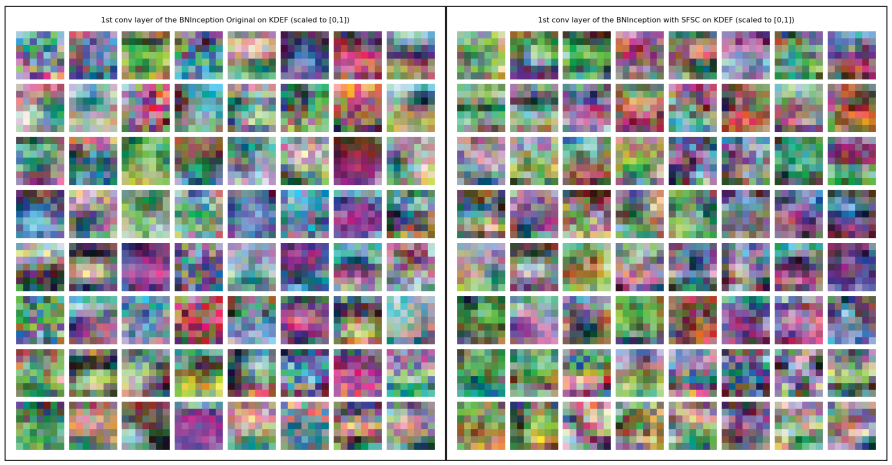


(a) Original (b) SFSC

(c) Filtre #1 (d) Filtre #2

(e) Filtre #3 (f) Filtre #4

Figure-A I-43 Représentation spatiale de chacun des filtres de la première couche de convolution du ResNet-50 sur KDEF selon les différentes méthodes



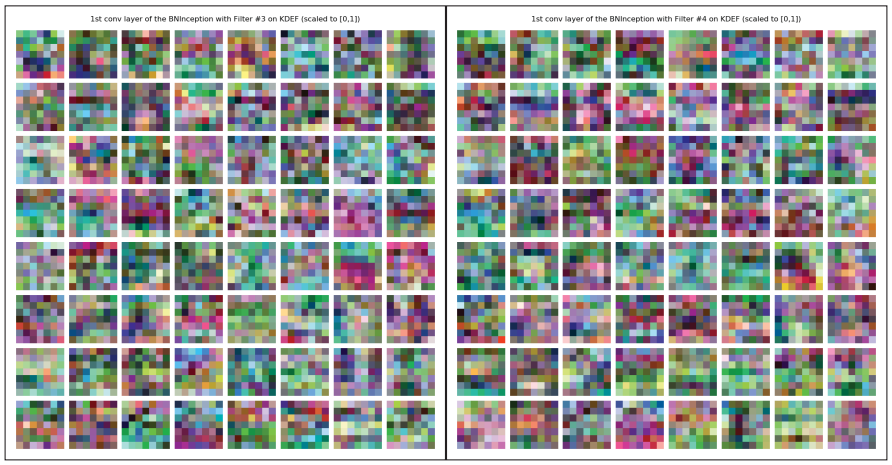
(a) Original

(b) SFSC



(c) Filtre #1

(d) Filtre #2



(e) Filtre #3

(f) Filtre #4

Figure-A I-44 Représentation spatiale de chacun des filtres de la première couche de convolution du BNInception sur KDEF selon les différentes méthodes

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

Adjobo, E. C., Mahama, A. T. S., Gouton, P. & Tossa, J. (2019). Proposition of Convolutional Neural Network Based System for Skin Cancer Detection. *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 35–39.

Andrade-Loarca, H., Kutyniok, G., Oktem, O. & Petersen, P. C. (2019). Extraction of digital wavefront sets using applied harmonic analysis and deep neural networks. *SIAM Journal on Imaging Sciences*, 12(4), 1936–1966.

Andrearczyk, V. (2017). *Deep learning for texture and dynamic texture analysis*. (Thèse de doctorat, Dublin City University, Dublin, Irlande).

Andreux et al. (2020). Kymatio : Scattering Transforms in Python. *Journal of Machine Learning Research*, 21(60), 1–6.

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

Asamwar, R. S., Bhurchandi, K. M. & Gandhi, A. S. (2010). Interpolation of images using discrete wavelet transform to simulate image resizing as in human vision. *International Journal of Automation and Computing*, 7(1), 9–16.

Ashwini, K., Amutha, R. & Harini, K. (2017). Sparse based image compression in wavelet domain. *2017 International Conference on Signal Processing and Communication (ICSPC)*, pp. 58-62.

Ayat, S. O., Khalil-Hani, M., Ab Rahman, A. A.-H. & Abdellatef, H. (2019). Spectral-based convolutional neural network without multiple spatial-frequency domain switchings. *Neurocomputing*, 364, 152–167.

Ba, J. L., Kiros, J. R. & Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv :1607.06450*.

Bouman, C. (2020). Digital Image Processing [Notes de cours]. Repéré à https://engineering .purdue.edu/~bouman/ece637/notes/pdf/RateConversion.pdf.

Bruna, J. & Mallat, S. (2013). Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1872–1886. Repéré à https:// ieeexplore.ieee.org/document/6522407.

Chaabane, C. B., Mellouli, D., Hamdani, T. M., Alimi, A. M. & Abraham, A. (2017). Wavelet convolutional neural networks for handwritten digits recognition. *International Conference on Health Information Science*, pp. 305–310.

Chatfield, K., Simonyan, K., Vedaldi, A. & Zisserman, A. (2014). Return of the devil in the details : Delving deep into convolutional nets. *arXiv preprint arXiv :1405.3531*.

Chen, T., Lin, L., Zuo, W., Luo, X. & Zhang, L. (2017a). Learning a wavelet-like auto-encoder to accelerate deep neural networks. *arXiv preprint arXiv :1712.07493*.

Chen, Y. et al. (2017b). Hyperspectral images classification with Gabor filtering and convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 14(12), 2355–2359.

Cheng, E.-J. et al. (2017). Deep learning based face recognition with sparse representation classification. *International Conference on Neural Information Processing*, pp. 665–674.

Chidester, B., Zhou, T., Do, M. N. & Ma, J. (2019). Rotation equivariant and invariant neural networks for microscopy image analysis. *Bioinformatics*, 35(14), i530–i537.

Chrabaszcz, P., Loshchilov, I. & Hutter, F. (2017). A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv :1707.08819*.

Chung, Y.-C., Cheng, P.-H. & Liu, C.-W. (2019). Energy efficient CNN inference accelerator using fast fourier transform. *2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pp. 1–4.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S. & Vedaldi, A. (2014). Describing textures in the wild. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613.

Cotter, F. & Kingsbury, N. (2018). Deep learning in the wavelet domain. *arXiv preprint arXiv :1811.06115*.

Dadashnialehi, A., Bab-Hadiashar, A. & Hoseinnezhad, R. (2017). Deep learning for texture classification via multi-wavelet fusion of scattering transforms. *2017 IEEE International Conference on Mechatronics (ICM)*, pp. 324–329.

Das, D. K. & Dutta, P. K. (2019). Efficient automated detection of mitotic cells from breast histological images using deep convolution neutral network with wavelet decomposed patches. *Computers in biology and medicine*, 104, 29–42.

Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A*, 2(7), 1160–1169. doi : 10.1364/JOSAA.2.001160.

De Silva, D., Fernando, S., Piyatilake, I. & Karunarathne, A. (2019). Wavelet based edge feature enhancement for convolutional neural networks. *Eleventh International Conference on Machine Vision (ICMV 2018)*, 11041, 110412R.

De Silva, D., Vithanage, H., Xavier, S., Piyatilake, I. & Fernando, S. (2020). Parameterized Wavelets for Convolutional Neural Networks. *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 170–176.

Deng, J. et al. (2009). ImageNet : A Large-Scale Hierarchical Image Database. *CVPR09*.

Deselaers, T. & Ferrari, V. (2010). Global and efficient self-similarity for object classification and detection. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1633–1640.

Dong, Z., Zhang, R. & Shao, X. (2019). A CNN-RNN Hybrid Model with 2D Wavelet Transform Layer for Image Classification. *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1050–1056.

Dragotti, P. L. (2015). Tutorial : Sparse Signal Processing. Part 1 : Sparse Signal Representation [Présentation PowerPoint]. Repéré à http://www.commsp.ee.ic.ac.uk/~pld/talks/TutorialSparseSP_Part1.pdf.

ducha aiki. (2016). caffenet-benchmark. Repéré à https://github.com/ducha-aiki/caffenet-benchmark/blob/master/batchnorm.md.

Dumic, E., Grgic, S. & Grgic, M. (2007). The use of wavelets in image interpolation : possibilities and limitations. *RADIOENGINEERING-PRAGUE-*, 16(4), 101.

Easley, G., Labate, D. & Lim, W.-Q. (2008). Sparse directional image representations using the discrete shearlet transform. *Applied and Computational Harmonic Analysis*, 25(1), 25–46.

edouardoyallon. (2017). Github. Repéré à https://github.com/edouardoyallon/scalingscattering/blob/master/CIFAR/models/Scatter_WRN.py.

Feng, Y., Lu, H., Bai, J., Cao, L. & Yin, H. (2020). Fully convolutional network-based infrared and visible image fusion. *Multimedia Tools and Applications*, 1–14.

Fogel, I. & Sagi, D. (1989). Gabor filters as texture discriminator. *Biological cybernetics*, 61(2), 103–113.

Fu, C.-Y. (2019). pytorch-vgg-cifar10. Repéré à https://github.com/chengyangfu/pytorch-vgg-cifar10.

Fujieda, S., Takayama, K. & Hachisuka, T. (2017). Wavelet convolutional neural networks for texture classification. *arXiv preprint arXiv :1707.07394*.

Fujieda, S., Takayama, K. & Hachisuka, T. (2018). Wavelet convolutional neural networks. *arXiv preprint arXiv :1805.08620*.

Gao, Y., Beijbom, O., Zhang, N. & Darrell, T. (2016). Compact bilinear pooling. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 317–326.

Gao, Y., Bregovic, R., Koch, R. & Gotchev, A. (2020). DRST : Deep Residual Shearlet Transform for Densely Sampled Light Field Reconstruction. *arXiv preprint arXiv :2003.08865*.

Gonzalez, R. C. & Woods, R. E. (2006). Filtering in the Frequency Domain. Dans *Digital Image Processing* (éd. 3, ch. 4, pp. 249–263). Upper Saddle River, NJ, USA : Prentice-Hall, Inc.

Goossens, B., Luong, H. & Philips, W. (2019). Wavelet/shearlet hybridized neural networks for biomedical image restoration. *Wavelets and Sparsity XVIII*, 11138, 111380H.

Guan, B., Zhang, J., Sethares, W. A., Kijowski, R. & Liu, F. (2019). SpecNet : Spectral Domain Convolutional Neural Network. *arXiv preprint arXiv :1905.10915*.

Guberman, N. (2016). On complex valued convolutional neural networks. *arXiv preprint arXiv :1602.09046*.

Guo, K., Kutyniok, G. & Labate, D. (2006). Sparse multidimensional representations using anisotropic dilation and shear operators. Citeseer.

Guo, Q., Cao, X. & Zou, Q. (2018). Enhanced wavelet convolutional neural networks for visual tracking. *Journal of Electronic Imaging*, 27(5), 053046.

He, K., Zhang, X., Ren, S. & Sun, J. (2015). Delving deep into rectifiers : Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

He, K., Zhang, X., Ren, S. & Sun, J. (2016a). Identity mappings in deep residual networks. *European conference on computer vision*, pp. 630–645.

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

Hermessi, H., Mourali, O. & Zagrouba, E. (2018). Convolutional neural network-based multimodal image fusion via similarity learning in the shearlet domain. *Neural Computing and Applications*, 30(7), 2029–2045.

Highlander, T. & Rodriguez, A. (2016). Very efficient training of convolutional neural networks using fast fourier transform and overlap-and-add. *arXiv preprint arXiv :1601.06815*.

Hosseini, S., Lee, S. H., Kwon, H. J., Koo, H. I. & Cho, N. I. (2018). Age and gender classification using wide convolutional neural network and Gabor filter. *2018 International Workshop on Advanced Image Technology (IWAIT)*, pp. 1–3.

Hou, R., Zhou, D., Nie, R., Liu, D. & Ruan, X. (2019). Brain CT and MRI medical image fusion using convolutional neural networks and a dual-channel spiking cortical model. *Medical & biological engineering & computing*, 57(4), 887–900.

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., Van Der Maaten, L. & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.

Huang, Q. et al. (2019). Blood cell classification based on hyperspectral imaging with modulated Gabor and CNN. *IEEE journal of biomedical and health informatics*, 24(1), 160–170.

Huang, X. & Gai, S. (2020). Banknote Classification Based on Convolutional Neural Network in Quaternion Wavelet Domain. *IEEE Access*, 8, 162141–162148.

Ioffe, S. & Szegedy, C. (2015a). Batch normalization : Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv :1502.03167*.

Ioffe, S. & Szegedy, C. (2015b). Batch normalization : Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv :1502.03167*.

Jabason, E., Ahmad, M. O. & Swamy, M. S. (2018). Shearlet based Stacked Convolutional Network for Multiclass Diagnosis of Alzheimer's Disease using the Florbetapir PET Amyloid Imaging Data. *2018 16th IEEE International New Circuits and Systems Conference (NEWCAS)*, pp. 344–347.

Jo, J. & Bengio, Y. (2017). Measuring the tendency of CNNs to learn surface statistical regularities. *arXiv preprint arXiv :1711.11561*.

John, V., Boyali, A. & Mita, S. (2017). Gabor filter and Gershgorin disk-based convolutional filter constraining for image classification. *Int. J. Mach. Learn. Comput*, 7(4), 55–60.

Kausar, T., Wang, M., Idrees, M. & Lu, Y. (2019). HWDCNN : Multi-class recognition in breast histopathology with Haar wavelet decomposed image based convolution neural network. *Biocybernetics and Biomedical Engineering*, 39(4), 967–982.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M. & Tang, P. T. P. (2016). On large-batch training for deep learning : Generalization gap and sharp minima. *arXiv preprint arXiv :1609.04836*.

Khatami, A. et al. (2020). Convolutional neural network for medical image classification using wavelet features. *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8.

Kingma, D. P. & Ba, J. (2014). Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*.

Krizhevsky, A., Hinton, G. et al. (2009). Learning multiple layers of features from tiny images.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, pp. 1097–1105.

Kwolek, B. (2005). Face detection using convolutional neural networks and Gabor filters. *International Conference on Artificial Neural Networks*, pp. 551–556.

Levinskis, A. (2013). Convolutional neural network feature reduction using wavelet transform. *Elektronika ir Elektrotechnika*, 19(3), 61–64.

Li, J., You, S. & Robles-Kelly, A. (2018). A frequency domain neural network for fast image super-resolution. *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8.

Li, Q., Shen, L., Guo, S. & Lai, Z. (2020). Wavelet Integrated CNNs for Noise-Robust Image Classification. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7245–7254.

Lin, C.-W., Ding, Q., Tu, W.-H., Huang, J.-H. & Liu, J.-F. (2019). Fourier dense network to conduct plant classification using UAV-based optical images. *IEEE Access*, 7, 17736–17749.

Lin, T.-Y. (2020). *Higher-Order Representations for Visual Recognition*. (Thèse de doctorat, University of Massachusetts Amherst, Amherst, Massachusetts).

Lin, T.-Y. & Maji, S. (2016). Visualizing and understanding deep texture representations. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2791–2799.

Lin, T.-Y., RoyChowdhury, A. & Maji, S. (2017). Bilinear convolutional neural networks for fine-grained visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6), 1309–1322.

LiQiufu. (2020). WaveCNet. Repéré à https://github.com/LiQiufu/WaveCNet.

Liu, C., Ding, W., Wang, X. & Zhang, B. (2018a). Hybrid Gabor Convolutional Networks. *Pattern Recognition Letters*, 116, 164–169.

Liu, P., Zhang, H., Zhang, K., Lin, L. & Zuo, W. (2018b). Multi-level wavelet-CNN for image restoration. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 773–782.

Lu, H., Wang, H., Zhang, Q., Won, D. & Yoon, S. W. (2018a). A dual-tree complex wavelet transform based convolutional neural network for human thyroid medical image segmentation. *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 191–198.

Lu, T., Wu, M. & Lu, T. (2018b). Face recognition via Gabor and convolutional neural network. *Ninth International Conference on Graphic and Image Processing (ICGIP 2017)*, 10615, 106150V.

Luan, S., Chen, C., Zhang, B., Han, J. & Liu, J. (2018). Gabor convolutional networks. *IEEE Transactions on Image Processing*, 27(9), 4357–4366.

Lundqvist, D., Flykt, A. & Öhman, A. (1998). The Karolinska directed emotional faces (KDEF). *CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet*, 91(630), 2–2.

Luo, P., Ren, J., Peng, Z., Zhang, R. & Li, J. (2018a). Differentiable learning-to-normalize via switchable normalization. *arXiv preprint arXiv :1806.10779*.

Luo, P., Ren, J., Peng, Z., Zhang, R. & Li, J. (2018b). Differentiable learning-to-normalize via switchable normalization. *arXiv preprint arXiv :1806.10779*.

Luo, P., Zhang, R., Ren, J., Peng, Z. & Li, J. (2019a). Switchable normalization for learning-to-normalize deep representation. *IEEE transactions on pattern analysis and machine intelligence*.

Luo, P., Zhang, R., Ren, J., Peng, Z. & Li, J. (2019b). Switchable normalization for learning-to-normalize deep representation. *IEEE transactions on pattern analysis and machine intelligence*.

Ma, Y., Luo, Y. & Yang, Z. (2020). PCFNet : Deep neural network with predefined convolutional filters. *Neurocomputing*, 382, 32–39.

Mallat, S. (2012). Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10), 1331–1398.

Mallikarjuna, P. et al. (2006). The kth-tips2 database. *Computational Vision and Active Perception Laboratory, Stockholm, Sweden*, 1–10.

Manjusha, K., Kumar, M. A. & Soman, K. (2018). Integrating scattering feature maps with convolutional neural networks for Malayalam handwritten character recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 21(3), 187–198.

Matlab. (2020a). Wavelet Scattering. Repéré à https://www.mathworks.com/help/wavelet/ug/wavelet-scattering.html.

Matlab. (2020b). Continuous and Discrete Wavelet Transforms. Repéré à https://www.mathworks.com/help/wavelet/gs/continuous-and-discrete-wavelet-transforms.html.

Matlab. (2020c). What is a Wavelet ? Repéré à https://www.mathworks.com/help/wavelet/gs/what-is-a-wavelet.html.

Matlab. (2020d). Shearlet Systems. Repéré à https://www.mathworks.com/help/wavelet/ug/shearlet-systems.html.

Mewada, H. K., Patel, A. V., Chaudhari, J., Mahant, K. & Vala, A. (2020). Wavelet features embedded convolutional neural network for multiscale ear recognition. *Journal of Electronic Imaging*, 29(4), 043029.

Mody, M., Ghone, C., Mathew, M. & Jones, J. (2017). Efficient frequency domain CNN algorithm. *2017 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pp. 22–25.

Molchanov, P., Tyree, S., Karras, T., Aila, T. & Kautz, J. (2016). Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv :1611.06440*.

Nava, R., Escalante-Ramírez, B., Cristóbal, G. & Estépar, R. S. J. (2014). Extended Gabor approach applied to classification of emphysematous patterns in computed tomography. *Medical & biological engineering & computing*, 52(4), 393–403. Repéré à https://link.springer.com/article/10.1007%2Fs11517-014-1139-9.

Oraibi, Z. A. (2019). *Local and Deep Texture Features for Classification of Natural and Biomedical Images*. (Thèse de doctorat, University of Missouri-Columbia, Columbia, Missouri).

Oulhaj, H., Jennane, R., Amine, A., El Hassouni, M. & Rziza, M. (2018). Study of the relative magnitude in the wavelet domain for texture characterization. *Signal, Image and Video Processing*, 12(7), 1403–1410. Repéré à https://link.springer.com/article/10.1007/s11760-018-1295-8.

Oyallon, E. & Mallat, S. (2015). Deep roto-translation scattering for object classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2865–2873.

Oyallon, E., Mallat, S. & Sifre, L. (2013). Generic deep networks with wavelet scattering. *arXiv preprint arXiv :1312.5940*.

Oyallon, E., Belilovsky, E. & Zagoruyko, S. (2017). Scaling the scattering transform : Deep hybrid networks. *Proceedings of the IEEE international conference on computer vision*, pp. 5618–5627.

Oyallon, E., Belilovsky, E., Zagoruyko, S. & Valko, M. (2018). Compressing the input for CNNs with the first-order scattering transform. *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 301–316.

Pandya, B. et al. (2018). Fingerprint classification using a deep convolutional neural network. *2018 4th International Conference on Information Management (ICIM)*, pp. 86–91.

Patrick, M. K., Weyori, B. A. & Mighty, A. A. (2020). Max-Pooled Fast Learning Gabor Capsule Network. *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pp. 1–8.

Peeples, J., Xu, W. & Zare, A. (2020). Histogram Layers for Texture Analysis. *arXiv preprint arXiv :2001.00215*.

Poojitha, U. P. & Sharma, S. L. (2019). Hybrid Unified Deep Learning Network for Highly Precise Gleason Grading of Prostate Cancer. *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 899–903.

Popa, C.-A. (2017). Complex-valued convolutional neural networks for real-valued image classification. *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 816–822.

Popa, C.-A. & Cernăzanu-Glăvan, C. (2018). Fourier transform-based image classification using complex-valued convolutional neural networks. *International Symposium on Neural Networks*, pp. 300–309.

Pragada, S. & Sivaswamy, J. (2008). Image denoising using matched biorthogonal wavelets. *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 25–32.

Pratt, H., Williams, B., Coenen, F. & Zheng, Y. (2017). Fcnn : Fourier convolutional neural networks. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 786–798.

Rahaman, N. et al. (2019). On the spectral bias of neural networks. *International Conference on Machine Learning*, pp. 5301–5310.

Rasti, R., Mehridehnavi, A., Rabbani, H. & Hajizadeh, F. (2018). Automatic diagnosis of abnormal macula in retinal optical coherence tomography images using wavelet-based convolutional neural network features and random forests classifier. *Journal of biomedical optics*, 23(3), 035005.

Reddi, S. J., Kale, S. & Kumar, S. (2019). On the convergence of adam and beyond. *arXiv preprint arXiv :1904.09237*.

Ren, Q., Zhou, B. & Guo, W. (2019). Image Classification with Hybrid Shearlet Scattering Networks. *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, 2, 304–307.

Ren, X., Guo, H., Di, C., Han, Z. & Li, S. (2016). Face recognition based on local Gabor binary patterns and convolutional neural network. *International Conference in Communications, Signal Processing, and Systems*, pp. 699–707.

Rezaeilouyeh, H., Mollahosseini, A. & Mahoor, M. H. (2016). Microscopic medical image classification framework via deep learning and shearlet transform. *Journal of Medical Imaging*, 3(4), 044501.

Rippel, O., Snoek, J. & Adams, R. P. (2015). Spectral representations for convolutional neural networks. *Advances in neural information processing systems*, pp. 2449–2457.

Rodriguez, M. X. B. et al. (2020). Deep Adaptive Wavelet Network. *The IEEE Winter Conference on Applications of Computer Vision*, pp. 3111–3119.

Rossetto, A. M. & Zhou, W. (2019). Improving Classification with CNNs using Wavelet Pooling with Nesterov-Accelerated Adam. *Proceedings of 11th International Conference*, 60, 84–93.

Ryu, J., Yang, M.-H. & Lim, J. (2018). Dft-based transformation invariant pooling layer for visual classification. *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 84–99.

Sarwar, S. S., Panda, P. & Roy, K. (2017). Gabor filter assisted energy efficient fast learning convolutional neural networks. *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 1–6.

Shahdoosti, H. R. & Rahemi, Z. (2019). Edge-preserving image denoising using a deep convolutional neural network. *Signal Processing*, 159, 20–32.

Sharma, V. et al. (2018). Classification-driven dynamic image enhancement. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4033–4041.

Shearlab. (2020). Shearlab : A rational design of a digital parabolic scaling algorithm. Repéré à http://shearlab.math.lmu.de/software.

Shechtman, E. & Irani, M. (2007). Matching local self-similarities across images and videos. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8.

Sifre, L. & Mallat, S. (2013). Rotation, scaling and deformation invariant scattering for texture discrimination. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1233–1240.

Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*.

Sinha, P., Psaromiligkos, I. & Zilic, Z. (2019). A Structurally Regularized Convolutional Neural Network for Image Classification Using Wavelet-Based Subband Decomposition. *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 649–653.

Soltan, E. A., El-Rabaie, S., El-Fishawy, N., Abd El-samie, F. E. & Elkhamy, S. E. (2010). Wavelet-based image interpolation with a least squares algorithm. *2010 International Computer Engineering Conference (ICENCO)*, pp. 103–106.

Tran, D. H., Meunier, M. & Cheriet, F. (2019). WaveM-CNN for Automatic Recognition of Sub-cellular Organelles. *International Conference on Image Analysis and Recognition*, pp. 186–194.

Tsuzuku, Y. & Sato, I. (2019). On the structural sensitivity of deep convolutional networks to the directions of fourier basis functions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 51–60.

Ulyanov, D., Vedaldi, A. & Lempitsky, V. (2016). Instance normalization : The missing ingredient for fast stylization. *arXiv preprint arXiv :1607.08022*.

Vasilyev, A. (2015). CNN optimizations for embedded systems and FFT. *Standford University Report*.

Velisavljević, V. (2008). Image interpolation using oriented wavelet transforms. *2008 16th European Signal Processing Conference*, pp. 1–5.

Venkataswamy, P., Omair Ahmad, M. & Swamy, M. N. S. (2018). Targeted Wavelet Based Image Aesthetics Classification Using Convolutional Neural Nets. *2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE)*, pp. 1-5.

Vishwakarma, A., Bhuyan, M. K. & Iwahori, Y. (2018). An Efficient Algorithm for Medical Image Fusion Using Nonsubsampled Shearlet Transform. *Proceedings of 2nd International Conference on Computer Vision & Image Processing*, pp. 243–252.

Wang, F., Zhang, B., Chai, S. & Xia, Y. (2018). An extreme learning machine-based community detection algorithm in complex networks. *Complexity*, 2018.

Wang, H., Wu, X., Huang, Z. & Xing, E. P. (2020). High-frequency Component Helps Explain the Generalization of Convolutional Neural Networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8684–8694.

Watanabe, T. & Wolf, D. F. (2020). Image classification in frequency domain with 2SReLU : a second harmonics superposition activation function. *arXiv preprint arXiv :2006.10853*.

Curvelet. (2018). Dans *Wikipedia*. Repéré le 16 novembre 2018 à : https://en.wikipedia.org/wiki/Curvelet.

Gabor filter. (2020a). Dans *Wikipedia*. Repéré le 15 Octobre 2020 à : https://en.wikipedia.org/wiki/Gabor_filter.

Ondelette. (2020b). Dans *Wikipedia*. Repéré le 29 Septembre 2020 à : https://fr.wikipedia.org/wiki/Ondelette.

Shearlet. (2020c). Dans *Wikipedia*. Repéré le 19 août 2020 à : https://en.wikipedia.org/wiki/Shearlet.

Williams, T. & Li, R. (2018). Wavelet pooling for convolutional neural networks. *International Conference on Learning Representations*.

Williams, T. L. (2017). *Advanced Image Classification Using Deep Neural Networks*. (Thèse de doctorat, North Carolina A&T State University, Greensboro, Caroline du Nord).

Wu, Y. & He, K. (2018). Group normalization. *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. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500.

Xiong, Y., Lu, T. & Jiang, Y. (2018). Chinese character recognition based on Gabor feature extraction and CNN. *MIPPR 2017 : Pattern Recognition and Computer Vision*, 10609, 106091N.

Xu, Z.-Q. J., Zhang, Y. & Xiao, Y. (2019). Training behavior of deep neural network in frequency domain. *International Conference on Neural Information Processing*, pp. 264–274.

Xu, Z. J. (2018). Understanding training and generalization in deep learning by fourier analysis. *arXiv preprint arXiv :1808.04295*.

Xue, J., Zhang, H. & Dana, K. (2018). Deep texture manifold for ground terrain recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 558–567.

Yang, G., Yang, J., Lu, Z. & Liu, D. (2020). A convolutional neural network with sparse representation. *Knowledge-Based Systems*, 106419.

Yang, J., Wu, Z. & Zhang, J. (2018). A Robust Fingerprint Identification Method by Deep Learning with Gabor Filter Multidimensional Feature Expansion. *International Conference on Cloud Computing and Security*, pp. 447–457.

Yao, H., Chuyi, L., Dan, H. & Weiyu, Y. (2016). Gabor feature based convolutional neural network for object recognition in natural scene. *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, pp. 386–390.

Yin, D., Gontijo Lopes, R., Shlens, J., Cubuk, E. D. & Gilmer, J. (2019a). A fourier perspective on model robustness in computer vision. *Advances in Neural Information Processing Systems*, 32, 13276–13286.

Yin, Q., Zhang, R. & Shao, X. (2019b). CNN and RNN mixed model for image classification. *MATEC Web of Conferences*, 277, 02001.

Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. (2014). How transferable are features in deep neural networks ? *Advances in neural information processing systems*, pp. 3320–3328.

Zagoruyko, S. & Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv :1605.07146*.

Zarka, J., Thiry, L., Angles, T. & Mallat, S. (2019). Deep network classification by scattering and homotopy dictionary learning. *arXiv preprint arXiv :1910.03561*.

Zhang, H., Xue, J. & Dana, K. (2017). Deep ten : Texture encoding network. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 708–717.

Zhang, H. & Ma, J. (2018). Hartley spectral pooling for deep learning. *arXiv preprint arXiv :1810.04028*.

Zheng, G. & Zhu, Q. (2019). Unsupervised Image Feature Extraction Based on Scattering Transform and Self-supervised Learning with Highly Training Efficiency. *Journal of Physics : Conference Series*, 1237(3), 032044.