EasyChair Preprint

№ 15851

Reducing Model Complexity for COVID-19 Classification: a Pruning-Based Approach

Sujata Shahabade and Renuka Londhe

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

February 20, 2025

I. INTRODUCTION

This Deep learning (DL) models have shown considerable promise in the medical field, particularly for the detection and classification of diseases such as COVID-19. However, the deployment of these models in real-world settings is often hindered by their high computational and memory requirements. To address this challenge, model pruning techniques, which reduce the size and complexity of neural networks by removing less important parameters, have been increasingly explored [2]. This paper investigates the application of magnitude-based pruning to a Deep Learning classification model for COVID-19 detection, as defined in a recent IEEE study.

The model [7], which leverage limited labelled data to improve performance, have demonstrated significant potential in medical image analysis [4]. In our study, we initially trained a weakly supervised learning model for the binary classification of COVID-19 cases, achieving a validation accuracy of 87%. To further enhance the model's efficiency and performance, we applied magnitude-based pruning, a technique that prunes weights with the smallest magnitudes, assuming they have less impact on the model's predictions [3].

We employed a Polynomial Decay pruning schedule to gradually increase the model's sparsity from 0% to 50% over 50 epochs. This dynamic adjustment allowed the model to adapt smoothly to the pruning process. After pruning, the model was fine-tuned, resulting in a significant improvement in validation accuracy to 95%.

Our study demonstrates the effectiveness of magnitude-based pruning in optimizing DL models for medical applications. By reducing model complexity while enhancing performance, magnitude-based pruning offers a practical solution for deploying efficient and accurate COVID-19 detection models in in resource-constrained environments.

The remainder of the manuscript is organized as follows: Section II discusses related work found in the literature. Section III discusses the datasets and methods used to implement classification of the COVID-19, Magnitude based pruning . Section IV provides discussion on the results obtained, and Section V concludes the study with a discussion on the merits and limitations of the proposed approach and future work directions.

II. LITERATURE SURVEY

Deep Learning for Medical Imaging

Deep learning (DL) has revolutionized medical imaging by enabling automated and accurate analysis of medical images. Convolutional neural networks (CNNs), a class of DL models, have been particularly successful in tasks such as disease detection, segmentation, and classification. Notably, CNNs have been employed in the detection of various diseases from chest X-ray images, including pneumonia, tuberculosis[5], and more recently, COVID-19 from chest X-ray and lung CT[6]. However, the deployment of these models in real-world clinical settings is often limited by their high computational and memory demands.

Weakly Supervised Learning in Medical Imaging

Weakly supervised learning, which leverages limited labelled data to train models, has gained traction in medical image analysis due to the scarcity and high cost of annotated medical data. Different papers has explained different weakly supervised methods for the classification of Localization of Common Thorax Diseases[7], infection detection and classification COVID-19 [8]] using weakly supervised learning and could achieved significant performance limited labelled data. This approach has shown promise in reducing the need for extensive annotation, making it more feasible for practical applications in medical imaging.

COVID-19 Detection Using DL Models

The COVID-19 pandemic has prompted extensive research into developing DL models for rapid and accurate detection of the virus from medical images. CNNs have been successfully employed to detect COVID-19 from chest X-rays and CT scans, achieving high accuracy and offering a valuable tool for aiding diagnosis [11]. However, the large size and complexity of these models present challenges for deployment in clinical settings, where computational resources may be limited.

Model Pruning Techniques

Model pruning is a widely used technique to reduce the size and complexity of DL models by removing redundant or less important weights and neurons. Pruning can be broadly categorized into structured and unstructured pruning. Structured pruning removes entire neurons or filters, whereas unstructured pruning removes individual weights. Detailed survey are found [9] on different pruning methods. Among the various pruning techniques, magnitude-based pruning has been extensively studied and applied due to its simplicity and effectiveness [10].

Magnitude-Based Pruning

Magnitude-based pruning zeroes out weights with the smallest magnitudes, under the assumption that these weights have less impact on the model's performance. This approach has been effective in significantly reducing model size without substantial loss in accuracy. [10] demonstrated that magnitude-based pruning could reduce the number of parameters in a CNN by up to 90% while maintaining its performance on tasks such as image classification. The technique involves iterative pruning and fine-tuning, which allows the model to adapt to the reduced set of weights.

Pruning Schedules

Pruning schedules are essential to the pruning process as they determine how sparsity is introduced into the model over time. Polynomial decay schedules, which gradually increase the sparsity level during training, have been shown to be effective in

allowing the model to adapt smoothly to pruning [12]. This gradual approach helps maintain model stability and performance throughout the pruning process.
By systematically reviewing these areas, this literature survey provides a comprehensive overview of the state-of-the-art techniques and their applications in the context of DL models for medical imaging and model pruning, setting the stage for the contributions of this study.

Contribution of This Study

Building on the existing research, this study applies magnitude-based pruning to a weakly supervised learning model for the binary classification of COVID-19 cases. By leveraging the Polynomial Decay pruning schedule, we aim to optimize the model for efficiency and performance. Our results demonstrate that pruning can enhance the model's accuracy from 87% to 95% while significantly reducing its size, highlighting the potential of this approach for practical deployment in resource-constrained environments.

III. MATERIALS AND METHOD

A. *Data Collection and Preprocessing*

The dataset referenced in this study is sourced from Kaggle [13]. It comprises CT images of 20 patients, including both COVID-19 positive and negative cases, in nii format. The CT scans cover both the left and right lungs, and they are labelled by radiologists and verified by experts. The dataset is organized into four folders: ct_scans/: Contains lung CT scans,infection_mask/: Includes infection masks for each CT slice. Masks are black for COVID-19 negative cases, lung_mask/: Contains lung masks for each CT scan, created manually by radiologists, lung_and_infection_mask/: Features images with lung masks superimposed on infection masks. The metadata.csv file provides the file paths for all files in these four directories.
The dataset contains a total of 3520 lung CT slices. Some slices lacked infection masks and were removed as part of the data cleaning process. Consequently, the number of CT slices after preprocessing is 2106.

The CLAHE (Contrast Limited Adaptive Histogram Equalization) [14] enhancement technique is applied to all images to enhance contrast and improve image quality. Additionally, the lung mask in the dataset is used to crop and obtain the region of interest (ROI) of each lung CT. This step focuses on the desired portion of the lung CT rather than processing the entire CT slice, thereby improving processing speed. SMOTE (Synthetic Minority Oversampling Technique) [16] method is also employed to overcome from data imbalance problem with dataset.

The entire dataset is split into a training set (70%) and a testing set (30%) for classification. Data Augmentation is employed since training dataset containing only 2106 CT slices and masks. To add more CT slices and increase the size of our dataset, data augmentation is performed on the :such as the left-right flip and up-down flip of the original images and original masks. The training process involves 50 epochs, and the Adam (Adaptive Moment Estimation) optimizer is used to update the model parameters along with advanced Cosine Annealing method to choose appropriate learning rate during the training process [15].

B. *Model Definition and Training*

Base Model: The DL classification model defined in the IEEE paper [7] is implemented for binary classification of COVID-19 cases and trained on a labeled dataset containing lung CT images. The model is trained for 50 epochs with Adam optimizer and Leaky ReLU is an activation function, and achieved validation accuracy of 87%.
Pruning pipeline steps used in this work is as showed in Figure 1. The Steps in the process are:1) The model defined is jointly pruned and trained on the given dataset using Magnitude-based pruning to reduce the model size by zeroing out weights with the smallest magnitudes. Pruning Schedule: A Polynomial Decay [12] schedule was used to dynamically adjust the sparsity from 0% to 50% over 50 epochs. The threshold is determined by sorting the absolute values of the weights and finding the value below which the target percentage of weights falls. Tensor Flow Model Optimization Toolkit: Handles the pruning process using a defined pruning schedule, updating masks during training, and stripping pruning wrappers after training. Weights below the threshold are masked (set to zero), and the pruning mask is updated accordingly. This approach ensures that less important weights are pruned, reducing the model size 3) Pruned model is saved for further processing. 4)load pruned model, fine tune the model from scratch using same learning rate, batch size 32 with RMSProp as optimizer with learning rate 0.0001at the start and updated using cosine Annearling schedule.

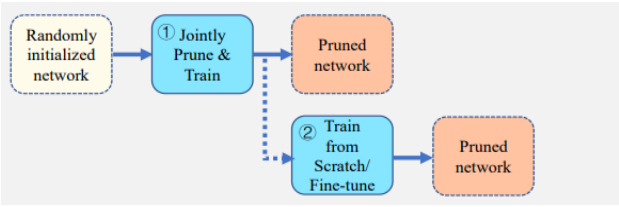
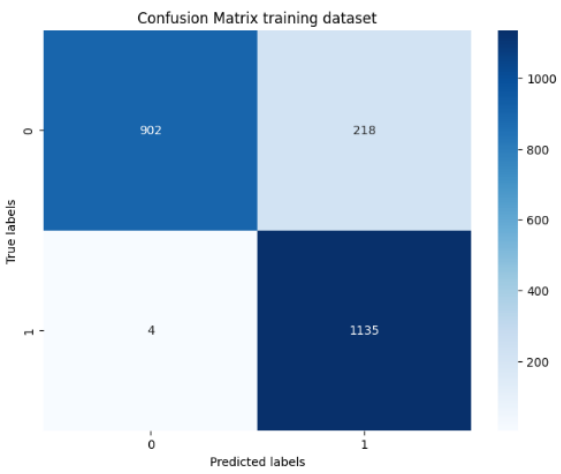


Figure 1: Pruning pipelining employed [9]

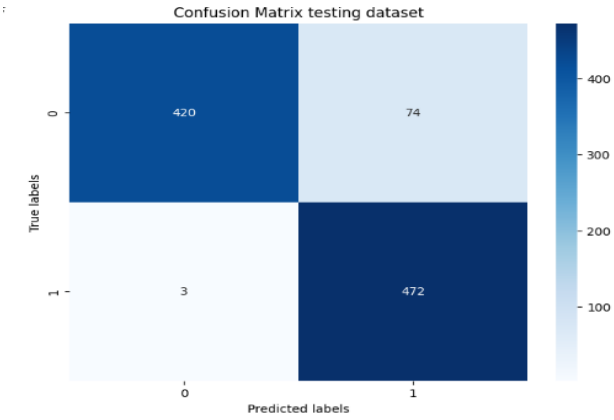
The model is then trained to restore and potentially enhance performance resulting in training Accuracy of 0.90, testing Accuracy of 0.92 and Overall Accuracy 0.91.

Confusion matrix in Figure 2 shows the performance of the model before and after training and testing dataset. The analysis of the bar graph plotted in the Figure 3 and Figure 4 is as shown below:

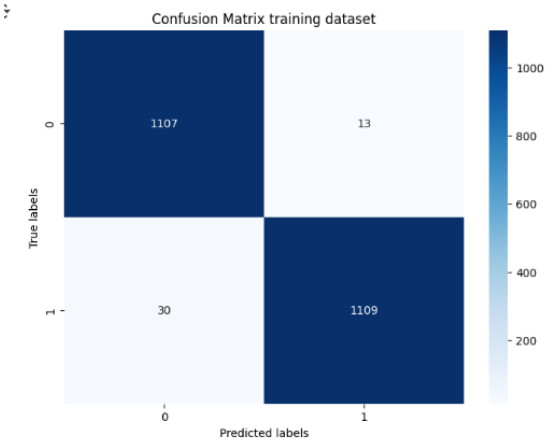
after pruning



dataset after pruning



before pruning



before pruning

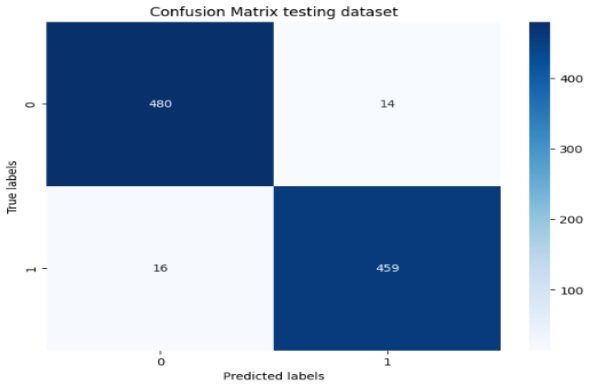


Figure 2: confusion Matrix analysis before and after pruning

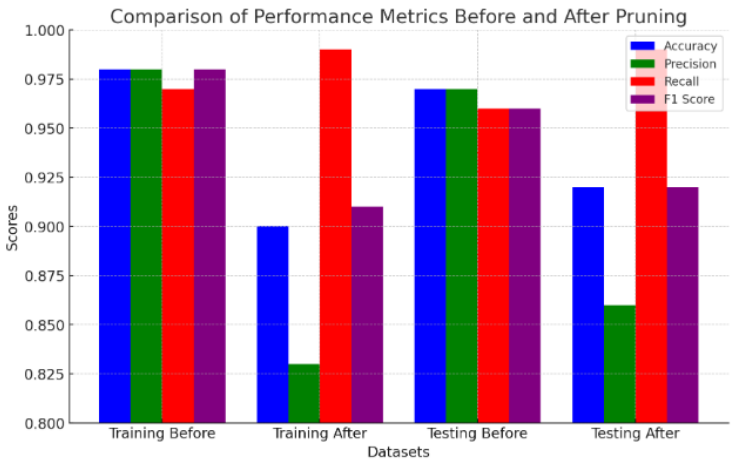


Figure 3: Comparison of performance metrics before and after pruning

Impact of Pruning on Training Performance: Accuracy, Precision, and F1 Score dropped significantly after pruning. The drop in Accuracy and Precision suggests that while the pruned model captures more true positives, it may also misclassify more

negative cases. Recall, however, increased slightly from 0.97 to 0.99, indicating that the pruned model became more sensitive in identifying positive cases.

Impact of Pruning on Testing Performance: Accuracy and Precision decreased but not as drastically as in the training set. Also Recall remained high (0.99) even after pruning, similar to the training results. The F1 Score dropped slightly from 0.96 to 0.92, which shows a trade-off between Precision and Recall.

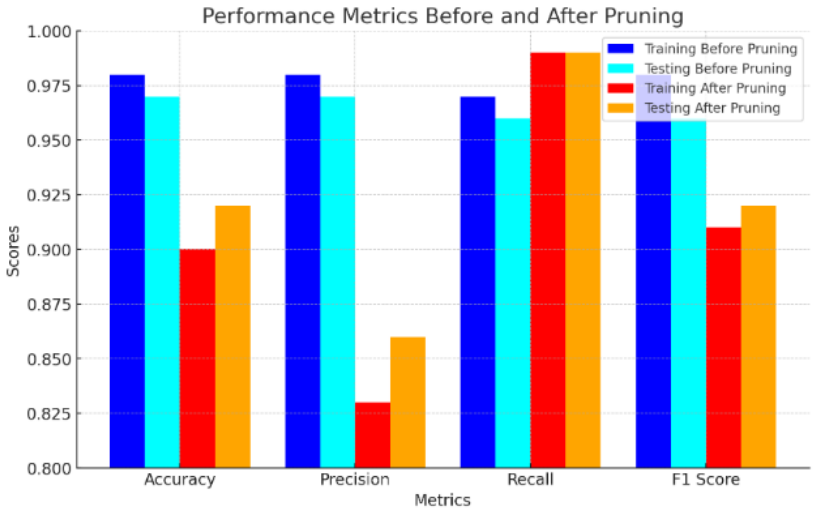


Figure 4: Comparison of performance of individual metric before and after pruning

We further analyzed each performance metric individually in details and its impact on the learning process of the model.

Accuracy: As per the bar graph showed in Figure 3, before pruning accuracy on the training dataset is 0.98 and testing dataset it is 0.97 which is high enough as compare to the training accuracy after pruning 0.90 and testing is 0.92. Reduction in the accuracy after pruning indicated that classification ability of the model is reduced due to removal of certain network parameters.

Precision: Value of the precision before pruning on training dataset was 0.98 and testing it was 0.97 and after pruning there is significant drop of 0.83 on Training and 0.86 on testing dataset. As a result prune**d** model produces more false positives, making it less confident in its positive predictions.

Recall: Before pruning recall metrics has value 0.97 on Training and 0.96 on testing dataset whereas after pruning recall value improved to 0.99 on both training and testing dataset. The increase in recall suggests that the pruned model is better at capturing true positive cases, even at the cost of lower precision. This is particularly useful for applications where missing a positive case (e.g., a disease diagnosis) is costly.

F1 Score: Before pruning: 0.98 in Training and 0.96 Testing**.** Reading after pruning: 0.91 for Training and 0.92 Testing**.** The F1 Score drop reflects the trade-off between precision and recall, indicating that while recall remains strong, the overall balance of the model's predictions is slightly affected.

Table 1 demonstrate comparison study of the model size and total number of parameters before and after pruning DL model.

1. Model Size Comparison

Metric	Before Pruning	After Pruning	Reduction
Total Parameters	7,854,454 (29.96 MB)	3,931,402 (15.00 MB)	**49.94%**
Trainable Parameters	3,928,458 (14.99 MB)	3,928,458 (14.99 MB)	**No Change**
Non-trainable Parameters	3,925,996 (14.98 MB)	2,944 (11.50 KB)	**99.92%**

Table 1: Comparison of DL model size and parameters before and after pruning

It was observed that total number of parameters decreased by 49.94%, reducing the overall model size from 29.96 MB to 15.00 MB. This indicates an efficient pruning strategy that significantly reduces storage requirements. Also the number of trainable parameters remains 3,928,458 and non-trainable parameters dropped from 3,925,996 to 2,944, a 99.92% reduction. This means pruning mainly affected non-trainable layers such as e.g., Batch Norm layers, biases, or frozen layers and hence the core learning capacity of the model is preserved.

IV. Result and Discussion

This paper implemented a classification model for the detection of COVID-19 to achieve a good accuracy of 98% in the training dataset and 97% of accuracy on the testing dataset before pruning. Overall Accuracy of the model before pruning 0.975 (97.5%). Implemented model then applied for the pruning using magnitude based pruning method aiming to reduce the size of the model without affecting its performance. After pruning result obtained is: Training Accuracy: 0.90, Testing Accuracy: 0.92 and Overall Accuracy 0.91 (91.0%).

After pruning, Recall improves, making the pruned model more sensitive to detecting positive cases. The Overall performance says that, the pruned model generalizes well, as seen from the similar accuracy levels in both training and testing after pruning. But as we know Pruning reduces model size and complexity but leads to a decrease in accuracy and precision. More fine-tuning e.g., using regularization or fine-pruning may help recover some of the lost precision while maintaining high recall. As showed in the Figure 2, the pruned and fine-tuned model has achieved validation accuracy of 95%. Pruning led to a reduction in overall model performance, especially in Precision, meaning the model is making more false positive predictions. However, since Recall remains high, the pruned model is still effective at identifying positive cases. The pruned model generalizes well, as the testing accuracy (0.92) is close to training accuracy (0.90), indicating no major overfitting.

With regard to the model size and parameters the model became nearly 50% smaller, making it more efficient for deployment while maintaining its learnable weights. Reduced non-trainable parameters indicate effective optimization, possibly improving inference speed. To further optimize pruned DL model, methods like quantization and fine-tuning can be employed**.**

V. Conclusion

This study explored impact of magnitude-based pruning on base model [7] which is a deep learning model for the binary classification of COVID-19 cases.. The pruned model, fine-tuned over 50 epochs, the findings demonstrate that even though there is an accuracy drop from 97.5% to 91.0% after pruning, the pruned model remains effective, especially since recall improved. The performance trade-off due to pruning, which helps reduce model complexity but slightly impacts its classification accuracy. Potential Trade-off demonstrated that the model may experience a slight accuracy drop due to parameter reduction, requiring fine-tuning.

By demonstrating the practical benefits of magnitude-based pruning, this study contributes to the ongoing efforts to develop efficient and accurate DL models for medical diagnostics, particularly in the context of the global pandemic. Significant reduced in the size of the model help to run model faster than the original model.

A. Limitations

Despite the promising results, several limitations of this study should be acknowledged:

Dataset Dependency: The model was trained and evaluated on the COVID-19 dataset modified for binary classification. The generalizability of the results to other datasets and real-world clinical data remains to be validated.

Model Architecture Specificity: The pruning techniques and schedules applied were tailored to the specific architecture of the weakly supervised learning model. Different architectures might require customized pruning strategies and may respond differently to pruning.

Structured pruning, which removes entire neurons or filters, could be more beneficial for certain hardware accelerators.

Initial Performance Drop: The initial application of pruning led to a performance drop, which was mitigated through fine-tuning. However, this initial degradation could be problematic in scenarios where interim model performance is critical.

Computational Overhead: The process of evaluating and pruning weights introduces additional computational overhead during training, which could extend the training time, particularly for larger models and datasets.

B. Future Recommendations

To address these limitations and further enhance the applicability of magnitude-based pruning, the following recommendations are proposed:

To improve accuracy after pruning, consider the following techniques: Fine-Tuning [17] the Pruned Model helps the remaining parameters, adjust and improve performance e.g. Reduce learning rate (e.g., lr=0.0001) and retrain the model for a few more epochs, Use Early Stopping to prevent overfitting while fine-tuning. Adding a small dropout [18] (e.g., 0.2-0.3) and data augmentation [19] can improve generalization. Instead of pruning a fixed percentage, structured pruning is better method to remove less important filters while keeping critical ones. Also try dynamic pruning [20], where less important neurons are removed during training rather than all at once.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] LeCun, Y., Denker, J. S., & Solla, S. A. (1990). Optimal Brain Damage. In Advances in Neural Information Processing Systems (pp. 598-605)..

[3] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. M. (2020). ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2097-2106)..

[4] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & van der Laak, J. A. (2017). A survey on deep learning in medical image analysis. In *Medical image analysis* (Vol. 42, pp. 60-88)..

[5] Shahabade, Sujata, and Renuka Londhe. "Advancement of Deep Learning and Its Substantial Impact on the Diagnosis of COVID-19 Cases." International Conference on Computing in Engineering & Technology. Singapore: Springer Nature Singapore, 2022..

[6] Rajaraman S, Siegelman J, Alderson PO, Folio LS, Folio LR, Antani SK. Iteratively Pruned Deep Learning Ensembles for COVID-19 Detection in Chest X-rays. IEEE Access. 2020;8:115041-115050. doi: 10.1109/access.2020.3003810. Epub 2020 Jun 19. PMID:

[7] Hu, S., et al.: Weakly supervised deep learning for COVID-19 infection detection and classification from CT images. IEEE Access 8, 118869–118883 (2020).

[8] Hu, S., et al.: Weakly supervised deep learning for COVID-19 infection detection and classification from CT images. IEEE Access 8, 118869–118883 (2020).

[9] Cheng, Hongrong, Miao Zhang, and Javen Qinfeng Shi. "A survey on deep neural network pruning-taxonomy, comparison, analysis, and recommendations." arXiv preprint arXiv:2308.06767 (2023).

[10] Han, S., Pool, J., Tran, J., & Dally, W. J. (2015). Learning both Weights and Connections for Efficient Neural Networks. In *Advances in Neural Information Processing Systems* (pp. 1135-1143).

[11] Zhou, T., Lu, H., & Yang, Z. (2020). The ensemble deep learning model for novel COVID-19 on CT images. In *IEEE Access* (Vol. 8, pp. 110850-110858).

[12] Zhu, M., & Gupta, S. (2017). To prune, or not to prune: Exploring the efficacy of pruning for model compression. In *arXiv preprint arXiv:1710.01878*.

[13] https://www.kaggle.com/datasets/andrewmvd/covid19-ct-scans?select=metadata.csv.

[14] Salem, N., Malik, H., & Shams, A,Medical image enhancement based on histogram algorithms. ProcediaComputer Science, 163, 300-311(2019).

[15] Ilya Loshchilov, Frank Hutter SGDR: Stochastic Gradient Descent with Warm Restarts, 2017, arXiv,https://doi.org/10.48550/arXiv.1608.03983.

[16] N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” Journal of artificial intelligence research, 321-357, 2002.

[17] Hubens, N., Mancas, M., Gosselin, B., Preda, M., & Zaharia, T. (2021). An Experimental Study of the Impact of Pre-training on the Pruning of a Convolutional Neural Network. arXiv preprint arXiv:2112.08227.

[18] Krogh, A., & Hertz, J. A. (1992). A Simple Weight Decay Can Improve Generalization. In Advances in Neural Information Processing Systems (pp. 950-957).

[19] Shorten, C., & Khoshgoftaar, T. M. (2019). A Survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1), 60.

[20] Liu, Z., Sun, M., Zhou, T., Huang, G., & Darrell, T. (2017). Rethinking the Value of Network Pruning. arXiv preprint arXiv:1810.05270.