Leveraging Data-Free Pruning through Gradient Analysis towards Lightweight Maize Disease Detection

Carlos PADEIRO[y], Tse-Wei CHEN[yy], Takahiro KOMAMIZU[yyy], and Ichiro IDE[yyyy]

y Nagoya University, Graduate School of Informatics, Department of Intelligent Systems
Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8601 Japan

E-mail: ypadeiroc@cs.is.i.nagoya-u.ac.jp, yytwchen@ieee.org, yyytaka-coma@acm.org, yyyyide@i.nagoya-u.ac.jp

Abstract The traditional method of diagnosing crop diseases by expert visual observations is expensive, time-consuming, and prone to errors. Convolutional Neural Network (CNN) models o□er promising alternatives, but are not easily accessible to farmers in rural areas due to high resource requirements and reliance on Internet connectivity. Lightweight models operating on resource-limited devices without Internet access are crucial to address this gap, where pruning is a promising approach. This report introduces a novel pruning method that uses gradient analysis. By examining the weight channel with a low magnitude value of gradient descent, we identify and eliminate less significant weights, without compromising its accuracy. Experiments show that the proposed method improves the accuracy, and reduces the number of parameters and inference time.

Key words Channel pruning, data-free pruning, crop diseasse, image classification

1. Introduction

Crop diseases pose a formidable challenge to global food security, causing substantial production and economic losses, estimated at $220 billion annually by the United Nations Food and Agriculture Organization (FAO) [12]. These diseases not only result in an insu□cient human food supply, but also disrupt farmers' income-generating activities. Maize, one of the dominant food crops, is particularly susceptible to various diseases, despite its global yield of $1.2 billion in 2020, with a productivity of 6.0 t/ha [13]. Conventional crop disease diagnosis methods rely on visual inspection by experts, utilizing their in-depth knowledge of crop diseases, and their symptoms. This process is not only time-consuming and expensive, but also prone to human error due to subjective perception.

The advent of Deep Neural Networks (DNNs), particularly in image processing techniques, has revolutionized precision agriculture promising to reduce labor costs, and high accuracy [43]. Previous studies, such as that by Zhang et al. [54], have proposed improved CNN (Convolutional Neural Network) models for crop disease detection. However, these models are computationally expensive and require significant memory due to over-parameterization, a common characteristic of DNNs [9]. This imposes high computational and memory demands for inference, making these solutions less accessible to farmers. Moreover, to address connectivity issues in remote cultivation areas, deploying these models on resource-limited devices is essential for wider adoption.

To address these challenges, we propose a lightweight object detection DNN model specifically designed to detect crop diseases. In this report, specifically, pruning [22] is the process of simplifying neural networks by removing unimportant and redundant weights, channels, filters, or neurons. It is usually divided into three types: unstructured [14], [45], semi-structured [17], [38], [36], and struc-

vestigates the contribution intermediate channels in layer or block to image variation through gradient-based analysis.

(2) **Validation**: We present an experimental evaluation conducted on a dataset for maize leaf disease detection. Results demonstrate that the propose BlockWIPG enhances the performance of data-free pruning models and serves as a valuable complement to existing structured pruning techniques.

2. Related Work

In computer vision, object detection methods balance accuracy and e□ciency. Two-stage methods excel in accuracy but are challenging to implement because of the sampling operations in Region-Of-Interest (ROI) pooling, hindering deployment on edge devices [23]. One-stage methods are computationally e□cient but cannot achieve high accuracy in some applications [10]. In this line of researches, this section analyzes combinations of pruning techniques for enhancing resource-limited crop disease detection applications.

2.1 Crop Disease Detection

Current techniques in precision agriculture are commonly preceded by analyzing images captured by devices and sensors to detect crop diseases. The problem of crop disease detection has been addressed in numerous studies. However, most of them focused only on the problem of classifying foliar diseases via image classification mechanism, such as the method by k-means clustering and deep learning to detect orange diseases, and predicting their names from image [25]. Their architectures are designed based on GoogLeNet's [42] inception networks and AlexNet [27] for identifying and recognizing apple leaf diseases as proposed by Liu et al. [31]. Unlike these works, we apply object detection networks to detect crop diseases on plant leaves.

2.2 Pruning

Pruning simplifies neural networks by removing unnecessary weights, channels, filters, or neurons. It can be categorized into three types: unstructured pruning, semi-structured pruning, and structured pruning. Unstructured pruning [14], [45] removes individual weights and achieves high compression and accuracy on specialized hardware. Semi-structured Pruning [17], [38], [36] balances high accuracy with structural regularity. Structured pruning [35], [11], [48] removes entire filters, channels, Transformer attention heads, or layers, and does not require specialized hardware. However, it may reduce accuracy compared to vanilla models. The possible reason is that unstructured pruning targets individual weights, while structured pruning enforces simultaneous pruning across multiple layers, ensuring all removed weights are consistently unimportant. This structural coupling makes structured pruning more compatible with hardware. The proposed method uses structured pruning due to its hardware compatibility and its practical benefits in terms of latency and memory improvements. Here, pruning is formalized as:

$$\arg\min_{W_p} \; L^1 x; W_p^{\circ} \quad \text{s.t.} \quad \|W_p\|_0 < N, \tag{1}$$

where L denotes the objective function for neural network training, x is the input, W_p are pruned weights, $\|W_p\|_0$ represents the number of none-zero weights in W_p, and N is the target number of none-zero weights in W_p. We remove the weights tensor in channel pruning. Eq. 2 is applied to calculate the channel to be kept.

$$c_{\text{op}} = c_o^{\,1} 1 \quad s^{\circ}, \quad c_{\text{ip}} = c_i^{\,1} 1 \quad s^{\circ}, \tag{2}$$

where c_{op} and c_{ip} denote the channel output and input sizes after pruning, c_o and c_i denote the original channel output and input sizes, and s is the sparsity pruning ratio ranging between 0 and 1.

a) Iterative Pruning vs. One-shot Pruning

Lin et al. [30] compare one-shot and iterative pruning methods through the lens of stochastic gradients. They find that iterative pruning computes a stochastic gradient at the pruned model, optimizing it for the compressed state. On the contrary, one-shot pruning computes the gradient at the original weights, aiming for the optimal dense model. The study by Santacroce et al. [44] supports iterative pruning theoretically, suggesting that the surrogate loss landscape, derived from a Taylor expansion, is only reliable locally, making it unsuitable for significant weight changes. We apply iterative pruning since it relies on gradient information to measure channel importance.

b) Pruning on Initialized vs. Pre-trained Weight

Recent studies [39], [55] indicate that pruning after pre-training [15], [17], [39] yields superior results compared to that before training (initialized weights) [18], [28]. Qiu and Suda [40] suggest that training weights can be divided into two aspects: the locations of weights and their exact values, with the locations holding most of the training information. This finding is also supported by Cheng et al. [6], which claim that pre-training pruning is crucial for achieving a pruned model with enhanced performance. Accordingly we adopt pre-training pruning.

2.3 Channel Importance Estimation for Pruning

The Channel Importance (CI) focus on evaluating the significance of channels in neural networks and then pruning less important channels according to predefined importance criteria. Various methods have been developed to establish these criteria, utilizing factors such as weight magnitude, activations, gradients, and others, as demonstrated in prior studies [19], [20], [52]. Li et al. [29] assess channel importance by computing the sum of absolute weight values. Liu et al. [33] minimize the reconstruction error of feature maps between the pruned and original pre-trained models. He et al. [21] and Tang et al. [46] highlight that the significance of channels is strongly influenced by the input data and suggest generating distinct subnetworks for each input instance. During inference, only channels with saliency values exceeding a predefined threshold are computed, while redundant features are omitted to enhance e□ciency. However, assessing the importance of individual channels in neural networks is challenging due to the complexity of operations such as activation functions and batch normalization. One potential approach is to evaluate a channel's significance based on the increase of the final loss caused by its removal. However, this method becomes impractical in very deep networks. In such cases, the shallow layers often exhibit limited discriminative capacity, as the long propagation path diminishes their influence on the final output.

Unlike previous studies, the proposed method estimates channel importance by considering the following key aspects: (1) Use of a deep model to e□ectively analyze the importance of channels within intermediate layers or blocks, (2) Division and computation of intermediate gradient analysis for each layer or block, and the investigation of contribution of intermediate channels to image variation through gradient-based analysis, and (3) Evaluation of channel importance by measuring changes in the output of intermediate layers or blocks when small weight perturbations are introduced.

3. Methodology: BlockWIPG

The core idea of the proposed method is to analyze critical information embedded in the gradients of each channel and leverage this information to estimate the channel importance. We introduce a novel approach, Block Weight and Image Perturbation-induced Gradients (BlockWIPG), to determine the channels to prune within specific layers or blocks of the model. This methodology is inspired by the works by Chung et al. [7] and Ilhan at al. [24].

[18] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and hu□man coding. *Computing Research Repository arXiv Preprints*, arXiv:1510.00149, 2016.

[19] S. Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for e□cient neural networks. *Advances in Neural Information Processing Systems*, 28:1135–1143, 2015.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[21] Y. He, X. Zhang, and J. Sun. Channel pruning for accelerating very deep neural networks. In *Proc. 16th IEEE International Conference on Computer Vision*, pages 1398–1406, 2017.

[22] H. Hu, R. Peng, Y. Tai, and C. Tang. Network trimming: A data-driven neuron pruning approach towards e□cient deep architectures. *Computing Reserch Repository arXiv Preprints*, arXiv:1608.08710, 2016.

[23] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, and S. Guadarrama. Speed/accuracy trade-o□s for modern convolutional object detectors. In *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 7310–7311, 2017.

[24] F. Ilhan, G. Su, S. F. Tekin, T. Huang, S. Hu, and L. Liu. Resource-e□cient transformer pruning for finetuning of large models. In *Proc. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16206–16215, 2024.

[25] K. Kaur and M. Kaur. Prediction of plant disease from weather forecasting using data mining. *International Journal on Future Revolution in Computer Science Communication Engineering*, 4(4):685–688, 2018.

[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *Computing Research Repository arXiv Preprints*, arXiv:1412.6980, 2017.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1106–1114, 2012.

[28] N. Lee, T. Ajanthan, and P. H. S. Torr. SNIP: Single-shot network pruning based on connection sensitivity. *Computing Research Repository arXiv Preprints*, arXiv:1810.02340, 2019.

[29] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for e□cient convnets. *Computing Research Repository arXiv Preprints*, arXiv:1608.08710, 2017.

[30] T. Lin, S. U. Stich, L. Barba, D. Dmitriev, and M. Jaggi. Dynamic model pruning with feedback. *Computing Research Repository arXiv Preprints*, arXiv:2006.07253, 2020.

[31] B. Liu and Y. Zhang. Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, 10(1):11, 2018.

[32] L. Liu, S. Zhang, Z. Kuang, A. Zhou, J.-H. Xue, X. Wang, Y. Chen, W. Yang, Q. Liao, and W. Zhang. Group Fisher pruning for practical network compression. In *Proc. 38th International Conference on Machine Learning*, pages 7021–7032, 2021.

[33] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang. Learning e□cient convolutional networks through network slimming. In *Proc. 16th IEEE International Conference on Computer Vision*, pages 2755–2763, 2017.

[34] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A ConvNet for the 2020s. In *Proc. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[35] X. Ma, G. Fang, and X. Wang. LLM-pruner: On the structural pruning of large language models. *Advances in Neural Information Processing Systems*, 36:21702–21720, 2023.

[36] X. Ma, W. Niu, T. Zhang, S. Liu, S. Lin, H. Li, W. Wen, X. Chen, J. Tang, K. Ma, B. Ren, and Y. Wang. An image enhancing pattern-based sparsity for real-time inference on mobile devices. In *Proc. 16th European Conference on Computer Vision*, pages 629–645, 2020.

[37] X. Men, M. Xu, Q. Zhang, B. Wang, H. Lin, Y. Lu, X. Han, and W. Chen. ShortGPT: Layers in large language models are more redundant than you expect. *Computing Research Repository arXiv Preprints*, arXiv:2403.03853, 2024.

[38] F. Meng, H. Cheng, K. Li, H. Luo, X. Guo, G. Lu, and X. Sun. Pruning filter in filter. *Advances in Neural Information Processing Systems*, 33:17629–17640, 2020.

[39] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz. Importance estimation for neural network pruning. *Proc. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11256–11264, 2019.

[40] Y. Qiu and R. Suda. Train-by-reconnect: Decoupling locations of weights from their values. *Advances in Neural Information Processing Systems*, 33:20952–20964, 2020.

[41] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28:91–99, 2015.

[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[43] M. H. Saleem, J. Potgieter, and K. M. Arif. Plant disease detection and classification by deep learning. *Plants*, 8(11):468–479, 2019.

[44] M. Santacroce, Z. Wen, Y. Shen, and Y. Li. What matters in the structured pruning of generative language models? *Computing Research Repository arXiv Preprints*, arXiv:2302.03773, 2023.

[45] H. Tanaka, D. Kunin, D. L. Yamins, and S. Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in Neural Information Processing Systems*, 33:6377–6389, 2020.

[46] Y. Tang, Y. Wang, Y. Xu, Y. Deng, C. Xu, D. Tao, and C. Xu. Manifold regularized dynamic network pruning. In *Proc. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5018–5028, 2021.

[47] L. Theis, I. Korshunova, A. Tejani, and F. Huszár. Faster gaze prediction with dense networks and fisher pruning. *Computing Research Repository arXiv Preprints*, arXiv:1801.05787, 2018.

[48] H. Wang and Y. Fu. Trainability preserving neural pruning. In *Proc. 11th International Conference on Learning Representations*, 2023.

[49] T. Wiesner-Hanks and M. Brahimi. Image set for deep learning: Field images of maize annotated with disease symptoms, 2019. `https://osf.io/p67rz/` (Visited: 04-July-2023).

[50] M. Worchel and M. Alexa. Di□erentiable shadow mapping for efficient inverse graphics. In *Proc. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 142–153, 2023.

[51] X. Wu, S. Gao, Z. Zhang, Z. Li, R. Bao, Y. Zhang, X. Wang, and H. Huang. Auto-train-once: Controller network guided automatic network pruning from scratch. In *Proc. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16163–16173, 2024.

[52] N. Yang, Y. Jang, H. Lee, S. Jeong, and K. Jung. Task-specific compression for multi-task language models using attribution-based pruning. In *Proc. 61st Findings of the Association for Computational Linguistics*, pages 594–604, 2023.

[53] Z. You, K. Yan, J. Ye, M. Ma, and P. Wang. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. *Computing Research Repository arXiv Preprints*, arXiv:1909.08174, 2019.

[54] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang. Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access*, 6:30370–30377, 2018.

[55] Y. Zhang, Y. Yao, P. Ram, P. Zhao, T. Chen, M. Hong, Y. Wang, and S. Liu. Advancing model pruning via bi-level optimization. *Advances in Neural Information Processing Systems*, 35:18309–18326, 2022.