Simposio CEA de Robótica, Bioingeniería, Visión Artificial y Automática Marina 2025

Efficient CNNs in computer vision from an explanatory perspective

López-González, C. I.[a,*], Besada-Portas, E.[b], Gómez-Silva, M. J.[b], Risco-Martín, J. L.[b], Pajares, G.[c]

[a] *Dpt. of Software Engineering and Artificial Intelligence, Complutense University of Madrid, Madrid, 28040, Spain.*
[b] *Dpt. of Computer Architecture and Automation, Complutense University of Madrid, Madrid, 28040, Spain.*
[c] *Institute for Knowledge Technology, Complutense University of Madrid, Madrid, 28040, Spain.*

Resumen

Las redes neuronales convolucionales son una herramienta indispensable en visión por computador. Sin embargo, su gran coste computacional limita su despliegue en dispositivos con recursos limitados, utilizados en el procesamiento de imágenes. A pesar de que existen métodos de compresión, la mayoría requieren procesos intensivos. Además, suelen obviar la naturaleza opaca de las redes, a riesgo de eliminar características relevantes y degradar el rendimiento. Se introducen dos estrategias de compresión de bajo coste computacional y se realiza un análisis explicativo de la relación entre las características preservadas, los criterios de poda y la eficiencia del modelo, aportando información novedosa. Los experimentos en segmentación de imágenes, donde la poda ha pasado desapercibida, demuestran la eficacia de las propuestas, reduciendo más del 90% los parámetros sin apenas perder precisión. Los experimentos en tareas de clasificación prueban su generalidad y su eficiencia en comparación con la literatura. Finalmente, el análisis explicativo guía las mejoras que incrementan la efectividad de los métodos.

Palabras clave: Visión por Computador, Inteligencia Artificial, Redes Neuronales, Reducción de Modelos, Inteligencia Artificial Explicativa.

Sesión: Visión por Computador.

Efficient CNNs in computer vision from an explanatory perspective

Abstract

Convolutional Neural Networks (CNNs) have become essential tools in computer vision. However, their high computational cost limits their deployment on resource-constrained devices commonly used in image processing tasks. Although various compression methods have been proposed, they require intensive computation or iterative procedures. Moreover, the opaque nature of CNNs is often overlooked, despite the risk of pruning the wrong features and thereby degrading performance. We introduce two low-effort compression strategies and present an explanatory analysis of the relationship between retained features, pruning criteria, and compressed model efficiency, providing novel insights. Experiments conducted on semantic segmentation, an area where pruning has received little attention, demonstrate the effectiveness of our methods, achieving over 90% parameter reduction with negligible loss in accuracy. Meanwhile, classification experiments prove their generality and efficiency compared to state-of-the-art approaches. Finally, the explanatory analysis guides improvements to enhance the effectiveness of the methods.

Keywords: Computer Vision, Artificial Intelligence, Neural Networks, Model Reduction, Explainable Artificial Intelligence.

Session: Computer Vision.

1. Introduction

Deep Learning (DL) models, in particular Convolutional Neural Networks (CNNs), have become some of the most powerful tools in computer vision. This field covers a wide range of domains and tasks, including classification and semantic segmentation in image processing. In many cases, these tasks must be performed in real time or on devices with limited computa-

*Corresponding author: claraisl@ucm.es

López-González, C. I., Besada-Portas, E., Gómez-Silva, M. J., Risco-Martín, J. L., Pajares, G., 2025. Efficient CNN in computer vision from an explanatory perspective. Simposio CEA de Robótica, Bioingeniería, Visión Artificial y Automática Marina 2025.

tional and memory resources, such as mobile devices or those onboarded in autonomous vehicles. In this context, two properties of CNNs are particularly noteworthy: (1) their high computational demand and (2) their lack of transparency.

On the one hand, the general tendency to scale up CNNs in order to improve their performance makes their use on resource-constrained devices impractical. Therefore, to carry out the aforementioned tasks, it is necessary to compress these models while preserving their effectiveness. This approach falls within the framework of model compression.

On the other hand, the black-box nature of CNNs poses a significant challenge to the interpretation of their predictions. To overcome this issue, explanatory methods have been developed within the field of eXplainable Artificial Intelligence (Minh et al., 2022, XAI). These methods provide insights into the intrinsic behavior of neural networks, particularly by identifying which features are relevant to the decisions and where they are encoded. This knowledge can be determinant for model compression. Since the relevance of the encoded features depends on the dataset, the absence of an explanatory analysis of the preserved and discarded features may cause the compression method to eliminate crucial information. Consequently, the performance of the compressed network can be significantly degraded.

The relationship between these two shortcomings in image processing motivates our work. We aim to compress CNNs, by reducing the number of parameters and FLOating Points operations (FLOPs[1]), and to analyze their performance in terms of the relevance of the preserved features.

Regarding model compression, recent research focuses on filter pruning to avoid sparsity and the reliance on specialized libraries (Han et al., 2016). These methods typically define an importance criterion to assess the relevance of each convolutional filter and remove those with the lowest values. The criteria may be based on information from filters themselves, such as magnitude (Li et al., 2017) or redundancy (He et al., 2019), or from activation maps (Shao et al., 2021). However, many filter pruning approaches require additional hyperparameters, intensive studies, or iterative processes (Liu et al., 2021; He et al., 2022). To address these drawbacks, we present two low-effort pruning strategies, already introduced in López-González et al. (2024a). The latter approach is applicable to complex convolutional neural networks and considers previously learned weights, in contrast to Garg et al. (2020).

Despite the opacity of CNNs, the literature offers little explanatory analysis of the pruning process. Nevertheless, data is expected to influence the selection of pruned filters, especially when the importance criterion relies on dataset-specific information. To overcome this limitation, we employ XAI to interpret the results. As a consequence, we extend the dependency analysis from López-González et al. (2024a) and provide novel insights into explanations of pruning performance.

Although semantic segmentation is one of the main processes in computer vision, model compression has not been evaluated as extensively for this task as it has for classification. To address this shortcoming, we focus on segmentation, particularly within the field of precision agriculture. This is a key area in computer vision where deploying image processing algorithms on devices with limited resources is often required. In addition, to demonstrate the generality of the compression methods, classification is also evaluated using widely known network and dataset. In this case, the explanatory analysis presented in our work is essential for understanding the performance of the pruned networks and for validating the modifications introduced to achieve state-of-the-art results.

2. Filter pruning methods

This section describes the proposed filter pruning methodology, detailed in López-González et al. (2024a) and illustrated in Figure 1. We focus on compressing pretrained networks, since in many cases the goal is to reduce the size of well-tested models without compromising their efficiency.

In this context, filter pruning strategies consist of three phases: (1) consider a pretrained CNN, (2) prune each convolutional layer, and (3) fine-tune the resulting model to recover generalization. The pruning phase of the proposed methods (phase 2 in Figure 1) is further divided in two steps: (A) computing the relevance of each channel[2] using the importance criterion, and (B) determining how many channels to delete. Pruning is carried out by removing the specified number of channels, starting with those with the lowest importance values.

In the following, we describe the main aspects of some steps of the process to end with an overview of all of it.

2.1. Channel importance calculation (box A, Figure 1)

Consider the lth convolutional layer C_l, whose weight tensor $W^l \in \mathbb{R}^{k_l \times k_l \times n_{l-1} \times n_l}$ has a kernel dimension of $k_l \times k_l$, where n_{l-1} is the number of input channels and n_l is the number of output channels or filters. Hence, the output activation map $a^l \in \mathbb{R}^{h_l \times w_l \times n_l}$ has h_l rows, w_l columns, and n_l channels.

Regarding the relevance of the channels, two importance criteria are defined:

- *Next convolution influence criterion*: the more influential a channel is on the activation map of the next convolution, the greater its importance. As demonstrated in López-González et al. (2024a), this influence can be measured by the l_1-norm of the weight tensor of the next convolution. In particular, the importance score of the jth channel is given by

$$\ell_j^l := \|W^{l+1}(\cdot, \cdot, j, \cdot)\|_1, \quad 1 \leq j \leq n_l, \tag{1}$$

  where $W^{l+1}(\cdot, \cdot, j, \cdot) \in \mathbb{R}^{k_{l+1} \times k_{l+1} \times n_{l+1}}$. As a result, we preserve those channels whose connections with layer C_{l+1} are more active[3].

- *Next convolution influence-Variance criterion*: the more homogeneous an activation map is, measured via its variance, the lower its relevance. Combining this with the

[1] We only consider FLOPs of convolutional operations.

[2] Note that there is a one-to-one correspondence between the filters of a convolutional layer and its output channels.

[3] Similarly, in the case of transposed convolutions ℓ_j^l is defined in terms of $W^{l+1}(\cdot, \cdot, \cdot, j) \in \mathbb{R}^{k_{l+1} \times k_{l+1} \times n_l}$.

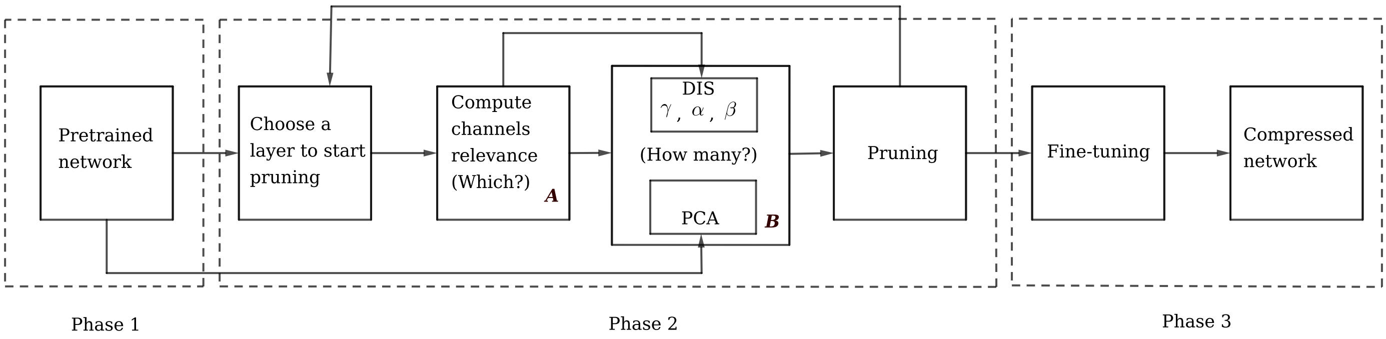

Figure 1: Flowchart of the proposed pruning methods.

previous criterion, we assign low importance to those channels that have little influence on the next convolution and, additionally, small variance. That is, the importance score is given by

$$\ell s^l_j := \ell^l_j \cdot s^l_j, 1 \leq j \leq n_k, \tag{2}$$

where $s^l_j = (1/N) \sum_{n=1}^{N} \sigma^n_{l,j}$ is the mean standard deviation of all the images in the dataset, and $(\sigma^n_{l,j})^2$ the variance of the jth activation map of the nth image.

Note that ℓ differs from the standard norm criterion (Li et al., 2017) in that, rather than relying on the same layer l, ℓ depends on layer $l+1$. Consequently, our strategy ensures that any adverse effects on the pruning of preceding layers do not impact the pruning of C_l. With respect to (2), scale unification is unnecessary, as shown in López-González et al. (2024a), since we are interested in the (ascending) order of the importance scores.

2.2. Channel number calculation (box B, Figure 1)

We use two methods to determine the number of channels to prune within C_l. We guarantee the low-effort of the proposals by omitting iterative algorithms and intensive optimizations.

- *Principal Component Analysis (PCA)*: by considering T mini-batches of size B, the output of the lth convolution can be seen as a matrix with $h_l \cdot w_l \cdot B \cdot T$ rows, where $h_l \times w_l$ is the dimension of the activation maps, and n_l columns, corresponding to the number of channels. In line with Garg et al. (2020), it suffices to take $T = \lceil 100 n_l/(h_l \cdot w_l \cdot B) \rceil$. Then, PCA is conducted regarding each row as a n_l-dimensional vector. This provides the optimal dimension of the layer n^{opt} as the number of eigenvalues that explain the 99.9% of the total variance. Thus, $n_l - n^{opt}$ is the number of channels to prune.

- *Distribution of channels Importance Score (DIS)*: since the channels of a layer C_l can be ordered by relevance according to their importance scores $\{u^l_j\}_{j=1}^{n_l}$, this method exploits the distribution of these scores to decide how many to prune. Without loss of generality, we assume that $\{u^l_j\}_{j=1}^{n_l}$ are sorted in ascending order and normalized by the maximum and minimum values.

  Let $U^l := \sum_{j=1}^{n_l} u^l_j$ be the total layer score and $U^l_k := \sum_{j=1}^{k} u^l_j$ the cumulative sum up to the kth score. We obtain the number of channels whose cumulative sum of importance scores does not exceed the ν percentage of the total layer score as

  $$k^l_\nu := \max\{k \in \mathbb{N} \ : \ 1 \leq k \leq n_l, \ U^l_k \leq \nu U^l\}, \ \nu \in [0,1]. \tag{3}$$

  Then, we select two thresholds: the maximum $\nu = \alpha$ and the minimum, $\nu = \beta$, which represent the bounds within which we are willing to prune any layer. That is, we aim to prune no more than k^l_α channels and no fewer than k^l_β. In addition, consider the number of sorted channels with scores lower than $\gamma \in [0,1]$, which is given by

  $$d^l_\gamma := \max\{d \in \mathbb{N} \ : \ 1 \leq d \leq n_l, \ u^l_d \leq \gamma\}. \tag{4}$$

  Finally, the number of channels to be removed is given by the middle value between k^l_α, k^l_β, and d^l_γ.

It is worth mentioning that DIS addresses the issue where PCA leads to poor pruning in low-resolution layers. Moreover, even though the same thresholds (α, β, γ) are used across all layers, this strategy provides different pruning rates for each layer. Regarding the parameters (α, β, γ), reasonable values can be selected to ensure that neither too many nor too few channels are deleted. The initial choice of (0.75, 0.1, 0.25) is based on the compression results from previous studies (Ayinde et al., 2019; Chen et al., 2022), and is consistently adjusted as needed. However, certain networks may require an exhaustive search. For more information, refer to López-González et al. (2024a).

2.3. Overview of the methodology (Figure 1)

Given a pretrained CNN, consider the first convolutional layer. To prune it, two strategies are proposed to determine which (A) and how many (B) channels to delete. The first approach combines PCA with the Next convolution influence criterion, called PCA-N hereafter. The second uses DIS and the Next convolution influence-Variance criterion, named DIS-NV hereafter. Once the selected channels are removed, we move to the next convolutional layer and repeat the procedure. When all the layers except the last one have been compressed, a fine-tuning is performed to recover generalization, using the original training hyperparameters expect for the epochs, which in the following section is indicated next to the method name (e.g., DIS-NV (10)). This retraining approach is chosen because iterative fine-tuning is time-consuming due to the complexity of the networks and datasets.

3. Experiments and results

The experiments conducted in this section aim to validate and demonstrate the effectiveness of the filter pruning proposals. In addition, comparisons with state-of-the-art methods and in-depth discussions are provided in Section 4.

3.1. Experimental setting

As explained in Section 1, we focus on semantic segmentation given the insufficient analysis in the compression literature. Specifically, two publicly available datasets in the field of precision agriculture are considered.

The Sugar Beet Dataset (Di Cicco et al., 2017, SBD) contains 1252 realistic synthetic images of sugar beet instances and random weeds. There are three classes (soil, crop, and weed) and pixel-level annotations. The Crop Row Benchmark Dataset (Vidovic et al., 2016, CRBD) has 281 color crop row images with ground truth data. Pixel-level annotations for soil and greenness classes are created by a semi-automatic process, recognizing greenness by a dominant G (green) component. We use a preprocessed variant based on vegetation indices. Defined as a combination of the RGB bands, these indices provide five additional pseudo-bands that we append to the original images. In both cases, U-Net (Ronneberger et al., 2015) is selected as the highest performance model, randomly dividing the dataset into 60% for training, 20% for validation, and 20% for testing.

To complete the analysis and prove the generality of the proposals, VGG-16 (Simonyan and Zisserman, 2015) is selected for classification on CIFAR-10 (Krizhevsky, 2009), as widely used architecture and dataset. The latter consists of 32×32 images, already split into training and test data sets, with 10 classes. We randomly select 10000 images from the training set for validation and perform reflections for data augmentation.

The models[4] are trained from scratch in Matlab on an i9-10900X CPU 3.70GHz with 46GM of RAM and NVIDIA GeForce RTX 3080 GPU. This device is also used for the remaining experiments. Table 1 specifies the training configurations, determined through trial and error.

Table 1: Hyperparameter configurations for the used pairs CNN-dataset, with – referring to non-applicable parameters.

Hyperparameter configurations	U-Net CRBD	U-Net SBD	VGG-16 CIFAR-10
Optimizer	Adam	Adam	sgdm
Momentum	–	–	0.9
Squared gradient decay factor	0.9	0.9	–
Learning rate	10^{-3}	10^{-3}	10^{-2}
Drop period	–	3	25
Drop factor	–	0.3	3×10^{-3}
Weight decay	5×10^{-3}	5×10^{-3}	10^{-3}
Training epochs	30	20	80
Mini-batch size	8	8	128
Validation patience	4	4	–

3.2. Pruning results

In the case of U-Net on CRBD, the DIS-NV strategy uses the whole dataset to compute s^l_j, while $(\alpha, \beta, \gamma) = (0.75, 0.1, 0.25)$ as initially set. For SBD, a random selection of 50% of the dataset is employed, while reducing α and setting $(\alpha, \beta, \gamma) = (0.5, 0.1, 0.25)$ yields great results.

Table 2 and 3 show the reduction in accuracy, parameters, and FLOPs with respect to the original models (included at the top row), as well as the mean time needed to segment an image in the last column. We achieve a reduction of over 90% in parameters, nearly 99% for SBD. This dataset also reaches a 90% decrease in FLOPs, while CRBD shows a 74.4% reduction. The accuracy loss is minimal in both methdos, at most 0.85% in PCA-N (10) with CRBD, even improving the performance of the baseline networks in some cases (as indicated by negative values).

Table 2: Pruning results for U-Net on CRBD. Acc. is short of accuracy and Param. of parameters. ↓ indicates the drop percent between the pruned models and the baseline model (included on the top). Best results are shown in bold.

Original: Acc = 95.21%, Param = 7.70M, FLOPs = 8.28×10^{10}, time = 0.0243s				
Model	Acc. ↓(%)	Param. ↓(%)	FLOPs ↓(%)	time (s)
PCA-N (10)	0.85	**92.7**	64.7	**0.0114**
PCA-N (30)	**-0.70**			0.0118
DIS-NV (10)	0.49	88.1	**74.4**	0.0116
DIS-NV (30)	0.004			**0.0114**

This success in pruning is reflected in time, which drops by more than a half. Although direct comparisons with the literature are not possible due to the limited study of semantic segmentation and U-Net, the results demonstrate the effectiveness of the proposed methods in reducing model size without compromising accuracy.

Table 3: Pruning results for U-Net on SBD.

Original: Acc = 99.05%, Param = 7.70M, FLOPs = 1.85×10^{11}, time = 0.0344s				
Model	Acc. ↓(%)	Param. ↓(%)	FLOPs ↓(%)	time (s)
PCA-N (7)	0.04	**99.86**	94.8	**0.0104**
PCA-N (20)	**-0.06**			0.0106
DIS-NV (7)	0.41	97.33	**98.5**	0.0107
DIS-NV (20)	0.23			0.0105

We move into classification, where the DIS strategy uses 1000 random images from the training set to compute s^l_j. Regarding the thresholds, great performance is achieved with $(\alpha, \beta, \gamma) = (0.5, 0.1, 0.25)$. Table 4 summarizes the pruning results. We observe that DIS-NV achieves greater compression, with a 40.9% drop in parameters and 50.8% in FLOPs. However, PCA-N shows less accuracy reduction, at most 0.73% versus 3.26% for DIS-NV, suggesting that the importance criterion used in the former is more appropriate. In both cases, retraining for half of the original epochs improves baseline performance, even by more than 0.5% for PCA-N (40). The further compression achieved by DIS compared to PCA aligns with the fact that the latter provides poor pruning in low-resolution activation maps, such as those of VGG-16, as explained in Section 2.2.

[4]The U-Net model is taken from MathWorks (2022) and VGG-16 corresponds to the version of Li et al. (2017).

Table 4: Pruning results for VGG-16 on CIFAR-10.

Original: Acc = 85.05%, Param = 14.9M, FLOPs = $6.26{\times}10^8$, time = 6.8×10^{-3} s				
Model	Acc. ↓(%)	Param. ↓(%)	FLOPs ↓(%)	time ($\times 10^{-3}$ s)
PCA-N (10)	0.73			6.31
PCA-N (25)	0.51	14.1	17.9	6.28
PCA-N (40)	**-0.55**			6.30
DIS-NV (10)	3.26			5.75
DIS-NV (25)	2.56	**40.9**	**50.8**	5.62
DIS-NV (40)	-0.12			**5.52**

4. Understanding pruning performance

It should be noted that classification results are not as favorable as those in the segmentation task, despite using the same pruning methods. Furthermore, Table 4 does not reach state-of-the-art performance, where the parameter drop exceeds 85%.

As previously explained, the DIS method is appropriate for computing the number of channels to prune in VGG-16, while the next convolution influence criterion ℓ seems more suitable for determining their importance. In order to gain insights and guide modifications to achieve state-of-the-art rates, we perform explanatory analysis. In particular, we analyze which features are prioritized and preserved by each of the importance criteria. As the first convolution is known to focus on essential features for decision-making, we limit the analysis to this layer.

The discussion in López-González et al. (2024a) illustrates that, while the variance in the next convolution influence-variance criterion ℓs attributes high importance to texture features for CRBD (which are essential in this kind of datasets), it prioritizes color over edge detectors for CIFAR-10 (despite the importance of the latter). This analysis elucidates the observed different pruning performances.

We extend this study by analyzing whether the importance criteria ℓ and ℓs preserve the essential features underlying the first convolutional layer. To this end, we use the ELFA explanatory method (López-González et al., 2024b), which reveals these intrinsic characteristics. In particular, we consider the correlation matrix Λ between the essential features and the original channels. This matrix is shown on the left side of Figure 2, where rows correspond to channels and columns represent features. Large absolute values indicate a strong influence of a feature on a given channel.

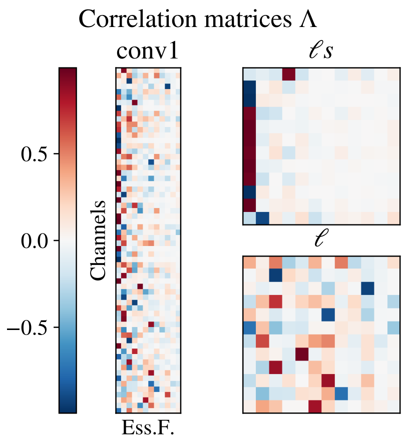

Figure 2: Original (left) and pruned (right) Λ matrices for the first convolution of VGG-16 on CIFAR-10. Rows correspond to channels, while columns represent essential features.

Letting m denote the number of essential features provided by ELFA, we retain the m most important channels in first convolutional layer according to each importance criterion. The resulting pruned Λ matrices represent the correlation between the essential features and the preserved channels, and illustrate which characteristics have been retained. These matrices are visualized in Figure 2, showing that ℓ retains nearly all the essential features, since almost every feature (column) is strongly correlated (large absolute values) with at least one channel (row). In contrast, ℓs primarily preserves the first features, forgetting the rest.

Alternatively, the Essential Feature Attribution Map (López-González et al., 2024b, EFAM) lets us highlight those regions of the input image that correspond to the most important preserved characteristics. Figure 3 displays the EFAM of the non-pruned layer, along with those obtained after pruning using each importance criterion. In line with the previous analysis, ℓ provides a more detailed representation, unlike ℓs, which focuses on fewer regions corresponding to essential features.

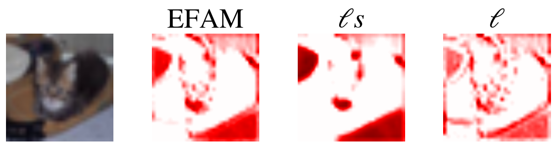

Figure 3: Essential features attribution maps of the first convolutional layer of VGG-16 on CIFAR-10. EFAM refers to the original attribution map.

All in all, these explanatory analyses demonstrate that, in the first convolutional layer of VGG-16 on CIFAR-10, ℓs not only prioritizes the wrong features but does not preserve the essential characteristics. On the contrary, ℓ gives importance to edge detectors and retains the intrinsic underlying features.

Consequently, an informed modification of the DIS-NV method could be introduced to improve pruning performance on VGG-16. The previous analysis indicates that ℓ is the appropriate importance criterion for the first layer. Taking into account that an analysis in López-González et al. (2024a) illustrates that using the ℓ criterion on the convolutional layer is analogous to considering ℓs on the ReLU layer following the convolution, we proceed as follows. To prune VGG-16 on CIFAR-10, we use the DIS-NV (40) proposal but computing the variance with respect to the ReLU layers after the convolutions. Besides, an exhaustive testing to find the best thresholds (α, β, γ) is needed, as detailed in López-González et al. (2024a).

Table 5 compares the results of this modified pruning method with state-of-the-art approaches. DIS-NV (40)-1, for which $(\alpha, \beta, \gamma) = (0.65, 0.05, 0.45)$, achieves an 86.4% reduction in parameters and an 82.2% decrease in FLOPs, along with a 0.78% improvement in accuracy. A higher pruning ratio is obtained with DIS-NV (40)-2, for which $(\alpha, \beta, \gamma) = (0.65, 0.1, 0.5)$, reaching a 91% reduction in parameters and an 89.4% decrease in FLOPs. As shown in Table 5, these results are comparable to those reported in the literature. Although our proposals are not the best in terms of parameter reduction, we do achieve a further decrease in FLOPs and gain in accuracy. Taking into account that our pruning approaches are remarkably simple, this certifies their validity.

Table 5: Results of pruned VGG-16 on CIFAR-10, taken from original papers, and of our experiments. The identified thresholds are (α, β, γ)-1 = $(0.65, 0.05, 0.45)$ and (α, β, γ)-2 = $(0.65, 0.1, 0.5)$.

Method	Acc. ↓(%)	Param. ↓(%)	FLOPs ↓(%)
l_1-Prune (Li et al., 2017)[5]	0.67	88.5	-
FStats (Li et al., 2020)	-0.33	90.7	56.3
FPGM (He et al., 2019)	0.36	88.5	-
FM info.(Shao et al., 2021)	0.39	93.3	73.2
N-Slim (Liu et al., 2017)	-0.06	88.5	51.1
Adapt-DCP (Liu et al., 2021)	-0.57	91.7	69.8
FDis (He et al., 2022)	-0.39	94.4	76.9
DIS-NV (40)-1	-0.78	86.5	82.2
DIS-NV (40)-2	2.26	91.7	89.4

5. Conclusions

Our work addresses the high computational cost of CNNs in computer vision in relation to their black-box nature. For this purpose, two low-effort filter pruning strategies are presented, and an explanatory analysis of compression performance is conducted. Our approach enables effective pruning of both segmentation and classification models without relying on intensive procedures, unlike many existing methods. We compare the obtained results against other pruning strategies. In addition, we gain insight into the features preserved by pruning criteria and their dependence on the dataset and pruning performance. The analysis also leads to improvements in the proposed compression methods.

In the future, we would like to conduct a more extensive study, involving additional CNNs and datasets, as well as comparisons with approaches from the literature. Furthermore, an extension of the explanatory analysis could be carried out to explore other parts of the networks and to incorporate additional explainability metrics and parameters.

Acknowledgments

The authors acknowledge support from the following Research Projects: INSERTION (PID20211-27648OB-C33) funded by the Spanish Ministry of Science, and SMART-BLOOMS (TED2021-130123B-I00) funded by the Spanish Ministry of Science and the European Union NextGeneration. The first author, Clara I. López-González, is supported by a FPU Ph.D. scholarship from the Spanish Ministry of Universities.

References

Ayinde, B. O., Inanc, T., Zurada, J. M., 2019. Redundant feature pruning for accelerated inference in deep neural networks. Neural Networks 118, 148–158.
DOI: 10.1016/j.neunet.2019.04.021

Chen, X., Zhang, Y., Wang, Y., 2022. MTP: Multi-Task Pruning for efficient semantic segmentation networks. In: IEEE International Conference on Multimedia and Expo. ICME'22. pp. 1–6.

Di Cicco, M., Potena, C., Grisetti, G., Pretto, A., 2017. Automatic model based dataset generation for fast and accurate crop and weeds detection. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'17. pp. 5188–5195.

Garg, I., Panda, P., Roy, K., 2020. A low effort approach to structured CNN design using PCA. IEEE Access 8, 1347–1360.
DOI: 10.1109/ACCESS.2019.2961960

Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M., Dally, W., 2016. EIE: Efficient Inference Engine on compressed deep neural network. In: ACM/IEEE Annual International Symposium on Computer Architecture. ISCA'16. pp. 243–254.

He, Y., Liu, P., Wang, Z., Hu, Z., Yang, Y., 2019. Filter pruning via geometric median for deep convolutional neural networks acceleration. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR'19. pp. 4335–4344.

He, Z., Qian, Y., Wang, Y., Wang, B., Guan, X., Gu, Z., Ling, X., Zeng, S., Wang, H., Zhou, W., 2022. Filter pruning via feature discrimination in deep neural networks. In: European Conference on Computer Vision. ECCV'22. pp. 245–261.

Krizhevsky, A., 2009. Learning multiple layers of features from tiny images.

Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H. P., 2017. Pruning filters for efficient convnets. ArXiv:1608.08710.

Li, H., Ma, C., Xu, W., Liu, X., 2020. Feature statistics guided efficient filter pruning. In: International Joint Conference on Artificial Intelligence. IJCAI'20. pp. 2619–2625.

Liu, J., Zhuang, B., Zhuang, Z., Guo, Y., Huang, J., Zhu, J., Tan, M., 2021. Discrimination-aware network pruning for deep model compression. IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (8), 4035–4051.
DOI: 10.1109/tpami.2021.3066410

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C., 2017. Learning efficient convolutional networks through network slimming. In: IEEE International Conference on Computer Vision. ICCV'17. pp. 2755–2763.

López-González, C. I., Gascó, E., Barrientos-Espillco, F., Besada-Portas, E., Pajares, G., 2024a. Filter pruning for convolutional neural networks in semantic image segmentation. Neural Networks 169, 713–732.
DOI: 10.1016/j.neunet.2023.11.010

López-González, C. I., Gómez-Silva, M. J., Besada-Portas, E., Pajares, G., 2024b. Layer factor analysis in convolutional neural networks for explainability. Applied Soft Computing 150, 111094.
DOI: 10.1016/j.asoc.2023.111094

MathWorks, 2022. U-Net layers.

Minh, D., Wang, H. X., Li, Y. F., Nguyen, T. N., 2022. Explainable artificial intelligence: a comprehensive review. Artificial Intelligence Review 55, 3503–3568.
DOI: 10.1007/s10462-021-10088-y

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention. MICCAI'15. pp. 234–241.

Shao, L., Zuo, H., Zhang, J., Xu, Z., Yao, J., Wang, Z., Li, H., 2021. Filter pruning via measuring feature map information. Sensors 21 (19), 6601.
DOI: 10.3390/s21196601

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations. ICLR'15.

Vidovic, I., Cupec, R., Željko Hocenski, 2016. Crop row detection by global energy minimization. Pattern Recognition 55, 68–86.
DOI: 10.1016/j.patcog.2016.01.013

[5] Results of Li et al. (2017) are obtained from Liu et al. (2017).