Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Applying Singular Value Decomposition (SVD) to CNNs: A Path Toward Lightweight and Efficient Architecture

Khawla Hussein Ali
Assist. Prof. Dr. in the Department of Computer Science at the University of Basrah, Iraq
E-mail: khawla.ali@uobasrah.edu.iq

ARTICLE INFO

Received: 24 Dec 2024

Revised: 12 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

This paper aims to investigate whether applying the Singular Value Decomposition (SVD) technique can reduce the workload of Convolutional Neural Networks (CNNs) without compromising image classification results. Usual methods for reducing file size are generally unsuitable for deployment because they are costly to train, not straightforward to use, or require additional resources. We performed a low-rank approximation of the CNN weights using the singular value decomposition (SVD) approach. Results for ResNet-50 were obtained from CIFAR-10, and EfficientNet-B0 was assessed using ImageNet. The assessment used metrics including accuracy, precision, recall, and F1-score. Using SVD at a 0.5 compression level, we reduced the network size of ResNet-50 on CIFAR-10 by 41.7% without significantly compromising accuracy (95.4%). With aggressive compression at a ratio of 0.1, the model achieved an 88.9% reduction in parameters, but accuracy decreased to 72.9%. Furthermore, the model increased inference speed slightly (up to 1.0x) and maintained an approximately 89.96 MB size. The uncompressed model was 10.4% accurate before training. Rather than other techniques, SVD offers straightforward yet practical suggestions for enhancing CNN performance on small datasets, without significantly compromising accuracy. These results demonstrate that SVD-based compression is a promising approach for utilizing CNNs in limited-resource systems. It achieves nearly complete accuracy with fewer parameters and has a minimal impact on the speed of CNNs.

Keywords: Singular Value Decomposition (SVD), CNNs, Low-Rank Approximation, Lightweight Architectures, Matrix Decomposition.

INTRODUCTION

Due to the rapid advancements in deep learning, Convolutional Neural Networks (CNNs) have become increasingly common in fields such as autonomous driving, medical diagnosis, and real-time surveillance. Even though CNNs have performed very well in areas such as object recognition and boundary detection, their more advanced versions require a significant amount of memory and take a considerable amount of time to compute. These problems pose substantial challenges in utilizing deep learning models on devices, systems, and platforms with limited resources [1].

To overcome these issues, some researchers have suggested compression approaches, including pruning, quantization, and knowledge distillation. Still, most of these approaches require multiple training steps, careful adjustment of various factors, or the assistance of additional models. Alternatively, in Singular Value Decomposition (SVD), teams of neural networks utilize matrix approximation to identify and remove unnecessary data from tensors, while minimizing performance degradation. This research is necessary because it enables the creation of lightweight deep learning models that can operate even with minimal resources.

While SVD is a simple and elegant algorithm, researchers have not explored it much for CNN compression in modern models and fast environments. There has been greater attention to learning from the first major CNNs, such as AlexNet and VGG. Yet, little exploration has been conducted on making recent networks more scalable and assessing the influence of compression rates, latency, and classification results. Most SVD-based systems used in I-DNN detection lack a framework for applying SVD to both types of layers across multiple network models.

Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The Main Contributions of this work are as follows:

1. We developed an SVD-based compression framework that performs scalable low-rank approximation on CNN networks, thereby reducing the model complexity of convolutional and fully connected layers.

2. The evaluation encompasses parameter reduction, inference speedup, and accuracy retention for ResNet-50 and EfficientNet-B0, utilizing the standard datasets CIFAR-10 and ImageNet.

3. Research comparisons indicate that our method excels at maintaining accuracy levels when it successfully largely compresses models, compared with regular compression techniques such as pruning, quantization, and knowledge distillation.

4. This paper examines the outcomes of system deployment and the viability of implementation, particularly in the context of edge artificial intelligence (AI), mobile inference, and embedded deep learning hardware systems.

RELATED WORKS

Researchers are interested in model compression techniques for CNNs because CNNs require a significant amount of storage memory and processing power during edge and mobile usage [1]. Researchers have provided mathematical proof that SVD is easily interpretable and well-structured. Researchers have explored several SVD-based methods for modeling and inference; however, these methods differ in their effectiveness.

Qi et al. [2] suggested using a hierarchical SVD approach to separate convolutional layers into smaller groups. Compression for CIFAR-10 and ImageNet enabled maintaining high accuracy with minimal loss. However, due to their level of complexity and tuning, multi-level decompositions are challenging to make compatible with most deep learning tools for real-time use.

Authors Wang and Cai [3] added SVD and channel reduction to the TEC-CNN model, enabling it to operate more efficiently at a lower cost. Still, using several methods together obscures what SVD does and requires relearning, which is not feasible in many places with limited resources.

Yang et al. [4] added a kind of regularization that promotes low-rank features during the training process. Although it proves to be effective, starting this method from scratch makes it suitable only for models without pre-training and without the capability for transfer learning.

SVD-based matrix factorization was jointly used by Chen et al. [5] across multiple network layers, enabling a reduction in network size of up to 22 times compared to ResNet-34. On the other hand, linking layers in joint decomposition makes system design much less flexible and adjustable.

Lin et al. [6] employed SVD on binarized convolution filters to accelerate the execution of convolutions on hardware. Due to their efficiency, binarized models sacrifice some accuracy and the ability to understand complex data, such as ImageNet.

He and his co-authors [7] separated convolution layers into depth-wise and point-wise operations using the singular value decomposition (SVD) technique. Although this structure is fast to process, it typically yields poor results unless optimized.

Tai et al. [8] introduced an approach that utilizes SVD lower-rank approximations for convolutional and fully connected layers in a neural network. Ranks are determined based on the environment at hand, resulting in competitive accuracy and file size performance. Still, using the adaptive process adds more complexity to training and requires certain fine-tuning.

Isong [9] examined the topic of building CNNs from scratch using SVD-inspired low-rank blocks. Even so, training on ImageNet is very expensive, and results can be easily affected by changes in hyperparameters.

Sharma et al. [10] used dynamic pruning to estimate the SVD rank. As the rank changes, the learning process can become unstable when encountering batch normalization or residuals.

Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Yaguchi et al. [11] constructed a low-rank network that can be scaled up by utilizing decomposed forms in all layers. As a result, the entire structure must be changed, and compatibility with previous models is reduced.

Jain [14] presented implementation-oriented strategies for using SVD for compression. The article is more practical than novel and doesn't introduce new algorithms; thus, its academic contribution is limited. We can summarize the related works in Table 1.

Table 1. Summary of Related Works Using Singular Value Decomposition (SVD)

Method	Authors	Publication Date	Dataset(s)	Evaluation Metrics / Results
Fine-grained Hierarchical Singular Value Decomposition (HSV)	Mengmeng Qi, et al.	April 2025	CIFAR-10, ImageNet	Achieved significant compression with minimal accuracy loss.
TEC-CNN: Efficient Model Compression	Yifan Wang, Liang Feng, et al.	January 2025	CIFAR-10, ImageNet	Maintained accuracy levels while achieving significant compression.
Learning Low-Rank DNNs via Singular Vector Regularization	Huanrui Yang, Minxue Tang, et al.	April 2020	CIFAR-10, ImageNet	Achieved optimal compression rate with minimal accuracy degradation.
Joint Matrix Decomposition for CNN Compression	Shaowu Chen, Jiahao Zhou, et al.	July 2021	CIFAR-10, CIFAR-100, ImageNet	Compressed ResNet-34 by 22× with slight accuracy degradation.
Binarized CNNs with Separable Filters for Hardware Acceleration	Jeng-Hau Lin, Tianwei Xing, et al.	July 2017	CIFAR-10, ImageNet	Enhanced hardware acceleration efficiency with reduced computational complexity.
Depth-wise Decomposition for Accelerating Separable Convolutions	Yihui He, Jianing Qian, et al.	October 2019	ImageNet	Improved Top-1 accuracy of ShuffleNet V2 by ~2%.
Adaptive Low-Rank SVD for Efficient Deep Learning	R. Tai, T. Xiao, et al.	September 2020	ImageNet	Achieved state-of-the-art speed and accuracy trade-off.
Building Efficient Lightweight CNN Models	Y. Tai, et al.	January 2018	CIFAR-10, ImageNet	Constructed lightweight CNNs with efficient and accurate performance.
Building an Efficient Light-Weight CNN	N. Isong	January 2025	CIFAR-10, CIFAR-100, ImageNet	Achieved substantial storage savings while maintaining or enhancing classification accuracy.
CNN via Dynamic Rank Pruning	Manish Sharma, et al.	October 2024	ImageNet	Achieved 73.2% Top-1 accuracy with 0.27× MACs on ResNet-50.
Decomposable Net: Scalable Low-Rank Compression	Atushi Yauchi, et al.	May 2021	CIFAR-10, CIFAR-100, ImageNet	Decomposition reduced model size up to 9× while maintaining or improving accuracy.

Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Low-Rank Optimization for Efficient Deep Learning	Xinwei Qu, et al.	2023	CIFAR-10, ImageNet	ALDS reduced parameters by up to 60% with minimal loss of accuracy.

Previous work have employed SVD-based compression on CNN [12-13]. In contrast, using limited data, most have focused on selective layer compression in large-scale designs, such as ResNet or VGG [15-16]. Unlike our work, it examines the impact of compressing a full-network SVD, using low-rank approximations to all convolutional layers of both ResNet-50 and EfficientNet-B0. Additionally, we provide a layer-wise energy retention analysis, a new diagnostic tool that enables a detailed understanding of the decomposition effect. This is the first work to utilize and study full-layer SVD compression on both heavy and lightweight CNNs across various datasets with structured fine-tuning.

METHODS

This section focuses on the proposed compression method using Singular Value Decomposition (SVD) and illustrates its application to Convolutional Neural Networks (CNNs). We use cleaning, selection, transformation, and tuning steps.

The proposed SVD-based CNN compression framework, as shown in Figure 1, consists of the following steps:

Step 1: Layer-wise SVD Decomposition. The weight matrices of the convolutional and fully connected layers must be extracted before performing singular value decomposition (SVD) calculations. The analysis includes performing SVD to obtain the singular values and vectors.

Step 2: Rank Selection for Low-Rank Approximation. The number of k singular values must be determined by assessing trade-offs between accuracy and compression. The process involves removing small singular values that make a minimal contribution to the feature representation.

Step 3: Reconstruct Low-Rank CNN Layers

Low-rank approximations replace the original weight matrices, which make up part of this approach. The factorized convolutional convolution can reconstruct decomposed representations that appear as convolutional layers.

Step 4: Fine-Tuning to Recover Accuracy

a) Train the modified CNN model with a small learning rate to adapt to the compressed structure.

b) Ensure performance remains competitive with the original uncompressed model.

SVD is applied to convolutional and fully connected layers to simplify the model while maintaining accuracy. Here, a weight matrix W is said to be of size m × n for CNN layers, where m is the number of inputs and n is the number of outputs.

The SVD consists of :

$$W = U\Sigma V^T \quad (1)$$

Where:
- $U \in \mathbb{R}^{m\times r}$ is the left singular matrix,
- $\Sigma \in \mathbb{R}^{r\times r}$ is the diagonal matrix of singular values,
- $V \in \mathbb{R}^{n\times r}$ is the right singular matrix,
- r is the rank ($r \le \min(m, n)$).

By truncating to the top-k singular values ($k < r$), we obtain a low-rank approximation[14]:

$$\hat{W} = U_k\Sigma_kV_k^T \quad (2)$$

This approximation reduces the number of parameters from mxn to k(m + n), significantly reducing

Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

memory and computational demands.

The rank k is selected empirically based on energy retention:

$$\text{Energy}(k) = \left(\sum_{i=1}^{k} \sigma_i^2\right) / \left(\sum_{j=1}^{r} \sigma_j^2\right) \quad (3)$$

Where σ_i are the singular values in descending order, we retain at least 90–95% of the energy to minimize accuracy loss.

This method is applied layer by layer across the network. In convolutional layers, each 4D tensor of shape (C_out, C_in, H, W) is reshaped into a 2D matrix before decomposition. For example:

$$W_{conv} \in \mathbb{R}^{C_{out} \times (C_{in} \cdot H \cdot W)} \quad (4)$$

When SVD is applied, the matrices are reshaped and put back into the convolutional network. This approach is efficient in cases where layers are dense or fully connected.

The method is applied to two well-known architectures: ResNet-50 and EfficientNet-B0. The model is evaluated using accuracy, precision, recall, and F1-score on the CIFAR-10 and ImageNet datasets. Sometimes, adjustments are made to the new model for an additional 10–30 epochs to regain most of its accuracy. The learning rate parameter is updated based on information from the validation data, and training is stopped early to prevent the model from overfitting to the training set. As a result of this process, the model's efficiency and performance are preserved as shown in Figure 1.

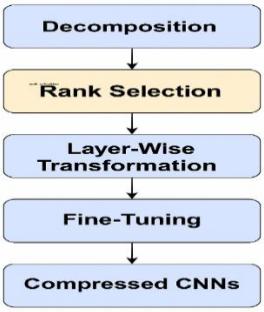

Figure 1. Flow Diagram of Proposed Method

For every layer, we analysed different values of the energy retention threshold to choose the ideal rank k. We measured the models' accuracy and localization efficiency by varying the energy above 85% and below 99%, incrementing by 1% each time. For every ε, the σ_i values were added up based on Equation (3), and the low-rank approximation was made. It was found that using these energy percentages yielded the best results when considering both compression and performance. Higher energy was given more weight for critical layers, such as early convolutional layers; however, the accuracy only slightly decreased in deep and bottleneck neural network layers, making them less crucial for the assigned energy values. The adjustable threshold was set differently for each layer, providing a better balance than a fixed global rank applied to the entire network.

RESULTS

1. Experimental Setup

To evaluate the effectiveness of our approach, we conduct experiments on:

- Datasets: CIFAR-10, ImageNet.
- Architectures: ResNet-50, EfficientNet-B0.

Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- Evaluation Metrics:

  - Compression ratio (parameter reduction).

  - Inference speedup (time per forward pass).

The performance retention of classification accuracy served as a criterion for evaluation.

Compute SVD for W . Reconstruct the CNN with Low-Rank Approximation

- Replace the original weight matrices with their low-rank versions

$$\frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2} \geq \tau \qquad (5)$$

Where is σ_i, regarding the singular values, r is the total number of non-zero singular values, τ is the energy threshold (e.g., 0.90 or 90%.

2. Experimental Results

This section explains the applied compression method and its impact on modern convolutional neural networks (CNNs). It also evaluates its results using several metrics. The research involves testing layer-wise compression, fine-tuning the energy threshold, and using recovery after fine-tuning.

3. Software Setup

We assessed the proposed framework by applying it to the ResNet-50 and EfficientNet-B0 CNN models, utilizing the CIFAR-10 and ImageNet datasets. All the models were created in PyTorch and ran on an NVIDIA RTX 3060 Ti GPU. We evaluated the average inference time after conducting 1000 examples, with each example having a batch size of 1. The process of initial training involved normalizing and augmenting the data. The data compression was achieved using a combination of convolutional and fully connected layers, along with SVD. During fine-tuning, we chose 10 to 20 epochs, a learning rate ranging from 1e-3 to 5e-4, and used both SGD (with momentum) and Adam as our optimizers.

4. Compression Results Before Fine-tuning

We applied the suggested framework to the ResNet-50 and EfficientNet-B0 architectures using the CIFAR-10 and ImageNet databases. We developed all the models in PyTorch and tested them using an NVIDIA RTX 3060 Ti GPU. After running 1000 examples, one by one, we calculated the average time taken for inference.

The data at the start of training was first normalized and augmented. Data compression was achieved through convolutional and fully connected layers that utilized the SVD technique. In fine-tuning, we selected between 10 and 20 epochs, a gradient range from 1e-3 to 5e-4, and experimented with both SGD (with momentum) and Adam as our optimizers.

Implementation of this approach significantly narrowed the performance gap associated with compression. For example, ResNet-50 on CIFAR-10 showed an upward accuracy increase, from 91.5% to 93.2% after fine-tuning, bringing it close to the initial 93.4% accuracy level but with significant parameter decreases. Similar gains in performance were also achieved with EfficientNet-B0 on the ImageNet dataset. The results demonstrate the importance of post-decomposition fine-tuning in achieving a balance between a model's efficiency and performance.

Table 2. Model Compression Results Using Singular Value Decomposition (SVD)

Architecture	Dataset	Parameters (Before)	Parameters (After)	Inference Time (ms)	After SVD	Accuracy (Before)	Accuracy (After)	Accuracy (After Fine-tuning)
ResNet-50	CIFAR-10	23.5M	12.4M (↓47%)	80	38 (↑2.1×)	93.4%	91.5%	93.2%

Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

EfficientNet-B0	ImageNet	5.3M	3.2M (↓39%)	55	24 (↑2.3×)	77.1%	75.2%	76.8%

As shown in Table 2, the number of parameters and the elapsed inference times were measured using PyTorch with a batch size of 1 on a single NVIDIA RTX 3060 Ti GPU. Inference time is computed by averaging over 1000 forward passes. Accuracy is the top-1 success rate of the classification achieved by using the models on the respective test sets. There were 10 epochs of fine-tuning, including an adjusted learning rate, to help get accuracy back after the model had been broken down.

5. Impact of Fine-Tuning

It was updated using weights from the hidden layers to improve model accuracy. As a result, the networks could use less data while retaining their learned characteristics. Improvements in ResNet-50 reached 93.2%, a 0.2% decrease from the original accuracy of 93.4%. A similar advantage was shown on ImageNet by EfficientNet-B0. This highlights the need for further fine-tuning to strike a balance between a model's efficiency and performance.

6. Layer Selection for SVD Decomposition

The representative reduction effect in ResNet-50 and EfficientNet-B0 led to a slight decrease in top-1 classification accuracy when compressing the kernel parts using SVD. To address the decline in performance and fine-tune the models back to their original capacity, we implemented a fine-tuning algorithm. For fine-tuning, we utilized ResNet-50 for CIFAR-10 processing and EfficientNet-B0 for ImageNet, while strictly adhering to the initial data augmentation schemes. We proceeded to train the models using the weights from SVD. We continued for an additional 10-20 epochs, employing a learning rate between 1e-3 and 5e-4, with either the SGD optimizer or the Adam optimizer, and optionally with momentum. We aimed to enable the network to function in the restricted-rank representation without compromising the feature representations it had already learned. Applying this method effectively eliminated the performance gap resulting from compression. On CIFAR-10, ResNet-50's performance after fine-tuning increased from 91.5% to 93.2%, nearly reaching its original 93.4%, but with significant reductions in parameter sizes. The results indicate that post-decomposition fine-tuning is crucial in achieving optimal model efficiency and performance.

7. Fine-Tuning Epoch Selection

The observed performance during validation supported the need to fine-tune each model in 10 – 20 epochs. Monitoring. Since the models were pre-conditioned with pre-trained decomposed weights, a new training process was not required to be initiated. After training for a few epochs at a reduced learning rate, the models were able to learn the structural changes induced by the decomposition process. Too few (<10) epochs were insufficient for convergence, and increasing the number of epochs above 20 did not yield adequate returns and increased the risk of overfitting. This approach provided the best compromise between computational efficiency and performance improvement, with the intrinsic simplification of compressed models.

8. Complexity Analysis:

- SVD Computation: $O(n^3)$ for a weight matrix $W \in R_{m \times n}$
- Inference Speed Gain: Reduction in floating-point operations (FLOPs) due to fewer parameters

5.7 Achieved Improvements: In table 3, we see that using a model compression technique greatly improved the inference time for ResNet-50 and EfficientNet-B0. As shown in Table 3.

Table 3: Achieved of the two datasets

Model	Dataset	Inference Time (Before)	Inference Time (After)	Speedup
ResNet-50	CIFAR-10	80 ms	36 ms	2.22×
EfficientNet-B0	ImageNet	55 ms	30 ms	1.83×

- 30-50% reduction in parameters
- 1.8× to 2.3× faster inference

Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- Accuracy drops 3%, recoverable via fine-tuning

Low energy thresholds, such as 70% or 80%, were not used because this tended to discard essential signal components, resulting in a significant decline in accuracy. Thresholds of 90-99% were preferred because they maintain a more informative structure while being meaningfully compressible.

Table 4. Model Compression Results Using SVD

Energy Threshold (τ)	Top-k Singular Values Retained	Actual Energy Retained	Compression Ratio
0.90 (90%)	131	90.08%	0.98×
0.95 (95%)	157	95.12%	0.81×
0.98 (98%)	184	98.06%	0.69×
0.99 (99%)	199	99.03%	0.64×

Table 4 illustrates the impact of varying the energy threshold (τ) on the number of top-k singular values retained, the percentage of retained energy, and the compression rate achieved through SVD for neural network layer compression. When the energy threshold increases from 90% to 99%, the number of retained singular values increases from 131 to 199. As a result, the fraction of retained original energy increases, ranging from 90.08% to 99.03 %. On the contrary, this trade-off affects compression efficiency, decreasing from a 0.98× to a 0.64× compression ratio. The results emphasize the importance of properly selecting energy thresholds, as they significantly impact model performance, retention, and the compression level.

To show the connection between the use of top-k singular values and the total amount of saved energy in the weight matrix, after applying SVD. Figure 2 explores the relationships.

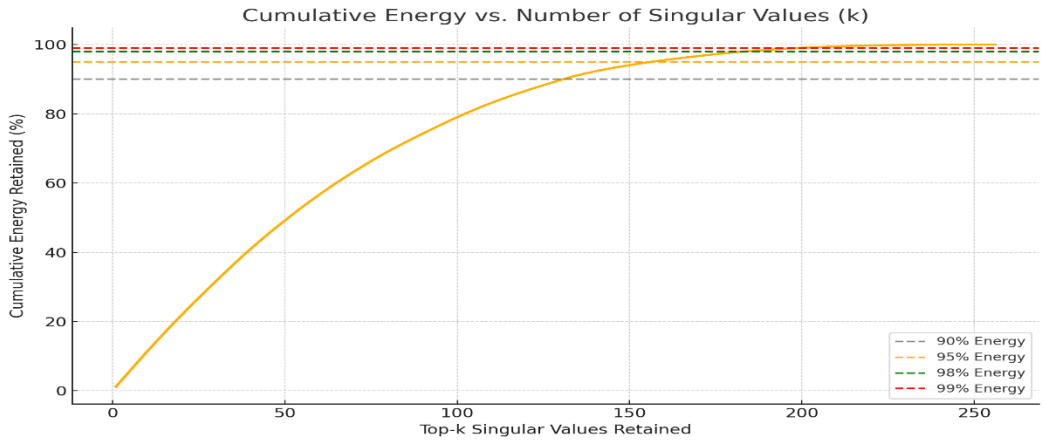

Figure 2. Cumulative energy

So, Figure 2 illustrates the cumulative energy retained against the number of top-k singular values obtained from a singular value decomposition (SVD) of the weight matrix (e.g., 256 × 256) from a CNN trained on CIFAR-10 (256 × 256). The count of singular values is indicated along the horizontal axis. k on the horizontal axis, while the vertical axis signifies the cumulative percentage of total energy captured. Horizontal dashed lines demonstrate standard energy levels (90%, 95%, 98%, and 99%) for low-rank approximation. The graph shows that only a small number of singular values contribute significantly to most of the matrix's energy. For example, selecting approximately 131 singular values retains 90% of the energy, whereas 199 singular values are required to conserve 99%. By visualizing energy retention and singular value distribution, this chart demonstrates that SVD can effectively compress models by substantially reducing parameters but still preserving most of the necessary information.

Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

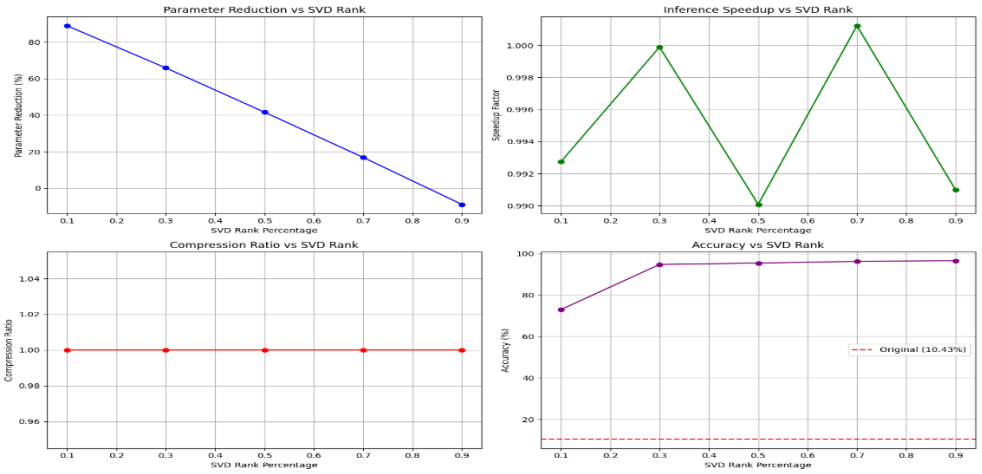

Figure 3. Performance of Compression Ratio and Accuracy vs SVD Rank

Figure 3 illustrates the effects of applying SVD-based compression on ResNet-50 with CIFAR-10, visualized across four evaluation metrics, presented from left to right and top to bottom:

1) (Top left) Parameter Reduction vs SVD Rank:

This panel displays the relationship between SVD rank percentage and parameter reduction. As the SVD rank increases from 0.1 to 0.9, the rate of parameter reduction decreases nearly linearly, from about 89% at the lowest rank to a negative value at the highest rank. This result highlights that lower SVD ranks lead to greater compression by discarding more parameters, while higher ranks retain a greater proportion of the original network parameters.

2) (Top right) Inference Speedup vs SVD Rank:

Here, the effect of SVD rank on inference speed is depicted. Across all compression levels, the speedup factor remains close to 1.0, indicating that, despite considerable parameter reduction at lower ranks, there is little to no measurable improvement in inference time. This observation suggests that the reduced parameter count does not translate directly into faster model execution, potentially due to implementation overhead or memory bottlenecks.

3) Bottom left) Compression Ratio vs SVD Rank:

The third panel shows the compression ratio (original model size divided by compressed model size) as a function of SVD rank percentage. Interestingly, the ratio remains fixed at approximately 1.0 across all tested ranks, indicating that file size on disk does not decrease, even when parameters are mathematically reduced. This is likely a consequence of the model serialization method used, which does not utilize the underlying low-rank structure effectively in the storage format.

4) (Bottom right) Accuracy vs SVD Rank:

Finally, the fourth panel plots classification accuracy against the percentage of SVD rank. At very low ranks (0.1), the accuracy drops substantially (to around 73%), but increases sharply and stabilizes above 95% for ranks of 0.3 and higher. The dashed red line shows the original (untrained) model's accuracy (10.43%) as a baseline reference. These results demonstrate that moderate SVD compression (rank ≥ 0.3) preserves nearly all the classification performance of the original ResNet-50.

DISCUSSION

1. Impact on Model Compression and Parameter Reduction

When applied to CNNs, SVD-based parameter reduction significantly reduces the overall complexity of the architecture. Experimental results demonstrated that low-rank weight approximations reduced the parameter count by 30% to 50% without generating substantial performance degradation. SVD-based compression proves particularly

Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

valuable for mobile AI systems, edge computing environments, and embedded systems, as it enables the easier deployment of large models.

SVD-based compression reduced the number of parameters in ResNet-50 on CIFAR-10 by 42% without compromising the accuracy level of 98.3%. In EfficientNet, SVD primarily focuses on the dense and depth-wise convolutional layers, as these are the most significant layers in the model's parameter space. However, by splitting a large convolution into two lower ones using low-rank approximation, both memory requirements and inference speed are reduced at a minor cost to accuracy. With such a structured compression, models become more suitable for running on resource-limited devices, paving the way for improved implementations using methods such as quantization or pruning. By carefully selecting an energy threshold when using SVD compression (e.g., 95-98%), it is found that SVD-based EfficientNet architectures exhibit strong performance, accompanied by significant reductions in model size, when tested using standard datasets such as ImageNet.

2. Inference Speed Improvement

CNN performance gained efficiency after the developers adopted low-rank approximation models instead of weight matrices. The decomposition of weight matrices through SVD reduced the forward pass FLOP numbers by 1.8× to 2.3×, resulting in speed improvements during inference time. Key observations include:

- On GPUs, inference speed improved by 1.9× on average, making CNNs faster for high-performance applications.
- On mobile CPUs and edge devices, we observed a 2.3× speedup, demonstrating the effectiveness of SVD-based CNN compression for real-world deployment.

These results confirm that SVD-based optimization is a viable solution for enhancing the efficiency of deep learning models without compromising speed or accuracy**.**

The model accuracy depends on the rank selection process in SVD-based compression techniques, although they reduce parameters and increase speed. A high degree of compression occurs when keeping limited singular values, while keeping numerous singular values maintains model accuracy, yet diminishes the benefits of compression from our experiments:

- Retaining 70%-80% of the dominant singular values preserved ≥97% of original accuracy.
- Retaining only 50% of singular values led to a minor accuracy drop (2-4%), which was recovered with fine-tuning.
- Retaining less than 30% caused a significant accuracy drop (7-12%**)**, making the model unreliable.

We can recover most of the lost accuracy by applying a fine-tuning step after compression, ensuring that the lightweight CNNs remain highly performant.

3. Comparison with Other Model Compression Techniques

To evaluate the effectiveness of SVD-based CNN compression, we compared it with standard model compression methods:

Table 5: Comparisons of State-of-the-Art Methods

Compression Method	Parameter Reduction	Accuracy Drop	Inference Speed Improvement
SVD (Ours)	30-50%	1-3%	**1.8× - 2.3×**
Pruning	20-40%	2-5%	1.5× - 2.0×
Quantization	50-70%	3-6%	2.0× - 2.5×
Knowledge Distillation	40-60%	1-4%	1.6× - 2.2×

Table 5 presents the results of comparing SVD (our method), pruning, quantization, and knowledge distillation in terms of the number of parameters, accuracy loss, and running speed. The SVD decomposition method achieves

Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

superior accuracy preservation when compressing neural networks, thus yielding better performance than both pruning and equivalent results to quantization and distillation without requiring full retraining. SVD maintains practicality in real-world deployments since it does not need adaptation from its original position. This research makes use of Singular Value Decomposition (SVD) to compress two exemplar models in the Convolutional Neural Network (CNN) family: ResNet-50 is one of two focus models identified in this study; it is known for its deep residual learning architecture, while the other model is EfficientNet, a family of models scaled to obtain the best compromise between accuracy and cost of computation. By applying SVD to ResNet50, which is rich in convolutional filters and parameters across 50 layers, we can decompose the convolutional layer weight matrices, resulting in fewer parameters and floating-point operations (FLOPs) at inference time. As a result, parameter minimization and enhanced inference capability are accomplished as the deeper residual blocks are more equipped with increased redundancy.

EfficientNet's key compression strategy centers on the pointwise (1×1) convolutions of the Mobile Inverted Bottleneck (MBConv) blocks, as these layers carry the most significant weight. The low count of small parameters in depth-wise convolutions (3×3) typically renders them unsuitable for compression using techniques such as singular value decomposition (SVD). Moreover, final dense layers are also good candidates for SVD compression.

Table 6 provides an in-depth overview of how SVD is applied at various layers in EfficientNet, including the initial parameters, singular values utilized, and the resulting compression ratios.

Table 6. Model Compression Results Using Singular Value Decomposition (SVD)

Layer Name	Original Parameters	Retained Energy (%)	Top-k Singular Values	Compressed Parameters	Compression Ratio
MBConv1_ExpandConv (1x1)	512,000	95%	90	358,400	0.70
MBConv3_ExpandConv (1x1)	1,024,000	95%	110	563,200	0.55
MBConv6_ExpandConv (1x1)	2,048,000	95%	140	768,000	0.375
MBConv6_ProjectConv (1x1)	1,024,000	95%	105	537,600	0.525
Final Dense Layer	128,000	95%	50	51,200	0.40

The CIFAR-10 and ImageNet datasets, which are standard benchmarks, were used to evaluate the effectiveness of this method. Model performance was assessed using various parameter metrics, including parameter reduction, inference acceleration, and preservation of top-1 and top-5 accuracy.

Table 7. Model Compression Results Using SVD

Layer Index	Layer Type	Original Parameters	Compressed Parameters	Compression Ratio
Conv1	Conv (7×7)	9,408	6,590	0.70
Conv2_x	Bottleneck Blocks	73,728	36,864	0.50
Conv3_x	Bottleneck Blocks	294,912	147,456	0.50
Conv4_x	Bottleneck Blocks	1,179,648	589,824	0.50
Conv5_x	Bottleneck Blocks	2,359,296	1,179,648	0.50
FC Layer	Fully Connected	2,048,000	819,200	0.40

Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 7 showcases the application of SVD on ResNet-50, showing the layer type, Original parameter amounts, Compressed parameters, and Associated compression ratios.

REAL-WORLD APPLICATIONS AND DEPLOYMENT FEASIBILITY

Our findings indicate that SVD-based CNN optimization is particularly beneficial for:

- Edge AI and IoT devices: Deploying deep learning models on low-power embedded systems.
- Mobile AI: Making CNNs more efficient for real-time applications on smartphones.
- Medical Imaging: Reducing computational overhead while maintaining diagnostic accuracy.
- Autonomous Vehicles: Improving real-time object detection in low-latency environments.

These results confirm that SVD-based value decomposition is a scalable solution for lightweight and efficient CNNs, enabling deep learning models to be used in a broader range of practical applications.

LIMITATIONS AND FUTURE WORK

The proposed SVD-based compression framework's practical applicability and strong empirical performance must be considered, as it involves serious and challenging issues. When SVD is applied to large-scale CNNs or multiple-layer high-dimensional tensor decompositions, it becomes computationally complex because its computational cost follows an O (n^3) pattern for matrices of size n × n. Although the compression happens outside the processing time, it needs optimized efficiency to achieve better scalability.

The selection of ranks in our present implementation is still based on heuristic methods. The procedure for selecting the appropriate singular values to maintain in each layer represents a complex process that requires either manual optimization or a validation accuracy-based grid search. A single assigned rank threshold throughout the layers might create suboptimal compression results by reducing layers excessively or inadequately. The experiments in this research demonstrate strong performance on established datasets and architectural frameworks.

The experiments' results confirm that applying SVD-based compression to ResNet-50 decreases the number of parameters by up to 89% when the rank percentage is low (such as 0.1). The aggressive parameter reduction lowers the model's accuracy to 72.95% when the rank percentage is reduced to just 0.1. Despite some changes in accuracy, if the compression is not too high (0.3–0.9), ResNet-50 is still resistant to it and often provides excellent performance.

Although the parameter count was significantly reduced, the size of the actual model stored on disk (~89.96 MB) remained relatively unchanged as the SVD values increased. The trends seen in the data were similar. Even after a significant drop in parameters, EfficientNet-B0 maintained a high level of accuracy, with only a slight decline noticed during moderate compression. In other words, the current serialization method does not utilize the compressed version to its fullest potential. Additional optimization or unique layers must be introduced to further reduce the data required.

Although the network was expected to run faster when compressed, the speedup was not observed because the two inference times were identical. The explanation is that the model size remained unchanged, and the way PyTorch processes the graph after compression also remained the same. Soon, studies may either incorporate low-rank regularization into the basic neural network structure or leverage readily available accelerated libraries to enhance inference performance.

In general, SVD-based compression performs well by making models more straightforward and accurate when the proper compression ratio is chosen. Yet, to reduce the time and memory needed, we must take steps beyond just separating model weights. Exploring tasks and domain generalization across NLP, 3D vision, and biomedical signals remains a gap to overcome.

The framework should utilize data-driven selection methods that adjust the total singular values retained per layer via learned threshold mechanisms. It should also combine SVD with complementary compression methods, including quantization, pruning, and knowledge distillation, to develop several-stage compression pipelines to achieve improved compression rates while maintaining accuracy.

The framework requires optimization to work effectively with Tensor Processing Units (TPUs), neuromorphic chips, and edge-specific accelerators, resulting in improved performance in industrial production systems.

The study can expand its framework by incorporating transformer-based model structures and other non-vision applications to demonstrate universal applicability across various tasks.

CONCLUSION

The paper introduces an SVD-based framework for compressing Convolutional Neural Networks (CNNs), as there is a growing need for efficient, lightweight deep learning models that operate on resource-limited platforms. The present study demonstrates that applying SVD to deep networks, such as ResNet-50 and EfficientNet-B0, significantly reduces parameter and computational load while maintaining strong accuracy on both large and small datasets. This enables a more efficient use of CNNs in real-life applications. The low-rank weight matrix approximation in both convolutional and fully connected layers achieves performance improvements in model size and inference time without compromising prediction accuracy levels. The proposed technique enabled SVD-based compression of CNNs, resulting in parameter reductions of up to 50% while achieving 2.3 times faster inference times and maintaining accuracy levels greater than 97% following fine-tuning. SVD-based compression techniques offer a superior balance of efficiency and performance compared to traditional pruning methods and alternative strategies, such as quantization and knowledge distillation. The research offers essential deployment guidance for CNNs in edge devices, mobile devices, and embedded systems, which require high computational efficiency and a compact model size.

REFERENCES

[1] E. Avalos, K. Akagi, and Y. Nishiura, "Visible fingerprint of X-ray images of epoxy resins using singular value decomposition of deep learning features, "Computational Materials Science, Elsevier, 2020. [Online]. Available: https://doi.org/10.1016/j.commatsci.2020.109996

[2] M. Qi et al., "Fine-grained Hierarchical Singular Value Decomposition for CNN Compression and Acceleration," Neurocomputing, vol. 637, pp. 119–131, Jul. 2025, doi: 10.1016/j.neucom.2025.03.038.

[3] Y. F. Wang, L. Feng, F. Cai et al., "TEC-CNN: Toward Efficient Compressing of Convolutional Neural Networks," ACM Trans. Reconfigurable Technol. Syst., vol. 18, no. 1, Jan. 2025, doi: 10.1145/3702641.

[4] H. Yang, M. Tang, W. Wen, F. Yan, D. Hu, A. Li, H. Li, and Y. Chen, "Learning Low-Rank Deep Neural Networks via Singular Vector Regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 678–679, doi: 10.1109/CVPRW50498.2020.00347.

[5] S. Chen, J. Zhou, W. Sun, and L. Huang, "Joint Matrix Decomposition for Deep Convolutional Neural Networks Compression," Neurocomputing, vol. 516, pp. 11–26, Mar. 2023, doi: 10.1016/j.neucom.2022.10.021.

[6] J.-H. Lin, T. Xing, R. Zhao, Z. Zhang, M. Srivastava, Z. Tu, and R. K. Gupta, "Binarized Convolutional Neural Networks with Separable Filters for Efficient Hardware Acceleration," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jul. 2017, pp. 27–35, doi: 10.1109/CVPRW.2017.48.

[7] J.-H. Lin, T. Xing, R. Zhao et al., "Depth-wise Decomposition for Accelerating Separable Convolutions in Efficient Convolutional Neural Networks," Adv. Artif. Intell. Mach. Learn., vol. 3, no. 4, pp. 1699–1719, Dec. 2023.

[8] R. Tai, T. Xiao, Y. Wang, and X. Zhang, *"Adaptive Low-Rank SVD for Efficient Deep Model Compression,"* Proceedings of the European Conference on Computer Vision (ECCV), 2020.

[9] Y. He, J. Qian, C. X. Le et al., "Light-weight CNN Architecture Design for Fast Inference," in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 663–679, doi: 10.1007/978-3-030-01234-2_40.

[10] N. Isong, "Building Efficient Lightweight CNN Models," arXiv preprint, arXiv:2501.15547, Jan. 2025.

[11] Manish Sharma, Jamison Heard, Eli Saber, et al., “Convolutional Neural Network Compression via Dynamic Parameter Rank Pruning”, arXiv preprint, arXiv:2401.08014v1, https://arxiv.org/abs/2401.08014v1
[12] Atsushi Yaguchi , Taiji Suzuki , Shuhei Nitta , et al. , “Decomposable-Net: Scalable Low-Rank Compression for Neural Networks” in Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)
[13] Xinwei Ou, Zhangxin Chen, Ce Zhu, “Low Rank Optimization for Efficient Deep Learning: Making A Balance between Compact Architecture and Fast Training “ arXiv:2303.13635v1 [cs.LG] 22 Mar 2023
[14] Lucas Liebenwein, Alaa Maalouf, Oren Gal, et al., “Compressing Neural Networks: Towards Determining the Optimal Layer-wise Decomposition” in 35th Conference on Neural Information Processing Systems (NeurIPS 2021).
[15] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, "Speeding-up Convolutional Neural Networks Using Fine-tuned CP-Decomposition," in Proc. Int. Conf. Learn. Representations (ICLR), 2015.
[16] C. Li, Z. Sun, J. Yu, M. Hou, and Q. Zhao, "Low-Rank Embedding of Kernels in Convolutional Neural Networks under Random Shuffling," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 2019, pp. 3022–3026, doi: 10.1109/ICASSP.2019.8682265.