Model Optimization by Destructive Methods: A Case Study Using Structured Pruning in Image Classification Models

Flávio Moura*
flavio.moura@itec.ufpa.br
Federal University of Pará
Belém, Pará, Brazil

Adriano Madureira†
adriano.madureira.santos@itec.ufpa.br
Federal University of Pará
Belém, Pará, Brazil

Lyanh Lopes
lyanh.pinto@itec.ufpa.br
Federal University of Pará
Belém, Pará, Brazil

Vinicius Neves
andre.neves.alves@itec.ufpa.br
Federal University of Pará
Belém, Pará, Brazil

Walter Oliveira
walter@inteceleri.com.br
Federal University of Pará
Belém, Pará, Brazil

Saulo Costa
saulo.costa@ifpa.edu.br
Federal University of Pará
Belém, Pará, Brazil

Jefferson Morais
jeffersonmorais@gmail.com
Federal University of Pará
Belém, Pará, Brazil

Marcos Seruffo
seruffo@ufpa.br
Federal University of Pará
Belém, Pará, Brazil

Abstract

The Deep Neural Network architectures' advances offer innovative and effective solutions to various complex challenges. In order to improve the models' effectiveness based on these architectures' requirements, the model hyperparameter optimization is an essential project stage. Despite this, the existing optimization techniques demand a high computational cost when directly applied to the state-of-the-art most complex architectures. In this sense, this study proposes a method for hyperparameter optimization leveraging low computational cost for Artificial Intelligence models through structured pruning techniques. For the purpose of verifying and demonstrating the method, three main hypotheses are investigated throughout the implementation of a case study, which consists of the models' fine-tuning and optimization for three-dimensional geometric shapes classification on real-world object images. The process includes hyperparameter optimization for pruned models, considering the posterior retraining, evaluation, analysis and comparison between the performance and efficiency of the original models. Finally, the results were promising, indicating an improvement that reaches up to 10.58% Precision by just focusing on the models' learning rates optimization.

Keywords

Model Optimization, Deep Neural Networks, Artificial Intelligence, Model Pruning, Structured Pruning

1 Introduction

The Deep Learning (DL) field has presented extensive advances in Computer Vision [31], Natural Language Processing [24] and Robotics [25] Machine Learning (ML) field areas. This revolution is mainly conducted by the Deep Neural Network (DNN) architecture improvements, which offer efficient and innovative solutions to various complex state-of-the-art challenges. The Convolutional Neural Network (CNN) [39], Residual Neural Network (ResNet) [28] and Vision Transformer (ViT) [40] are examples of popular DNNs most used for complex ML tasks, such natural language processing, computer vision and reinforcement learning. The DNN progress had been a component of the quality model development. In this context, the need for strategic architectural configurations to extract the maximum DNN performance emerges. Thus, the models' hyperparameter optimization has an essential role in exploring the complete architectural potential through the ideal DNN setups founds [21].

The DNN hyperparameter optimization field is also in constant progress, being the Grid Search [1], the Random Search [10] and the Bayesian Optimization [32] some of the most used techniques. In contrast, the Random Search adopts an agile approach while randomly selecting the hyperparameter combinations, which can provide fast and satisfactory solutions. On the other hand, Bayesian Optimization presents a substantial advance and is frequently considered as most effective [32] by using probabilistic models to predict the model performance on different hyperparameter setups.

Although these techniques are efficient, hyperparameter optimization is a highly computational demand approach when applied to the most complex state-of-the-art model architectures [5]. This turns even worse the practical challenges, especially in computational scarcity environments, such as Yasir et al. (2021) addresses the need for insights based on the non-exhausting optimization methods for these environments. In this scenario, the Architectural Neural Destructive Search (ANDS) methods offers an ideal approach, ensuring the DNN architectural parameters reduction and simplifying strategically the DNN components to better evaluate the performances' impacts on the general model [20]. Among these methods, it is highlighted that pruning is an adequate approach, which leverages the high computational demand and allows the simplification of effective DNNs' size reduction without compromising its performance.

Inside the pruning techniques spectral, the literature provides a variety of methods, such as the iterative pruning approach [13], in which the components removal occurs on a series of stages; the one-shot pruning approach [9], performed in just one stage at once; the random pruning approach [18], in which the components

*Both authors contributed equally to this research.
†Both authors contributed equally to this research.

are removed randomly; the structured and non-structured pruning approaches [36]. Among these, the structured pruning is applied posterior to the model training process, eliminating the channels or neurons' unities based on its performance relevance to the general model [27]. This technique reduces the DNN complexity in a sorted way and is capable of reducing the model training computational cost while maintaining its most relevant weights.

Considering the significance of pruning techniques, particularly structured pruning, in reducing complexity and computational cost in DNNs, this work aims to extend these benefits by proposing a new method for hyperparameter optimization. This method is specifically tailored for models in computationally constrained environments. The application of this method involves four distinct steps: (i) training the model (pre-pruning); (ii) applying the one-shot structured pruning technique to the trained model (post-pruning); (iii) optimizing the hyperparameters of the post-pruning model; and finally, (iv) retraining the pre-pruning model with the best hyperparameters obtained in step iii, resulting in the optimized post-pruning model. This idea consists of using the best hyperparameter found in the optimized pruning model, ensuring that the computational cost is reduced and, at the same time, improving the performance of the pre-pruning model.

This work was made based on three main hypotheses, whose are thoroughly investigated: (a) whether the pruned DNN optimized hyperparameters found using a 90% pruning approach would be relevant to the original model; (b) whether directly optimize the original model is less efficient than optimize its 90% pruned version, leveraging the inference time, solution convergence and model performance; (c) whether exists a variation interval range intersection that provides satisfactory precision among the three evaluated pruned models. The goal of these hypotheses is to better explore the DNNs' optimization effectiveness, while directly comparing the pre-pruning models' performance concerning its optimized pruned version performances.

In order to demonstrate the method application, it is proposed a case study which aims to address Brazil's educational gap in basic geometry teaching by integrating AI into a mobile application using model optimization through destructive methods. This strategy is essential for mobile environments with limited computing resources, ensuring the application is both lightweight and efficient. In the field of mathematics teaching, previous work has strongly pointed out that the traditional methodologies are contributing to the low educational indices on Mathematics [26]. Data from the International Program of Students Assessment (PISA) and the Basic Education Assessment Index (IDEB) indicate Brazilian students struggle with basic Mathematics proficiency, with expectations of further decline. Education 5.0 offers an innovative solution by integrating digital technologies like the Metaverse and ML to revolutionize mathematics teaching [12].

Leveraging this, the work presents a case study involving the three-dimensional geometric shapes classification models fine-tuning and optimization through real-world object images, using Metaverse and ML technologies. The goal is to integrate these technologies on the Geometa[1] application, which is an educational application designed to teach basic Mathematics through the Metaverse, proposing an immersive and experimental teaching method. The study will validate the proposed hypotheses, applying a model low computational cost optimization method and discuss the research founds and their implications for Mathematical teaching.

The main research contributions involve the following concerns: (i) the development of a low computational cost hyperparameter optimization method designed for DNN through a structured pruning technique; (ii) a case study implementation for the method demonstration; (iii) the method validation through the hypothesis that are deeply investigated throughout the work; (iv) a general and definitive approach for methodology replication on different Artificial Intelligence contexts.

The remaining work is divided into four main sections, which present: in Section 2 the literature review through related works; in Section 3 the adopted methodology description to the method development and implementation, contextualizing the proposed case study; in Section 4 the experiments and results are discussed; and finally, in Section 5 the final research considerations.

2 Related Works

In this research context, it was adopted a term search protocol, centered on four key concepts: "pruning", "structured", "non-structured" and "optimization pruning". The "pruning" term is used to describe an unnecessary or redundant component elimination process on a structure, which is an often performed approach in a diversity of areas, such as Data Science, Software Development and Artificial Neural Networks. The "structured" and "non-structured" terminology refers to systems or data that present a clear organization and definition, while also including the systems that lack standardization and predictability.

The "optimization pruning" key represents a specific approach that aims to improve the systems or algorithms' performance and efficiency, through unnecessary component removal without compromising its essential functionalities. These terms are particularly pertinent in an Artificial Intelligence context, where the models and algorithms optimization is fundamental to the continual improvement of DNN effectiveness. Furthermore, this terms set was employed based on structured searches and subsequent analyses, aiming to provide an actual and comprehensive understanding involving the most recent practices and innovations in the field.

Although the pruning techniques are not a recent advance in the DNN field, it is possible to notice that its application growth is a tendency nowadays. This occurs due to the DNN diffusion to the mobile devices, as demonstrated by Hubens et al. (2020) , where it is clarified that to execute the models on a lighter way it is necessary to perform the redundant parameter pruning. This approach allows providing lighter models for mobile devices without the need for a model retraining stage.

In this sense, Fang et al. (2023) presents the Torch Pruning library[2], which was designed to provide high-level and low-level pruning application methods, leveraging different pruning techniques and the effective model pruning complexity reduction. Besides this, Blalock et. al. (2020) highlighted the absence of standardized benchmarks and metrics for DNN pruning. Furthermore, Crowley et al. (2018) explore the structured pruning and emphasize that the structured pruning produced architectures would provide

[1] https://play.google.com/store/apps/details?id=com.Inteceleri.Geometa&pli=1

[2] https://github.com/VainF/Torch-Pruning

superior improvement to the ones that were first conducted to the pruning process to posterior fits.

When the structured and non-structured pruning techniques are compared to each other, it would be noticed that the structured pruning can improve the processing speed and provide computational efficiency benefits on a trade-off with the models precision reduction [35]. On the other hand, non-structured pruning approaches would have disadvantages concerning computational efficiency. For this reason, Zhao et al. (2023) identified two main non-structured pruning-related disadvantages: the weight matrix turns into a sparse matrix and the weight connections distribution is randomly removed.

Thus, the structured pruning would be able to provide the best optimization results. This is also emphasized by Xiao et. al. (2022) , in which is treated that the structured pruning-related optimization would provide benefits on the realistic model efficiency improvement. In contrast, Cai et. al. (2022) demonstrate that the non-structured pruning-related optimization would not be able to obtain an increase in the real models' efficiency, besides the model preserves a low sparsity posterior to the pruning process.

When the actual literature is compared to this work's contributions, it is proposed the connection between the model optimization and the structured pruning field. Based on the research carried out, the literature lacks model hyperparameter optimization-related approaches based on computational efficiency. Thus, this work introduces an innovative approach to incorporate optimized hyperparameters obtained from structured pruning techniques on non-pruned models. This implementation aims to provide substantial improvements on the models' optimization process computational cost and performance. Different from other works where they only apply the pruning method, in this work hyperparameters obtained through pruning are used in other models, seeking optimization and contributing to the state of the art.

3 Methodology

The proposed method was organized into two main pipeline blocks, the CPU Server Pipeline and GPU Pipeline, as shown in Figure 1. Both the pipeline blocks have inner implementation blocks. On the CPU Server Pipeline, the Geometa Vision AI models are used for Model Pruning and subsequent Hyperparameter Optimization. Further, on the GPU Pipeline, the original Model Retraining is performed using the optimized hyperparamters. In sequence, the original Model Evaluation is performed through machine learning and inference performance metrics, for the detailing on the Model Analysis of the different performances obtained. Finally, the original Model Launch is carried out to store the best models for future application integration. Each pipeline block and its will be further detailed in the following subsections.

3.1 CPU Server Pipeline

The CPU Server Pipeline flux starts with the Geometa Vision AI models, which are models previously developed by the Inteceleri company based on the three-dimensional geometric shapes classification through real-world objects. In sequence, the structured Model Pruning approach is applied to turn these models as fast and lighter as possible. Finally, the Bayesian Optimization technique is used to search the hyperparameters closest to the optimal solutions on a range interval defined for experimental hyperparameter selection.

3.1.1 Geometa Vision AI In a previous paper, Santos et al. (2023) trained CNN, MobileNet, ViT, BEiT, ResNet, and ResNeXt models for the classification of three-dimensional geometric shapes classes using a reclassified ObjectNet database [3], intituled Solidos-V1. For this work, the ResNet and BEiT models, which obtained the best results in the previous article, were selected for the implementation of the proposed method. Besides these, the EfficientNet model was selected for a better comparison of the results with the other models. The aim is to improve the performance of the trained models and implement them in the GeoMeta application to obtain the best performance for users in future versions. For this, the same database (i.e. ObjectNet) was used to train, validate and test the models pruned and not pruned.

In this study, the ResNet architecture, particularly Microsoft ResNet-50, is utilized for its ability to train deep neural networks efficiently, maintaining low computational costs and high performance in varying conditions such as different poses and lighting in human face identification [15, 19]. Additionally, Microsoft's state-of-the-art BEiT, a successor to Google ViT and part of the ViT category, stands out for its semi-supervised learning approach, combining supervised and unsupervised learning to enhance tasks like image classification and object detection [2]. Furthermore, Nvidia's EfficientNet was used, introduced in 2019, innovates CNNs for efficient image processing, employing a compound scaling method that balances network width, depth, and resolution, thereby achieving high accuracy and efficiency in image classification and object detection tasks [33].

3.1.2 Model Pruning To address the computational challenges in deep neural networks, pruning techniques have been developed, focusing on reducing computational demands without sacrificing performance. These methods are based on the principle that not all neurons or layers contribute equally to the model's effectiveness. By identifying and removing less critical parameters, structured pruning can significantly decrease the model's size and computational requirements, even before the training process begins, aligning with model optimization strategies [16]. This approach is particularly effective in streamlining the training, evaluation, and application phases of deep learning models.

One notable variant of this technique is Taylor Pruning, which utilizes Taylor series expansions to assess the impact of individual parameters on the network's output. By expressing the network function as an infinite sum of its derivatives, this method provides a nuanced understanding of how each parameter influences the overall performance. The parameters with minimal contributions, as indicated by lower-order terms in the Taylor series, are identified as candidates for pruning. This iterative process not only enhances computational efficiency but also necessitates subsequent fine-tuning to offset any potential performance losses. The effectiveness of Taylor Pruning is ultimately evaluated based on the improved machine learning and inference performance of the models [29, 30].

[3] https://objectnet.dev/

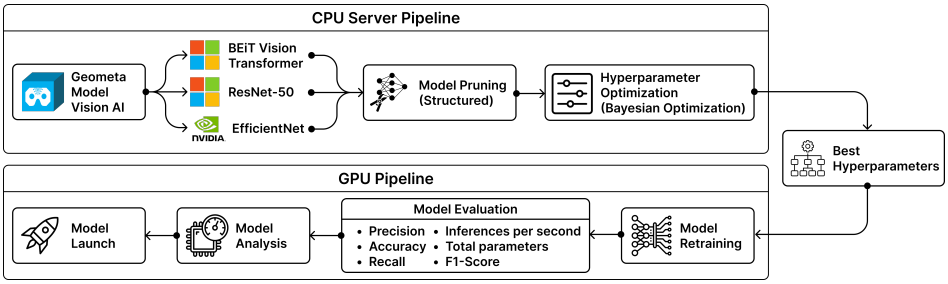

Figure 1: Pipeline model for CPU server and GPU

3.1.3 Hyperparameter Optimization The enhancement of model performance in AI is increasingly reliant on the optimization of hyperparameters, particularly for pruned models. Traditional methods like Grid Search and Random Search, while effective, often incur high computational costs and may not always converge to an optimal solution. A more efficient alternative is Bayesian Optimization, which utilizes probabilistic surrogate models like the Gaussian Process Regression model. This non-parametric method excels in handling complex, non-linear functions and in quantifying uncertainties in predictions, making it ideal for optimizing intricate and uncertain objective functions [23].

The Bayesian Optimization operates iteratively, updating the surrogate model with new data to refine its mean and covariance functions, thereby enhancing the accuracy of the posterior distribution in estimating the true objective function. A key component of this process is the Upper Confidence Bound (UCB) acquisition function, which strategically balances the exploration of uncertain areas with the exploitation of promising regions. The ultimate goal of this optimization is to pinpoint the global optimum with the fewest possible evaluations of the true function, achieved through a continuous cycle of model fitting, acquisition function optimization, and objective function evaluation [14].

Due to the drastic initial lose in machine learning metrics performance, the pruned models evaluation was conducted under the improvement ratios, which combines a pruned initial state performance and the pruned optimized performance for a defined maximization metric, considering each of the pruned models' hyperparameter setups. In this sense, the calculations performed are shown in sequence in the Equations 1 and 2, where:

$$T_P = \zeta_T \times P_{ISP} \tag{1}$$

$$PIR = \begin{cases} 0 & , P_{OP} < P_{ISP} \\ 1 & , P_{OP} \geq \zeta_T \times P_{ISP} \\ \frac{P_{OP} - P_{ISP}}{TP} & , \text{other cases} \end{cases} \tag{2}$$

The **Performance Improvement Ratio** (PIR) is related to any defined metric for the case study. This metric is influenced by the **Pruned Initial State Performance** (P_{ISP}) which is evaluated before the optimization process, the **Pruned Optimized Performance** (P_{OP}) which is evaluated after the pruned model training with a hyperparameter setup and the **Target Performance** (T_P) which is evaluated based on the specific research goals. The T_P is highly influenced by the ζ_T specialists defined value, which is a multiplier of the P_{ISP}.

3.2 GPU Pipeline

The GPU Pipeline flux initiates with the optimal hyperparameters found on the CPU Server Pipeline being used for original Model Retraining, Thus, these models have their machine learning and inference performances evaluated through original Model Evaluation and collected for discussion on the original Model Analysis. Finally, these models are launched for future research and integration on the Inteceleri models storage.

3.2.1 Model Retraining Subsequent to the Geometa Vision AI pruned models optimization, the optimal hyperparameters are found and stored in a hyperparameter optimization dataset. This dataset is used to identify the Top10 optimized configurations for each original pruned model and these configurations are conducted to the retraining process of the original model. This is done in order to perform a future comparison between the models' original optimized version and its original initial version, which could provide insights concerning the performance improvements achieved considering the applied methodology.

3.2.2 Model Evaluation In integrating Artificial Intelligence with user-centric applications, the trained models were evaluated using machine learning metrics and inference performance indicators. Machine learning metrics, such as precision, accuracy, recall, and F1-Score, assess the model's capability to recognize three-dimensional geometric shapes' patterns, crucial for minimizing inference errors and discarding irrelevant patterns [38]. Inference performance metrics, on the other hand, focus on the model's computational complexity, including total and per-inference time, inferences per second (IPS), total parameters, and model size. These metrics are essential for ensuring a satisfactory user experience, as they reflect the model's responsiveness and efficiency in processing data streams [8]. The balance between these two sets of metrics is vital for the effective application of AI in user-focused environments, where both Precision and computational efficiency in pattern recognition are key to performance.

3.2.3 Model Analysis In the process of retraining and evaluating original models using optimal hyperparameter configurations, a comprehensive analysis was conducted to identify the best-performing model for integration with the Geometa application. This analysis encompassed various aspects: evaluating the performance of original models trained with default library hyperparameters over a hundred epochs; assessing the optimal pruned model performance with hyperparameter optimization over three epochs for each setup; comparing the initial pruned model with the top five best pruned

optimized models; contrasting the performance of the initial original model with its optimized version; and summarizing general research findings and hypothesis refutation. These analyses are crucial for extracting insights into hyperparameter optimization using pruning methods, contributing to future optimization research in deep learning and enhancing the scientific community's understanding of these methods for reproducibility and real applications [22].

3.2.4 Model Launch Posterior to the performed analysis the original optimized models are stored on Google Drive as Launch. The main purpose behind the storage of these models is the future AI model integration with the Geometa application. In this sense, the models will be compared among different three-dimensional geometric shapes' classification datasets and their baseline performance will be available for future research. The other purposes include the best analysis strategies throughout the models' possibilities in the Deep Learning field. This will turn in the future Geometa's Augmented Reality module available for students' and teachers' experiments, focusing on proposing better quality Mathematics education for these individuals.

4 Experiments and Results

Later than the performed experiments, the results were evaluated and presented for further investigation of the hypothesis and selection of the best model for future integration with the Geometa application. It is also worth noting that the models involved in this study will be stored as a launch for future research of the best approaches found to improve the educational process and also intend to improve the recent deep learning state-of-the-art optimization.

Due to the structured pruning's low efficiency on models' parameters and inference time reduction when considering lower pruning amounts, the pruning amount was inserted as high as possible while still attempting to represent the original models. For this reason, the 90% pruning approach was performed as it proved to be satisfactory through a pruning amount performance analysis carried out leveraging the pruned models' precision and size decay, in addition to inferences per second increase.

This value was selected due to even in 10%, 30% and 60% cases the original model still loses considerable Precision and still presents low IPS. The Table 1 summarizes the mean pruning amount effects on performance between the models considering the mean and standard deviation for Precision, IPS and Total Parameters. Thus, in sequence, the CPU Server computational setup will be described.

The CPU Server Pipeline computational modeling was carried out using Python 3.11 programming language, and its libraries such as: PyTorch, Torchvision, HuggingFace and BoTorch. The CPU Server Pipeline was executed under an experimental setup involving the use of a computer with Intel(R) Core(TM) i7-6700F CPU 3.40 GHZ, 16GB DDR3 RAM and 80GB HD partition. The results were collected during one day of pruned models' training and processing. After obtaining the best-found hyperparameters the GPU Pipeline flow was carried out in an attempt to find the best hyperparameters for further original model retraining.

An early exploration set of hyperparameters was defined for initial searching the pruned models optimization space and it led to the early precision scores filling onto the hyperparameter dataset, considering a total of 130 different setups. The hyperparameters set includes multipliers scalars of 1, 3, 6 and 9 combined with order base values around 10^{-2} to 10^{-11}. It is worth noting that every base learning rate was multiplied by the following constants: 1, 1.25, 1.75, 2.25, 2.75, 3.25, 3.75 and 4.25. Then, the values were normalized to stay between 10^{-1} and 10^{-12}. The reason for these distant extremes was just to evaluate whether the models were able to present a satisfactory performance just on the extremes.

The hyperparameter search progressed with Bayesian Optimization, leveraging Gaussian Processes to analyze previously explored hyperparameters, generating a dataset with hyperparameters, PIR, and ζ for each learning rate of the pruned models, followed by retraining. This retraining utilized Inteceleri Vision AI models from prior studies to explore potential performance enhancements through a swift optimization process, comparing the machine learning metrics of the optimized pruned models against their original counterparts, with subsequent detailing of GPU computational setups.

The computational setup for the GPU pipeline was based on Python 3.11 and utilized similar libraries as the initial pipeline, excluding BoTorch, which is specific to Bayesian Optimization. The hardware comprised an Intel(R) Core(TM) i3-10100F CPU, NVIDIA GeForce RTX 2060, 16GB DDR4 RAM, and a 1TB HD, with results gathered from a day's training and processing. This approach aimed for performance improvements despite the models being significantly pruned, indicating a strategic move towards refining the optimization process for future enhancements, with forthcoming sections to provide a detailed performance analysis.

4.1 Original Models Performance Analysis

The performance of the models was evaluated onto six main metrics, that is, Precision, Accuracy, Recall, F1-Score, IPS and Total Parameters. These performances represent the baseline for this experiment, as it is the initial and brute state of the models after the training process considering the default hyperparameters of the libraries used, which was: 10^{-3}, 100 epochs, AdamW optimizer. Table 2 shows a brief evaluation of the ResNet, BEiT and EfficientNet models on the test dataset.

The ResNet model showed the best performance on machine learning metrics, although it had intermediate performances on inference performance metrics. When compared to EfficientNet, which is the second better model, the ResNet achieved an improvement of: 9.45% on Precision; 4.77% on Accuracy; 5.56% on Recall; 4.11% on F1-Score. Whilst this, the ResNet model had reduced values on inference performance metrics when compared with EfficientNet, which achieved: 95.72% better IPS and 82.93% less Total Parameters.

When BEiT is compared with these two models, it had the lowest performances both on machine learning and inference performance metrics. When compared to the ResNet model on machine learning metrics, the ResNet presents an improvement of: 30.9% on Precision; 43.87% on Accuracy; 49.78% on Recall; 45.6% on F1-Score. On the other hand, when the comparison is made concerning the EfficientNet model on inference performance metrics, the EfficientNet presents an improvement of: 85.06% on IPS and 95.33% less Total Parameters.

Pruning Amount	Original	10%	30%	60%	90%
Precision	68.56 ± 12.93	20.08 ± 5.65	6.59 ± 1.70	2.17 ± 1.33	2.68 ± 1.11
IPS	16 ± 10	17 ± 10	22 ± 12	35 ± 18	75 ± 41
Total Parameters	37.8 M ± 34.9 M	31.4 M ± 29.4 M	20.6 M ± 19.9 M	8.4 M ± 8.8 M	1.2 M ± 1.6 M

Table 1: Mean pruning effects on Precision, IPS and Total Parameters

Model	Precision	Accuracy	Recall	F1-Score	IPS	Total Parameters
ResNet	82.01	77.27	75.27	72.64	14.94	23.5 M
BEiT	51.11	33.40	25.49	27.04	4.37	85.77 M
EfficientNet	72.56	72.5	69.71	68.53	29.24	4.01 M

Table 2: The original models evaluation through machine learning and inference performance metrics

4.2 Pruned Models Performance Analysis

The hyperparameter optimization process started considering about a thousand different pruned models' hyperparameter setups before evaluating the best configurations. In order to show the benefits of the structured pruning process, the models' specifics IPS and Total Parameters on 90% pruning amount are shown in Table 3. The best model on IPS after the pruning process was the ResNet model with 105 IPS, even having 73.91% more Total parameters than the EfficientNet model. Inversely, the BEiT model presents the lowest IPS of 16 and the highest Total Parameters, being 98.31% bigger than the EfficientNet model. After the pruned models inference performance metrics evaluation, the Hyperparameter Optimization process was performed.

Model	IPS	Total Parameters
ResNet	105	0.23 M
BEiT	16	3.56 M
EfficientNet	102	0.06 M

Table 3: Pruned initial state models evaluation on inference performance metrics

During the hyperparameter optimization process, the setups and the scores were constantly updated on a dataset, which provides the analysis of the configurations that better improved the initial performance of the pruned models on the Precision metric, which was defined as the PIR metric. Thus, each pruned model was trained for three epochs. It is worth noting that the ζ defined for the experiments was equal to 10, which means that the experiments expected to achieve up to 10 times the pruned model initial Precision on the third epoch. In this sense, the Top5 better setups were extracted from the dataset and are presented to conduct an evaluation of the PIR metric and ζ of the pruned models on these hyperparameters. The goal is to evaluate the capability of the Top5 better setups to improve the original models during the retraining process. The Table 4 presents the Top5 hyperparameter setups and the correspondent PIR and ζ to the ResNet, BEiT and EfficientNet pruned models.

In the evaluation of pruned models, the top five setups based on the PIR metric showed mean values of 18.6%, 98.2%, and 12.4% for ResNet, BEiT, and EfficientNet, with standard deviations of 6.4%, 2.7%, and zero, respectively. Additionally, the mean ζ evaluations were 2.86, 11.47, and 2.24 for these models, with standard deviations of 0.64, 1.06, and zero. This analysis highlights BEiT pruned models' significant performance improvement, exceeding expectations with an 11.47-fold increase in initial precision. In contrast, ResNet and EfficientNet pruned models did not meet ζ expectations as closely but still showed efficient optimization, enhancing initial performances by about 2.86 and 2.24 times in the best setups.

Finally, the BEiT model achieved better performance improvement, presenting on mean PIR metric 79.6% increase when compared to ResNet and 85.8% concerning EfficientNet. The ζ multiplier surpassed the expectations for BEiT, exceeding 12.82% of the defined target multiplier, while being 75.07% better than ResNet and also outperforming the EfficientNet on 80.47% on the multiplier. Thus, in sequence with the pruned models' performance analyses, the Top5 hyperparameters obtained were conducted to the original models' optimization.

4.3 Original Optimized Models Performance Analysis

Posterior to the pruned models analysis, the best-found hyperparameters were used to train the original Geometa Vision AI models for a hundred epochs and its performance was constantly evaluated on the validation dataset. The model retraining process collects the model at the epoch that presents the best performance on the precision metric for testing subsequent evaluation on the test dataset. Table 5 presents the original optimized models' machine learning metrics performance on the test dataset, leveraging the best-found hyperparameters. It is highlighted that the original models maintain the same IPS and Total Parameters presented before even when optimized, as the goal is to optimize the machine learning metrics performances.

In this sense, it was identified that the best optimized BEiT was obtained with LR equal to $3.88652794 \cdot 10^{-2}$ after 54 epochs. Furthermore, the best optimized ResNet was obtained with LR equal to $8.85630921 \cdot 10^{-2}$ after 63 epochs. In sequence, the best performance-optimized EfficientNet was obtained with LR equal to $6.00000005 \cdot 10^{-3}$ after 18 epochs. The model evaluation on the test dataset provides insights involving the improvement of the three models on their optimized version concerning the original initial models.

The optimized ResNet was the better evaluated model and concerning the original version it was able to achieve an improvement of: 2.18% on Precision; 1.14% on Accuracy; 1.17% on Recall; and 0.79% on F1-Score. On the other hand, the optimized BEiT had the best improvement on every evaluated metric to the original model, achieving an increase of: 10.58% Precision; 5.92% Accuracy; 6.96% Recall; and 5.29% F1-Score. In closing, the performance improvement obtained through the optimized EfficientNet with its original version was 7.62% on Precision. However, this model loses performances on the other metrics, having a decrease of: 2.73% on Accuracy; 2.55% on Recall and 4.44% on F1-Score.

The model that still presents the best results when compared among others, posterior to the optimization process, was the ResNet model with 4.01% Precision, 8.64% Accuracy, 9.28% Recall and 9.34%

	ResNet			BEiT			EfficientNet		
Rank	**Learning Rate**	**PIR**	ζ	**Learning Rate**	**PIR**	ζ	**Learning Rate**	**PIR**	ζ
1	$1.78256611 \cdot 10^{-4}$	0.284	3.84	$2.24332325 \cdot 10^{-3}$	1	13.30	$6.00000005 \cdot 10^{-3}$	0.124	2.24
2	$1.47153260 \cdot 10^{-4}$	0.229	3.29	$3.88652794 \cdot 10^{-2}$	1	11.95	$4.23757359 \cdot 10^{-2}$	0.124	2.24
3	$9.36821848 \cdot 10^{-2}$	0.164	2.64	$2.02499996 \cdot 10^{-5}$	1	11.01	$4.14208360 \cdot 10^{-2}$	0.124	2.24
4	$8.85630921 \cdot 10^{-2}$	0.154	2.54	$5.95125835 \cdot 10^{-3}$	0.981	10.81	$5.09522446 \cdot 10^{-2}$	0.124	2.24
5	$9.88561288 \cdot 10^{-2}$	0.100	2.00	$3.68676335 \cdot 10^{-3}$	0.929	10.29	$4.13979366 \cdot 10^{-2}$	0.124	2.24

Table 4: Top5 optimized setups for the models ResNet, BEiT and EfficientNet

Model	Precision	Accuracy	Recall	F1-Score
ResNet	**84.19**	**78.41**	**76.44**	**73.43**
BEiT	61.69	39.32	32.45	32.33
EfficientNet	80.18	69.77	67.16	64.09

Table 5: Machine learning metrics performance post optimization process

F1-Score better performances concerning the EfficientNet model, which was the second highest performance evaluated model. Finally, the BEiT model proved to be the most complex and unstable model on every evaluated machine learning metric, being surpassed by the ResNet model, which had 22.5% on Precision, 39.09% on Accuracy, 43.99% on Recall and 41.1% on F1-Score.

4.4 Research Founds and Hypothesis Discussion

This section consolidates the findings from model analysis, revealing that: (i) Pruning techniques effectively optimize hyperparameters in complex models, aligning with hypothesis (a); (ii) Direct optimization of complex models is significantly more resource-intensive than optimizing pruned models, supporting hypothesis (b); and (iii) Optimal learning rates for similar DNN optimization experiments fall between 10^{-2} and 10^{-5}, corroborating hypothesis (c). These insights lead to a discussion on the research hypotheses informed by the findings.

The (a) hypothesis was confirmated through the original optimized model analysis, in which the improvement in every machine learning metric was observed, considering the research method for using only the pruned models during the hyperparameter optimization process. In this sense, the pruning techniques demonstrated an effective way for considerably reducing the time and dimension of the models and searching the hyperparameters setup possibilities while still preserving the posterior-most complex and original models retraining.

The (b) hypothesis was confirmated through the comparison between the inference performance metrics applied to the original and optimized models to obtain the IPS and Total Parameters. It was demonstrated that the models had an improvement in their IPS and a reasonable reduction in Total Parameters when the pruned models were compared to the original models. Whilst the mean IPS for the original models is 16, the mean of the same metric for the pruned models is 75. In addition, the mean Total Parameters of the original models were 37.8 M, while this metric for the pruned models was 1.2 M. These improvements turned the models 78.29% faster and 96.83% lighter during the training and evaluation process in comparison to leading with the original ones.

The (c) hypothesis was confirmated through Table 4, as the best learning rates found for the three DNN models were between 10^{-2} and 10^{-5}, although the performances were not close, this interval range was able to provide satisfactory performances to both models, whilst also being an intersection for the three models. It is also valid to mention that it was not maintained the proportionality of the zeta values, although the metric was crucial for identifying the best hyperparameters based on Precision improvement.

In closing, the retrained models leveraging the best hyperparameter setups will be launched as a baseline for future studies involving optimization techniques and model complexity reduction. This will be done by focusing on proposing even faster and lighter models for a variety of research areas in the deep learning field. The actual research carried out the case study of the Geometa Vision AI models and the future intention is to integrate these models into the Geometa application, proposing a more immersive and simple educational experience in teaching and learning for teachers and students.

5 Conclusion

This study introduced a structured pruning technique for optimizing DNN hyperparameters at low computational cost. A case study was conducted to explore three main hypotheses by: (i) training and evaluating ResNet, BEiT, and EfficientNet models on the Solidos-V1 dataset, followed by structured pruning; (ii) optimizing and evaluating the pruned models to identify the best hyperparameter setups; (iii) retraining the original models with the best hyperparameters; and (iv) comparing the performance of the retrained models against their pre-pruned counterparts. Key metrics such as IPS, Total Parameters, and evaluation results were recorded for each model throughout the process.

Conforming the experiments performed and the obtained results presented in Section 4, it was identified that the (a) hypothesis was accepted, such that it was noticed an improvement in every machine learning metric when the original optimized model was compared to its initial state original version. Furthermore, it was also observed that the direct original model optimization is less efficient than its 90% pruned version optimization, which confirmed the (b) hypothesis. Finally, the best learning rates were found between 10^{-2} and 10^{-5} for the three state-of-the-art deep learning models, emphasizing the acceptation of the (c) hypothesis.

Future work will focus on developing an enhanced evaluation metric for pruned models to capture precision growth more accurately across all training epochs, rather than solely the third epoch as currently practiced. Additionally, the introduction of adaptive learning rates after a set number of epochs aims to enhance both the optimization and performance of pruned and original retrained models. Lastly, research will be directed towards determining the optimal pruning threshold, moving away from arbitrary values to find a balance between reducing computational costs and minimizing precision loss.

References

[1] Hussain Alibrahim and Simone A. Ludwig. 2021. Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization. In *2021 IEEE Congress on Evolutionary Computation (CEC)*. 1551–1559. https://doi.org/10.1109/CEC45853.2021.9504761

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254* (2021).

[3] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. 2020. What is the state of neural network pruning? *Proceedings of machine learning and systems* 2 (2020), 129–146.

[4] Yaohui Cai, Weizhe Hua, Hongzheng Chen, G Edward Suh, Christopher De Sa, and Zhiru Zhang. 2022. Structured pruning is all you need for pruning CNNs at initialization. *arXiv preprint arXiv:2203.02549* (2022).

[5] Daeyoung Choi, Hyunghun Cho, and Wonjong Rhee. 2018. On the Difficulty of DNN Hyperparameter Optimization Using Learning Curve Prediction. In *TENCON 2018 - 2018 IEEE Region 10 Conference*. 0651–0656. https://doi.org/10.1109/TENCON.2018.8650070

[6] Elliot J Crowley, Jack Turner, Amos Storkey, and Michael O'Boyle. 2018. A closer look at structured pruning for neural network compression. *arXiv preprint arXiv:1810.04622* (2018).

[7] Adriano Madureira Dos Santos, Flávio Rafael Trindade Moura, André Vinicius Neves Alves, Lyanh Vinicios Lopes Pinto, Fernando Augusto Ribeiro Costa, Walter Dos Santos Oliveira Júnior, Diego Lisboa Cardoso, and Marcos Cesar Da Rocha Seruffo. 2023. Artificial Intelligence in Education 5.0: a methodology for three-dimensional geometric shape classification for an educational tool. In *2023 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*. 1–6. https://doi.org/10.1109/LA-CCI58595.2023.10409430

[8] Miriam Elia. and Bernhard Bauer. 2023. A Methodology Based on Quality Gates for Certifiable AI in Medicine: Towards a Reliable Application of Metrics in Machine Learning. In *Proceedings of the 18th International Conference on Software Technologies - ICSOFT*. INSTICC, SciTePress, 486–493. https://doi.org/10.5220/0012121300003538

[9] Sara Elkerdawy, Mostafa Elhoushi, Abhineet Singh, Hong Zhang, and Nilanjan Ray. 2020. One-Shot Layer-Wise Accuracy Approximation For Layer Pruning. In *2020 IEEE International Conference on Image Processing (ICIP)*. 2940–2944. https://doi.org/10.1109/ICIP40778.2020.9191238

[10] Ahmad Esmaeili, Zahra Ghorrati, and Eric T. Matson. 2023. Agent-Based Collaborative Random Search for Hyperparameter Tuning and Global Function Optimization. *Systems* 11, 5 (2023). https://doi.org/10.3390/systems11050228

[11] Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. 2023. Depgraph: Towards any structural pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16091–16101.

[12] Carla Denize Ott Felcher and Vanderlei Folmer. 2021. Educação 5.0: Reflexões e perspectivas para sua implementação. *Revista Tecnologias Educacionais em Rede (ReTER)* (2021), e5–01.

[13] N. Fnaiech, F. Fnaiech, and M. Cheriet. 2002. A new feedforward neural network pruning algorithm: SSM-iterative pruning (SSMIP). In *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 4. 6 pp. vol.4–. https://doi.org/10.1109/ICSMC.2002.1173310

[14] Roman Garnett. 2023. *Bayesian Optimization*. Cambridge University Press. https://doi.org/10.1017/9781108348973

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[16] Yang He and Lingao Xiao. 2023. Structured Pruning for Deep Convolutional Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023), 1–20. https://doi.org/10.1109/TPAMI.2023.3334614

[17] Nathan Hubens, Matei Mancas, Marc Decombas, Marius Preda, Titus Zaharia, Bernard Gosselin, and Thierry Dutoit. 2020. An Experimental Study of the Impact of Pre-Training on the Pruning of a Convolutional Neural Network. In *Proceedings of the 3rd International Conference on Applications of Intelligent Systems (APPIS 2020)*. ACM. https://doi.org/10.1145/3378184.3378224

[18] Foroozan Karimzadeh, Ningyuan Cao, Brian Crafton, Justin Romberg, and Arijit Raychowdhury. 2020. Hardware-Aware Pruning of DNNs using LFSR-Generated Pseudo-Random Indices. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–5. https://doi.org/10.1109/ISCAS45731.2020.9181101

[19] Brett Koonce and Brett Koonce. 2021. ResNet 50. *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization* (2021), 63–72.

[20] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. 2021. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing* 461 (2021), 370–403.

[21] Roman Liessner., Jakob Schmitt., Ansgar Dietermann., and Bernard Bäker. 2019. Hyperparameter Optimization for Deep Reinforcement Learning in Vehicle Energy Management. In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*. INSTICC, SciTePress, 134–144. https://doi.org/10.5220/0007364701340144

[22] Ning Liu. 2020. *Real-World Applicable Deep Learning Techniques: From Efficient Modeling to Automated Model Optimization*. Ph. D. Dissertation. Northeastern University.

[23] Peng Liu. 2023. *Bayesian Optimization Overview*. Apress, Berkeley, CA, 1–32. https://doi.org/10.1007/978-1-4842-9063-7_1

[24] Yogesh Malhotra. 2018. AI, machine learning & deep learning risk management & controls: beyond deep learning and generative adversarial networks: model risk management in AI, machine learning & deep learning: princeton presentations in AI-ML risk management & control systems (presentation slides). In *Machine Learning & Deep Learning: Princeton Presentations in AI-ML Risk Management & Control Systems (Presentation Slides)(April 21, 2018). Princeton Presentations in AI & Machine Learning Risk Management & Control Systems, 2018 Princeton Fintech & Quant Conference, Princeton University*.

[25] Eduardo F Morales, Rafael Murrieta-Cid, Israel Becerra, and Marco A Esquivel-Basaldua. 2021. A survey on deep learning and deep reinforcement learning in robotics with a tutorial on deep reinforcement learning. *Intelligent Service Robotics* 14, 5 (2021), 773–805.

[26] Evelyn do Amaral Oliveira. 2022. Dificuldades de matemática no 5º ano: análise dos resultados da rede estadual do RS na Prova Brasil 2021. (2022).

[27] Halit Orenbas and Wang Min. 2021. Analysing the Lottery Ticket Hypothesis on Face Recognition for Structured and Unstructured Pruning. In *2021 Innovations in Intelligent Systems and Applications Conference (ASYU)*. 1–4. https://doi.org/10.1109/ASYU52992.2021.9599030

[28] Asmaa Rassil, Hiba Chougrad, and Hamid Zouaki. 2022. Augmented Graph Neural Network with hierarchical global-based residual connections. *Neural Networks* 150 (2022), 149–166. https://doi.org/10.1016/j.neunet.2022.03.008

[29] Yasufumi Sakai, Akinori Iwakawa, Tsuguchika Tabaru, Atsuki Inoue, and Hiroshi Kawaguchi. 2022. Automatic Pruning Rate Derivation for Structured Pruning of Deep Neural Networks. In *2022 26th International Conference on Pattern Recognition (ICPR)*. 2561–2567. https://doi.org/10.1109/ICPR56361.2022.9956644

[30] Tuanjie Shao and Dongkun Shin. 2022. Structured Pruning for Deep Convolutional Neural Networks via Adaptive Sparsity Regularization. In *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*. 982–987. https://doi.org/10.1109/COMPSAC54236.2022.00151

[31] Eliott Simon. and Alexia Briassouli. 2022. Vision Transformers for Brain Tumor Classification. In *Proceedings of the 15th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2022) - BIOIMAGING*. INSTICC, SciTePress, 123–130. https://doi.org/10.5220/0010834300003123

[32] Prashant Singh and Andreas Hellander. 2018. HYPERPARAMETER OPTIMIZATION FOR APPROXIMATE BAYESIAN COMPUTATION. In *2018 Winter Simulation Conference (WSC)*. 1718–1729. https://doi.org/10.1109/WSC.2018.8632304

[33] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 6105–6114.

[34] Mengzhou Xia, Zexuan Zhong, and Danqi Chen. 2022. Structured pruning learns compact and accurate models. *arXiv preprint arXiv:2204.00408* (2022).

[35] Zhengwu Yang and Han Zhang. 2021. Comparative Analysis of Structured Pruning and Unstructured Pruning. In *International Conference on Frontier Computing*. Springer, 882–889.

[36] Zhengwu Yang and Han Zhang. 2022. Comparative Analysis of Structured Pruning and Unstructured Pruning. In *Frontier Computing*, Jason C. Hung, Neil Y. Yen, and Jia-Wei Chang (Eds.). Springer Nature Singapore, Singapore, 882–889.

[37] Muhammad Yasir, Li Chen, Amna Khatoon, Muhammad Amir Malik, Fazeel Abid, et al. 2021. Mixed script identification using automated DNN hyperparameter optimization. *Computational intelligence and neuroscience* 2021 (2021).

[38] Jiye Zeng, Tsuneo Matsunaga, Nobuko Saigusa, Tomoko Shirai, Shin-ichiro Nakaoka, and Zheng-Hong Tan. 2017. Evaluation of three machine learning models for surface ocean CO 2 mapping. *Ocean Science* 13, 2 (2017), 303–313.

[39] Quan Zhang. 2018. Convolutional Neural Networks. In *3rd International Conference on Electromechanical Control Technology and Transportation - ICECTT*. INSTICC, SciTePress, 434–439. https://doi.org/10.5220/0006972204340439

[40] Song Zhang, Lin Li, and Jiangxuan Qiao. 2023. PT-MVSNet: Overlapping Attention Multi-view Stereo Network with Transformers. In *2023 4th International Conference on Computer Vision, Image and Deep Learning (CVIDL)*. 589–592. https://doi.org/10.1109/CVIDL58838.2023.10167367

[41] Jiaqi Zhao, Ying Chen, Yufeng Zhong, Yong Zhou, Rui Yao, Lixu Zhang, and Shixiong Xia. 2023. Filter pruning based on evolutionary algorithms for person re-identification. *Multimedia Tools and Applications* (2023), 1–18.