

Feature map quantification: An efficient approach for active trachoma image classification

Author

Zewudie, Mulugeta Shitie, Xiong, Shengwu, Yu, Xiaohan, Wu, Xiaoyu

Published

2025

Journal Title

Computers in Biology and Medicine

Version

Version of Record (VoR)

DOI

[10.1016/j.combiomed.2025.111295](https://doi.org/10.1016/j.combiomed.2025.111295)

Rights statement

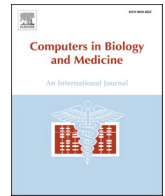
© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Downloaded from

<https://hdl.handle.net/10072/440329>

Griffith Research Online

<https://research-repository.griffith.edu.au>



Feature map quantification: An efficient approach for active trachoma image classification

Mulugeta Shitie Zewudie^{a,b}, Shengwu Xiong^b, Xiaohan Yu^{c,*}, Xiaoyu Wu^d

^a Department of Information Technology, Debark University, Debark, Ethiopia

^b School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, China

^c School of Computing, Macquarie University, Sydney, Australia

^d Institute for Integrated and Intelligent Systems, Griffith University, QLD, Australia

ARTICLE INFO

Keywords:

Active trachoma
Feature map
Feature similarity
Filter pruning
Singular value decomposition

ABSTRACT

Convolutional neural networks (CNNs) classify inverted eyelid images related to active trachoma. However, using these complex networks in medical service centers faces challenges due to computational resource constraints. To overcome this challenge, we propose a quantified feature map-based filter pruning framework (FSIM-SVD) that relies on feature similarity and feature map contributions. Based on these insights, our approach involves quantifying redundant feature maps using feature similarity (FSIM) and assessing the contribution of each feature map through singular value decomposition (SVD). By analyzing the impact of each component on the overall model performance, less significant filters can be identified and pruned. The experiment uses VGG16, ResNet56, and ResNet110 on the active trachoma and CIFAR10 datasets. The results reveal that VGG16 achieved an accuracy of 86.9 % (+0.43 % from the baseline) for active trachoma classification while reducing FLOPs by 28.6 % and parameters by 33.4 %. In the CIFAR10 classification, ResNet110 achieved an accuracy of 94.31 % (+0.73 % from the baseline) with a 43.8 % reduction in FLOPs and a 43.1 % reduction in parameters. Compared to state-of-the-art compression techniques, the proposed approach achieves a higher pruning rate and improved classification performance.

1. Introduction

The success of deep neural networks, especially in computer vision, has resulted in the development of complicated and multi-stage models [1–4]. Some methods, such as convolutional neural networks (CNN), require a significant number of parameters and extensive floating point operations (FLOPs) to achieve satisfactory performance [5]. For example, VGG16, among the most renowned and efficient deep learning models, has approximately 138.34 million parameters and requires more than 30.94 billion FLOPs to recognize a single 224×224 input image. While deep learning models greatly enhance prediction outcomes, training and deploying CNN models of this nature often require high-performance computational devices like graphics processing units (GPUs). However, there is a significant demand for developing and deploying CNNs on mobile phones and Internet of Things (IoT) devices, which have limited computational resources. Achieving satisfactory performance on such devices is highly desirable but presents notable challenges [2–4,6,7]. Access to high-performance computing resources

and adequate training data can often be limited in rural tropical areas, exacerbating the challenge of detecting active trachoma. Despite rapid progress in medical image analysis, most previous works have primarily focused on classification accuracy while only briefly addressing the computational cost of the underlying architectures. Recent studies on breast cancer histopathology [8], COVID-19 CT-scan analysis [9], and filter pruning strategies [10] reveal that state-of-the-art CNNs are often over-parameterized, requiring millions of weight parameters and billions of FLOPs, which limits deployment on resource-constrained devices. Even when pruning is applied, computational efficiency is often treated as a secondary benefit rather than a primary design goal. For example, pruning-based methods have reported FLOP reductions of up to 63 % in VGG19 while retaining competitive accuracy, highlighting the importance of balancing accuracy with computational efficiency [8]. These findings underscore the need for approaches that explicitly integrate computational cost considerations into the design and evaluation of CNN models. To address this issue, researchers have proposed various model compression techniques such as model pruning [1,3,11],

* Corresponding author.

E-mail address: xiaohan.yu@mq.edu.au (X. Yu).

<https://doi.org/10.1016/j.combiomed.2025.111295>

Received 21 October 2024; Received in revised form 25 September 2025; Accepted 9 November 2025

Available online 14 November 2025

0010-4825/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

quantization [12,13], and knowledge distillation [1–3,14]. Model pruning reduces model size and computation by removing redundant or less important parameters, including weights, filters, or neurons from the network, effectively creating a sparse architecture without significantly affecting accuracy. Quantization compresses models by reducing the numerical precision of weights and activations, such as converting from 32-bit floating point to 8-bit integer representations, thereby lowering memory usage and computational costs. Knowledge distillation offers another approach by transferring knowledge from a large, high-performing "teacher" model to a smaller "student" model, where the student is trained to mimic the teacher's output or feature representations, enabling efficient deployment without replicating the teacher's full complexity. These techniques collectively provide various strategies for making deep learning models more practical for real-world deployment scenarios. Filter pruning is one of the most common techniques among various model pruning methods and has shown promising results in a wide range of applications. It operates on the principle that certain filters within a network are less critical than others in carrying information from the input to the output. Removing these less significant filters should minimally affect the prediction results. Filter pruning has proven effective in reducing the number of FLOPs needed in a CNN [15]. The main techniques for filter pruning can be categorized into weight-based filter pruning [16,17] and feature map-based filter pruning [18–20]. Filter pruning based on weights is a data-independent technique known for its robust generalization abilities. Nevertheless, achieving optimal performance across diverse datasets proves challenging without prior data insights. In contrast, methods that use feature map-based filter pruning leverage information about the data distribution to refine the pruning strategy and improve efficiency. The relevance of a particular feature map is directly associated with its corresponding filter. Consequently, feature map-based pruning approaches conduct filter pruning by introducing an evaluation function to measure the importance of the feature maps [21].

In recent studies, researchers have introduced effective methods for compressing CNN models. However, there is a significant gap in the literature regarding integrating two critical factors related to feature maps: feature similarity (FSIM) of low-level features and Singular Value Decomposition (SVD). Using SVD helps identify crucial information within feature maps while considering the similarity of low-level features between feature maps, which is essential for preserving valid features. For instance, if two feature maps have highly similar low-level features, it becomes reasonable to consider one redundant. Unfortunately, previous research has neglected to examine the similarity of low-level features and the SVD of feature maps, resulting in the ineffective elimination of redundant parameters from CNN models. Our research proposes an innovative approach to active trachoma detection and classification through filter pruning to address this gap. We chose to focus on active trachoma detection due to both its significant public health relevance and the unique technical challenges it presents. Clinically, active trachoma is the leading infectious cause of blindness worldwide, with the highest prevalence in resource-limited regions where timely diagnosis is essential but access to trained ophthalmologists is limited [22]. Automated detection systems can play a key role in large-scale screening and early intervention in such settings. Additionally, the high computational requirements of many CNN architectures limit their deployment in low-resource medical centers [8]. Our motive is to develop a model with minimal parameters that can achieve high accuracy in detecting active trachoma while being easily integrable without imposing significant computational requirements.

The process begins by obtaining the output feature maps for each layer in the model architecture. Subsequently, we analyze the similarity of low-level features within each output feature map using FSIM. Additionally, we evaluate the information content of each output feature map through SVD. The FSIM-SVD approach measures the significance of each output feature map through a two-step process. First, the Feature Similarity (FSIM) is used to evaluate redundancy among feature maps by

capturing low-level perceptual information such as phase congruency and gradient magnitude. This step ensures that structurally similar or redundant feature maps can be identified. Second, Singular Value Decomposition (SVD) is applied to feature maps to quantify their contribution. The distribution of singular values reflects the amount of unique and discriminative information preserved in each feature map: higher singular values correspond to more informative maps, whereas lower values indicate weaker contributions or redundancy. By combining these two measures, FSIM-SVD generates a significance score for each feature map, enabling the framework to retain highly informative filters while pruning less significant ones. This strategy ensures that computational efficiency is improved without sacrificing classification performance. This intentional removal of filters streamlines the model, and ensures that computational efficiency is improved without sacrificing classification performance.

The following are the primary contributions of this research.

1. The study acknowledges the challenges of implementing complex CNNs in medical service centers due to limited computational resources. To address this challenge, the proposed FSIM-SVD framework focuses on quantified feature map-based filter pruning, effectively reducing model parameters and FLOPs.
2. We introduce a novel approach that combines FSIM and SVD to assess the significance of each feature map. The methodology achieves a more accurate evaluation of feature maps by leveraging the human visual system's sensitivity to image features (FSIM) and identifying crucial information within feature maps (SVD).
3. We evaluated the FSIM-SVD method using the active trachoma and CIFAR10 datasets. The results indicate that FSIM-SVD is more effective in accuracy, has reduced FLOPs, and has decreased parameters.

The paper follows the following format: Section 2 presents the related work, while Section 3 describes the proposed method. Dataset description and parameter setting are presented in Section 4, followed by the result and discussion in Section 5. Finally, Section 6 concludes the paper.

2. Related work

Deep Learning (DL) has been proven successful in various medical image classification tasks, surpassing clinical classification performance. Researchers have also implemented DL image detection and classification systems for trachoma, specifically targeting active trachoma image classification. For instance, Kim et al. [22] Proposed a CNN for detecting clinical trachoma signs, focusing on two stages of active trachoma: trachomatous inflammation follicular (TF) and trachomatous inflammation intense (TI) [23,24]. The model achieved 70 % accuracy for TF and 85 % for TI cases. Yenegeta and Assabie [25] proposed a texture-feature-based CNN for the detection and grading of trachoma. They extracted salient texture features from eye images using Gabor filters and achieved a 97.9 % accuracy. However, their approach only classified three stages of trachoma (TS, TT, and CO) and did not cover active trachoma. Socia et al. [26] proposed trachoma classifiers using ResNet101 and VGG16 CNN models and employed over-sampling techniques on positive images to balance the data. Their study focused only on TF, one of the five stages of trachoma, and yielded 95 % recall and 68 % precision values.

Previous research has focused on achieving excellent outcomes through deep neural networks with numerous stages. However, the high cost of training these models raises concerns about their practicality in a clinical environment. In clinical settings, the main goal is to accurately identify and classify active trachoma images using cost-effective and easily accessible devices, such as smartphones or embedded devices. This goal proves challenging with deep models and multi-stage techniques. There is a need to develop learning methods that utilize less

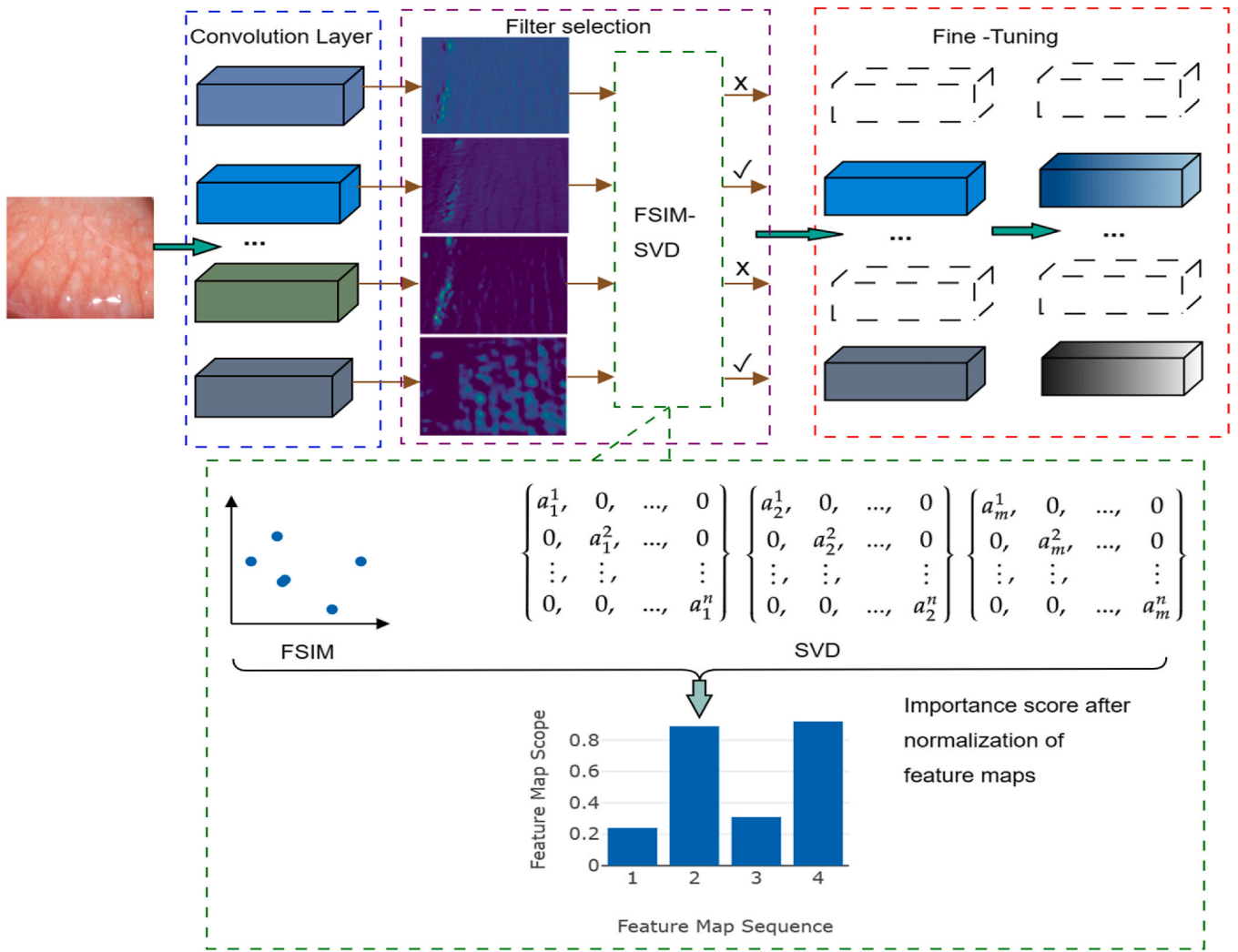


Fig. 1. Shows the proposed method. Within the convolutional layer, convolution is executed, resulting in output feature maps for each model layer. Subsequently, FSIM calculates low-level feature similarity, and SVD is applied to determine the contribution of each feature map. The FSIM-SVD method is then employed to measure the importance of each feature map, generating a set of quantified values. Ultimately, the network structure undergoes pruning and fine-tuning guided by these quantified values.

expensive hardware while maintaining or surpassing the performance of deep or multi-stage frameworks to address this challenge. One potential approach for compressing networks involves training smaller models with reduced parameters using effective filter pruning methods. For example, Hu et al. [27] determined the importance of a filter by calculating the percentage of zero activations in the feature map. Chen Z et al. [28] introduced a dynamic channel pruning technique that removes less crucial channels at the initial stages of training. The evaluation of channel importance directly influences the final accuracy of the network. Unlike conventional attribute-based methods that prune fixed ratios layer by layer, this approach ensures adaptability, avoiding a less flexible model. The adaptive importance methods alter the network’s loss function, causing the importance indicators of specific channels to converge to zero. This alteration is achieved by assigning importance scores to filters based on the network’s learning of the input-output mapping and comparing these scores across all convolutional filters [29]. By dynamically removing unimportant channels, these methods can significantly reduce the computational burden while maintaining performance [30].

Additionally, some methods utilize a threshold of the L1-norm of pruned weights to ensure no degradation of the loss function, allowing for adaptive pruning rates per layer [31]. While these methods can achieve a substantial compression ratio, they often necessitate

additional hyperparameter training, leading to less efficient model pruning. Lin et al. [32] introduced the HRank filter pruning method, which targets filters with low-rank feature maps. This approach substantially reduces FLOPs and parameters while experiencing minimal accuracy loss. Although high-rank feature maps may also contain redundant information, they remain untouched. Chen et al. [33] presented FPC, a filter pruning method centered on the contribution of the output feature map. SVD decomposes the output feature map and effectively eliminates filters with low contribution, thereby maintaining model performance. However, this method overlooks the possibility that low-contributing output feature maps may still contain valuable information. Yajun et al. [19] proposed techniques for pruning filters that measure features’ similarity and the entropy of feature maps. The importance of filters is determined by considering the richness of information in the feature maps and the similarity among underlying features. However, it’s important to note that entropy cannot offer insights into the significance of specific features in the data.

Based on the gaps above, we propose filter pruning using feature maps. It involves selectively removing filters from a neural network based on the similarity between feature maps and their contribution to the overall model. This process typically includes techniques such as FSIM to quantify the similarity between feature maps and SVD to measure the contribution of each feature map. By analyzing the contribution

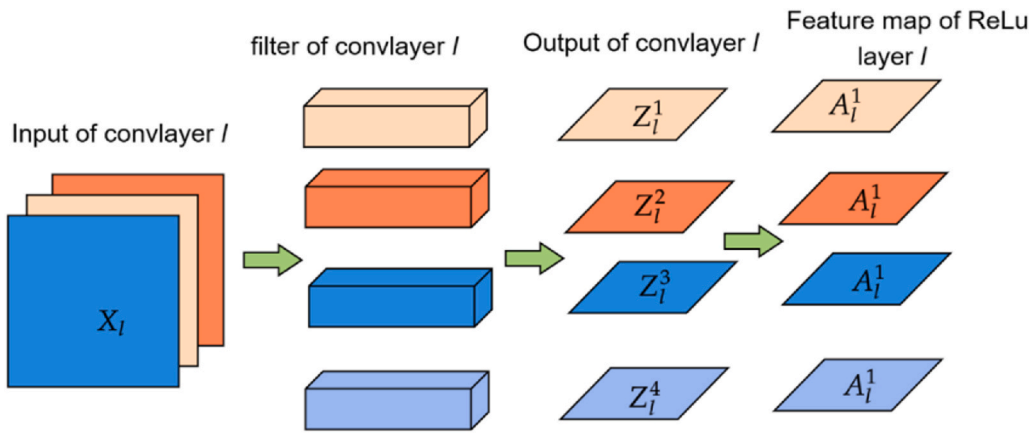


Fig. 2. Internal computation of a convolutional layer in a neural network, illustrating how filters convolve with input feature maps to produce activated output feature maps for hierarchical feature learning. This operation serves as the foundation for the process depicted in Fig. 1, where the resulting feature maps are further analyzed using FSIM and SVD for importance quantification, pruning, and network optimization.

of each component to the overall model performance, less significant filters can be identified and pruned. This strategic filter removal aims to streamline the model, reduce computational complexity, and potentially enhance generalization capabilities by retaining only the most essential features for accurate predictions.

3. Method

3.1. Motivation

The typical use of CNNs in medical image analysis often results in models with excessive parameters, rendering them computationally expensive and unsuitable for deployment in resource-constrained settings, such as clinics or remote areas with limited access to high-performance computing resources. The main reason for this lies in the convolutional operation, which generates various feature maps; some of these feature maps may be redundant and offer different contributions. Feature maps with similar or redundant features and fewer contributions are not crucial for model accuracy; instead, they only increase model parameters and FLOPs. Feature maps that are highly similar or redundant often capture overlapping structural patterns, meaning their informational content is already represented elsewhere in the network. As a result, retaining all such maps does not significantly improve the model's discriminative ability. Similarly, feature maps with lower contributions, as indicated by their singular value distribution, contain limited unique information and add little to the decision-making process. Pruning these redundant or weakly informative maps reduces computational complexity without compromising accuracy, since the remaining feature maps preserve the critical and diverse representations needed for classification. To address this problem, various studies propose methods to measure the importance of feature maps and apply filter pruning, as the relevance of a particular feature map is directly associated with its corresponding filter [21]. Chen et al. [33] eliminate filters based on the contribution of the output feature map, but their method overlooks the potential value in low-contributing output feature maps. Yajun et al. [19] proposed techniques that consider features' similarity and the entropy of feature maps for filter pruning, thereby offering a holistic approach to filter selection. However, entropy cannot provide insights into the significance of specific features in the data. In response to these gaps, we propose a novel FSIM-SVD framework that differs from conventional pruning and dimensionality reduction techniques by jointly addressing feature redundancy and contribution in a unified manner. Specifically, FSIM is employed to quantify structural and textural similarity among feature maps, moving beyond purely statistical or weight-based criteria, while SVD is applied to assess the

representational contribution of each feature map through a rank-based evaluation of information content. By integrating these two complementary measures, the framework ensures that only feature maps with both low uniqueness and low contribution are pruned, thereby minimizing the risk of discarding informative features. Moreover, the FSIM-SVD approach is explicitly designed with practical deployment in mind, aiming to maintain or even enhance classification accuracy while significantly reducing FLOPs and parameters, making it particularly suitable for medical imaging applications in resource-constrained healthcare settings.

Fig. 1 illustrates the proposed FSIM-SVD-based pruning framework, which operates across all convolutional layers of the backbone CNN (e.g., VGG16, ResNet variants). The framework consists of three main stages. First, the output feature maps are extracted from each convolutional layer in the network. Second, for each layer, FSIM is applied to evaluate the low-level structural similarity between feature maps, while SVD is used to measure the individual contribution of each feature map to the network's representational capacity. These two complementary metrics are then combined to produce a quantified importance score for every feature map. Finally, filters corresponding to feature maps with low importance scores are pruned, and the resulting network is fine-tuned to recover performance. This layer-wise approach ensures that redundancy and low-contribution filters are identified and removed throughout the network, enhancing compression efficiency while preserving accuracy.

3.2. Convolution operation

As shown in Fig. 2, in the l -th convolutional layer of a neural network, the computation process involves convolving filters with the output feature maps from the $(l-1)$ -th layer. Each filter, represented by W_l^i , undergoes a convolution operation with the input feature maps X_l^i , producing intermediate feature maps Z_l^i as shown in equation (1). An activation function f is applied, resulting in the feature map A_l^i . If pooling is employed, the feature maps are further downsampled. The final output of the l -th layer, denoted as A_l^i , as shown in equation (2), consists of a set of feature maps obtained through these operations. The subscript denotes the index of the convolutional layer, while the superscript represents the index within that layer. This process is repeated for subsequent layers, allowing the network to learn hierarchical representations from the input data, ultimately contributing to the network's ability to capture complex patterns and make accurate predictions. We evaluate the similarity and contribution of the output feature map A_l^i in each layer. Based on the combined similarity and contribution values of the feature map, we can determine the importance of the filters that

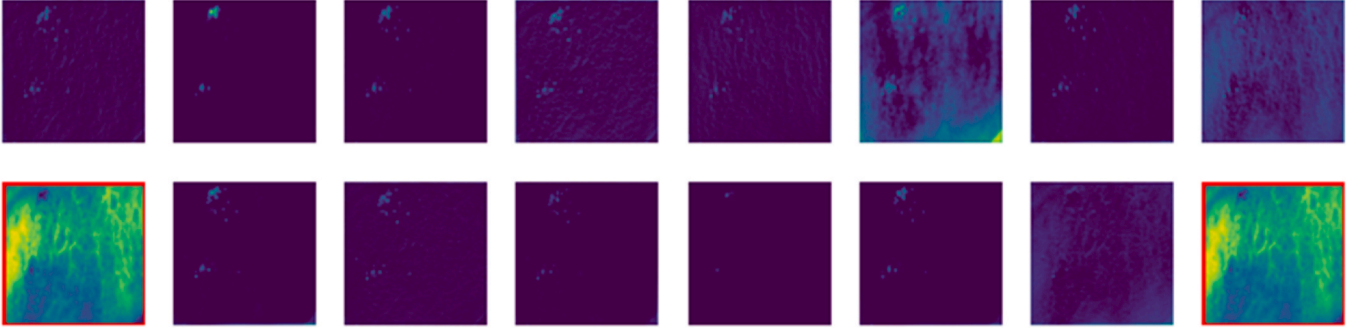


Fig. 3. Show the feature map from the initial convolutional layer of VGG16, revealing a notable similarity among specific feature maps. The visual examination of the two feature maps highlighted in red indicates a significant similarity.

generate it.

In the proposed FSIM-SVD framework, this convolutional filter layer serves not only as the standard feature extractor but also as a dedicated pre-processing stage for the similarity and decomposition steps. By learning to emphasize salient low-level structures, such as edges, textures, and local gradients, the layer produces clean and information-rich feature maps A_i^l . These refined maps provide a stable basis for the next stage, where FSIM quantifies the similarity among feature maps to detect redundancy and SVD decomposes the selected maps to capture their intrinsic low-rank structure. Evaluating the similarity and contribution of each output feature map A_i^l therefore allows the method to determine the importance of the filters that generate it and to prune or retain filters based on their true representational value.

$$Z_i^l = X_i^l W_i^l \quad (1)$$

$$A_i^l = f(Z_i^l) \quad (2)$$

3.3. Feature similarity (FSIM)

The FSIM technique demonstrates a remarkable ability to capture low-level features, including edges, textures, contrast, and shape, within an image by maintaining phase consistency. This unique characteristic makes FSIM a strong candidate for quantifying visual quality in a manner that closely resembles the perception of the human visual system. Specifically, the phase consistency (PC) component captures structural and shape information, while the gradient magnitude (GM) component reflects contrast and texture variations. Together, these low-level visual cues define the geometry and fine details in feature maps. When applied to CNN feature maps, FSIM exclusively assesses low-level feature similarity because it was originally designed for image quality evaluation based on the human visual system's sensitivity to structural details such as edges, textures, and color patterns. In this context, FSIM measures the degree of similarity among feature maps within the same convolutional layer by focusing on structural and textural similarities while disregarding high-level semantic content. This ensures that redundancy is identified purely at the level of visual details rather than object-level interpretations. Fig. 3 provides an example of the output feature maps from the first convolutional layer in VGG16, which consists of 16 maps and illustrates the presence of similarity among them.

A similarity measure is valuable in tasks such as image quality assessment, where understanding the similarity of feature maps can contribute to evaluating redundant features. To assess FSIM between the j -th and k -th feature maps within the same layer, let's consider the output, i.e., feature maps, of the l -th convolutional layer of the CNN model, denoted as $F^l = \{f_1^l, f_2^l, \dots, f_j^l, \dots, f_{N_l}^l\} \in \mathbb{R}^{N_l \times H_l \times W_l}$. The similarity between the two feature maps is measured in Equation (3):

$$FSIM_{j,k}^l = FSIM(f_j^l, f_k^l) = \frac{\sum_{f_j^l, f_k^l \in \mathbb{R}} S_L(f_j^l, f_k^l) \cdot PC_m(f_j^l, f_k^l)}{\sum_{f_j^l, f_k^l \in \mathbb{R}} PC_m(f_j^l, f_k^l)} \quad (3)$$

where $FSIM_{j,k}^l$ denotes the similarity between the j -th and k -th feature maps of the l -th convolutional layer.

The weighting factor $PC_m(f_j^l, f_k^l) = \max(PC(f_j^l), PC(f_k^l))$ is employed to assign a weight to the similarity of feature maps f_j^l and f_k^l . This weighting factor is determined by selecting the maximum value between the phase consistency information of f_j^l and f_k^l denoted as $PC(f_j^l)$ and $PC(f_k^l)$ respectively.

Here, $PC(f_j^l)$ and $PC(f_k^l)$ signify the phase consistency information related to the feature maps f_j^l and f_k^l respectively. The $S_L(f_j^l, f_k^l)$ represents the similarity between the feature maps f_j^l and f_k^l . This similarity is computed using the following formula:

$$S_L(f_j^l, f_k^l) = [S_{PC}(f_j^l, f_k^l)]^\alpha \cdot [S_{GM}(f_j^l, f_k^l)]^\beta \quad (4)$$

where $S_{PC}(f_j^l, f_k^l)$ indicates the feature similarity between the feature maps f_j^l and f_k^l , capturing how closely their structural characteristics align. On the other hand, $S_{GM}(f_j^l, f_k^l)$ presents the gradient similarity between these feature maps, emphasizing the resemblance in their gradient distributions, α and β are positive numbers, α adjusts the weight assigned to feature similarity. At the same time, β controls the weight associated with gradient similarity, and $\alpha = \beta = 1$ [19] in this study.

$$S_{PC}(f_j^l, f_k^l) = \frac{2PC(f_j^l) \cdot PC(f_k^l) + T_1}{PC^2(f_j^l) \cdot PC^2(f_k^l) + T_1} \quad (5)$$

$$S_{GM}(f_j^l, f_k^l) = \frac{2GM(f_j^l) \cdot GM(f_k^l) + T_2}{GM^2(f_j^l) \cdot GM^2(f_k^l) + T_2} \quad (6)$$

where T_1 and T_2 are both positive constants that prevent the denominator from becoming zero, thereby increasing the stability of $S_{PC}(f_j^l, f_k^l)$ and $S_{GM}(f_j^l, f_k^l)$. In this study, we take the values $T_1 = 0.85$ and $T_2 = 160$ were taken directly from the original FSIM work by Ref. [19], where these values were empirically determined to yield stable and accurate low-level feature similarity measurements in image quality assessment tasks. In our FSIM-SVD framework, we adopted these same constants because they are widely validated in prior literature and ensure consistency and comparability with established FSIM-based methods.

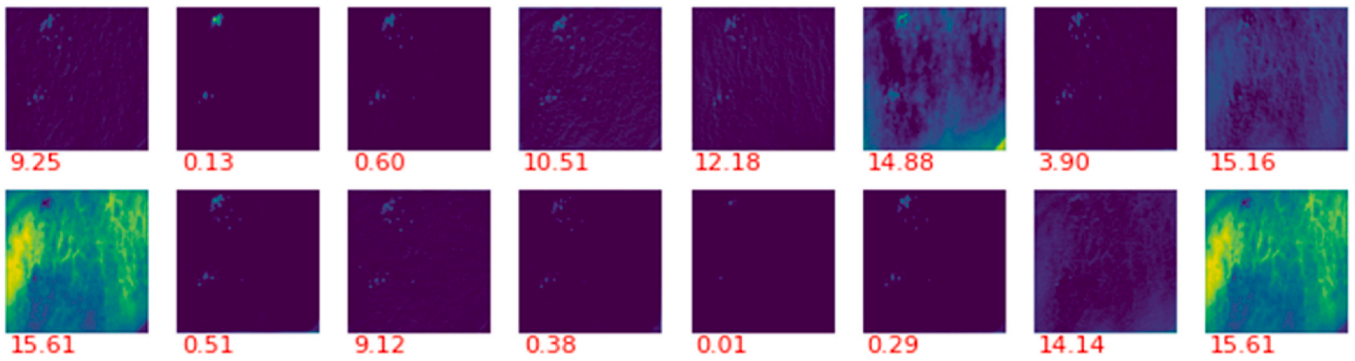


Fig. 4. The figure shows the distribution of information within feature maps of different layers. The SVD provides a quantitative measure of the diversity or uniformity of information in each channel of the feature maps.

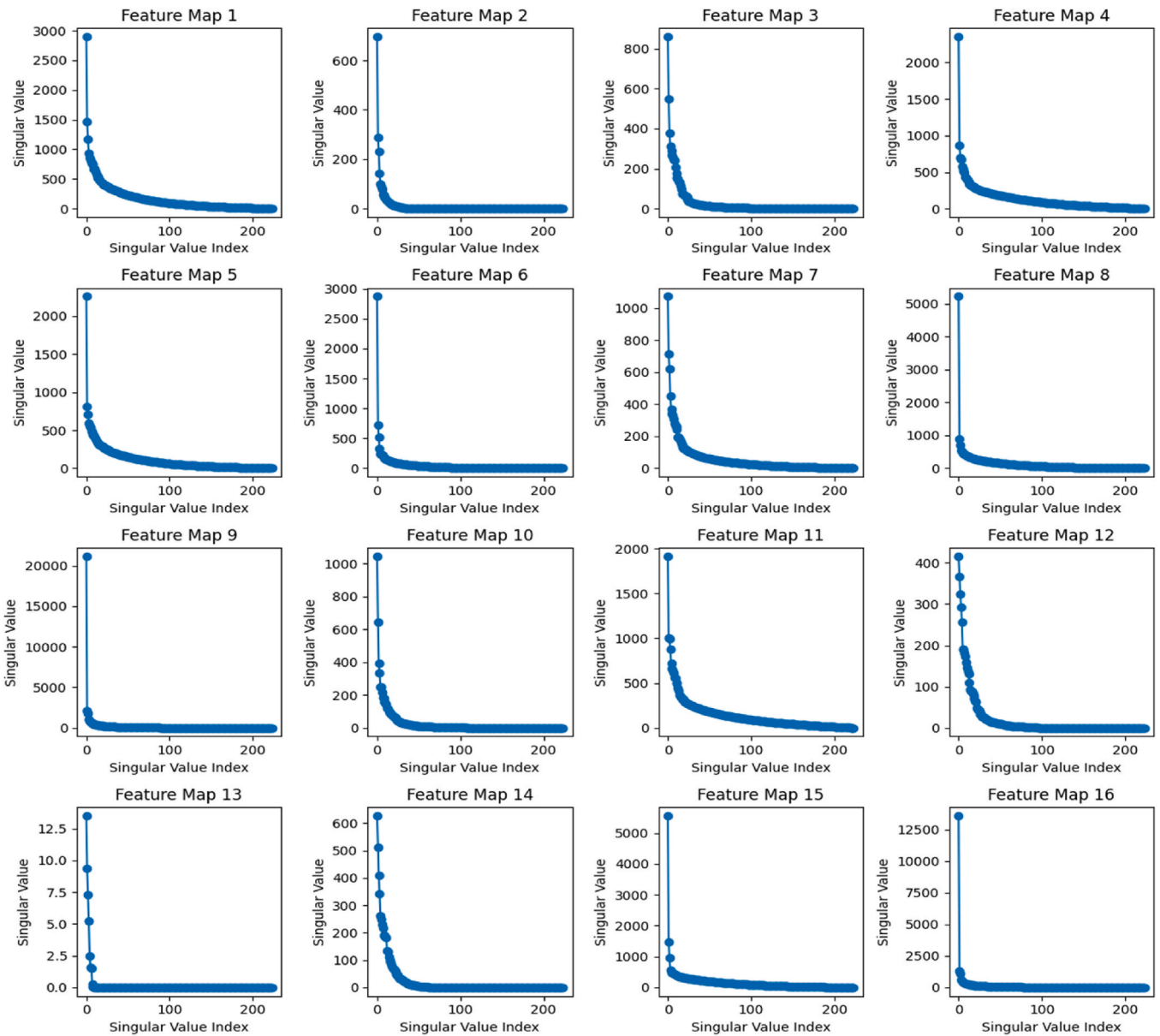


Fig. 5. The singular values are calculated using SVD for each feature map. The singular values are then visualized by plotting them against their index, highlighting the importance of each feature in a feature map. If a few singular values dominate (e.g., Feature maps 8, 9, and 16), it suggests that the corresponding features play a crucial role in representing patterns in the data.

$GM(f_j^l)$ and $GM(f_k^l)$ denote the gradient amplitude information of the feature maps f_j^l and f_k^l , respectively.

The result denoted as $FSIM_{j,k}^l$ is obtained by computing the low-level feature similarity between two distinct feature maps. For the l -th convolutional layer, comprising N_l output feature maps, the FSIM value of the j -th output feature map is determined using the following formula:

$$FSIM_j^l = \sum_{k=1}^{N_l} FSIM(f_j^l, f_k^l), s.t. j \neq k \quad (7)$$

Following Equation (7), we acquire the FSIM value for each feature map within the same layer. However, these values exhibit significant variation, posing a risk of errors in computation. To address this, we employ max–min normalization on the computed FSIM values to facilitate subsequent index quantification.

3.4. Singular value decomposition (SVD)

In a neural network model, all feature maps in a specific layer undergo identical transformations, incurring the same amount of FLOPs. Additionally, each weight matrix takes up a comparable amount of storage space. Despite these similarities, output feature maps' importance and impact on overall model performance vary significantly. Fig. 4 visually represents the output feature map obtained from the middle layer of the VGG-16 network, along with extracting their singular values, highlighting clear distinctions among these feature maps. In Fig. 5, the singular values are then visualized by plotting them against their index, highlighting the importance of each feature in a feature map. Those feature maps that contribute minimally do not enhance the network's performance and can be safely eliminated. To assess the contribution of an output feature map, we utilize SVD to decompose each feature map $A_i^l \in R^{m \times n}$ as represented in Equation (8):

$$A_i^l = U_i^l S_i^l V_i^{lT} \quad (8)$$

The matrix A_i^l represents the entity being decomposed, which typically corresponds to a feature map. The decomposition is achieved through singular value decomposition (SVD), where U_i^l denotes a unitary (orthogonal) matrix containing the left singular vectors of A_i^l . The diagonal matrix S_i^l comprises the singular values of A_i^l , representing the magnitudes of the principal components. Finally, V_i^{lT} is the transpose of a unitary (orthogonal) matrix containing the right singular vectors of A_i^l . This decomposition facilitates the analysis and manipulation of the feature representation within the model. Larger singular values signify a greater amount of conveyed information, considering these singular values facilitates the assessment of the output feature map's contribution. We aggregate all the singular values within the diagonal matrix to streamline and efficiently measure this contribution. This assessment is expressed through Equation (9):

$$G_i^l = \sum S_i^l \quad (9)$$

In equation (9), G_i^l is derived from the singular value decomposition of the i th feature map A_i^l at layer l . The singular values, contained in S_i^l , are summed to produce G_i^l , which reflects the aggregated significance of the feature map's components.

FSIM-SVD: The FSIM-SVD method is defined in this paper to guide the pruning of filters based on the principle of quantifying feature map importance. A feature map's importance is influenced by its low-level features and more significant information content. In this study, we begin by normalizing $FSIM_j^l$ and G_i^l to the range [0,1] using max–min normalization, ensuring they are on the same scale. Afterward, we merge FSIM and SVD to measure the significance of the output feature map in each layer. The formula for calculating the importance of the

output feature map is provided below.

$$C_j^l = \lambda \cdot FSIM_j^l + (1 - \lambda) \cdot G_i^l \quad (10)$$

This Equation represents the combined score (C_j^l) for the j -th feature map in the l -th convolutional layer, combining the feature similarity ($FSIM_j^l$) and the contribution (G_i^l) calculated through SVD. The weight λ balances the influence of FSIM and SVD.

3.5. Pruning and fine-tuning

The detailed pruning process involves a systematic approach to identifying and removing less important filters from a convolutional layer in a neural network, guided by the combined metric $C_j^l = \lambda \cdot FSIM_j^l + (1 - \lambda) \cdot G_i^l$. First, for each filter j in layer l , the combined metric C_j^l is calculated, considering the weights λ that balance the contributions of FSIM and the SVD measure (G_i^l). Next, a pruning threshold (τ) is determined based on a specified criterion, such as a percentile of C_j^l values. Filters with C_j^l values below this threshold are identified as candidates for pruning.

The actual pruning operation involves removing the identified filters from layer l and adjusting the dimensions of subsequent layers to maintain connectivity within the neural network architecture. The fine-tuning process is a crucial step following filter pruning, aimed at adapting the pruned neural network model to a specific task or dataset. This process involves training the pruned model on the target task with a smaller learning rate and refining its parameters to enhance performance. The fine-tuning process is considered a crucial step because pruning alters the network architecture by removing filters associated with less important feature maps, which can cause a temporary drop in model accuracy due to the loss of some learned parameters. Fine-tuning allows the remaining filters to adapt and re-optimize their weights based on the reduced architecture, thereby recovering and often improving performance. This step enables the network to re-learn feature representations that compensate for the pruned components, ensuring that essential discriminative information is preserved. As shown in our experiments, fine-tuning after pruning with FSIM-SVD consistently restores or even surpasses the baseline accuracy, confirming its necessity for stable and effective deployment of the compressed model. This model's specific setup of hyperparameters, including parameters such as epoch and learning rate, will be explained in detail in section 4.2.

4. Dataset and experimental setting

4.1. Dataset

Our experiment used two datasets: active trachoma inverted eyelid [22] and CIFAR10 [34]. Active trachoma is an open dataset of field-collected conjunctival images from clinical trial participants in Niger and Ethiopia used by Kim et al.'s [22]. Trained field workers followed a standardized protocol to capture the images. Three experts independently classified each image using the World Health Organization's simplified system for trachomatous inflammation. The dataset comprised 1656 labeled conjunctival images, with 39 % showing trachomatous changes. Among them, 22 % had TF, 7 % had TI, and 10 % had both TF and TI. The original authors obtained ethical approval. The dataset was divided into three sets to evaluate the model's performance: 80 % for training, 10 % for validation, and 10 % for testing. The CIFAR-10 dataset consists of color images with dimensions of 32×32 , classified into 10 categories that encompass animals and vehicles. It encompasses a total of 50,000 training images and 10,000 test images.

4.2. Experimental setting

The FSIM-SVD pruning method is employed to prune the model filters, followed by fine-tuning using stochastic gradient descent (SGD) with a reduced learning rate. We conducted experiments on three models: VGG16 [35] and ResNet56/110 [36]. We then compared the pruned models obtained from our method with the baseline models and other state-of-the-art pruning techniques. In the experimental setups for training models on the active trachoma dataset, the training batch size is set to 64, and the learning rate remains fixed at 0.001. Each experiment involves training for 90 epochs for the VGG16 model and 90 epochs for the ResNet56 and ResNet110 models using a momentum of 0.9 and weight decay of $1e-4$. For the CIFAR10 experiments, the training parameter settings described in Ref. [19] are adopted. This includes training for 300 epochs with a batch size 256, weight decay of $5e-4$, and momentum of 0.9. The initial learning rate starts at 0.01 and is divided by 10 at the 150th and 225th epochs. To compute feature similarity and determine the contribution of the output feature map for each, we randomly selected 50 images as samples from the active trachoma dataset and 150 sample images from the CIFAR10 dataset. The pre-training and implementation of the pruning algorithm for all fundamental network models in this study rely on the PyTorch framework. The experiments were executed utilizing the A100-SXM4-40 GB GPU on the Google Colab Pro platform, offering 40 GB of graphics memory, and the virtual machine possesses a RAM size of 81 GB.

4.3. Evaluation criteria

We utilize three different metrics to assess the effects of pruning thoroughly. First, managing the model's classification accuracy within a suitable range after pruning is crucial. However, a direct comparison of accuracy after pruning might be somewhat biased due to variations in the baseline accuracy of other methods. The outcomes detailed in section 5 demonstrate that our method can achieve better classification performance than the baseline. Secondly, we employ the reduction in FLOPs to measure the acceleration effects on the model, and the ratio of reduced parameters to the original parameters measures the compression efficiency.

5. Results and discussion

In our experiments, the FSIM-SVD technique was employed as a filter pruning framework to evaluate both the active trachoma and CIFAR-10 datasets. Specifically, output feature maps were analyzed using FSIM to detect redundancy based on perceptual similarity and SVD applied to quantify the contribution of each feature map by decomposing its energy distribution. Feature maps with lower FSIM-SVD significance scores were pruned, and the networks were fine-tuned to recover accuracy. For the active trachoma dataset, this enabled efficient classification of inverted eyelid images while substantially reducing FLOPs and parameters, thereby supporting deployment in resource-constrained medical service centers. For CIFAR-10, the same pruning process validated the generalization ability of FSIM-SVD on a large-scale benchmark, demonstrating its effectiveness in compressing models such as ResNet56 and ResNet110 without degrading, and in some cases even improving, classification performance.

5.1. Results and discussion on active trachoma

The purpose of combining FSIM and SVD is not limited to pruning VGG16 layers; rather, it provides a general framework for quantifying feature map importance across different architectures. We applied FSIM-SVD to VGG16 for active trachoma classification as a representative case, but we also validated it on ResNet56 and ResNet110 with the CIFAR-10 dataset. In all cases, FSIM-SVD consistently reduced FLOPs and parameters while maintaining or improving accuracy,

Table 1

Presents the pruning outcomes for active trachoma, enabling a comparison with different pruning ratios against the baseline model. The table includes top-1 accuracy, FLOPs pruning ratio (FPR), and parameter pruning ratio (PPR).

Model	Top-1 (%)	FLOPs (FPR)	Parameters (PPR)	Model Size (MB)
Original VGG16	86.56	313.86 M (0.0 %)	14.72 M (0.0 %)	56.30
FSIM-SVD-1	86.99	224.09 M (28.6 %)	9.8 M (33.4 %)	37.55
FSIM-SVD-2	86.35	140.92 M (55.1 %)	5.4 M (63.3 %)	20.71
FSIM-SVD-3	84.47	61.83 M (80.3 %)	2.84 M (80.7 %)	10.87
Original ResNet56	86.23	126.55 M (0.0 %)	0.85 M (0.0 %)	3.24
FSIM-SVD-1	86.41	87.83 M (30.6 %)	0.55 M (35.4 %)	2.07
FSIM-SVD-2	86.20	62.90 M (50.3 %)	0.41 M (51.6 %)	1.57
FSIM-SVD-3	83.52	23.79 M (81.2 %)	0.14 M (83.3 %)	0.53
Original ResNet110	86.51	254.72 M (0.0 %)	1.73 M (0.0 %)	6.61
FSIM-SVD-1	86.64	149.52 M (41.3 %)	1.06 M (38.5 %)	4.02
FSIM-SVD-2	86.02	88.13 M (65.4 %)	0.67 M (61.3 %)	2.56
FSIM-SVD-3	85.32	49.16 M (80.7 %)	0.35 M (79.5 %)	1.34

demonstrating that its role extends beyond a single network and can be effectively generalized to other CNN architectures. Table 1 presents the results of pruning experiments conducted on the active trachoma classification task using VGG16, ResNet56, and ResNet110 neural network architectures. The evaluation of pruned models encompasses accuracy, FLOPs, and parameters. The findings highlight the benefits of utilizing FSIM and SVD as quantitative indicators of feature map importance.

Table 2

Presents the pruning result of VGG16 on CIFAR10.

Model	Top-1 (%)	FLOPs (FPR)	Parameters (PPR)	Model Size (MB)
Original VGG16	93.96	313.86 M (0.0 %)	14.72 M (0.0 %)	56.30
L1 [37]	93.40	206.00 M (34.3 %)	5.40 M (63.6 %)	20.71
Zhao et al. [38]	93.18	190.00 M (39.1 %)	3.92 M (73.3 %)	15.05
GAL-0.05 [39]	92.03	189.49 M (39.6 %)	3.36 M (77.6 %)	12.83
SSS [40]	93.02	183.13 M (41.6 %)	3.93 M (73.8 %)	15.00
HRank [32]	93.43	145.61 M (53.5 %)	2.51 M (82.9 %)	9.57
FSIM-E [19]	93.65	131.61 M (58.1 %)	3.31 M (77.5 %)	13.11
FSIM-SVD	93.72	121.15 M (61.4 %)	3.59 M (75.6 %)	14.18
HRank [32]	92.34	108.61 M (65.3 %)	2.64 M (82.1 %)	10.11
AKECP [41]	92.68	63.18 M (79.8 %)	3.21 M (78.6 %)	12.30
FSIM-E [19]	92.84	59.90 M (80.9 %)	2.32 M (84.2 %)	8.85
FSIM-SVD	92.91	57.43 M (81.7 %)	2.16 M (85.3 %)	8.28
GAL-0.1 [39]	90.73	171.89 M (45.2 %)	2.67 M (82.2 %)	10.88
FSIM-E [19]	92.51	53.63 M (82.9 %)	1.59 M (89.2 %)	6.08
FSIM-SVD	92.62	49.90 M (84.1 %)	1.72 M (88.3 %)	6.50

Table 3
Presents the pruning result of ResNet on CIFAR10.

Model	Top-1 (%)	FLOPs (FPR)	Parameters (PPR)	Model Size (MB)
Original ResNet56	93.30	126.55 M (0.0 %)	0.85 M (0.0 %)	3.24
L1 [37]	93.06	90.90 M (27.6 %)	90.90 M (27.6 %)	3.44
HRank [32]	93.52	88.72 M (29.3 %)	0.71 M (16.8 %)	2.73
FilterSketch [42]	93.65	88.05 M (30.4 %)	0.68 M (20.6 %)	2.60
FSIM-E [19]	94.10	88.01 M (30.5 %)	0.56 M (34.1 %)	2.14
FSIM-SVD	94.23	85.92 M (32.1 %)	0.54 M (36.6 %)	2.07
GAL-0.6 [39]	92.98	78.30 M (37.6 %)	0.75 M (11.8 %)	2.87
FilterSketch [42]	93.19	73.36 M (41.5 %)	0.50 M (41.2 %)	1.91
AKECP [41]	93.21	69.28 M (44.8 %)	0.47 M (44.9 %)	1.80
Feng et al. [43]	93.05	63.79 M (49.6 %)	0.48 M (43.5 %)	1.82
HRank [32]	93.17	62.72 M (50.0 %)	0.49 M (42.4 %)	1.87
FSIM-E [19]	93.48	59.24 M (53.2 %)	0.39 M (54.1 %)	1.50
FSIM-SVD	93.59	56.06 M (55.7 %)	0.36 M (57.1 %)	1.38
CP [44]	90.80	62.00 M (50.6 %)	0.29 M (65.9 %)	1.14
GAL-0.8 [39]	90.36	49.99 M (60.2 %)	0.29 M (65.9 %)	1.14
AKECP [41]	91.86	38.02 M (69.7 %)	0.26 M (69.9 %)	1.06
Fan et al. [45]	91.13	33.99 M (72.9 %)	0.23 M (72.9 %)	0.91
HRank [32]	90.72	32.52 M (74.1 %)	0.27 M (68.1 %)	1.09
FilterSketch [42]	91.20	32.47 M (74.4 %)	0.24 M (71.8 %)	0.92
FSIM-E [19]	91.96	31.08 M (75.4 %)	0.21 M (75.3 %)	0.80
FSIM-SVD	92.01	29.99 M (76.3 %)	0.22 M (73.1 %)	0.84
Original ResNet110	93.58	254.72 M (0.0 %)	1.73 M (0.0 %)	6.61
L1 [37]	93.30	155.00 M (38.7 %)	1.16 M (32.6 %)	4.58
AKECP [41]	93.90	152.10 M (39.9 %)	1.03 M (40.0 %)	4.07
SFP [16]	93.38	150.00 M (40.8 %)	–	–
FSIM-E [19]	94.16	145.56 M (42.9 %)	1.07 M (38.2 %)	4.22
FSIM-SVD	94.31	143.15 M (43.8 %)	0.98 M (43.1 %)	3.87
GAL-0.5 [39]	92.55	130.20 M (48.5 %)	0.95 M (44.8 %)	3.74
HRank [32]	93.36	105.70 M (58.2 %)	0.70 M (59.2 %)	2.75
AKECP [41]	93.54	101.67 M (59.8 %)	0.69 M (59.9 %)	2.71
HRel [46]	93.03	95.72 M (62.4 %)	0.62 M (63.8 %)	2.42
FilterSketch [42]	93.44	92.84 M (63.3 %)	0.69 M (59.9 %)	2.71
FSIM-E [19]	93.68	73.59 M (71.1 %)	0.58 M (66.5 %)	2.28
FSIM-SVD	93.87	68.01 M (73.3 %)	0.53 M (69.6 %)	2.07
HRank [32]	92.65	79.30 M (68.6 %)	0.53 M (68.7 %)	2.07
FSIM-E [19]	92.72	51.74 M (79.7 %)	0.43 M (75.1 %)	1.65
FSIM-SVD	92.88	51.45 M (79.8 %)	0.41 M (76.3 %)	1.57

FSIM demonstrates its capability to identify similarities between feature maps, particularly in low-level features, leading to the discovery of redundant information. Simultaneously, SVD quantifies the essential information within feature maps from an informatics perspective, enabling effective compression of the model structure. For VGG16, applying FSIM-SVD pruning results in a 28.6 % decrease in FLOPs, a reduction of 33.4 % in parameters, and an accuracy of 86.99 % (with a +0.43 % improvement from the baseline). Moreover, it only experiences a 0.21 % accuracy drop when pruning 55.1 % of FLOPs and 63.3 % of parameters. There is also a 2.09 % accuracy decrease when pruning 80.3 % of FLOPs and 80.7 % of parameters. For ResNet56, employing FSIM-SVD pruning results in a 30.6 % decrease in FLOPs, a reduction of 35.4 % in parameters, and an accuracy of 86.41 % (with a +0.18 % improvement from the baseline). Notably, an accuracy loss of 0.03 % is observed when pruning 50.3 % of FLOPs and 51.6 % of parameters, and a 2.71 % accuracy decrease occurs when pruning 81.2 % of FLOPs and 83.3 % of parameters. In the case of ResNet110, the results demonstrate a 41.3 % reduction in FLOPs, 38.5 % fewer parameters, and an accuracy of 86.64 % (with a +0.13 % improvement from the baseline). Additionally, pruning 65 % of FLOPs and 61 % of parameters results in a 0.49 % accuracy loss, while pruning 80.7 % of FLOPs and 79.5 % of parameters leads to a 1.19 % accuracy loss.

5.2. Results and discussion on CIFAR10

The pruning results of VGG16 on CIFAR10 are shown in Table 2. With a 61.4 % reduction in FLOPs and a 75.6 % reduction in model parameters, our method can obtain a 0.24 % higher accuracy than the baseline model. For the pruning rate (61.47 % for FLOPs and 75.6 % for parameters), the proposed method, compared with L1 [37], Zhao et al. [38], GAL-0.05 [39], SSS [40], HRank [32] and FSIM-E [19], excels in all accuracy, FLOPs, and parameters. However, it has a lower parameter pruning rate than GAL-0.05 [39], HRank [32] and FSIM-E [19]. For a higher pruning rate (81.7 % for FLOPs and 85.3 % for parameters), our method still maintains a significant accuracy advantage (92.91 % vs. 92.68 % by AKECP [41]). For an even higher pruning rate (84.1 % for FLOPs and 88.3 % for parameters), our method still demonstrates a significant accuracy advantage (92.62 % vs. 92.51 % by FSIM-E [19] and 90.73 % by GAL-0.1 [39]).

Table 3 displays the results of various pruning strategies on the CIFAR10 dataset for ResNet56 and ResNet110 models. For the ResNet56 model, our approach demonstrates an accuracy that surpasses the baseline model by 0.93 %. This enhancement is accompanied by a 32.1 % reduction in FLOPs and a 36.6 % decrease in the number of parameters. At this pruning rate, our model surpasses L1 [37], HRank [32], FilterSketch [42], and FSIM-E [19] in terms of accuracy, FLOPs, and parameter pruning rate. As the compression rate increases, with a 55.7 % reduction in FLOPs and a 57.1 % decrease in parameters, our model maintains a 0.29 % higher accuracy than the baseline model. It also outperforms GAL-0.6 [39], FilterSketch [42], AKECP [41], Feng et al. [43], HRank [32] and FSIM-E [19] across all evaluation aspects. Further increasing the compression rate, with a 76.3 % drop in FLOPs and a 77.1 % decrease in parameters, our model exhibits a 1.58 % accuracy loss compared to the base model. Nevertheless, it demonstrates superior accuracy improvement over previous works such as CP [44], GAL-0.8 [39], AKECP [41], Fan et al. [45], HRank [32], FilterSketch [42] and FSIM-E [19].

For the ResNet110 model, our method continues to perform well. It achieves a higher accuracy than the baseline model by 0.73 %, with a 43.8 % decrease in FLOPs and a 43.1 % reduction in parameters. Our model outperforms L1 [37], AKECP [41], SFP [16], and FSIM-E [19] in accuracy, FLOPs, and parameter pruning rate. With a 73.3 % reduction in FLOPs and a 69.6 % decrease in parameters, our model maintains a 0.29 % higher accuracy than the baseline. It also outperforms GAL-0.5 [39], HRank [32], AKECP [41], HRel [46], FilterSketch [42] and FSIM-E [19] across all evaluation aspects. Further increasing the

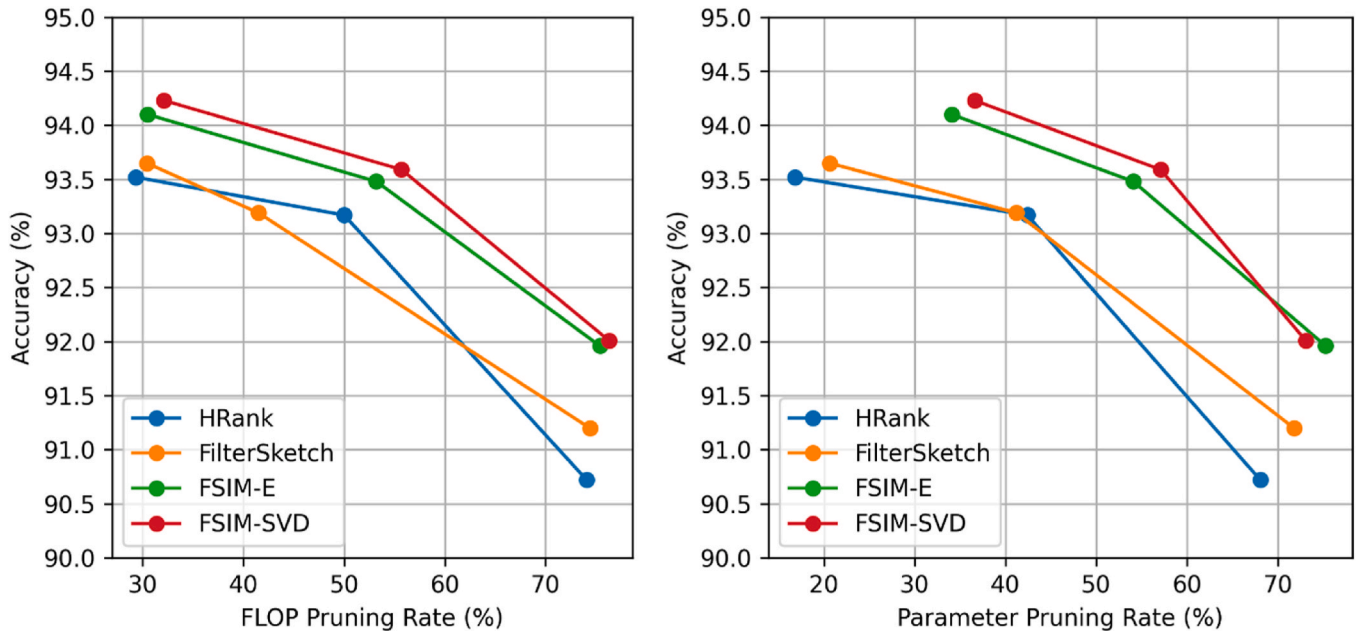


Fig. 6. Depicts the impact of FLOP and parameter pruning rates on accuracy for both the current and proposed pruning models applied to ResNet56. Note: The Y-axis starts at 90 % (not zero) to better illustrate small performance differences between methods.

pruning rate, with a 79.8 % FLOPs and a 76.3 % parameters reduction, our model experiences a 0.7 % accuracy loss compared to the baseline but still outperforms HRank [32] and FSIM-E [19] in accuracy, FLOPs, and parameter pruning rate. An increase in the compression rate directly leads to a reduction in FLOPs and a decrease in parameters because pruning removes redundant or less informative filters from the network. Each pruned filter eliminates its associated weights (parameters) and also removes the corresponding computations required for generating its output feature maps, thereby reducing the number of floating-point operations (FLOPs). As the compression rate increases, a larger proportion of filters are removed, which proportionally lowers both the parameter count and the computational cost. This relationship is evident in our experimental results, where higher pruning ratios achieved by FSIM-SVD consistently correspond to significant reductions in FLOPs and parameters while maintaining competitive accuracy.

The FSIM-SVD pruning method substantially reduces both FLOPs and parameter counts across all evaluated models and datasets. For example, pruning VGG16 on the active trachoma dataset with FSIM-SVD-3 decreases FLOPs by 80.3 % and parameters by 80.7 %, reducing the model size from 56.30 MB to 10.87 MB. Similarly, ResNet56 on CIFAR10 is reduced from 3.24 MB to 1.38 MB with a 55.7 % reduction in FLOPs. These reductions lead to fewer operations per forward/backward pass, decreasing training and inference time. The lower parameter counts also reduce memory consumption, enabling deployment on GPUs or embedded devices with limited resources. Therefore, FSIM-SVD not only improves theoretical efficiency, as reflected in FLOPs, but also provides tangible computational cost savings. The updated “Model Size (MB)” column in Tables 1–3 quantifies these memory savings alongside FLOPs and accuracy.

We extensively assessed the effectiveness of our proposed FSIM-SVD method by comparing its Top-1 accuracy with that of several state-of-the-art techniques across different pruning rates for FLOPs and parameters. The results of these experiments are illustrated in Fig. 6. Notably when considering ResNet56 on CIFAR10, FSIM-SVD consistently maintains a high Top-1 accuracy even with varying pruning rates for FLOPs and parameters. This consistent performance is compelling evidence for utilizing FSIM and SVD to evaluate the importance of feature maps quantitatively.

Table 4

Ablation study to provide insights into the individual contributions of FSIM and SVD indicators in guiding filter pruning.

Model	Top-1 (%)	FLOPs (FPR)	Parameters (PPR)
Original ResNet56	93.30	126.55 M (0.0 %)	0.85 M (0.0 %)
FSIM	93.46	56.06 M (55.7 %)	0.36 M (57.1 %)
SVD	93.41	56.06 M (55.7 %)	0.36 M (57.1 %)
FSIM-SVD	93.59	56.06 M (55.7 %)	0.36 M (57.1 %)

5.3. Ablation study

5.3.1. Feature map indicators

In this section, we conduct a comprehensive ablation study to show the distinct and combined contributions of FSIM and SVD indicators in guiding filter pruning. Initially, we examine the model’s performance when feature maps are pruned based solely on FSIM scores. Furthermore, we explore the impact of using the SVD indicator alone for guiding filter pruning. FSIM alone does not account for how much unique or discriminative information each feature map contributes. In contrast, SVD analyzes the intrinsic energy distribution of a feature map, where dominant singular values reflect the richness of unique content. This makes SVD well suited for quantifying contribution, but it lacks sensitivity to redundancy across maps. By combining FSIM and SVD, the FSIM-SVD framework jointly considers redundancy (FSIM) and contribution (SVD), ensuring that pruning decisions retain both diverse and informative feature maps. The experiments were conducted on the CIFAR10 dataset, utilizing the experimental parameter settings outlined in Table 3. This complementarity explains why the combined FSIM-SVD consistently outperforms either method alone, as shown in our ablation study (Table 4), where FSIM-SVD achieved higher accuracy (93.59 %) than FSIM-only (93.46 %) or SVD-only (93.41 %) under identical pruning rates. Thus, the integration of FSIM and SVD is not redundant but rather necessary to balance perceptual distinctiveness with information contribution.

5.3.2. Weight parameter (λ)

In this ablation study, we systematically explore the impact of the weight parameter (λ) on filter pruning, governing the balance between

Table 5
Impact of weight parameter (λ) on ResNet56 filter pruning.

Weight parameter (λ)	Top-1 accuracy (%)
$\lambda = 0.2$	93.42
$\lambda = 0.3$	93.43
$\lambda = 0.4$	93.45
$\lambda = 0.5$	93.59
$\lambda = 0.6$	93.46
$\lambda = 0.7$	93.49
$\lambda = 0.8$	93.51

FSIM and SVD metrics in the combined filter importance metric C_f^j . Understanding how different λ values influence pruning outcomes is crucial for optimizing the trade-off between preserving feature similarity and leveraging singular value information. We varied λ while maintaining ResNet56 experimental settings (93.59 % accuracy, Table 3). Results in Table 5 show that ResNet56 accuracy varies with λ . At $\lambda = 0.5$, post-pruning accuracy is 93.59 %, while other λ values yield lower accuracies. After consideration, we set the FSIM-SVD indicator's λ to 0.5, offering a balanced approach; $\lambda = 0$ emphasizes SVD, $\lambda = 1$ prioritizes FSIM, and intermediate values assess a balanced contribution from both metrics.

6. Conclusion

- ✓ The research provides a viable solution to the challenges of deploying complex CNNs for active trachoma detection.
- ✓ The proposed FSIM-SVD framework effectively quantifies the significance of feature maps, enabling substantial model compression without sacrificing accuracy.
- ✓ The methodology addresses computational constraints in medical service centers and is tailored to the needs of active trachoma detection.
- ✓ The study contributes to the advancement of CNN compression techniques and their application to medical image classification, especially in resource-limited settings.
- ✓ The findings demonstrate the potential for developing compact yet accurate models that can be integrated into clinical workflows to improve trachoma diagnosis.

While the proposed method demonstrates strong performance on our curated dataset, deploying it in real-world settings may present several challenges. These include limitations in computational resources, variability and quality of field-acquired data, integration with existing diagnostic workflows, and scalability for larger datasets. Addressing these challenges will be essential for practical applications. Future work will focus on field validation to assess the method's effectiveness in operational environments and to optimize the approach for resource-constrained settings.

CRedit authorship contribution statement

Mulugeta Shitie Zewudie: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Formal analysis, Data curation, Conceptualization. **Shengwu Xiong:** Visualization, Validation, Project administration, Methodology, Funding acquisition, Conceptualization. **Xiaohan Yu:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Conceptualization. **Xiaoyu Wu:** Software, Resources, Data curation.

Ethics approval

The original authors obtained ethical approval for the dataset.

Ethics Statement

No ethics concern is related to this work.

Declaration of competing interest

We wish to submit an original research article entitled “Feature Map Quantification: An Efficient Approach for Active Trachoma Classification” for consideration by the Computers in Biology and Medicine. We declared that no known competing financial interests or personal relationships could have appeared to influence the work reported in this paper.

Acknowledgment

This work was in part supported by the Project of Sanya Science and Education Innovation Park of Wuhan University of Technology (Grant No. 2021KF0031)

Data availability

Original images for this study are available on figshare:<https://doi.org/10.6084/m9.figshare.7551053.v1>. We can provide processed data via email contact.

References

- [1] K. Xu, L. Rui, Y. Li, L. Gu, Feature normalized knowledge distillation for image classification, in: European Conference on Computer Vision, Springer, 2020, pp. 664–680.
- [2] S. Fu, Z. Li, Z. Liu, X. Yang, Interactive knowledge distillation for image classification, *Neurocomputing* 449 (2021) 411–421.
- [3] J. Kang, J. Gwak, Ensemble learning of lightweight deep learning models using knowledge distillation for image classification, *Mathematics* 8 (10) (2020) 1652.
- [4] J. Gou, B. Yu, S.J. Maybank, D. Tao, Knowledge distillation: a survey, *Int. J. Comput. Vis.* 129 (2021) 1789–1819.
- [5] Y. Wang, Y. Wang, J. Cai, T.K. Lee, C. Miao, Z.J. Wang, Ssd-kd: a self-supervised diverse knowledge distillation method for lightweight skin lesion classification using dermoscopic images, *Med. Image Anal.* 84 (2023) 102693.
- [6] M.S. Khan, K.N. Alam, A.R. Dhruva, H. Zunair, N. Mohammed, Knowledge distillation approach towards melanoma detection, *Comput. Biol. Med.* 146 (2022) 105581.
- [7] Y. Liu, W. Zhang, J. Wang, Adaptive multi-teacher multi-level knowledge distillation, *Neurocomputing* 415 (2020) 106–113.
- [8] T. Choudhary, V. Mishra, A. Goswami, J. Sarangapani, A transfer learning with structured filter pruning approach for improved breast cancer classification on point-of-care devices, *Comput. Biol. Med.* 134 (2021) 104432.
- [9] T. Choudhary, S. Gujar, A. Goswami, V. Mishra, T. Badal, Deep learning-based important weights-only transfer learning approach for COVID-19 CT-scan classification, *Appl. Intell.* 53 (6) (2023) 7201–7215.
- [10] J. Li, H. Shao, S. Zhai, Y. Jiang, X. Deng, A graphical approach for filter pruning by exploring the similarity relation between feature maps, *Pattern Recognit. Lett.* 166 (2023) 69–75.
- [11] N. Dong, Y. Zhang, M. Ding, S. Xu, Y. Bai, One-stage object detection knowledge distillation via adversarial learning, *Appl. Intell.* (2022) 1–17.
- [12] K. Wang, Z. Liu, Y. Lin, J. Lin, S. Han, Haq: hardware-aware automated quantization with mixed precision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8612–8620.
- [13] H. Peng, J. Wu, Z. Zhang, S. Chen, H.-T. Zhang, Deep network quantization via error compensation, *IEEE Transact. Neural Networks Learn. Syst.* 33 (9) (2021) 4960–4970.
- [14] C. Xu, W. Gao, T. Li, N. Bai, G. Li, Y. Zhang, Teacher-student collaborative knowledge distillation for image classification, *Appl. Intell.* 53 (2) (2023) 1997–2009.
- [15] M. Zarski, B. Wójcik, K. Książek, J.A. Mischczak, Finicky transfer learning—A method of pruning convolutional neural networks for cracks classification on edge devices, *Comput. Aided Civ. Infrastruct. Eng.* 37 (4) (2022) 500–515.
- [16] Y. He, G. Kang, X. Dong, Y. Fu, Y. Yang, Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks, 2018 *arXiv preprint arXiv:1808.06866*.
- [17] Y. He, P. Liu, Z. Wang, Z. Hu, Y. Yang, Filter pruning via geometric median for deep convolutional neural networks acceleration, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4340–4349.
- [18] C. Zhang, C. Li, B. Guo, N. Liao, Neural network compression via low frequency preference, *Remote Sens.* 15 (12) (2023) 3144.
- [19] Y. Liu, K. Fan, D. Wu, W. Zhou, Filter pruning by quantifying feature similarity and entropy of feature maps, *Neurocomputing* 544 (2023) 126297.
- [20] H. Yang, Y. Liang, W. Liu, F. Meng, Filter pruning via attention consistency on feature maps, *Appl. Sci.* 13 (3) (2023) 1964.

- [21] J.-H. Luo, J. Wu, Autopruner: an end-to-end trainable filter pruning method for efficient deep model inference, *Pattern Recogn.* 107 (2020) 107461.
- [22] M.C. Kim, et al., Sensitivity and specificity of computer vision classification of eyelid photographs for programmatic trachoma assessment, *PLoS One* 14 (2) (2019) e0210463.
- [23] M.M. Alambo, E.A. Lake, S. Bitew Workie, A.Y. Wassie, Prevalence of active trachoma and associated factors in Areka town, south Ethiopia, 2018, *Interdisciplinary Perspectives on Infectious Diseases* 2020 (2020).
- [24] M.J. Burton, D.C. Mabey, The global burden of trachoma: a review, *PLoS Neglected Trop. Dis.* 3 (10) (2009) e460.
- [25] B. Yenegeta, Y. Assabie, TrachomaNet: detection and grading of trachoma using texture feature based deep convolutional neural network, *Multimed. Tool. Appl.* (2022) 1–26.
- [26] D. Socia, C.J. Brady, S.K. West, R.C. Cockrell, Detection of trachoma using machine learning approaches, *PLoS Neglected Trop. Dis.* 16 (12) (2022) e0010943.
- [27] H. Hu, R. Peng, Y.-W. Tai, C.-K. Tang, Network Trimming: a data-driven Neuron Pruning Approach Towards Efficient Deep Architectures, 2016 *arXiv preprint arXiv:1607.03250*.
- [28] Z. Chen, T.-B. Xu, C. Du, C.-L. Liu, H. He, Dynamical channel pruning by conditional accuracy change for deep neural networks, *IEEE Transact. Neural Networks Learn. Syst.* 32 (2) (2020) 799–813.
- [29] S. Pan, L. Zhang, J. Zhang, X. Li, L. Hou, X. Tu, Layer-adaptive structured pruning guided by latency, *arXiv preprint (2023) arXiv:2305.14403*.
- [30] S. Wu, C. Xiao, J. Yang, W. An, Dynamic Channel Pruning with Adaptive Weight Learning, *Authorea Preprints*, 2022.
- [31] M. Mondal, B. Das, S.D. Roy, P. Singh, B. Lall, S.D. Joshi, Adaptive CNN filter pruning using global importance metric, *Comput. Vis. Image Understand.* 222 (2022) 103511.
- [32] M. Lin, et al., Hrank: filter pruning using high-rank feature map, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1529–1538.
- [33] Y. Chen, X. Wen, Y. Zhang, Q. He, FPC: filter pruning via the contribution of output feature map for deep convolutional neural networks acceleration, *Knowl. Base Syst.* 238 (2022) 107876.
- [34] A. Krizhevsky, G. Hinton, Learning Multiple Layers of Features from Tiny Images, 2009.
- [35] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for large-scale Image Recognition, 2014 *arXiv preprint arXiv:1409.1556*.
- [36] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [37] H. Li, A. Kadav, I. Durdanovic, H. Samet, H.P. Graf, Pruning filters for efficient convnets, *arXiv preprint arXiv:1608.08710* (2016).
- [38] C. Zhao, B. Ni, J. Zhang, Q. Zhao, W. Zhang, Q. Tian, Variational convolutional neural network pruning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2780–2789.
- [39] S. Lin, et al., Towards optimal structured cnn pruning via generative adversarial learning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2790–2799.
- [40] Z. Huang, N. Wang, Data-driven sparse structure selection for deep neural networks, in: *Proceedings of the European Conference on Computer Vision, ECCV*, 2018, pp. 304–320.
- [41] H. Zhang, L. Liu, H. Zhou, W. Hou, H. Sun, N. Zheng, Akecp: adaptive knowledge extraction from feature maps for fast and efficient channel pruning, in: *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 648–657.
- [42] M. Lin, et al., Filter sketch for network pruning, *IEEE Transact. Neural Networks Learn. Syst.* 33 (12) (2021) 7091–7100.
- [43] K.-Y. Feng, X. Fei, M. Gong, A. Qin, H. Li, Y. Wu, An automatically layer-wise searching strategy for channel pruning based on task-driven sparsity optimization, *IEEE Trans. Circ. Syst. Video Technol.* 32 (9) (2022) 5790–5802.
- [44] Y. He, X. Zhang, J. Sun, Channel pruning for accelerating very deep neural networks, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1389–1397.
- [45] F. Fan, Y. Su, P. Jing, W. Lu, A dual rank-constrained filter pruning approach for convolutional neural networks, *IEEE Signal Process. Lett.* 28 (2021) 1734–1738.
- [46] C. Sarvani, M. Ghorai, S.R. Dubey, S.S. Basha, Hrel: filter pruning based on high relevance between activation maps and class labels, *Neural Netw.* 147 (2022) 186–197.