Deep neural networks compression using low-rank approximations

Van Tien Pham

Université de Toulon, Aix Marseille University, CNRS, LIS, UMR 7020, France

Abstract

This report updates the outcomes of the first year and outlines the progress made during the second year of my PhD. The manuscripts submitted from the two first-year projects, CORING and NORTON, are published in Neural Networks and IEEE Transactions on Neural Networks and Learning Systems, respectively. In the second year, I focused on two new projects, contributing to advancements in neural network compression and acceleration through filter pruning and tensor decomposition techniques. The first project led to the development of SLIMING (Singular vaLues-drIven autoMated prunING), an automated filter pruning method based on singular value analysis. SLIMING divides the pruning task into two stages: determining the pruning configuration and selecting the filters to prune for each layer. By formulating these as optimization problems, the method ensures the structural integrity of filters and has been validated through extensive experiments on various architectures and datasets. The second project introduced coupled filter decomposition, a novel low-rank approximation approach. By jointly factorizing filters using coupled tensor decompositions, specifically coupled Canonical Polyadic Decomposition (CCPD), we enable the sharing of a common factor matrix across filters, thereby significantly reducing both parameter count and computational complexity. A custom distance metric for K-means clustering further improves the grouping of similar filters, enhancing the overall compression performance. The approach was tested across multiple vision tasks, showing competitive results in model compression. Potential avenues include delving deeper into the low-rank approximation domain, expanding to encompass other techniques, exploring a broader spectrum of neural network architectures, and applying these efficient models to diverse applications.

Keywords: model compression, low-rank approximation, filter pruning

This second-year report has been examined by the Thesis Follow-up Committee (CSI):

Thesis Supervisor:	Thanh Phuong Nguyen	Professor	Université Côte d'Azur
Thesis Co-supervisor:	Yassine Zniyed	Associate Professor	Université de Toulon
CSI Member:	Seong G. Kong	Professor	Sejong University
CSI Member:	Mohammed Nabil El Korso	Professor	Université Paris-Saclay

This research is conducted within the Laboratoire d'Informatique et des Systèmes, Université de Toulon (UTLN), Aix Marseille University, CNRS, with funding provided through a doctoral contract with UTLN.

1. Introduction

Neural network compression has become a critical area of research as deep learning models grow in complexity and size, requiring significant computational and memory resources. Reducing the model size and improving inference speed without sacrificing accuracy is essential for deploying neural networks in resource-constrained environments, such as mobile devices and edge computing [3, 18]. Among various techniques, filter pruning and low-rank approximation have emerged as effective approaches for reducing the number of parameters and computational costs

Email address: van-tien-pham@etud.univ-tln.fr (Van Tien Pham)

in neural networks. Filter pruning [4] focuses on identifying and removing unnecessary filters in convolutional layers, while low-rank approximation [2] leverages the low-rank nature of the weights, decomposing them into smaller factors.

During the first year of my PhD, I concentrated on network compression through filter pruning and tensor decomposition, which resulted in the following publications:
Conference:

1. [9] Pham, V.T., Zniyed, Y., Nguyen, T.P., 2023. Élagage efficace des filtres basé sur les décompositions tensorielles, in: XXIXème Colloque Francophone de Traitement du Signal et des Images, GRETSI, 937–940.

2. [12] Pham, V.T., Zniyed, Y., Nguyen, T.P., 2024c. Hybrid network compression through tensor decompositions and pruning, in: 32nd European Signal Processing Conference, EUSIPCO 2024, pp. 1052–1056.

Journal:

1. [10] Pham, V.T., Zniyed, Y., Nguyen, T.P., 2024a. Efficient tensor decomposition-based filter pruning. Neural Networks 178, 106393.

2. [15] Pham, V.T., Zniyed, Y., Nguyen, T.P., 2024b. Enhanced network compression through tensor decompositions and pruning. IEEE Transactions on Neural Networks and Learning Systems, 1–13.

In the second year, I have continued to focus on developing novel techniques for model compression. This resulted in two research projects, which have led to the following outcomes:

1. Pham, V.T., Zniyed, Y., Nguyen, T.P., 2024. Singular values-driven automated filter pruning, *submitted*.

2. Pham, V.T., Zniyed, Y., Nguyen, T.P., 2024. Coupled filters decomposition for model compression, *to be submitted*.

In the following sections, I will introduce the second-year projects, SLIMING and CCPD, in Sections 2 and 3. Section 4 provides conclusions and discusses potential directions for future research.

2. The SLIMING Framework

2.1. Introduction

This work introduces a method that leverages tools from linear and multilinear algebra to provide a new solution for automated filter pruning. We propose to detect network redundancy hinging on the dynamics of singular values. In this work, we illuminate the intricate relationship between filter redundancy within neural networks and the observable variations in their singular values. This observation underpins our approach, which we articulate as a constrained optimization problem. Recognizing the inherent complexity and combinatorial nature of filter pruning, we strategically decompose the overarching challenge into two manageable sub-problems. Both are methodically designed to leverage the same underlying principle—the variation of singular values—thereby ensuring consistency in our methodology. Central to our formulation is the consideration of the filters multidimensional structure, a critical aspect that preserves the integrity of information throughout the pruning process. We show that our approach guarantees that the essential information encoded within the filters is retained, preventing any loss of crucial information. To navigate the complexities of this optimization landscape, we introduce two heuristic methods, each tailored to effectively tackle one of the sub-problems. The first method focuses on estimating the optimal pruning configuration, determining the precise number of filters to be retained across different layers under a global constraint. The second method is dedicated to the selection of filters, identifying which filters to preserve in order to maximize filter independence. Together, these methods embody a comprehensive solution to automatic filter pruning, called SLIMING, for Singular vaLues-drIven autoMated prunING. Briefly, the contributions of this work are three-fold. First, based on the observation of the relation between the distribution of the singular values of the weight tensors and the network's redundancy, we formulate automated structured pruning as constrained optimization problems, aiming to maximize the pruned model filter independence. Secondly, we introduce two algorithms to address these problems, one automatically determines the pruning configuration, while the other effectively selects filters. Thirdly, we assess our framework across diverse vision tasks. Through a comprehensive comparison with related works, we validate the effectiveness of our approach.

2.2. Approach

2.2.1. Classical Problem Formulation of Filter Pruning

Let us consider a convolutional neural network (CNN) model comprising L convolutional layers. The l-th layer possesses a weight tensor $\boldsymbol{\mathcal{W}}^l$ of dimensions $C_l \times C_{l-1} \times d_l \times d_l$, where C_l, C_{l-1}, and d_l represent, respectively, the number of output channels, the number of input channels, and the kernel size. The primary objective of filter pruning can be expressed as

$$\underset{\left\{\boldsymbol{\mathcal{W}}^l\right\}_{l=1}^{L}}{\arg\min}\ \mathcal{L}\Big(\left\{\boldsymbol{\mathcal{W}}^l\right\}_{l=1}^{L}, \mathcal{D}\Big), \quad \text{s.t.} \quad \mathcal{C}\left(\left\{\boldsymbol{\mathcal{W}}^l\right\}_{l=1}^{L}\right) \leq \mathcal{C}_{\text{desired}}, \tag{1}$$

where $\mathcal{L}$ is the loss function, and $\mathcal{D}$ is the considered dataset. $\mathcal{C}(\cdot)$ is the function that computes the model's configuration, and $\mathcal{C}_{\text{desired}}$ is the desired resource budget, representing flexible constraints such as the number of retained filters, target parameters, or MACs. Without loss of generality, let $\mathcal{C}_{\text{desired}}$ represent the desired total number of filters to retain N, and the function $\mathcal{C}(\cdot)$ now computes the sum of the number of filters to be kept across all layers. For the l-th layer, let $\mathcal{K}^l = \left\{k_1^l, k_2^l, \ldots, k_{N_l}^l\right\}$ be the set of the indices of the retained filters, where k_n^l is the index of the n-th retained filter, $1 \leq n \leq N_l$, and N_l is the total number of the retained filters. $\boldsymbol{\mathcal{W}}^l_{\mathcal{K}^l,:,:,:}$ is the weight tensor of the l-th pruned layer. Thus, the problem formulated in (1) can be expressed as

$$\underset{\left\{\mathcal{K}^l\right\}_{l=1}^{L}}{\arg\min}\ \mathcal{L}\Big(\left\{\boldsymbol{\mathcal{W}}^l_{\mathcal{K}^l,:,:,:}\right\}_{l=1}^{L}, \mathcal{D}\Big), \text{s.t.} \sum_{l=1}^{L} \left|\mathcal{K}^l\right| \leq N. \tag{2}$$

2.2.2. "Multilinear" Singular Values and Filters Dependencies

In the case of matrices, the concept of redundancy, *i.e.*, the dependency among columns and/or rows, culminates in rank degeneracy, signifying that the matrix does not possess full rank. This phenomenon of rank degeneracy is palpable through the lens of singular values. Specifically, if the tally of non-zero singular values falls short of $\min(m, n)$ for a matrix of dimensions $m \times n$, it heralds the existence of dependencies between the matrix's columns and/or rows. Transitioning to our context, where the weights $\boldsymbol{\mathcal{W}}^l$ manifests as a multidimensional 4-order tensors rather than matrices, the conventional approach of flattening these tensors into matrices of size $C_l \times (C_{l-1} \cdot d_l^2)$, which encapsulates the vectorized rendition of the C_l filters across their rows, poses the risk of obfuscating the filters inherent multidimensional characteristics.

It appears most prudent, therefore, to explore the variation of a certain "multilinear" singular values counterpart pertaining to the 4-order tensor, particularly across the filters mode, *i.e.,* mode-1 of tensor $\boldsymbol{\mathcal{W}}^l$. This mode quintessentially embodies the filters variation. However, the application of the HOSVD does not yield 'multidimensional" singular values per se, given the core tensor's non-diagonal structure. Nevertheless, this core tensor boasts two important properties [1]: (i) the all orthogonality property (*i.e.*, all the slice matrices of a given mode of the core tensor are mutually orthogonal), and (ii) the ordering property. The latter ensures that the Frobenius norm of the mode-1 matrix slices of the core tensor $\boldsymbol{\mathcal{G}}$ in Tucker decomposition, articulated as: $\left\|\boldsymbol{\mathcal{G}}(i,:,:,:)\right\|_F \geq \left\|\boldsymbol{\mathcal{G}}(j,:,:,:)\right\|_F \quad \text{if} \quad i \leq j$. This insight remains true for other modes as well, and allows an analysis of filter dependencies via the enduring Frobenius norms of the mode-1 slices of the core tensor $\boldsymbol{\mathcal{G}}$. This quantity, namely the Frobenius norm of the slice matrices, can be seen as a kind of "multilinear" singular values. Moreover, an interesting property highlighted in [1] reveals that, given the HOSVD of the 4-order tensor $\boldsymbol{\mathcal{W}}^l$ as $\boldsymbol{\mathcal{W}}^l = \boldsymbol{\mathcal{G}} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \times_4 \mathbf{U}^{(4)}$, and concurrently, the SVD of mode-1 unfolding of $\boldsymbol{\mathcal{W}}^l$ as $\text{unfold}_1\boldsymbol{\mathcal{W}}^l = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$, then it follows that $\left\|\boldsymbol{\mathcal{G}}(i,:,:,:)\right\|_F = \sigma_i$. In essence, this analysis elucidates that pursuing the variation of "multilinear" singular values relative to a specific mode, whilst preserving the tensor's multidimensional structure, is equivalent to analyzing the singular values fluctuation of the tensor's unfolding in accordance to that mode. Leveraging these properties, we propose to formulate our pruning problem using the singular values variation, without compromising their intrinsic dimensional interdependencies. In the sequel and in light of these explanations, we now define the nuclear norm of a tensor $\boldsymbol{\mathcal{W}}^l$ as the nuclear norm of its mode-1 unfolding matrix as $\left\|\boldsymbol{\mathcal{W}}^l\right\|_* = \left\|\text{unfold}_1\boldsymbol{\mathcal{W}}^l\right\|_*$.

2.2.3. Proposed Problem Formulation

Drawing on the insights from the preceding subsubsection, it now appears intuitive that if we establish a general pruning ratio or a fixed number N of filters to retain, then the following two assertions become equivalent: "Retaining

the least redundant filters in a layer of a CNN" is synonymous with "Preserving the set of filters that maximizes the sum of singular values". This equivalence arises because filters exhibiting similarities drive their corresponding singular values toward zero. Consequently, we can express the singular values-driven pruning problem as follows:

$$\underset{\left\{\mathcal{K}^l\right\}_{l=1}^{L}}{\arg\max} \sum_{l=1}^{L} \left\| \boldsymbol{\mathcal{W}}^l_{\mathcal{K}^l,:,:,:} \right\|_*, \quad \text{s.t.} \quad \sum_{l=1}^{L} \left|\mathcal{K}^l\right| \leq N. \tag{3}$$

This optimization challenge seeks to identify a set of sets, $\mathcal{K}^l$ for $1 \leq l \leq L$, that maximizes the nuclear norm (thereby ensuring maximal independence among filters) while ensuring that the aggregate cardinalities of these sets do not exceed a predetermined value N. One should note that the number of possibilities for the problem in (3) is bounded by $\sum_{n=1}^{N} \binom{\sum_{l=1}^{L} C_l}{n}$. Given its combinatorial nature and the vast search space, we approach this problem by decomposing it into two subproblems: (i) determining the optimal cardinalities for each layer that maximize the sum of nuclear norms across all layers, and (ii) once the cardinalities for each layer are established, solving L independent optimization problems where, for each layer, we seek the filter selection that maximizes the nuclear norm subject to the given cardinality constraint.

For the first challenge, namely, finding the optimal configuration, the optimization problem can succinctly be represented as:

$$\underset{\left\{N_l\right\}_{l=1}^{L}}{\arg\max} \sum_{l=1}^{L} \sum_{i=1}^{N_l} \sigma_i^l, \quad \text{s.t.} \quad \begin{cases} \sum_{l=1}^{L} N_l \leq N, \\ 1 \leq N_l \leq C_l \ \text{ for } \ 1 \leq l \leq L, \end{cases} \tag{4}$$

where σ_i^l are the i-th singular values of $\text{unfold}_1 \boldsymbol{\mathcal{W}}^l$, and N_l is the cardinal of the set $\mathcal{K}^l$, essentially the number of filters to retain in each layer.

Upon determining the optimal configuration, the problem of filter selection then diverges into L distinct optimization challenges, one for each layer, defined as follows:

$$\underset{\mathcal{K}^l}{\arg\max} \left\| \boldsymbol{\mathcal{W}}^l_{\mathcal{K}^l,:,:,:} \right\|_*, \quad \text{s.t.} \quad \left|\mathcal{K}^l\right| = N_l, \tag{5}$$

for $1 \leq l \leq L$, with N_l being known. This phase shifts focus to identifying, for each layer, the precise subset of filters $\mathcal{K}^l$ that, when retained, maximizes the nuclear norm, hence adhering to the pre-established cardinality N_l. To succinctly summarize the two sub-problems delineated in (4) and (5), which collectively supplant the initial problem posited in (3), it can be articulated that the problem in (4) addresses the query, *"How many filters will each layer contribute?"*, whereas the problem in (5) resolves the inquiry, *"Which filters will each layer contribute?"*.

2.3. Pruning Configuration Estimation

We introduce Algorithm 1, a method aimed at determining the optimal filter configuration across all layers while adhering to a constraint on the total number of retained filters, as described by (4). The core concept revolves around iteratively selecting the largest singular values from all layers until the desired number of retained filters is met. Initially, the algorithm initializes the retained filter count for each layer to 1. The set $\mathcal{S}$ is used to store the remaining largest singular value from each layer, initialized with the second singular values σ_2^l in step 5. This set serves as the pool of candidate singular values to be considered in each iteration. The algorithm iteratively selects the largest candidate singular value from $\mathcal{S}$ and increments the retained filter count of the corresponding layer. This simple process repeats until the desired total number of retained channels is achieved.

2.3.1. Filters Selection

We introduce Algorithm 2, designed to address the solution to the optimization problem formulated in (5). It is imperative to note that this stage assumes a predefined pruning configuration, with the optimal number of filters to be retained across each layer already determined. The objective of Algorithm 2 is to identify, for each layer, the set $\mathcal{K}^l$ of size N_l, which epitomizes maximal filters independency. This aims to preserve the core functionality of the network by retaining a subset of filters that collectively contribute to its predictive capacity, devoid of redundancy. Given the combinatorial essence of the problem, an exhaustive search approach for deep CNNs is impractical due to

the colossal search space it necessitates. To circumvent this challenge, Algorithm 2 employs a heuristic strategy that iteratively eliminates filters until only the N_l most significant filters, as measured by our objective function, remain. The underlying idea of our filter elimination methodology hinges on the observation that the removal of a filter, which results in negligible or no reduction in the overall nuclear norm, implies substantial redundancy of this filter with respect to the retained set.

Algorithm 1 Pruning configuration estimation

Require: $\{\mathcal{W}^l\}_{l=1}^L$, N.
Ensure: Optimal configuration $\{N_l\}_{l=1}^L$ for (4).
for $l = 1$ to L **do**
  $N_l = 1$
  $\text{unfold}_1 \mathcal{W}^l \overset{\text{SVD}}{=} \mathbf{U}_l \mathbf{\Sigma}_l \mathbf{V}_l^T$, with $\sigma_i^l = \mathbf{\Sigma}_l(i,i)$.
end for
$\mathcal{S} = \{\sigma_2^l\}_{l=1}^L$
for $n = 1$ to $N - L$ **do**
  $j = \arg\max \mathcal{S}_j$
  $N_j = N_j + 1$
  $\mathcal{S}_j = \sigma_{N_j+1}^j$
end for
Return $\{N_l\}_{l=1}^L$

Algorithm 2 Filters selection

Require: $\mathcal{W}^l$, number of retained filters N_l.
Ensure: Optimal set $\mathcal{K}^l = \{k_1^l, k_2^l, \ldots, k_{N_l}^l\}$ for (5).
Initialize $\mathcal{K}^l = \{1, 2, \ldots, C_l\}$
for $i = 1$ to $(C_l - N_l)$ **do**
  Initialize $\Delta_{\text{min}} = \infty$
  for δ_{current} in $\mathcal{K}^l$ **do**
    $\mathcal{R}^l = \mathcal{K}^l \setminus \{\delta_{\text{current}}\}$
    $\Delta_{\text{current}} = \left\|\mathcal{W}^l_{\mathcal{K}^l,:,:,:}\right\|_* - \left\|\mathcal{W}^l_{\mathcal{R}^l,:,:,:}\right\|_*$
    if $\Delta_{\text{current}} < \Delta_{\text{min}}$ **then**
      $\Delta_{\text{min}} = \Delta_{\text{current}}$
      $\delta_{\text{min}} = \delta_{\text{current}}$
    end if
  end for
  $\mathcal{K}_l = \mathcal{K}^l \setminus \{\delta_{\text{min}}\}$
end for
Return $\mathcal{K}^l$

2.4. Experiments

We validate the versatility of our method through an experiment conducted on a multi-branch architecture, GoogLeNet, on CIFAR-10, as detailed in Table 1. SLIMING exhibits superior accuracy with reduced overhead when compared to similarity-based and hand-crafted methods [10, 4]. Notably, SLIMING surpasses automatic pruning ratio approaches like CC [3] across all evaluated metrics.

3. The CCPD Framework

3.1. Introduction

In this work, we highlight a key observation: within a convolutional layer, all filters have identical dimensions and are convolved with a common input tensor to generate their respective output feature maps. This observation raises two important points. First, all filters are of equal standing and some can be partially identical by sharing a common factor in a subspace. Therefore, they can potentially and technically be decomposed jointly. Second, as they extract information from a common input, partially similar filters may produce partially similar output features. To enhance computational efficiency, the redundant computation of these similar parts should be avoided. Building on these insights, we introduce, for the first time in the context of low-rank approximation for model compression, the concept of *coupled filters decomposition*. This method is accompanied by clustering, as a preprocessing phase, to improve the approximation quality, as depicted in Fig. 1. In the coupled filters decomposition scheme, multiple filters are approximated jointly using coupled tensor decompositions. To demonstrate the use of this method, we employ coupled CPD [17] as

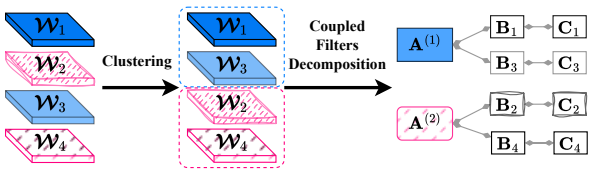

Figure 1: Coupled filters decomposition.

a representative example due to its simplicity and efficiency, although our approach can be adapted to other decomposition techniques. Specifically, instead of decomposing each filter individually, we propose jointly factorizing them along a specific mode. After decomposition, the filters within a cluster share a common factor matrix in the selected mode while retaining their unique factor matrices in other modes. This approach suggests that filters possess both common and unique characteristics. Notably, since all filters share the same input tensor, the computation between the decomposed common factor and the input must only be performed once. Briefly, the contributions of this work are three-fold. First, we introduce the coupled filters decomposition approach for CNN compression. Using the coupled CPD, we factorize a conventional convolution into a novel compact operator called CCPDBlock. Secondly, we propose the projected distance for the K-means clustering algorithm, enhancing the overall performance. Thirdly, we assess our framework across diverse vision tasks. Through a comprehensive comparison with related methods, we validate the effectiveness of our approach.

3.2. Approach

3.2.1. Clustering

Consider a convolutional layer, parameterized by the weight tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{O \times I \times D_{\text{h}} \times D_{\text{w}}}$, where O, I D_{h} and D_{w} represent the number of output channels, input channels, height, and width of the kernel, respectively. $\boldsymbol{\mathcal{W}}$ consists of O 3-order filters denoted as $\left\{\boldsymbol{\mathcal{W}}_1, \boldsymbol{\mathcal{W}}_2, \ldots, \boldsymbol{\mathcal{W}}_O\right\}$, where the i-th filter $\boldsymbol{\mathcal{W}}_i \in \mathbb{R}^{I \times D_{\text{h}} \times D_{\text{w}}}$ is the sub-tensor $\boldsymbol{\mathcal{W}}_{i,:,:,:}$. These weights map an input tensor $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{H_{\text{in}} \times W_{\text{in}} \times I}$ into an output tensor $\boldsymbol{\mathcal{O}} \in \mathbb{R}^{H_{\text{out}} \times W_{\text{out}} \times O}$, where H_{in}, W_{in}, H_{out}, and W_{out} are the height and width of the input and output tensors, respectively.

We propose to cluster the O filters into K separate clusters, where $1 \leq K \leq O$. The k-th cluster ($1 \leq k \leq K$) consists of $o^{(k)}$ filters. The total number of filters across all clusters equals the total number of filters of the layer. $\sum_{k=1}^{K} o^{(k)} = O$. Our intuition is that grouping similar filters in a common cluster will improve the quality of the coupled decomposition. To this end, we may consider any clustering method, however, we employ the K-means clustering approach as it is simple and effective.

The objective of clustering is to gather the filters that own similar first-mode factor matrices into a common cluster. Therefore, a measurement of the similarity between the two filters in the first-mode factor subspace is essential. However, the standard K-means algorithm uses the Euclidean distance which is designed to measure the distance between two points in Euclidean space. Therefore, we propose a custom distance that aligns with the context of coupled CPD. In our custom K-means algorithm, we use this projected distance to measure the distance between a filter and a centroid in the assignment step, as well as to compute the inertia for an initialization.

3.2.2. Treat One Cluster

In CNNs, the mapping of the input tensor $\boldsymbol{\mathcal{I}}$ to the output tensor $\boldsymbol{\mathcal{O}}$ is achieved through the convolution operation. This convolution is expressed as

$$\boldsymbol{\mathcal{O}}_i(x, y) = \sum_{m=0}^{D_{\text{h}}-1} \sum_{n=0}^{D_{\text{w}}-1} \sum_{p=0}^{I-1} \boldsymbol{\mathcal{I}}(x+m, y+n, p) \cdot \boldsymbol{\mathcal{W}}_i(p, m, n), \tag{6}$$

where, for $1 \leq i \leq o$, and $\boldsymbol{\mathcal{O}}_i \in \mathbb{R}^{H_{\text{out}} \times W_{\text{out}}}$. The coupled CPD of $\boldsymbol{\mathcal{W}}_i$ is given as

$$\boldsymbol{\mathcal{W}}_i(p, m, n) = \sum_{r=0}^{R-1} \mathbf{A}(p, r) \cdot \mathbf{B}_i(m, r) \cdot \mathbf{C}_i(n, r). \tag{7}$$

By substituting (7) into (6), we obtain a new CPD-based approach to compute the convolution. This approach involves a sequence of mappings using factor matrices instead of high-order tensors. The resulting equation is:

$$\boldsymbol{\mathcal{O}}_i(x, y) = \sum_{m=0}^{D_{\text{h}}-1} \sum_{n=0}^{D_{\text{w}}-1} \sum_{p=0}^{I-1} \sum_{r=0}^{R-1} \boldsymbol{\mathcal{I}}(x+m, y+n, p) \cdot \mathbf{A}(p, r) \cdot \mathbf{B}_i(m, r) \cdot \mathbf{C}_i(n, r). \tag{8}$$

3.2.3. Treat All Clusters

The decomposition process is conducted in parallel across all K clusters. One way to implement the coupled CPDBlock on all the K clusters is to treat each sub-layer as a set of K parallel convolutions. The final outputs of these convolutions are then concatenated as follows $\boldsymbol{\mathcal{O}} = \left\{\boldsymbol{\mathcal{O}}^{(k)}\right\}_{k=1}^{K}$.

3.3. Experiments

To assess CCPD, we conduct experiments on the ImageNet dataset by addressing ResNet-50 as shown in Table 2. Across all compression levels, our method consistently outperforms other approaches in terms of both performance and complexity reduction.

Table 1: Results of SLIMING for GoogLeNet on CIFAR10

Method	Auto	Top-1	MACs (CR)	Params (CR)
GoogLeNet		95.05	1.52B (00)	6.15M (00)
CC-0.5 [3]	✓	95.18	0.76B (50)	2.83M (54)
CORING [10]	✗	95.32	0.65B (57)	2.85M (54)
SLIMING (Ours)	✓	**95.48**	**0.62B (60)**	**2.68M (57)**
DCFF [4]	✗	94.92	0.46B (70)	2.08M (66)
CORING [10]	✗	95.03	0.39B (74)	2.10M (66)
SLIMING (Ours)	✓	**95.31**	**0.29B (81)**	**1.41M (77)**

Table 2: Results of CCPD for ResNet on ImageNet

Method	Top-1	Top-5	MACs (CR)	Params (CR)
ResNet-50	76.13	92.87	4.12B (00)	25.56M (00)
CEPD [18]	75.82	92.84	1.53B (63)	9.38M (63)
NORTON [11]	75.95	92.91	1.49B (64)	10.52M (59)
CCPD (Ours)	**75.96**	**92.91**	**1.42B (66)**	**8.81M (66)**
LRPET [2]	70.97	90.16	0.98B (76)	7.74M (70)
DCFF [4]	73.81	91.59	1.02B (75)	6.56M (74)
CCPD (Ours)	**73.88**	**92.07**	**0.92B (78)**	**5.21M (80)**

4. Conclusion and Future Works

In summary, this report updates the first-year outcomes and outlines the progress made during the second year of my PhD. Key contributions include the novel methods SLIMING and CCPD for network pruning and tensor decomposition, which effectively achieve model compression while preserving essential features and performance.

Looking ahead, there are many exciting opportunities for future research:

- Deeper exploration in pruning and decomposition: Building on our initial works, there's room for delving deeper into the pruning and tensor decomposition domain. Developing more advanced techniques and algorithms could involve exploring different criteria for pruning or devising more efficient decomposition methods.
- Broadening to other techniques: Extending our research to encompass a wider range of model compression techniques beyond pruning and tensor decomposition is a promising direction. Investigating methods like quantization, knowledge distillation, and NAS will allow us to explore their potential to reduce model size while maintaining performance.
- Widening to other model architectures: Applying compression techniques to various neural network architectures other than CNNs opens up exciting possibilities. Experimenting with models like transformers, recurrent neural networks, graph neural networks, and large language models will help assess the effectiveness of our methods in different domains [5, 7, 8, 6].

Appendix A. Lists of publications

Update: October 25, 2025. The research carried out during this PhD has resulted in the following publications [9, 12, 10, 15, 16, 13, 14, 19]:

International journals:

- **[J1] V.T. Pham**, Y. Zniyed, T.P. Nguyen, *Enhanced network compression through tensor decompositions and pruning*, IEEE Transactions on Neural Networks and Learning Systems, vol. 36, no. 3, pp. 4358-4370, March 2025.
- **[J2] V.T. Pham**, Y. Zniyed, T.P. Nguyen, *Efficient tensor decomposition-based filter pruning*, Neural Networks, vol. 178, p. 106393, October 2024.
- **[J3]** N. Tokcan[1], S.S. Sofi[1], **V.T. Pham**[1], C. Prévost[1], S. Kharbech[1], B. Magnier, T.P. Nguyen, Y. Zniyed, L.D. Lathauwer, *Tensor decompositions for signal processing: theory, advances, and applications*, Signal Processing, vol. 238, p. 110191, January 2026.
- **[J4] V.T. Pham**, Y. Zniyed, T.P. Nguyen, *Singular values-driven automated filter pruning*, Neural Networks, vol. 192, p. 107857, December 2025.
- **[J5] V.T. Pham**, Y. Zniyed, T.P. Nguyen, *Coupled tensor decomposition for compact network representation*, IEEE Transactions on Neural Networks and Learning Systems, 2025, to appear.

[1]These authors contributed equally to this work as co-first authors.

Preprint:

- **[J6] V.T. Pham**, Y. Zniyed, T.P. Nguyen, *Decoupling matrix-valued function using zeroth and first-order information and its application in neural network compression*, to be submitted.

International conference:

- **[C1] V.T. Pham**, Y. Zniyed, T.P. Nguyen, *Hybrid network compression through tensor decompositions and pruning*, in 32nd European Signal Processing Conference, Lyon, France, pp. 1052-1056, August 2024.

National conference:

- **[C2] V.T. Pham**, Y. Zniyed, T.P. Nguyen, *Élagage efficace des filtres basé sur les décompositions tensorielles*, in XXIXème Colloque Francophone de Traitement du Signal et des Images, Grenoble, France, pp. 937-940, August 2023.

Seminar:

- **[S1] V.T. Pham**, Y. Zniyed, T.P. Nguyen, *Hybrid network compression through tensor decompositions and pruning*, in Journées Apprentissage Signal Image, Aix-en-Provence, France, pp. 1-20, June 2024.
- **[S1] V.T. Pham**, Y. Zniyed, T.P. Nguyen, *Coupled tensor decomposition for compact network representation*, in Journées Apprentissage Signal Image du LIS, Châteauneuf-le-Rouge, France, November 2025.

References

[1] De Lathauwer, L., De Moor, B., Vandewalle, J., 2000. A multilinear singular value decomposition. SIAM Journal on Matrix Analysis and Applications 21, 1253–1278.

[2] Guo, K., Lin, Z., Chen, C., Xing, X., Liu, F., Xu, X., 2024. Compact model training by low-rank projection with energy transfer. IEEE Transactions on Neural Networks and Learning Systems .

[3] Li, Y., Lin, S., Liu, J., Ye, Q., Wang, M., Chao, F., Yang, F., Ma, J., Tian, Q., Ji, R., 2021. Towards compact cnns via collaborative compression, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6434–6443.

[4] Lin, M., Chen, B., Chao, F., Ji, R., 2023. Training compact cnns for image classification using dynamic-coded filter fusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 45, 10478–10487.

[5] Pham, V.T., Le, T.L., Tran, T.H., Nguyen, T.P., 2020. Hand detection and segmentation using multimodal information from kinect, in: 2020 International Conference on Multimedia Analysis and Pattern Recognition (MAPR), pp. 1–6.

[6] Pham, V.T., Nguyen, T.P., 2023. Identification and localization covid-19 abnormalities on chest radiographs, in: The International Conference on Artificial Intelligence and Computer Vision, Springer. pp. 251–261.

[7] Pham, V.T., Tran, C.M., Zheng, S., Vu, T.M., Nath, S., 2021a. Chest x-ray abnormalities localization via ensemble of deep convolutional neural networks, in: 2021 International Conference on Advanced Technologies for Communications (ATC), pp. 125–130.

[8] Pham, V.T., Tran, T.H., Vu, H., 2021b. Detection and tracking hand from fpv: benchmarks and challenges on rehabilitation exercises dataset, in: 2021 RIVF International Conference on Computing and Communication Technologies (RIVF), IEEE. pp. 1–6.

[9] Pham, V.T., Zniyed, Y., Nguyen, T.P., 2023. Élagage efficace des filtres basé sur les décompositions tensorielles, in: XXIXème Colloque Francophone de Traitement du Signal et des Images, GRETSI - Groupe de Recherche en Traitement du Signal et des Images. pp. 937–940.

[10] Pham, V.T., Zniyed, Y., Nguyen, T.P., 2024a. Efficient tensor decomposition-based filter pruning. Neural Networks 178, 106393.

[11] Pham, V.T., Zniyed, Y., Nguyen, T.P., 2024b. Enhanced network compression through tensor decompositions and pruning. IEEE Transactions on Neural Networks and Learning Systems , 1–13.

[12] Pham, V.T., Zniyed, Y., Nguyen, T.P., 2024c. Hybrid network compression through tensor decompositions and pruning, in: 32nd European Signal Processing Conference, EUSIPCO 2024, pp. 1052–1056.

[13] Pham, V.T., Zniyed, Y., Nguyen, T.P., 2025a. Coupled tensor decomposition for compact network representation. IEEE Transactions on Neural Networks and Learning Systems , 1–15.

[14] Pham, V.T., Zniyed, Y., Nguyen, T.P., 2025b. Decoupling matrix-valued function using zeroth and first-order information and its application in neural network compression. to be submitted .

[15] Pham, V.T., Zniyed, Y., Nguyen, T.P., 2025c. Enhanced network compression through tensor decompositions and pruning. IEEE Transactions on Neural Networks and Learning Systems 36, 4358–4370.

[16] Pham, V.T., Zniyed, Y., Nguyen, T.P., 2025d. Singular values-driven automated filter pruning. Neural Networks 192, 107857.

[17] Sørensen, M., Domanov, I., De Lathauwer, L., 2015. Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-(l_r,n,l_r,n,1) terms—part ii: Algorithms. SIAM Journal on Matrix Analysis and Applications 36, 1015–1045.

[18] Sui, Y., Yin, M., Gong, Y., Yuan, B., 2024. Co-exploring structured sparsification and low-rank tensor decomposition for compact dnns. IEEE Transactions on Neural Networks and Learning Systems .

[19] Tokcan, N., Sofi, S.S., Pham, V.T., Prévost, C., Kharbech, S., Magnier, B., Nguyen, T.P., Zniyed, Y., De Lathauwer, L., 2026. Tensor decompositions for signal processing: Theory, advances, and applications. Signal Processing 238, 110191.