Modélisation Parcimonieuse de CNNs avec des Paquets d'Ondelettes Dual-Tree

Hubert Leterme, Kévin Polisano, Valérie Perrier, Karteek Alahari

Modélisation Parcimonieuse de CNNs avec des Paquets d'Ondelettes Dual-Tree
Sparsifying Convolutional Layers with Dual-Tree Wavelet Packets

H. Leterme[1] K. Polisano[2] V. Perrier[2] K. Alahari[1]

[1] Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP*, LJK, 38000 Grenoble, France
[2] Univ. Grenoble Alpes, CNRS, Grenoble INP, LJK, 38000 Grenoble, France

Résumé

Nous proposons d'améliorer l'interprétabilité mathématique des réseaux neuronaux convolutifs (CNNs) pour la classification d'images. Pour cela, nous remplaçons les premières couches de réseaux tels qu'AlexNet ou ResNet par un opérateur faisant intervenir une version complexe, orientée et redondante de la transformée en paquets d'ondelettes discrète, appelée en anglais dual-tree wavelet packet transform. Nous montrons expérimentalement que ces réseaux modifiés se comportent de manière très similaire aux modèles originaux une fois entraînés. L'objectif est ensuite d'étudier, d'un point de vue théorique, l'opérateur mathématique ainsi introduit, et d'identifier des leviers d'optimisation. Nous souhaitons analyser ses principales propriétés telles que la sélectivité directionnelle, la stabilité par translation et rotation, qui permettent de discriminer des images de nature différente tout en atténuant les sources de variabilité au sein d'une même classe d'images. Ce travail est un pas vers une description plus complète des CNNs existants à l'aide d'opérateurs mathématiques bien définis, caractérisés par un faible nombre de paramètres arbitraires, les rendant de fait plus aisés à interpréter.

Mots Clef

Apprentissage profond, classification d'images, transformée en paquets d'ondelettes discrète

Abstract

We propose to improve the mathematical interpretability of convolutional neural networks (CNNs) for image classification. In this purpose, we replace the first layers of existing models such as AlexNet or ResNet by an operator containing the dual-tree wavelet packet transform, i.e., a redundant decomposition using complex and oriented waveforms. Our experiments show that these modified networks behave very similarly to the original models once trained. The goal is then to study this operator from a theoretical point of view and to identify potential optimizations. We want to analyze its main properties such as directional selectivity, stability with respect to small shifts and rotations, thus retaining discriminant information while decreasing intra-class variability. This work is a step toward a more complete description of CNNs using well-defined mathematical operators, characterized by a small number of arbitrary parameters, making them easier to interpret.

Keywords

Deep learning, image classification, dual-tree wavelet packet transform

1 Introduction

Les réseaux de neurones convolutifs (CNNs) ont considérablement repoussé les limites de l'état de l'art dans des domaines tels que la reconnaissance vocale, la classification ou la détection d'images [12]. Cependant, ces algorithmes sont très gourmands en ressources. De plus cette approche, très empirique, manque d'une compréhension mathématique approfondie.

D'autre part, en traitement d'images, la théorie des ondelettes et l'analyse multi-résolution sont construites sur une base mathématique solide. Elles ont prouvé leur efficacité dans de nombreux domaines, en particulier la compression de signaux et la réduction du bruit [16]. De plus, les filtres d'ondelettes ont été largement utilisés comme extracteurs de caractéristiques pour la classification de signaux, d'images et de textures [7, 11, 20, 31].

Ces approches, bien que se basant toutes deux sur des filtres de convolution pour atteindre leurs objectifs, diffèrent en un point essentiel. En théorie des ondelettes, les filtres sont spécifiquement conçus pour vérifier certaines conditions bien précises, tandis que les CNNs utilisent des filtres librement appris sans préjuger de leur comportement. Néanmoins, bien souvent en vision par ordinateur, les paramètres appris par les CNNs prennent dans la première couche l'allure de filtres de Gabor orientés [2, 32]. Ce phénomène suggère que les premières couches extraient des caractéristiques génériques comme des contours orientés ou des formes simples, qui sont indépendantes du problème à résoudre.

*Institute of Engineering Univ. Grenoble Alpes

Approche proposée Dans cet article, nous proposons de contraindre la première couche du réseau en remplaçant les filtres librement appris par une transformation en paquets d'ondelettes complexes et orientées, appelée en anglais dual-tree wavelet packet transform (DT-$\mathbb{C}$WPT) [1]. De tels paquets d'ondelettes peuvent être vus comme la version discrète et orthogonale des filtres de Gabor. Afin de conserver la structure originale du réseau, cette transformation est suivie d'une couche de convolution 1×1 effectuant des combinaisons linéaires entre les matrices de coefficients. Nous introduisons ainsi des hypothèses a priori, ce qui a pour effet de guider la phase d'apprentissage et de réduire le nombre de paramètres appris tout en conservant le pouvoir prédictif du réseau.

Le principal objectif de notre travail est de décrire et interpréter le comportement observé des premières couches de CNNs à l'aide d'un modèle parcimonieux. De façon schématique, si l'on note Θ l'ensemble des états que peut prendre un réseau standard, la phase d'apprentissage vise à sélectionner $\theta_0 \in \Theta$ minimisant la fonction de coût. Comme nous le verrons, notre modèle restreint implicitement l'ensemble des configurations possibles à $\Theta' \subset \Theta$. En notant $\theta'_0 \in \Theta'$ l'état sélectionné après apprentissage, nous montrons que, sous certaines conditions, $\theta'_0 \approx \theta_0$. L'avantage de notre modèle est qu'il tire avantage des propriétés d'extraction de la DT-$\mathbb{C}$WPT telles que la sélectivité directionnelle ou la stabilité par translation, propriétés pour lesquelles il existe des garanties théoriques.

Comme preuve de concept, nous avons testé notre approche sur AlexNet [10]. Notre choix est motivé par la taille importante des noyaux de convolution, permettant une comparaison visuelle plus aisée avec nos propres filtres. Cependant, les motifs oscillants se retrouvent sur la plupart des CNNs entraînés sur des bases de données d'images naturelles. Des expériences supplémentaires menées sur une architecture ResNet [6] montrent que notre méthode se généralise bien à d'autres modèles.

Travaux apparentés Dans un esprit similaire, un certain nombre de tentatives de concilier ces deux champs de recherche ont été menées par le passé. Les réseaux de diffusion d'ondelettes [1] [3] effectuent sur les images d'entrée une cascade de transformations en ondelettes suivies par une non-linéarité (calcul du module des coefficients). Les représentations obtenues sont stables par déformation et conservent les informations de haute fréquence. Ces réseaux ont par la suite été améliorés pour des images plus complexes [18, 19, 33], et par ailleurs adaptés dans le cas d'une transformée discrète [25].

Un des objectifs visés est de construire des réseaux de type CNN, structurés en opérateurs mathématiques bien définis, permettant ainsi d'en étudier les propriétés et offrant des garanties de performance. Notre travail s'inscrit dans cette démarche. Cependant, au lieu de construire un réseau *ad hoc*, nous cherchons à fournir une description mathématique de modèles *existants*.

1. En anglais, wavelet scattering networks.

Par ailleurs, [23] utilisent des filtres de Gabor dans la première couche pour réduire la complexité du modèle au prix d'une faible dégradation de performance. Toutefois, la moitié des filtres de convolution restent librement appris ; de plus le modèle n'est testé que sur de petites images.

Enfin, d'autres travaux visent à améliorer les modèles existants en ajoutant aux CNNs des filtres d'ondelettes, de Gabor ou de cosinus discrets [5, 14, 15, 27, 29, 30]. Cependant, ces approches ne visent pas à reproduire le comportement de réseaux existants, et poursuivent donc un objectif différent du nôtre.

A notre connaissance, nous sommes les seuls à utiliser la DT-$\mathbb{C}$WPT pour guider l'apprentissage d'un CNN. Tout comme les filtres de Gabor, les paquets d'ondelettes sont bien localisés en fréquence et présentent le même facteur de sous-échantillonnage sur l'ensemble des matrices de coefficients. L'approche *dual-tree* permet de plus d'extraire des caractéristiques orientées et stables par translation. Deux avantages majeurs par rapport aux filtres de Gabor sont la parcimonie et l'exhaustivité : une seule paire de filtres 1D suffit à caractériser l'ensemble du processus ; de plus la représentation ainsi calculée, tout en présentant une redondance limitée, permet la reconstruction parfaite de l'image d'origine.

2 Contexte

Notations Dans cet article, nous considérons des séquences 2D infinies, appelées *images* : $\mathrm{X} \in \mathbb{U} = \mathbb{R}^{\mathbb{Z}^2}$. L'indexation se fait entre crochets : $\forall \boldsymbol{n} \in \mathbb{Z}^2$, $\mathrm{X}[\boldsymbol{n}] \in \mathbb{R}$. La convolution entre deux images est définie par $(\mathrm{X} * \mathrm{Y})[\boldsymbol{n}] = \sum_{\boldsymbol{k} \in \mathbb{Z}^2} \mathrm{X}[\boldsymbol{n} - \boldsymbol{k}]\, \mathrm{Y}[\boldsymbol{k}]$. On considère de plus l'image miroir : $\overline{\mathrm{X}}[\boldsymbol{n}] = \mathrm{X}[-\boldsymbol{n}]$; sur-échantillonnée : $(\mathrm{X} \uparrow d)[\boldsymbol{n}] = \mathrm{X}[\boldsymbol{n}/d]$ si $\boldsymbol{n}/d \in \mathbb{Z}^2$ ($= 0$ sinon) ; sous-échantillonnée : $(\mathrm{X} \downarrow d)[\boldsymbol{n}] = \mathrm{X}[d\boldsymbol{n}]$. Enfin, pour tout scalaire $b \in \mathbb{R}$, on note $b + \mathrm{X} = b\mathrm{J} + \mathrm{X}$, où J désigne la séquence de uns.

Par ailleurs, on travaillera également avec des *tenseurs*, i.e., des batchs d'images multicanales, que l'on notera $\mathbf{X} = (\mathrm{X}_{p,\,k}) \in \mathbb{U}^{P \times K}$.

Transformée en paquets d'ondelettes (WPT) Ce paragraphe donne un aperçu très succinct de l'algorithme WPT [16]. L'idée est de construire une base orthonormale multirésolution de $\mathbb{U}$, composée de motifs oscillants orientés. A partir d'une paire de filtres miroirs conjugués (CMFs) h et $\mathrm{g} \in \mathbb{R}^{\mathbb{Z}}$, nous considérons un banc de filtres 2D séparables, composé d'un filtre passe-bas $\mathrm{G}^{(0)} = \mathrm{h} \otimes \mathrm{h}$ et de trois filtres passe-haut $\mathrm{G}^{(1)} = \mathrm{h} \otimes \mathrm{g}$, $\mathrm{G}^{(2)} = \mathrm{g} \otimes \mathrm{h}$ et $\mathrm{G}^{(3)} = \mathrm{g} \otimes \mathrm{g}$.

Soit $\mathrm{X} \in \mathbb{U}$. La décomposition commence avec $\mathrm{X}_0^{(0)} = \mathrm{X}$. En considérant $j > 0$, supposons que nous ayons calculé 4^{j-1} séquences de coefficients à l'échelle $j-1$, notées $\mathrm{X}_{j-1}^{(k)}$ pour chaque $k \in \{0 \,..\, 4^{j-1} - 1\}$. Elles constituent une représentation de X à partir de laquelle l'image d'origine peut être parfaitement reconstruite. Nous calculons alors les coefficients de paquets d'ondelettes à l'échelle j

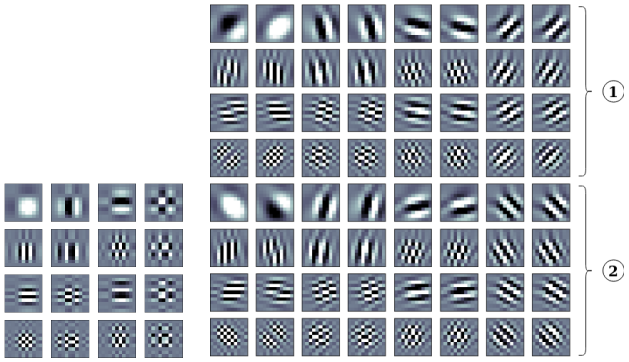

FIGURE 1 – Filtres résultants de la WPT (gauche) et de la DT-$\mathbb{C}$WPT (droite) avec $J = 2$, générés à partir de CMFs de type Q-shift de longueur 10 [9]. L'image de droite est composée de 32 séquences complexes, représentées alternativement par leur parties réelle et imaginaire. Les blocs ① et ② sont respectivement associés à $\mathrm{Z}_J^{\nearrow(k)}$ et $\mathrm{Z}_J^{\nwarrow(k)}$.

en décomposant chaque $\mathrm{X}_{j-1}^{(k)}$ en quatre sous-séquences :

$$\forall l \in \{0\,..\,3\},\ \mathrm{X}_j^{(4k+l)} = \left(\mathrm{X}_{j-1}^{(k)} * \overline{\mathrm{G}^{(l)}}\right) \downarrow 2. \quad (1)$$

La WPT consiste donc en une succession de convolutions séparables, qui peut s'écrire comme une unique convolution dont les filtres sont montrés en Figure 1.

Transformée par *dual-tree* (DT-$\mathbb{C}$WPT) Malgré d'intéressantes propriétés, la WPT est instable par translation et ne présente que deux orientations : verticale et horizontale. Afin de remédier à ces faiblesses, N. Kingsbury [8] a conçu une transformée en ondelettes discrète basée sur deux paires de CMFs. Les images sont décomposées dans un *frame* (i.e., une base redondante) composé de signaux analytiques, autrement dit des motifs oscillants complexes et orientés. Cet algorithme a plus tard été généralisé aux paquets d'ondelettes [1].

En considérant une échelle fixée $J > 0$, supposons que nous ayons décomposé l'image d'entrée X en quatre représentations $\left\{\mathrm{X}_{\mathrm{a},\,J}^{(k)}\right\}$, $\left\{\mathrm{X}_{\mathrm{b},\,J}^{(k)}\right\}$, $\left\{\mathrm{X}_{\mathrm{c},\,J}^{(k)}\right\}$ et $\left\{\mathrm{X}_{\mathrm{d},\,J}^{(k)}\right\}$, avec $k \in \left\{0\,..\,4^J - 1\right\}$. Pour cela nous avons appliqué l'algorithme WPT avec quatre bancs de filtres méticuleusement choisis : $\left\{\mathrm{G}_{\mathrm{a}}^{(l)}\right\}$, $\left\{\mathrm{G}_{\mathrm{b}}^{(l)}\right\}$, $\left\{\mathrm{G}_{\mathrm{c}}^{(l)}\right\}$ et $\left\{\mathrm{G}_{\mathrm{d}}^{(l)}\right\}$, avec $l \in \{0\,..\,3\}$. Les 2×4^J séquences de coefficients complexes, notées $\mathrm{Z}_J^{\nearrow(k)}$ et $\mathrm{Z}_J^{\nwarrow(k)}$, sont alors calculés de cette manière :

$$\begin{pmatrix} \mathrm{Z}_J^{\nearrow(k)} \\ \mathrm{Z}_J^{\nwarrow(k)} \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{X}_{\mathrm{a},\,J}^{(k)} \\ \mathrm{X}_{\mathrm{d},\,J}^{(k)} \end{pmatrix} + i \cdot \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \mathrm{X}_{\mathrm{c},\,J}^{(k)} \\ \mathrm{X}_{\mathrm{b},\,J}^{(k)} \end{pmatrix}. \quad (2)$$

En plus de rendre compte de diverses orientations, cette représentation de X est stable par translation, si l'on considère le module des coefficients complexes.

De même que pour la WPT, les filtres résultants de cette succession d'opérateurs linéaires sont montrés en Figure 1.

Couches de convolution Dans un CNN, une couche de convolution 2D prend en entrée un batch de P images multicanales, chacune d'entre elles étant représentée par un ensemble de K séquences 2D (ex. : $K = 3$ pour les images couleur en entrée du réseau). Notons également L le nombre de canaux de sortie.

Définition 1. Un opérateur de convolution multicanal avec hyperparamètres [2] $s, d, q \in \mathbb{N}^*$, muni d'un tenseur de poids $\mathbf{W} \in \mathbb{U}^{\frac{K}{q} \times L}$ et d'un vecteur de biais $\boldsymbol{b} \in \mathbb{R}^L$, est une application linéaire

$$\begin{aligned} \mathcal{C}_{s,\,d}^{(q)}(\mathbf{W},\,\boldsymbol{b}) : \mathbb{U}^{P \times K} &\to \mathbb{U}^{P \times L} \\ \mathbf{X} &\mapsto \mathbf{Y}, \end{aligned} \quad (3)$$

telle que pour chaque $p \in \{1\,..\,P\}$, $l \in \{1\,..\,L\}$,

$$\mathrm{Y}_{p,\,l} = b_l + \sum_{k=1}^{K/q} \left(\mathrm{X}_{p,\,k_0(l)+k} * \left(\overline{\mathrm{W}_{k,\,l}} \uparrow d\right)\right) \downarrow s, \quad (4)$$

où $k_0(l) = \lfloor lq/L \rfloor \cdot K/q$ désigne le premier canal d'entrée connecté au l-ième canal de sortie.

Remarque. En pratique, les images sont de taille finie. Pour se ramener à la définition ci-dessus, elles peuvent être considérées comme des séquences infinies à support compact, ou bien périodiques. Les conséquences d'un tel choix ne sont pas discutées ici.

3 Modèles proposés

Soit $\mathbf{X} \in \mathbb{U}^{P \times K}$ un batch d'images couleurs ($K = 3$). Dans un premier temps, nous introduisons une architecture basée sur AlexNet, que nous adaptons ensuite au ResNet.

3.1 DT-$\mathbb{C}$WPT AlexNet

Dans un réseau AlexNet standard, la première couche est modélisée par un opérateur de convolution multicanal, tel que le tenseur de sortie $\mathbf{Y}_{\mathrm{alex}}$ s'écrive

$$\mathbf{Y}_{\mathrm{alex}} = \mathcal{C}_{4,\,1}^{(1)}(\mathbf{W}_{\mathrm{alex}},\,\boldsymbol{b}_{\mathrm{alex}}) \cdot \mathbf{X}, \quad (5)$$

avec $\mathbf{W}_{\mathrm{alex}} \in \mathbb{U}^{3 \times 64}$ et $\boldsymbol{b}_{\mathrm{alex}} \in \mathbb{R}^{64}$. De plus les noyaux de convolution ont leur support inclus dans une région de taille 11×11.

L'opérateur que nous proposons à la place, illustré en Figure 2, est constitué d'une DT-$\mathbb{C}$WPT suivie d'une couche de convolution 1×1. Nous verrons ensuite que, sous certaines conditions, il peut être modélisé sous la forme (5).

Bancs de filtres WPT La première étape du DT-$\mathbb{C}$WPT consiste à effectuer quatre WPTs en parallèle. Nous allons montrer qu'une WPT peut en fait s'écrire comme une succession d'opérateurs de convolution multicanaux.

Soit $J > 0$ l'échelle de la décomposition. Considérant $j \in \{1\,..\,J\}$, on note $K_j = 3 \times 4^{j-1}$ et $L_j = 4K_j$

2. Respectivement : stride, facteur de dilation, nombre de groupes contrôlant les connexions entre les canaux d'entrée et de sortie.

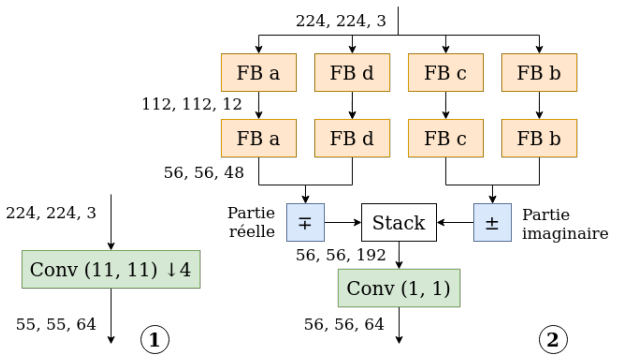

FIGURE 2 – ① Première couche d'AlexNet ; ② Modèle proposé en remplacement de la couche standard, avec $J = 2$. Chaque module orange (FB, pour *filter bank*) correspond à une étape de décomposition WPT. Seuls les modules verts (Conv) contiennent des paramètres appris. Les nombres entre chaque couche indiquent la hauteur et la largeur des images (i.e., taille du support) ainsi que le nombre de canaux.

les nombres respectifs de canaux d'entrée et de sortie à l'étape j. L'expression (1) est implémentée par un opérateur de convolution multicanal $\mathcal{C}^{(q_j)}_{s_j,1}(\mathbf{W}_j, \mathbf{0})$, avec $s_j = 2$ (sous-échantillonnage), $q_j = K_j$ (chaque canal d'entrée est traité séparément) et $\mathbf{W}_j \in \mathbb{U}^{1\times L_j}$. Les tenseurs de poids contiennent les filtres passe-haut et passe-bas introduits en Section 2, mais nous n'allons pas détailler précisément de quelle manière. En sortie de la décomposition, on obtient $\mathbf{X}' \in \mathbb{U}^{P\times L_J}$.
Par exemple, avec $J = 2$, on a $\mathbf{X}' \in \mathbb{U}^{P\times 48}$, avec

$$\mathbf{X}' = \mathcal{C}^{(12)}_{2,1}(\mathbf{W}_2, \mathbf{0}) \cdot \left(\mathcal{C}^{(3)}_{2,1}(\mathbf{W}_1, \mathbf{0}) \cdot \mathbf{X}\right). \tag{6}$$

Calcul des coefficients complexes Notons $\mathbf{X}'_{\mathrm{a}}$, $\mathbf{X}'_{\mathrm{b}}$, $\mathbf{X}'_{\mathrm{c}}$ et $\mathbf{X}'_{\mathrm{d}}$ les tenseurs de coefficients obtenus avec (6), en utilisant les quatre bancs de filtres introduits en Section 2. Le tenseur des coefficients complexes, noté $\mathbf{Z} \in \mathbb{U}^{P\times(4L_J)}$, est un empilement de quatre tenseurs $\mathbf{Z}^{\nearrow}_{\mathrm{re}}$, $\mathbf{Z}^{\nwarrow}_{\mathrm{re}}$, $\mathbf{Z}^{\nearrow}_{\mathrm{im}}$ et $\mathbf{Z}^{\nwarrow}_{\mathrm{im}} \in \mathbb{U}^{P\times L_J}$, tels que

$$\begin{aligned}
\mathbf{Z}^{\nearrow}_{\mathrm{re}} &= \mathbf{X}'_{\mathrm{a}} - \mathbf{X}'_{\mathrm{d}};\\
\mathbf{Z}^{\nwarrow}_{\mathrm{re}} &= \mathbf{X}'_{\mathrm{a}} + \mathbf{X}'_{\mathrm{d}};\\
\mathbf{Z}^{\nearrow}_{\mathrm{im}} &= \mathbf{X}'_{\mathrm{c}} + \mathbf{X}'_{\mathrm{b}};\\
\mathbf{Z}^{\nwarrow}_{\mathrm{im}} &= \mathbf{X}'_{\mathrm{c}} - \mathbf{X}'_{\mathrm{b}}.
\end{aligned} \tag{7}$$

Cette expression est une formulation tensorielle de (2), dans laquelle les parties réelles et imaginaires des coefficients complexes sont stockées séparément. Elle peut également être modélisée par un opérateur de convolution multicanal ; cela nécessite quelques détails techniques que nous n'expliciterons pas dans cet article.

Couche de convolution 1×1 Nous cherchons maintenant à sélectionner les paquets d'ondelettes les plus pertinents pour la classification d'images. Pour cela nous introduisons en sortie de DT-$\mathbb{C}$WPT une couche de convolution 1×1 librement apprise :

$$\mathbf{Y}_{\text{dt-alex}} = \mathcal{C}^{(1)}_{1,1}(\mathbf{W}_{\mathrm{mix}}, \boldsymbol{b}_{\mathrm{mix}}) \cdot \mathbf{Z}, \tag{8}$$

où $\mathbf{W}_{\mathrm{mix}} \in \mathbb{U}^{(4L_J)\times 64}$, le support de chaque noyau de convolution étant restreint au singleton $\{\mathbf{0}\}$. Ainsi, pour chaque $p \in \{1\,..\,P\}$ et $l \in \{1\,..\,64\}$, $\mathrm{Y}_{\text{dt-alex } p,\,l}$ est une combinaison linéaire des séquences de coefficients $\mathrm{Z}_{p,\,l'}$ pour $l' \in \{1\,..\,4L_J\}$.
Une autre motivation pour l'utilisation de cette couche est que le nombre de canaux de sortie doit être égal à 64 comme dans AlexNet. Qui plus est, chaque canal de sortie doit être influencé par l'ensemble des canaux RGB – c'est le sens du $^{(1)}$ en exposant dans l'expression (5). Or ce n'est pas le cas si l'on se restreint à la DT-$\mathbb{C}$WPT, qui se calcule séparément sur chacun des canaux d'entrée.

3.2 DT-$\mathbb{C}$WPT ResNet

Nous proposons ici une adaptation de notre modèle au ResNet. Dans le réseau standard, la première couche est modélisée par

$$\mathbf{Y}_{\mathrm{res}} = \mathcal{C}^{(1)}_{2,1}(\mathbf{W}_{\mathrm{res}}, \mathbf{0}) \cdot \mathbf{X}, \tag{9}$$

avec $\mathbf{W}_{\mathrm{res}} \in \mathbb{U}^{3\times 64}$ tel que les noyaux de convolution ont leur support inclus dans une région de taille 7×7.
L'opérateur proposé à la place est identique à la version proposée pour AlexNet, à l'exception de la couche de convolution 1×1, qui cette fois n'est pas biaisée :

$$\mathbf{Y}_{\text{dt-res}} = \mathcal{C}^{(1)}_{1,1}(\mathbf{W}_{\mathrm{mix}}, \mathbf{0}) \cdot \mathbf{Z}. \tag{10}$$

Nous verrons également que le choix de l'échelle de décomposition J varie entre les deux modèles.

3.3 Equivalence des opérateurs

Les modèles proposés ci-dessus sont conçus comme une succession d'opérateurs de convolution multicanaux. La proposition ci-dessous, dont la preuve est donnée en annexe, énonce qu'un tel enchaînement peut s'exprimer à l'aide d'un unique opérateur. Elle s'appuie sur le résultat bien connu que deux convolutions successives sont équivalentes à une seule convolution avec un noyau de plus grande taille. Cependant, nous présentons ici une approche plus générale, utilisant le formalisme introduit dans la Définition 1.
Soit $\mathcal{C}^{(q)}_{s,d}(\mathbf{W}, \boldsymbol{b})$ un opérateur de convolution multicanal, avec $(d = q = 1)$, $\mathbf{W} \in \mathbb{U}^{K\times L}$ et $\boldsymbol{b} \in \mathbb{R}^L$. On considère un second opérateur $\mathcal{C}^{(r)}_{t,1}(\mathbf{V}, \boldsymbol{a})$, avec $\mathbf{V} \in \mathbb{U}^{(L/r)\times L'}$ et $\boldsymbol{a} \in \mathbb{R}^{L'}$. On suppose de plus que L et L' sont tous deux divisibles par r.

Proposition 1. *La composition des deux opérateurs définis ci-dessus peut s'écrire comme un unique opérateur de convolution multicanal :*

$$\mathcal{C}^{(r)}_{t,1}(\mathbf{V}, \boldsymbol{a}) \circ \mathcal{C}^{(q)}_{s,d}(\mathbf{W}, \boldsymbol{b}) = \mathcal{C}^{(q')}_{s',d'}(\mathbf{W}', \boldsymbol{b}'), \tag{11}$$

avec $(s' = st)$ *et* $(d' = q' = 1)$. *De plus, le tenseur de poids résultant se calcule de la façon suivante :*

$$\overline{\mathbf{W}'} = \mathcal{C}^{(q_{\mathrm{w}})}_{s_{\mathrm{w}},\, d_{\mathrm{w}}}\left(\mathbf{V},\, \mathbf{0}\right) \cdot \overline{\mathbf{W}}, \tag{12}$$

avec $(s_{\mathrm{w}} = 1)$, $(d_{\mathrm{w}} = s)$ *et* $(q_{\mathrm{w}} = r)$. *Enfin, nous avons* $\boldsymbol{b}' = \boldsymbol{a} + f\left(\mathbf{V},\, \boldsymbol{b}\right)$, *où* f *est telle que* $f(\cdot,\, \mathbf{0}) = 0$.

En utilisant la Proposition 1, on montre par récurrence que les modèles proposés peuvent s'écrire sous la forme

$$\mathbf{Y}_{\text{dt-alex}} = \mathcal{C}^{(1)}_{(2^J),\,1}\left(\mathbf{W}_{\text{dt-alex}},\, \boldsymbol{b}_{\text{mix}}\right) \cdot \mathbf{X}; \tag{13}$$

$$\mathbf{Y}_{\text{dt-res}} = \mathcal{C}^{(1)}_{(2^J),\,1}\left(\mathbf{W}_{\text{dt-res}},\, \mathbf{0}\right) \cdot \mathbf{X}, \tag{14}$$

où $\mathbf{W}_{\text{dt-alex}}$ et $\mathbf{W}_{\text{dt-res}} \in \mathbb{U}^{3\times 64}$ sont obtenus par des itérations successives de (12). Ces résultats sont à comparer respectivement aux expressions (5) et (9) pour les modèles originaux.

Remarque. La Proposition 1 est utilisés à des fins d'analyse, mais ne doit pas en pratique se substituer aux algorithmes de décomposition.

Échelle de décomposition Une condition nécessaire pour obtenir l'identification avec les réseaux standards est d'avoir le même coefficient de sous-échantillonnage ; i.e., $2^J = 4$ pour AlexNet et 2 pour ResNet. En première approche, nous avons donc fixé l'échelle des coefficients à $J = 2$ pour AlexNet et $J = 1$ pour ResNet [3].
Cependant, comme on le verra plus loin, certains filtres appris par les réseaux standards ont des fréquences plus basses que celles obtenues par DT-$\mathbb{C}$WPT à ces échelles. Nous avons donc, dans un second temps, ajouté un niveau de décomposition pour chaque modèle. Afin de conserver le coefficient de sous-échantillonnage souhaité, le dernier niveau de décomposition se fait sans décimation, au prix d'une redondance plus élevée. On s'approche ici du concept de transformée en ondelettes stationnaire [17].

3.4 Réduction du nombre de paramètres

Seule la couche de convolution 1×1 placée après la décomposition en paquets d'ondelettes contient des paramètres qui vont évoluer pendant la phase d'apprentissage (voir Figure 2). En ce qui concerne la version DT-$\mathbb{C}$WPT d'AlexNet, le nombre de paramètres appris avec $J = 3$ est égal à $768 \times 64 = 49\,152$. C'est deux fois plus que pour l'architecture standard.
Cependant, il est possible de drastiquement diminuer ce chiffre sans dégrader les performances du modèle, en considérant 1. que la plupart des filtres DT-$\mathbb{C}$WPT génèrent des coefficients de très basse énergie, et 2. que l'on peut imposer des contraintes supplémentaires sur la couche de convolution 1×1. Il n'est en effet a priori pas souhaitable de combiner des filtres de fréquences et/ou d'orientations différentes.

3. Une DT-$\mathbb{C}$WPT nécessite en théorie au moins deux niveaux de décomposition. Nous avons contourné le problème en sur-échantillonnant les images d'entrée. Cela n'affecte pas l'expression (14).

Le modèle que nous avons implémenté à $J = 3$ a donc été modifié de la manière suivante :

- les filtres générant les séquences de coefficients de plus basse énergie ont été ignorés. Par conséquent, $\mathbf{Z} \in \mathbb{U}^{P\times 384}$ au lieu de $\mathbb{U}^{P\times 768}$;
- la couche de convolution 1×1 est éclatée en quatre couches parallèles afin de séparer les filtres ayant des échelles et des orientations différentes.

Une fois ces modifications faites, le modèle ne contient plus que $6\,144$ paramètres appris.
De même, sur la version DT-$\mathbb{C}$WPT du ResNet, nous avons modifié la couche de convolution 1×1 de manière à ce que le nombre de paramètres appris avec $J = 2$ soit égal à $3\,072$, contre $9\,408$ pour l'architecture standard.

4 Expériences

4.1 Détails de l'implémentation

Notre implémentation est basée sur PyTorch et CUDA. La DT-$\mathbb{C}$WPT est générée à partir de CMFs de type Q-shift de longueur 10 [9], qui possèdent les propriétés suffisantes pour obtenir des filtres approximativement analytiques.

Bases de données Nos modèles, construits sur les architectures AlexNet et ResNet34, ont été entraînés et évalués sur la base ImageNet ILSVRC2012 [22]. Le serveur d'évaluation en ligne n'étant plus disponible, nous avons mis de côté $100\,000$ images de la base d'entraînement (100 par classe) pour constituer une base de validation. Cette base est utilisée pour calculer la taux de précision au cours de l'apprentissage. Quant à la base de validation fournie par ImageNet, nous l'avons transformée en base de test sur laquelle nos modèles entraînés sont évalués.
Afin de montrer la généralisation de la performance de nos modèles, nous les avons également entraînés sur les bases PASCAL VOC 2012 [4] et COCO 2014 [13], en considérant le problème de classification multilabel. Pour ce faire, nous avons initialisé les réseaux avec les paramètres appris sur ImageNet (pré-entraînement). Une fois de plus nous n'avons pas eu accès aux serveurs d'évaluation en ligne ; c'est pourquoi la base de validation a été scindée en deux parties de tailles environ égales pour les phases de validation et de test.

Phase d'entraînement Pour chaque base de données, les réseaux ont été entraînés à l'aide d'un seul GPU. La procédure est inspirée de nombreux articles ILSVRC [6, 10, 24, 26]. Plus précisément, la fonction de coût (entropie croisée) est minimisée en utilisant l'algorithme de descente de gradient stochastique. Le réseau est alimenté par des batchs aléatoires de 256 images ($P = 256$), jusqu'à atteindre 100 cycles à travers la base d'entraînement (i.e., 100 époques, ce qui correspond à $461.4K$ itérations pour ImageNet). De plus, le moment est fixé à 0.9 et le facteur de *weight decay* à $5 \cdot 10^{-4}$. Quant au taux d'apprentissage, il est initialisé à 10^{-2}, et réduit d'un facteur 10 toutes les 25 époques.

TABLE 1 – Taux d'erreur sur AlexNet (test).

Modèle	*Nb params*	*ImageNet*		*VOC*	*COCO*
		Top-1	*Top-5*	*Erreur moyenne*	
$J = 2$	$12.3K$	42.3%	20.2%	23.1%	43.4%
$J = 3$	$6.1K$	**42.1%**	**20.0%**	**22.7%**	**43.0%**
STANDARD	$23.2K$	42.2%	**20.0%**	22.9%	43.4%
GELÉ	–	51.1%	27.8%	31.2%	51.3%

TABLE 2 – Taux d'erreur sur ResNet34 (test).

Modèle	*Nb params*	*ImageNet*		*VOC*	*COCO*
		Top-1	*Top-5*	*Erreur moyenne*	
$J = 1$	$3.1K$	26.3%	8.3%	11.9%	29.7%
$J = 2$	$3.1K$	26.1%	**8.1%**	**11.2%**	**29.2%**
STANDARD	$9.4K$	**26.0%**	**8.1%**	11.7%	29.6%
GELÉ	–	27.3%	8.8%	12.7%	31.0%

Afin de palier le problème de surapprentissage, nous avons suivi une procédure de *data augmentation* utilisée dans les réseaux Inception [26]. Les images sont d'abord normalisées à une moyenne et un écart-type fixé pour chaque canal RGB. Elles sont ensuite aléatoirement inversées (avec une probabilité de 50%) et rognées entre 8% et 100% de leur taille d'origine, avec un rapport de forme variant de $\frac{3}{4}$ à $\frac{4}{3}$, avant d'être redimensionnées à 224×224 pixels en utilisant une interpolation bilinéaire.

Évaluation du modèle La phase de test suit la procédure décrite par [10]. Tout d'abord, dix patchs sont extraits de l'image d'entrée. Pour chaque patch, un vecteur de probabilités est obtenu en calculant le *softmax* de la sortie du CNN. Enfin, ces vecteurs sont moyennés afin d'obtenir la prédiction finale. Les métriques utilisées sont d'une part les taux de précision top-1 et top-5 pour ImageNet, et d'autre part la précision moyenne (*AP score*) pour la classification multilabel [4].

Comparaison avec les modèles existants La performance de nos modèles est comparée aux réseaux standards AlexNet et ResNet34, entraînés selon la même procédure. De plus, nous souhaitions isoler la contribution spécifique de notre opérateur au pouvoir prédictif global du réseau. Pour ce faire, nous avons également entraîné des CNNs dans lesquels la première couche de convolution est gelée à ses paramètres initiaux.

4.2 Résultats et discussion

Pouvoir prédictif des modèles L'évolution de l'erreur au cours de l'apprentissage avec ImageNet est montrée en Figure 3, tandis que l'évaluation des réseaux entraînés sur l'ensemble de test est détaillée dans les Tables 1 et 2. Les meilleures performances apparaissent en gras. Concernant la classification multilabel (VOC et COCO), nous avons défini l'erreur moyenne par $\mathrm{E} = 1 - \Pi \in [0, 1]$, où Π désigne la précision moyenne.

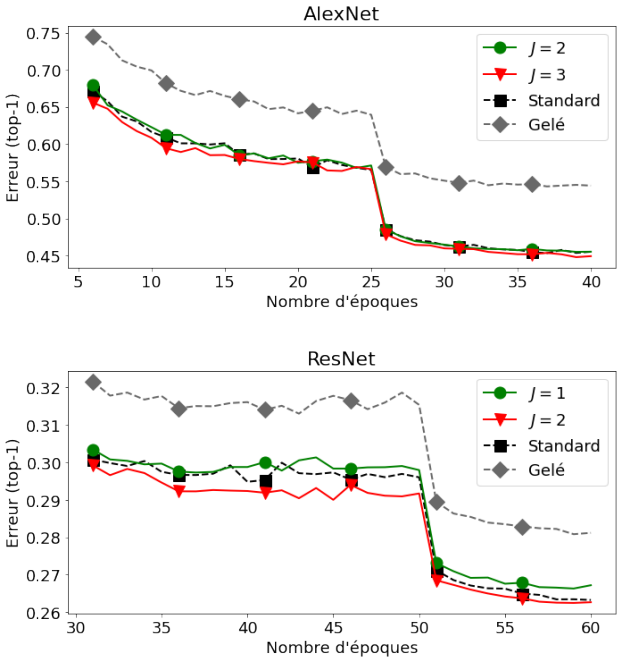

FIGURE 3 – Évolution de l'erreur (top-1) au cours de l'apprentissage, pour AlexNet (haut) et ResNet34 (bas). Les courbes sont tronquées pour plus de lisibilité. L'évaluation a été faite sur l'ensemble de validation, en redimensionnant le plus petit côté de l'image à 224 pixels et en extrayant un unique patch de taille 224×224 au centre.

On peut noter que les modèles proposés atteignent, voire dépassent la précision des architectures standard. Par ailleurs, les bons résultats obtenus sur VOC et COCO attestent de leur généralisabilité. Les résultats surpassant ceux des réseaux originaux sont indiqués en rouge.
Bien que les modèles proposés prétendent imiter le comportement des architectures standard, on y observe parfois de meilleurs résultats. Ceci peut s'expliquer par un apprentissage plus rapide dû au nombre réduit de paramètres, ou encore par la taille du support des filtres, qui dépasse celle des réseaux originaux (voir paragraphe ci-dessous).

Visualisation des filtres de convolution Une visualisation des filtres découlant de la Proposition 1 pour la version DT-$\mathbb{C}$WPT d'AlexNet, à savoir $\mathbf{W}_{\text{dt-alex}}$ introduit en (13), est proposée en Figure 4. Pour tout $l \in \{1\,..\,64\}$, les trois filtres reliant chaque canal d'entrée RGB au l-ième canal de sortie sont affichés sous forme d'image en couleurs. Des images similaires peuvent être obtenues avec ResNet34.
Nous insistons sur le fait que le calcul de ces filtres est utile à des fins d'analyse, mais n'entre pas en jeu dans la phase d'apprentissage. De plus, la taille du support des noyaux de convolution est plus grande que celle de la version standard. Pour des raisons de comparaison visuelle, nous n'en avons affiché que des patchs centraux de taille 11×11.

Fréquences caractéristiques Nous allons maintenant comparer les fréquences caractéristiques de nos modèles

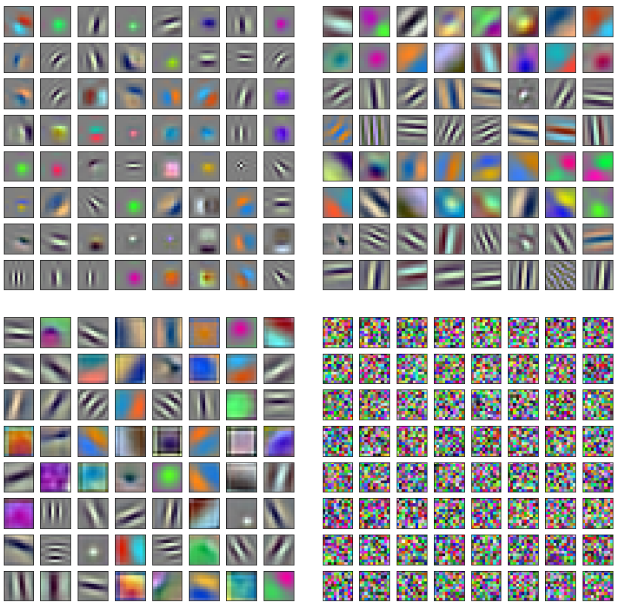

FIGURE 4 – Haut : $\mathbf{W}_{\text{dt-alex}}$ (DT-$\mathbb{C}$WPT AlexNet) pour $J = 2$ (gauche) et $J = 3$ (droite). On remarquera que dans ce dernier cas, quatre groupes de filtres sont reconnaissables (voir section 3.4). Bas : $\mathbf{W}_{\text{alex}}$ (AlexNet standard) après entraînement (gauche) et sans entraînement – couche gelée (droite). Le modèle avec $J = 3$ est très proche visuellement du réseau standard.

avec les architectures standard. Le calcul d'une telle métrique a été réalisé comme suit. Dans un premier temps, nous déterminons l'orientation principale du filtre, notée $\boldsymbol{n} \in \mathbb{R}^2$ ($\|\boldsymbol{n}\|_2 = 1$), en utilisant les valeurs propres du tenseur de structure [21]. Puis la fréquence caractéristique, notée $\boldsymbol{\nu}$, est estimée par le calcul d'un centre de masse :

$$\boldsymbol{\nu} = \frac{1}{\Pi(\boldsymbol{n})} \iint_{[-\pi,\,\pi]^2} \delta(\boldsymbol{\omega},\,\boldsymbol{n}) \widehat{\mathrm{X}}(\boldsymbol{\omega})^2\, \boldsymbol{\omega}\, \mathrm{d}^2\boldsymbol{\omega}, \qquad (15)$$

où

— $\delta(\boldsymbol{\omega},\,\boldsymbol{n}) \in [0,\,1]$ représente l'écart entre l'orientation principale et celle du vecteur fréquence $\boldsymbol{\omega}$: $\delta(\boldsymbol{\omega},\,\boldsymbol{n}) = \left\langle \boldsymbol{n},\, \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \right\rangle \times \left| \left\langle \boldsymbol{n},\, \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \right\rangle \right|$;

— $\widehat{\mathrm{X}}$ désigne la transformée de Fourier à temps discret de X ;

— $\Pi(\boldsymbol{n}) = \iint_{[-\pi,\,\pi]^2} \delta(\boldsymbol{\omega},\,\boldsymbol{n}) \widehat{\mathrm{X}}(\boldsymbol{\omega})^2\, \mathrm{d}^2\boldsymbol{\omega}$.

Les fréquences caractéristiques de nos modèles basés sur AlexNet sont affichées sous forme de nuages de points en Figure 5, et comparées avec celles du réseau standard. Nous n'avons conservé que les filtres passe-bande (i.e., dont la somme des éléments est quasi-nulle) qui présentent une orientation bien définie. Cette sélection se fait grâce à l'indice de cohérence, qui est une mesure de directionnalité d'un motif, également déduite des valeurs propres du tenseur de structure.

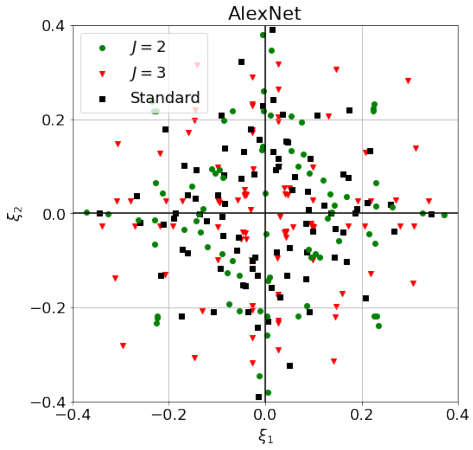

FIGURE 5 – Fréquences 2D caractéristiques de $\mathbf{W}_{\text{dt-alex}}$ pour $J = 2$ et $J = 3$, comparées à celles du réseau AlexNet standard. Les nuages de points ont été symétrisés pour plus de lisibilité. On remarque que pour $J = 2$, les filtres n'atteignent pas des fréquences suffisamment basses.

On remarque que pour $J = 2$, les filtres n'atteignent pas des fréquences suffisamment basses ; d'où l'introduction d'un niveau de décomposition supplémentaire. Une observation similaire peut être faite sur ResNet34.

5 Conclusion

Après une période de course effrénée vers la performance, de nombreuses recherches se tournent maintenant vers la compréhension des mécanismes d'apprentissage dans les CNNs. Dans cette optique, nous avons proposé de modéliser la première couche de certaines architectures populaires par un opérateur mathématique bien défini. Cet opérateur contient une version complexe et orientée de la transformée discrète en paquets d'ondelettes.
La prochaine étape consiste à étudier cet opérateur plus en détail. En particulier, nous cherchons à établir un lien quantitatif entre la sortie de la première couche d'activation du CNN et le module de coefficients complexes, en s'appuyant sur les travaux de [28]. Le but est ainsi de montrer la stabilité du réseau par translation, et d'identifier des leviers d'amélioration. Une autre piste de travail consiste à identifier l'espace de couleur optimal dans lequel effectuer la décomposition en paquets d'ondelettes.
Notre travail constitue ainsi une première étape vers une compréhension plus complète des réseaux existants, en se basant notamment sur la théorie des ondelettes.

Annexe – Preuve de la proposition 1

Démonstration. Soit $\mathbf{X} \in \mathbb{U}^{P \times K}$ un tenseur d'entrée. On note $\mathbf{Y} \in \mathbb{U}^{P \times L}$ and $\mathbf{Y}' \in \mathbb{U}^{P \times L'}$ les sorties des deux opérateurs de convolution introduits dans l'énoncé :

$$\mathbf{Y} = \mathcal{C}^{(1)}_{s,\,1}(\mathbf{W},\,\boldsymbol{b}) \cdot \mathbf{X}; \qquad (16)$$

$$\mathbf{Y}' = \mathcal{C}^{(r)}_{t,\,1}(\mathbf{V},\,\boldsymbol{a}) \cdot \mathbf{Y}. \qquad (17)$$

Soit $p \in \{1\,..\,P\}$. En utilisant la Définition 1, l'expression (17) devient, pour tout $l' \in \{1\,..\,L'\}$,

$$\mathrm{Y}'_{p,\,l'} = a_{l'} + \sum_{l=1}^{L/r} \left(\mathrm{Y}_{p,\,l_0(l')+l} * \overline{\mathrm{V}_{l,\,l'}} \right) \downarrow t, \tag{18}$$

où $l_0(l') = \lfloor l'r/L' \rfloor \cdot L/r$. De plus, pour tout $l \in \{1\,..\,L\}$, l'expression (16) s'écrit :

$$\mathrm{Y}_{p,\,l} = b_l + \sum_{k=1}^{K} \left(\mathrm{X}_{p,\,k} * \overline{\mathrm{W}_{k,\,l}} \right) \downarrow s. \tag{19}$$

La prochaine étape nécessite les deux lemmes suivants.

Lemme 1. *Soient* $\mathrm{U},\ \mathrm{V} \in \mathbb{U}$ *et* $b \in \mathbb{R}$. *On a :*

$$(b + \mathrm{U}) * \mathrm{V} = \left(b \cdot \sum_{\boldsymbol{n} \in \mathbb{Z}^2} \mathrm{V}[\boldsymbol{n}] \right) + (\mathrm{U} * \mathrm{V}). \tag{20}$$

Lemme 2. *Soient* $\mathrm{U},\ \mathrm{V} \in \mathbb{U}$ *et* $s,\ t \in \mathbb{N}^*$. *On a :*

$$((\mathrm{U} \downarrow s) * \mathrm{V}) \downarrow t = (\mathrm{U} * (\mathrm{V} \uparrow s)) \downarrow (st). \tag{21}$$

En injectant l'expression (19) dans (18) et en utilisant le Lemme 1, on obtient

$$\mathrm{Y}'_{p,\,l'} = b'_{l'} + \sum_{l=1}^{L/r} \left(\sum_{k=1}^{K} \left(\mathrm{X}_{p,\,k} * \overline{\mathrm{W}_{k,\,l_0(l')+l}} \right) \downarrow s * \left(\overline{\mathrm{V}_{l,l'} \downarrow u} \right) \right) \downarrow t, \tag{22}$$

où

$$b'_{l'} = a_{l'} + \sum_{l=1}^{L/r} \left(b_{l_0(l')+l} \cdot \sum_{\boldsymbol{n} \in \mathbb{Z}^2} \mathrm{V}_{l,\,l'}[\boldsymbol{n}] \right). \tag{23}$$

Finalement, en intervertissant les deux sommes et en utilisant le Lemme 2, on obtient

$$\mathrm{Y}'_{p,\,l'} = b'_{l'} + \sum_{k=1}^{K} \left(\mathrm{X}_{p,\,k} * \sum_{l=1}^{L/r} \left(\overline{\mathrm{W}_{k,\,l_0(l')+l}} * \overline{\mathrm{V}_{l,l'} \uparrow s} \right) \right) \downarrow (st). \tag{24}$$

En notant $\mathbf{W}' \in \mathbb{U}^{K \times L'}$ tel que pour tous k, l',

$$\overline{\mathrm{W}'_{k,\,l'}} = \sum_{l=1}^{L/r} \left(\overline{\mathrm{W}_{k,\,l_0(l')+l}} * \overline{\mathrm{V}_{l,l'} \uparrow s} \right) \tag{25}$$

l'expression (24) devient

$$\mathrm{Y}'_{p,\,l'} = b'_{l'} + \sum_{k=1}^{K} \left(\mathrm{X}_{p,\,k} * \overline{\mathrm{W}'_{k,\,l'}} \right) \downarrow (st). \tag{26}$$

En appliquant la Définition 1 à l'expression (26), on obtient

$$\mathbf{Y}' = \mathcal{C}^{(1)}_{s',\,1}(\mathbf{W}',\, \boldsymbol{b}') \cdot \mathbf{X}, \tag{27}$$

où $s' = st$.
Ceci est vrai pour tout $\mathbf{X}$, ce qui prouve (11).
Finalement, en appliquant la Finally, by applying Définition 1 à l'expression (25), on obtient

$$\overline{\mathbf{W}'} = \mathcal{C}^{(r)}_{1,\,s}\,(\mathbf{V},\, \mathbf{0}) \cdot \overline{\mathbf{W}}, \tag{28}$$

ce qui prouve (12).
En notant $f\,(\mathbf{V},\, \boldsymbol{b}) = \boldsymbol{b}' - \boldsymbol{a} \in \mathbb{R}^{L'}$, on obtient $f\,(\mathbf{V},\, \mathbf{0}) = \mathbf{0}$, comme précisé dans l'énoncé de la Proposition 1. □

Remerciements

Ces travaux ont été partiellement financés par le LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01, soutenu par le programme Investissement d'avenir), ainsi que le projet AVENUE (ANR-18-CE23-0011).

Références

[1] Ilker Bayram and Ivan W. Selesnick. On the dual-tree complex wavelet packet and M-band transforms. *IEEE Transactions on Signal Processing*, 2008.

[2] Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *CVPR*, 2010.

[3] Joan Bruna and Stéphane Mallat. Invariant Scattering Convolution Networks. *TPAMI*, 35(8) :1872–1886, 2013.

[4] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge : A Retrospective. *IJCV*, 111(1) :98–136, 2015.

[5] Shin Fujieda, Kohei Takayama, and Toshiya Hachisuka. Wavelet Convolutional Neural Networks for Texture Classification. *arXiv 1707.07394*, 2017.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[7] Ke Huang and Selin Aviyente. Wavelet feature selection for image classification. *IEEE Transactions on Image Processing*, 2008.

[8] Nick Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Applied and computational harmonic analysis*, 10(3) :234–253, 2001.

[9] Nick Kingsbury. Design of Q-shift complex wavelets for image processing using frequency domain energy minimization. In *ICIP*, 2003.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[11] Andrew Laine and Jian Fan. Texture Classification by Wavelet Packet Signatures. *TPAMI*, 1993.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553) :436–444, 2015.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO : Common Objects in Context. *arXiv 1405.0312*, 2015.

[14] Hongya Lu, Haifeng Wang, Qianqian Zhang, Daehan Won, and Sang Won Yoon. A Dual-Tree Complex Wavelet Transform Based Convolutional Neural Network for Human Thyroid Medical Image Segmentation. In *Intl. Conf. Healthcare Informatics*, 2018.

[15] Shangzhen Luan, Chen Chen, Baochang Zhang, Jungong Han, and Jianzhuang Liu. Gabor Convolutional Networks. *IEEE Transactions on Image Processing*, 27(9) :4357–4366, 2018.

[16] Stéphane Mallat. *A Wavelet Tour of Signal Processing : The Sparse Way*. Elsevier/Academic Press, 2009.

[17] G. P. Nason and B. W. Silverman. The Stationary Wavelet Transform and some Statistical Applications. In *Wavelets and Statistics*, Lecture Notes in Statistics, pages 281–299. Springer, New York, NY, 1995.

[18] Edouard Oyallon, Eugene Belilovsky, and Sergey Zagoruyko. Scaling the Scattering Transform : Deep Hybrid Networks. In *ICCV*, 2017.

[19] Edouard Oyallon, Eugene Belilovsky, Sergey Zagoruyko, and Michal Valko. Compressing the Input for CNNs with the First-Order Scattering Transform. In *ECCV*, 2018.

[20] Stefan Pittner and Sagar V. Kamarthi. Feature extraction from wavelet coefficients for pattern recognition tasks. *TPAMI*, 21(1) :83–88, 1999.

[21] Kévin Polisano. *Modélisation de Textures Anisotropes Par La Transformée En Ondelettes Monogéniques*. Doctoral thesis, Université Grenoble Alpes (ComUE), 2017.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3) :211–252, 2015.

[23] Syed Shakib Sarwar, Priyadarshini Panda, and Kaushik Roy. Gabor filter assisted energy efficient fast learning Convolutional Neural Networks. In *IEEE/ACM International Symposium on Low Power Electronics and Design*, 2017.

[24] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

[25] Amarjot Singh and Nick Kingsbury. Dual-Tree wavelet scattering network with parametric log transformation for object classification. In *ICASSP*, 2017.

[26] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[27] Matej Ulicny, Vladimir A. Krylov, and Rozenn Dahyot. Harmonic Networks for Image Classification. In *BMVC*, 2019.

[28] Irène Waldspurger. *Wavelet Transform Modulus : Phase Retrieval and Scattering*. Doctoral thesis, Ecole normale supérieure, Paris, 2015.

[29] Travis Williams and Robert Li. Advanced Image Classification Using Wavelets and Convolutional Neural Networks. In *Intl. Conf. Machine Learning and Applications*, 2016.

[30] Travis Williams and Robert Li. Wavelet Pooling for Convolutional Neural Networks. In *ICLR*, 2018.

[31] Gary G. Yen. Wavelet packet feature extraction for vibration monitoring. *IEEE Transactions on Industrial Electronics*, 2000.

[32] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks ? In *NeurIPS*, 2014.

[33] John Zarka, Louis Thiry, Tomás Angles, and Stéphane Mallat. Deep Network Classification by Scattering and Homotopy Dictionary Learning. In *ICLR*, 2020.