Low Parameters UNet for Energy-Efficient Cloud Detection

Chun-Fu Chen
Dept. of Elec. and Comp. Engineering
NTUST
Taipei, Taiwan (R.O.C)
ORCID: 0009-0009-9100-6005

Pei-Jun Lee
Dept. of Elec. and Comp. Engineering
NTUST
Taipei, Taiwan (R.O.C)
ORCID: 0000-0003-2010-0853

Trong-An Bui
Institute of Aerospace and System Engineering
NTUT
Taipei, Taiwan (R.O.C)
ORCID: 0000-0001-7660-4060

Abstract[1]—**Applying cloud detection algorithms to edge devices using semantic segmentation poses significant challenges due to the high computational complexity and power consumption. This paper proposes a parameter reduction method that combines linear channel pruning and adaptive pruning techniques to effectively compress the large UNet model. Linear channel pruning efficiently reduces model redundancy, while adaptive pruning further enhances the model's sparsity. This approach not only maintains performance comparable to the original UNet in terms of Intersection over Union (IoU) and other key metrics, but also significantly reduces the model's parameter count, thereby lowering DRAM access power consumption. Experimental results show that the first and second stages of parameter reduction decreased the model's parameter count by 98.5% and 70.3%, respectively, while reducing memory access by 67.5% and 65%, respectively.**

Keywords—Cloud detection, Low parameter, remote sensing.

I. Introduction

Cloud detection on satellites has rapidly developed, with only meaningful and useful data being transmitted to the end-user on Earth with low transition bandwidth. Compared to traditional methods, Neural network (NN)-based segmentation networks can effectively differentiate cloud-similar white backgrounds .Among them, the UNet[1] model strikes a better balance between computational complexity and accuracy compared to FCN[2] and SegNet[3], making it the chosen segmentation algorithm for this task. Implementing the cloud segmentation model on nano-satellites (CubeSAT missions) is highly challenging. The main difficulty lies in UNet's high computational complexity and large number of parameters, which not only increases processing time but also leads to frequent memory access, significantly raising power consumption. As mentioned in [4] , the power consumption of accessing memory accounts for a large proportion of total power consumption in edge devices. This paper proposes an on satellite cloud segmentation system designed to process the model with minimal resource consumption. The proposed model includes two-stage **parameter reduction method**: linear channel pruning and adaptive pruning techniques. These methods aim to streamline the model and improve energy efficiency on the satellite payload.

II. Proposed Algorithm

This study aims to reduce the model's complexity and redundancy, making it suitable for on satellite. Figure 1 shows the proposed flowchart, which includes a two-stage **parameter reduction method**. Stage 1 effectively reduces the number of parameters linearly while maintaining the UNet encoder-decoder architecture. Stage 2 effectively increases the sparsity of the model while maintaining the IoU level. This paper found that when using Gaofen-1 (GF1) [5] images for cloud segmentation tasks, the model's ability to capture thin clouds is significantly better when the overall IoU is above 80 compared to when it is below 80. Therefore, this paper sets an IoU of 80 as the benchmark for model compression.

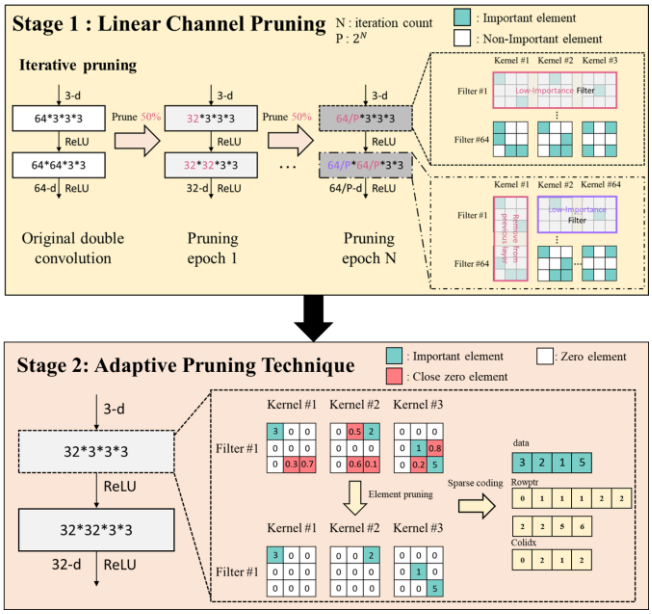

Figure 1. *Proposed reduce parameter Flowchart*

Stage 1 : Linear Channel Pruning

Deep learning models for structured segmentation tasks include encoder and decoder processes, including the Unet-based model. The proposed architecture proportionally adjusts the number of output channels in each convolutional layer to analyze and determine the most suitable model size for cloud segmentation tasks. As shown in Figure 2, the initial model to the stage1 input is a huge UNet model, with both the encoder and decoder composed of D-Conv(double convolution layers). The linear channel pruning method proposed in this paper is applied to the D-Conv in both the encoder and decoder. With each pruning iteration, half of the channels are removed, ensuring that the model retains the structural integrity of the encoder and decoder after pruning. After each iteration, the system evaluates the model's IoU performance on the cloud segmentation task. If the IoU falls below 80, the pruning process is stop; otherwise, the pruning continues iteratively.

This work was supported in part by the National Science and Technology Council of Taiwan under Contract No. NSTC 112-2218-E-006-015, NSTC 112-2622-E-011-022 and NSTC 113-2221-E-011 -102 -MY2.

Chun-Fu Chen (ORCID: 0009-0009-9100-6005) and Pei-Jun Lee (ORCID: 0000-0003-2010-0853) are with the Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan (R.O.C). Trong-An Bui (ORCID: 0000-0001-7660-4060) is with the Institute of Aerospace and System Engineering, National Taipei University of Technology, Taipei, Taiwan (R.O.C).

2025 IEEE International Conference on Consumer Electronics (ICCE) | 979-8-3315-2116-5/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICCE63647.2025.10929940

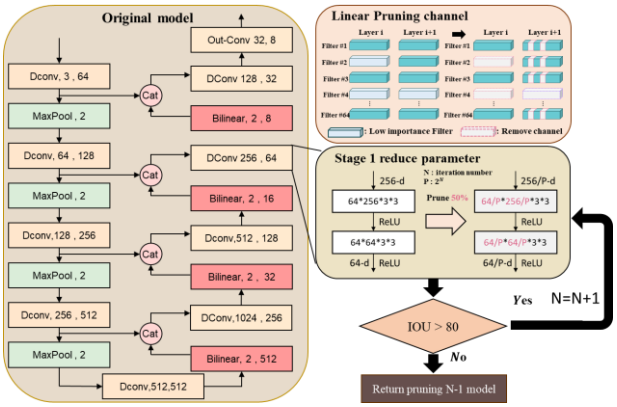

Figure 2. *Stage 1 - Linear Channel Pruning*

Stage 2: Adaptive Pruning Technique

To effectively simplify computational matrices and reduce memory access, this paper proposes an adaptive pruning technique to enhance the sparsity of each layer in UNet. The algorithm evaluates the weight sparsity of each convolutional layer by analyzing its impact on the IoU, and is applied to an unstructured pruning model. Figure 3 illustrates the proposed adaptive pruning technique. The method sets a sparsity level for each layer, assessing its sensitivity to the final IoU output, and adjusts accordingly based on a sensitivity threshold. Layers with higher sparsity contribute to improved computational efficiency, while those with lower sparsity retain more critical information. Once the sparsity level for each layer is determined, unimportant weights are locked and set to zero. The model is then fine-tuned through training to restore accuracy while maintaining high sparsity. For the sparse model generated using this method, we applied the CSR encoding proposed in [6] for compression estimation.

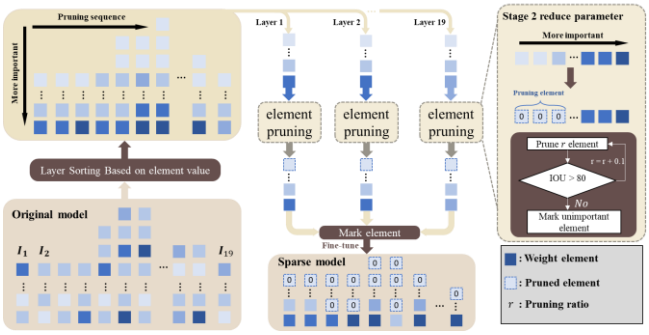

Figure 3. *Stage 2 - adaptive pruning technique*

III. Simulation Results

Table 1 presents the performance indicators inferred from the models after the two-stage parameter reduction method. As shown in the table, after the two-stage reduction, the indicators of the models are comparable to those of the original UNet model, as illustrated in Figure 5. The first stage compression reduced the number of parameters by 98.5% compared to the original UNet, and the second stage further reduced the parameters by an additional 70.3%. Moreover, as previously mentioned, the power consumption required for memory access is significant. By considering both the weights and input activations and applying sparse CSR encoding, memory usage is significantly reduced, and energy efficiency is improved by minimizing the frequency of memory access during algorithm execution. The first stage can reduce memory access by 67.5%, and after applying the second stage of encoding, access frequency is further reduced by 65%.

Table 1. The comparison of the performance on GF1 dataset

	UNet	Propose Stage 1	Propose Stage 2
PARAMETER	17.3 MB	0.27MB	0.08MB
PRECISION	93.69%	93.12%	92.37%
RECALL	92.83%	90.96%	92.75%
F1-SCORE	92.57%	91.79%	91.19%
IOU	85.89%	83.53%	82.82%
MEMORY ACCESS TIME	1x	0.325x	0.114x

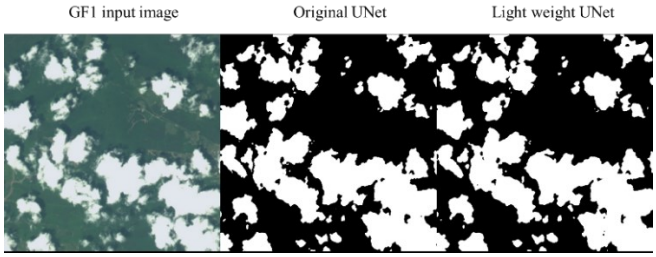

Figure 5 . *Comparison of Inference between Original UNet and Lightweight UNet .*

IV. Conclusion

This paper proposes a parameter reduction method combined with a high-efficiency algorithm to reduce memory access, which is a key factor in achieving energy savings. The method effectively reduces the number of model parameters in two steps. This enables the large UNet model to run on edge devices such as satellites, achieving high-efficiency Edge AI. As a result, satellites can directly perform AI functions on the edge device, using these functions to determine whether or not to download images.

References

[1] M. Yin, P. Wang, C. Ni, and W. Hao, "Cloud and snow detection of remote sensing images based on improved Unet3+," *Scientific Reports 2022 12:1*, vol. 12, no. 1, pp. 1–13, Aug. 2022

[2] S. Acer, O. Selvitopi, and C. Aykanat, "Improving performance of sparse matrix dense matrix multiplication on large-scale parallel systems," *Parallel Comput*, vol. 59, pp. 71–96, Nov. 2016

[3] D. Chai, S. Newsam, H. K. Zhang, Y. Qiu, and J. Huang, "Cloud and cloud shadow detection in Landsat imagery based on deep convolutional neural networks," *Remote Sens Environ*, vol. 225, pp. 307–316, May 2019

[4] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017,

[5] "GF1_WHU: GF-1 Cloud and Cloud Shadow Cover Validation Dataset – SENDIMAGE." Accessed: Sep. 03, 2024.

[6] X. Feng, H. Jin, R. Zheng, K. Hu, J. Zeng, and Z. Shao, "Optimization of sparse matrix-vector multiplication with variant CSR on GPUs," *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS*, pp. 165–172, 2011