# Sujet de thèse
# Apprentissage profond basé sur la conception de modèles efficaces : applications à la surveillance maritime

## 1 Introduction

Ces dernières années, les réseaux de neurones profonds (DNNs pour Deep Neural Networks) ont considérablement repoussé les limites de l'intelligence artificielle dans un large éventail de tâches, notamment la reconnaissance d'objets à partir d'images [1], la reconnaissance vocale [2], la traduction automatique [3], etc. Les réseaux de neurones profonds nécessitent beaucoup de calcul et de mémoire, ce qui les rend difficiles à déployer sur des équipements embarqués avec des ressources de calcul limitées. Ces réseaux profonds sont caractérisés par des millions voire des milliards de paramètres et sont presque exclusivement entraînés en utilisant une ou plusieurs cartes graphique (GPU) très rapides et gourmandes en énergie. Considérons un exemple avec le modèle de pointe VGG-16 [4], il est constitué de 138,34 millions de paramètres, occupant plus de 500 Mo d'espace de stockage, 15,5 milliards d'opérations de cumul (MAC) et nécessite 30,94 milliards d'opérations en virgule flottante (FLOP) pour classer une seule image. Cela prend plusieurs minutes dans la phase d'inférence sur un appareil mobile ayant une capacité de calcul et des ressources de mémoire limitées. Ces réseaux profonds nécessitent donc énormément de données, de calcul, de mémoire et d'énergie, ce qui les rend difficiles à utiliser et à déployer dans des applications réelles sur des équipements tels que smartphones, tablettes et systèmes embarqués. La compression des modèles de réseaux profonds et la réduction de la consommation d'énergie, tout en préservant les performances prédictives, revêt une importance cruciale pour le déploiement de réseaux profonds dans un tel contexte. C'est pour cela que les tendances récentes se concentrent sur le déploiement d'applications en temps réel telles que YOLO [5] ou sur des ressources limitées (par exemple, MobileNet [6]). Dans le cadre de cette thèse, nous nous concentrerons sur la compression des réseaux de neurones pour surmonter ce défi en réduisant les besoins en stockage, en consommation d'énergie, et la complexité de calcul dans la phase d'inférence des réseaux de neurones sans que cela n'affecte leur précision. Le but est de déployer les modèles compressés sur des équipements embarqués tels que les caméras intelligentes ou les drones (AUV, ROV, *etc*). Ces systèmes seront ensuite utilisés pour des tâches de vision par ordinateur telles que l'analyse de scènes dynamiques [7, 8, 9, 10, 11], ou la détection/reconnaissance d'objets dans des scènes maritimes ou sous-marines. Cela aura un lien avec d'autres projets portés par notre équipe, notamment le projet Rapid DGA UHV-MANTA et le projet ANR Astrid ROV-Chasseur.

## 2 Etat de l'art

Les travaux récents dans la littérature visent à concevoir des modèles DNNs pour optimiser la précision tout en minimisant l'énergie utilisée et les coûts/temps de calculs. Nous rappelons ci-après les divers axes qui ont été explorés pour compresser les réseaux de neurones. Ceux-ci peuvent être brièvement classés en deux groupes principaux.

## 2.1 Réduire la précision des calculs

Le premier groupe vise à réduire la précision des opérations et des opérandes dans le fonctionnement des réseaux de neurones tels que la quantification non uniforme, le partage du poids, la réduction de la largeur de bit, la conversion de représentation des nombres. Les recherches sur la réduction de précision se sont souvent concentrées sur la réduction de la précision des paramètres, ceux-ci étant directement liés à l'empreinte de mémoire des modèles DNNs. Une autre approche consiste à traiter la quantification sur les couches d'activation dans la phase d'inférence.
*La quantification*, qui cartographie les données sur un plus petit ensemble de niveaux quantifiés, est le moyen le plus utilisé dans cette direction pour réduire les coûts de stockage et la complexité de calcul des DNNs. Cette direction peut également être référée en terme de *partage de poids*, ce qui oblige alors plusieurs poids à partager une seule et même valeur. Il peut être appliqué directement en utilisant une distance uniforme entre les niveaux de quantification ou une fonction logarithmique sur une distribution non uniforme lorsque la distance entre les niveaux varie. Il peut également être appris à partir de données en utilisant des techniques d'apprentissage non supervisées.

La quantification uniforme est prise en compte lorsque les niveaux sont uniformément espacés. La représentation en point fixe dynamique [12] est adoptée pour représenter une virgule flottante de 32 bits par un point fixe dynamique de 8 bits afin de réduire le poids des paramètres et les couches d'activation dans les DNNs. L'utilisation d'un point fixe 8 bits a aussi de nombreux impacts bénéfiques sur les coûts d'énergie et de mémoire, car une opération de multiplication consomme 15-18 fois plus d'énergie et 12-27 fois plus de mémoire par rapport à l'opération correspondante sur une virgule fixe ou flottante 32 bits [13]. Dans le cas de la quantification non uniforme, il existe deux approches populaires. La quantification du domaine de journalisation est prise en compte lorsque les niveaux de quantification sont attribués en fonction d'une distribution logarithmique. Dans [14], les poids sont quantifiés sur des puissances de deux, donc la multiplication peut être remplacée par un décalage de bit. D'autre part, la quantification peut être apprise à partir des données. Cela peut être fait en utilisant une fonction de hachage [15] ou regroupement k-moyen [16].
*Réseaux binaires (BNN) :* La quantification de poids peut même aller jusqu'à un seul bit, cette direction de recherche est souvent orientée vers des réseaux binaires inspirés par les modèles binaires locaux. Binary Connect [17] propose des poids binaires ($-1$ et $1$) pour éviter les opérations de multiplication et n'utiliser que des opérateurs d'addition et de soustraction. Cette idée est ensuite étendue dans les BNN [18] pour traiter des activations binaires. Différents travaux [19, 20, 21, 22] explorent cette direction pour améliorer la précision en réduisant la plage dynamique d'activation [19], en prenant en compte 2 bits dans les cartes d'activation [20, 21] ou en prenant en compte du codage ternaire [23].

## 2.2 Réduire la complexité ou la taille du modèle

Alors que la première approche consistait à se concentrer sur la réduction de la taille de chaque opération ou opérande, telle que le poids ou l'activation, la seconde direction de recherche a pour but de réduire le nombre d'opérations et la taille du modèle. Cela inclut différentes techniques telles que l'élagage du réseau, la conception d'architecture de réseau et la distillation des connaissances.

*Élagage du réseau :* Un réseau profond est souvent surparamétrisé pour faciliter sa formation. Ses poids sont alors redondants et une grande quantité de poids pourrait être supprimée. C'est le principe d'élagage d'un réseau. Il existe différents critères pour supprimer les poids qui sont moins importants. Dans un premier travail, appelé lésions cérébrales optimales [24], la saillance du poids est calculée en basant sur la performance d'entraînement. En raison des calculs onéreux, cette métrique est simplement remplacée par la magnitude du poids [25] dans les DNNs récents à grande échelle. Cependant, le nombre de poids retirés n'implique pas toujours une grande diminution en matière d'énergie consommée. Par exemple, dans le réseau VGG-16, les poids dans les couches entièrement connectées représentent 90 % du poids total, mais les poids dans les couches à convolution consomment 90 % de l'énergie totale. Par conséquent, dans [26], l'élagage en fonction du poids est régi par une métrique d'énergie, appelée "conscience de l'énergie", qui permet de réduire l'énergie totale consommée des DNN. Après un processus d'élagage, il a également été envisagé le stockage efficace des poids épars en utilisant des techniques de compression conventionnelles (par exemple, le codage de Huffman) [16] ou en favorisant les multiplications matrice-vecteur.

*Approximation de rang faible :* Il s'agit d'une approche prédominante qui consiste à modéliser les poids par des tableaux de données multidimensionnelles, aussi appelés *tenseurs*, et à les décomposer en une combinaison de tenseurs de rang et d'ordre plus faibles. Cela permet de compresser efficacement les DNNs car elle traite une couche convolutive comme un tenseur 4D et une couche entièrement connectée comme un tenseur 2D. Cette approche a été explorée en utilisant des décompositions en train de tenseurs [27], des décompositions de Tucker [28] et des décompositions canoniques polyadiques [29].

*Apprentissage des architectures de réseau :* Bien que la plupart des DNNs de pointe soient généralement conçus par des experts, une architecture de réseau efficace peut être obtenue en apprenant automatiquement les architectures de réseau. Cela comprend trois composants principaux : (1) l'espace de recherche, (2) l'algorithme d'optimisation et (3) l'évaluation des performances. Certains travaux récents [30] proposent d'apprendre automatiquement l'architecture neuronale grâce à l'apprentissage par renforcement. Cependant, l'espace de recherche de ces méthodes étant extrêmement vaste, il est nécessaire de former des centaines de modèles pour pouvoir les évaluer.

*Distillation de connaissances :* Il transfère les connaissances acquises par le modèle complexe ou la moyenne des prédictions de différents modèles (“enseignant”) au modèle plus simple (“élève”). Le réseau d'étudiants peut alors atteindre une meilleure précision par rapport à une formation directe à partir de la même base de données d'apprentissage [31].

# 3 Objectifs des travaux de recherche menés dans le cadre de cette proposition de thèse

Les objectifs de cette thèse sont :

— De développer de nouveaux algorithmes pour la compression des réseaux de neurones afin de les embarquer sur des appareils mobiles disposant de ressources limitées en matière de mémoire et de capacité de calcul

— De parvenir à réduire la consommation d'énergie sur ces équipements tout en préservant la précision quant aux résultats obtenus. Cela facilitera l'utilisation généralisée du DNN au niveau d'un plus grand nombre d'applications courantes, nécessitant une mise en oeuvre sur des équipements embarqués (réseaux de capteurs, etc.).

— De considérer des applications en surveillance maritime en déployant des modèles profonds efficaces sur des équipements embarqués dédiés (drone, ROV, etc.).

## 3.1 Axes de recherche

À cet effet, plusieurs axes de recherche seront donc considérés :

1. **Réduction de la complexité de calcul des modèles profonds**

   Plusieurs voies seront considérées. Tout d'abord, **la quantification** sera étudiée afin de réduire les coûts de stockage des paramètres du modèle et d'accélérer les opérations de convolution. Nous considérerons des techniques de quantification adaptative ou bien logarithmique pour obtenir une réduction efficace du volume de poids des paramètres et du coût d'exécution des opérations multiplicatives. Une autre piste consistera à étudier la quantification apprise à partir des valeurs des paramètres. Les méthodes existantes, qui utilisent une fonction de hachage [15] ou un regroupement $k$-moyennes [16] imposent certaines restrictions quant à la qualité de la quantification. Nous étudierons dans cette thèse plusieurs autres techniques de regroupement afin d'éviter certaines limitations. D'autre part, certaines techniques classiques de compression avec perte seront également exploitées afin de compresser les poids de chacune des couches. La deuxième piste s'intéressera à **l'élagage du réseau de neurones** pour diminuer sa taille. Ce principe est considéré pour réduire les synapses entre des neurones, supprimer des neurones inutiles, ou bien éliminer des canaux convolutifs ainsi que des couches d'activation inefficaces. Les couches convolutives consomment jusqu'à 90 % d'énergie des DNNs. L'élagage du réseau, qui réduit les structures de réseau (poids, filtre, neurone, par exemple), constitue alors une bonne solution pour réduire le stockage en mémoire et la consommation d'énergie. Notre objectif sera d'étudier des nouvelles métriques afin d'estimer l'importance des poids et/ou des neurones afin de retailler un réseau efficacement en évitant les limitations des métriques existantes [24, 16]. L'objectif étant la suppression des poids et des neurones non importants, permettant ainsi une réduction significative de la taille du modèle et de la complexité de calcul des DNNs. En outre, déterminer la redondance entre les filtres de chaque couche permet également d'éliminer les filtres inutiles.

La suppression d'un filtre peut être efficacement réalisée en abordant des mesures statistiques (par exemple : corrélation, mesures de similarité) ou la théorie de l'information (par exemple : information mutuelle, entropie) ou encore en appliquant des techniques d'apprentissage automatique (par exemple : ACP - analyse en composants principales, ACI - analyse en composantes indépendantes) sur ces filtres.

2. **Représentations parcimonieuses et de rang faible pour la modélisation des DNNs**

   Un noyau de convolution dans des réseaux de neurones est typiquement un tenseur d'ordre quatre, i.e., un tableau de données à 4 dimensions. Le constat évident est qu'il y a souvent une forte redondance d'information au niveau de ces tenseurs pourtant entièrememnt caractérisés par leur variables latentes (matrices facteurs). Les décompositions tensorielles [32, 33] constituent donc une piste particulièrement prometteuse dans la conception de modèles profonds efficaces, i.e., permettant d'éliminer la redondance dans les noyaux convolutifs et les couches d'activation. Le principe est de trouver des décompositions tensorielles de rang faible, permettant de décomposer une couche convolutive en plusieurs couches plus petites. Bien qu'il y ait plus de couches après la décomposition, le nombre total d'opérations à virgule flottante et de poids sera plus petit. Ce principe permet de rendre le poids du modèle profond plus petit et donc plus efficace en terme de complexité algorithmique et de volume de stockage. Les méthodes existantes dans cette direction reposent souvent sur la décomposition en valeurs singulières, la décomposition canonique polyadique (CP), ou bien encore la décomposition de Tucker. Notre objectif sera d'introduire une méthode efficace de décomposition tensorielle et de l'utiliser pour entraîner et compresser des CNNs.

   D'autre part, dans les modèles DNNs modernes, la fonction d'activation ReLU [34] est souvent utilisée comme transformation non linéaire pour construire des couches d'activation en raison de son efficacité de calcul et de sa vitesse de convergence dans la phase d'apprentissage. D'autre part, cela crée une représentation parcimonieuse au niveau de la couche d'activation, toutes les valeurs négatives étant converties en 0. Notre objectif est de proposer une représentation efficace des couches d'activation clairsemées / parcimonieuses. Une piste possible consistera à utiliser une méthode de compression de données sans perte, telle que les algorithmes de codage LZW ou de Huffman, pour réduire efficacement le stockage des mappages d'activation. Pour l'étape suivante, nous nous concentrerons sur les opérations efficaces dans les couches de convolution afin de traiter directement ce type de représentation.

3. **Développement d'architectures efficaces de réseau de neurones**

   Dans cette approche, nous nous concentrerons tout d'abord sur **la conception d'une architecture efficace** de modèles profonds en nous appuyant sur l'amélioration des concepts intelligents tels que : convolution $1 \times 1$ [35], convolution séparable en profondeur [6], réseau de compression et d'excitation [36], convolution clairsemée structurée entrelacée [37], convolution de groupe appris [38], la recherche des architectures de réseau [39], etc. Ensuite, nous nous intéresserons à concevoir des **réseaux de neurones binaires** ou ternaires qui ont un coût de calcul très faible par rapport à celui

TABLE 1 – Planning de travail et résultats espérés

| **Milestones** | **Durée** | **Output** |
|---|---|---|
| État de l'art en apprentissage profond, et en compression des réseaux de neurones | T0+6 (6 mois) | Rapport d'avancement |
| Étude et proposition d'architectures efficaces des modèles DNNs | T0+15 (8 mois) | Rapport d'avancement, 1 article de conférence |
| Étude et proposition de méthodes efficaces pour réduire la complexité et/ou la taille des modèles DNNs | T0+24 (8 mois) | Rapport d'avancement, 1 article de revue, 1 article de conférence |
| Étude et proposition de méthodes efficaces pour compresser des DNN en s'appuyant sur des décompositions tensorielles | T0+30 (8 mois) | Rapport d'avancement, 1 article de revue, 1 article de conférence, éventuel brevet ou transfert de technologie |
| Validation et rédaction du manuscrit de thèse | T0+36 (6 mois) | Thèse de doctorat, Applications industrielles |

des réseaux convolutifs conventionnels. Nous nous concentrerons sur la décomposition d'un filtre convolutif en un ensemble de filtres convolutifs fixes et prédéfinis qui ne seront pas mis à jour pendant le processus d'apprentissage pour réduire efficacement le stockage de poids. Ces filtres ne sont pas forcément binaires comme dans l'algorithme LBCNN [40] pour ne pas affecter les performances du modèle. Nous nous intéresserons également à des techniques de quantification vectorielle (pas seulement la quantification binaire/ternaire comme dans [23] qui donne la perte de précision) afin de compresser plus efficacement le réseau.

### 3.2 Applications à la surveillance maritime

Finalement, cette thèse vise également à déployer les modèles profonds efficaces sur des équipements embarqués tels que drones ou ROV pour les applications en surveillance maritime qui sont actuellement développées dans notre équipe. Dans le cadre du projet DGA Rapid Manta, porté par Nadège THIRION-MOREAU, nous nous intéressons à développer un drone intelligent permettant d'éviter automatiquement des obstacles sur la mer. D'autre part, le projet ANR Astrid ROV-Chasseur, porté par Thanh Phuong NGUYEN, s'intéresse à la détection et la reconnaissance des objets spécifiques sous-marins. Des résultats théoriques dans le cadre de cette thèse mènent directement aux progrès considérables en surveillance maritime liées à ces projets ci-dessus.

## 4 Planning de travail prévisionnel

Pour atteindre les objectifs proposés, voici le plan de mise en œuvre et les principaux résultats souhaités (voir Table 1).

## 5 Supervision

La thèse sera co-dirigée par :

Thanh Phuong NGUYEN, Maître de conférences (HDR)
Université de Toulon, France
Équipe SIIM, laboratoire LIS
email : tpnguyen@univ-tln.fr
page web : `http://tpnguyen.univ-tln.fr`

et

Yassine ZNIYED, Maître de conférences
Université de Toulon, France
Équipe SIIM, laboratoire LIS
email : zniyed@univ-tln.fr
page web : `https://yzniyed.blogspot.com/p/about-me.html`

**Thanh Phuong NGUYEN** a obtenu son doctorat en informatique à l'Université de Nancy (maintenant l'Université de Lorraine) en 2010 et puis son diplôme HDR (Habilitation à Diriger des Recherches) en 2021. Il a travaillé comme chercheur postdoctoral au CMM, Mines Paristech (2011) et puis à l'ENSTA Paristech (2012-2015). Il est actuellement Maître de conférences (HDR) en informatique à l'Université de Toulon et au laboratoire LIS UMR CNRS 7020. Ses recherches portent sur la classification des motifs, la vision par ordinateur, l'analyse de formes et, plus récemment, la texture dynamique, la reconnaissance d'action. Il a publié 24 articles dans des revues internationales présentant des facteurs d'impact élevés et plus de 30 conférences internationales.

**Yassine ZNIYED** a obtenu son diplôme d'ingénieur et M2R en traitement du signal de l'EHTP (Maroc) et de l'école Centrale de Marseille, en 2016. En 2019, il obtient un doctorat en traitement du signal et des images de l'université de Paris-Saclay. De décembre 2019 à août 2021, il a travaillé en tant que chercheur postdoctorant au laboratoire CRAN à Nancy. Depuis septembre 2021, il est maître de conférences à l'université de Toulon, et est membre du Laboratoire d'Informatique et Systèmes (LIS) UMR CNRS 7020. Ses recherches portent sur le développement de nouvelles méthodes tensorielles pour le traitement du signal et l'apprentissage automatique. Ses activités de recherche sont principalement axées sur l'algèbre linéaire/multilinéaire et le traitement statistique du signal.

## Références

[1] He, K., Zhang, X., Ren, S., Sun, J. : Deep residual learning for image recognition. In : CVPR. (2016) 770–778

[2] Graves, A., Mohamed, A., Hinton, G.E. : Speech recognition with deep recurrent neural networks. In : IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013. (2013) 6645–6649

[3] Devlin, J., Chang, M., Lee, K., Toutanova, K. : BERT : pre-training of deep bidirectional transformers for language understanding. In : Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). (2019) 4171–4186

[4] Simonyan, K., Zisserman, A. : Very deep convolutional networks for large-scale image recognition. In : ICLR. (2015)

[5] Redmon, J., Farhadi, A. : Yolov3 : An incremental improvement. arXiv (2018)

[6] Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L. : Mobilenetv2 : Inverted residuals and linear bottlenecks. In : CVPR. (2018) 4510–4520

[7] Bechar, I., Lelore, T., Bouchara, F., Guis, V., Grimaldi, M. : Object segmentation from a dynamic background using a pixelwise rigidity criterion and application to maritime target recognition. In : 2014 IEEE International Conference on Image Processing, ICIP 2014, Paris, France, October 27-30, 2014. (2014) 363–367

[8] Nguyen, X.S., Mouaddib, A., Nguyen, T.P., Jeanpierre, L. : Action recognition in depth videos using hierarchical gaussian descriptor. Multimedia Tools Appl. **77**(16) (2018) 21617–21652

[9] Nguyen, X.S., Nguyen, T.P., Charpillet, F., Vu, N. : Local derivative pattern for action recognition in depth images. Multimedia Tools Appl. **77**(7) (2018) 8531–8549

[10] Nguyen, T.T., Nguyen, T.P., Bouchara, F. : Completed statistical adaptive patterns on three orthogonal planes for recognition of dynamic textures and scenes. JEI **27**(05) (2018) 053044

[11] Nguyen, T.T., Nguyen, T.P., Bouchara, F. : Smooth-invariant gaussian features for dynamic texture recognition. In : ICIP. (2019) 4400–4404

[12] Ma, Y., Suda, N., Cao, Y., Seo, J., Vrudhula, S.B.K. : Scalable and modularized RTL compilation of convolutional neural networks onto FPGA. In : FPL. (2016) 1–8

[13] Horowitz, M. : Computing's energy problem (and what we can do about it). In : ISSCC. (2014)

[14] Gysel, P., Motamedi, M., Ghiasi, S. : Hardware-oriented approximation of convolutional neural networks. CoRR **abs/1604.03168** (2016)

[15] Chen, W., Wilson, J.T., Tyree, S., Weinberger, K.Q., Chen, Y. : Compressing neural networks with the hashing trick. (2015)

[16] Han, S., Mao, H., Dally, W.J. : Deep compression : Compressing deep neural networks with pruning, trained quantization and huffman coding. In : ICLR. (2016)

[17] Courbariaux, M., Bengio, Y., David, J. : Binaryconnect : Training deep neural networks with binary weights during propagations. In : NIPS. (2015)

[18] Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., Bengio, Y. : Binarized neural networks : Training deep neural networks with weights and activations constrained to +1 or -1. arXiv preprint arXiv :1602.02830 (2016)

[19] Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A. : XNOR-Net : ImageNet Classification Using Binary Convolutional Neural Networks. In : ECCV. (2016) 525–542

[20] Cai, Z., He, X., Sun, J., Vasconcelos, N. : Deep learning with low precision by half-wave gaussian quantization. CoRR **abs/1702.00953** (2017)

[21] Zhou, S., Ni, Z., Zhou, X., Wen, H., Wu, Y., Zou, Y. : Dorefa-net : Training low bitwidth convolutional neural networks with low bitwidth gradients. CoRR **abs/1606.06160** (2016)

[22] Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y. : Quantized neural networks : Training neural networks with low precision weights and activations. CoRR **abs/1609.07061** (2016)

[23] Li, F., Liu, B. : Ternary weight networks. CoRR **abs/1605.04711** (2016)

[24] LeCun, Y., Denker, J.S., Solla, S.A. : Optimal brain damage. In : NIPS. (1990)

[25] Han, S., Pool, J., Tran, J., Dally, W.J. : Learning both weights and connections for efficient neural networks. In : NIPS. (2015)

[26] Yang, T., Chen, Y., Sze, V. : Designing energy-efficient convolutional neural networks using energy-aware pruning. CoRR **abs/1611.05128** (2016)

[27] Novikov, A., Podoprikhin, D., Osokin, A., Vetrov, D. : Tensorizing neural networks. In : Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, Cambridge, MA, USA (2015) 442–450

[28] Kim : Compression of deep convolutional neural networks for fast and low power mobile applications. CoRR (2015)

[29] Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I.V., Lempitsky, V.S. : Speeding-up convolutional neural networks using fine-tuned cp-decomposition. In Bengio, Y., LeCun, Y., eds. : ICLR, San Diego, CA, USA. (2015)

[30] He, Y., Han, S. : ADC : automated deep compression and acceleration with reinforcement learning. CoRR **abs/1802.03494** (2018)

[31] Hinton : Distilling the knowledge in a neural network. NIPS (2014)

[32] Kolda, T.G., Bader, B.W. : Tensor decompositions and applications. SIAM Review **51**(3) (2009) 455–500

[33] Miron, S. : Tensor methods for multisensor signal processing. IET Signal Processing **14** (December 2020) 693–709(16)

[34] Krizhevsky, A., Sutskever, I., Hinton, G.E. : Imagenet classification with deep convolutional neural networks. In : NIPS. (2012) 1106–1114

[35] Lin, M., Chen, Q., Yan, S. : Network in network. In Bengio, Y., LeCun, Y., eds. : 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings. (2014)

[36] Hu, J., Shen, L., Sun, G. : Squeeze-and-excitation networks. In : 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA,

June 18-22, 2018, Computer Vision Foundation / IEEE Computer Society (2018) 7132–7141

[37] Xie, G., Wang, J., Zhang, T., Lai, J., Hong, R., Qi, G. : IGCV2 : interleaved structured sparse convolutional neural networks. CoRR **abs/1804.06202** (2018)

[38] Huang, G., Liu, S., van der Maaten, L., Weinberger, K.Q. : Condensenet : An efficient abs-1812-00332densenet using learned group convolutions. CoRR **abs/1711.09224** (2017)

[39] Cai, H., Zhu, L., Han, S. : Proxylessnas : Direct neural architecture search on target task and hardware. CoRR **abs/1812.00332** (2018)

[40] Juefei-Xu, F., Boddeti, V.N., Savvides, M. : Local binary convolutional neural networks. In : CVPR. (2017) 4284–4293