ISSN 2959-6157

# Optimizing Lightweight Convolutional Neural Networks for Hyperspectral Image Fusion

**Weiheng Hong**

Southwest Petroleum University
Email:1923800837@qq.com

**Abstract:**

This paper addresses the challenges associated with processing complex scenes and high-precision images on resource-constrained devices, particularly within the domain of hyperspectral image processing and super-resolution reconstruction techniques. We present an optimized model for existing hyperspectral image fusion models, leveraging network lightweight and channel pruning. Our proposed model, NestFuse Small, employs NestFuse as the primary network architecture and integrates a quantization pruning module. Experimental findings demonstrate that, in comparison to the original NestFuse model, NestFuse Small's computation speed is 164.8% of the pre-optimization speed, a decrease in memory usage of 20.65%, resulting in a slight decrease in performance. This optimized model facilitates more efficient image processing on resource-constrained devices.

**Keywords:** hyperspectral image fusion, embedded devices,8bit quantization, channel pruning

## 1. Introduction

Hyperspectral imaging, renowned for its capacity to capture detailed spectral information across a broad range of wavelengths, has garnered widespread utility across diverse domains. However, the practical implementation of hyperspectral image fusion technology is beset by formidable challenges. These encompass its inherent intricacy, limited adaptability to specific image types, and the prevalence of redundant spectral information. Moreover, the computational demands and algorithmic intricacies stemming from its complex network structures curtail its practical viability, particularly in scenarios necessitating swift processing of extensive datasets and when deployed on resource-constrained devices such as embedded or mobile platforms[1].

Given the escalating volume of image data and the burgeoning emphasis on energy efficiency and emission reduction, there exists an increasingly compelling imperative to devise lightweight networks tailored for hyperspectral image processing on resource-constrained devices. This paper endeavors to address these challenges by proposing an optimized model for hyperspectral image fusion predicated on network lightweight and structural pruning. The proposed model seeks to ameliorate computational costs and memory usage while upholding exemplary performance, thereby facilitating more efficient image processing on resource-constrained devices.

## 2. Model Selection

To enable the efficient operation of hyperspectral image fusion technology on mobile devices, embedded systems, and other resource-constrained environments, this study employs lightweight convolutional neural networks to optimize the model. Lightweighting involves designing and optimizing network structures to reduce computational and storage resource consumption while maintaining high performance under limited computing resources. Compared to general networks, lightweight networks are optimized in terms of model structure, parameter quantity, computational complexity, and energy efficiency, which is crucial for reducing device battery life and energy consumption in IoT devices.

The lightweight of hyperspectral image fusion models can be achieved by incorporating mature model compression techniques into the model to reduce its parameter quantity and network complexity, thereby improving inference speed. Techniques such as pruning, quantization, and knowledge distillation are employed. Pruning involves trimming redundant parameters in the network to reduce network size, while quantization reduces the bit width of network parameters to compress the model and enable efficient computation. Knowledge distillation utilizes the knowledge of a large teacher model to supervise the training of a smaller student model. Currently, some methods design corresponding network compression modules and

apply them to network models.

For example, Rao et al. [2]utilized a CPU-FPGA heterogeneous system architecture and various network compression techniques to further conserve system computing and storage resources, enabling the deployment of hyperspectral image-matching detection networks on low-power heterogeneous systems. However, the integration of multiple networks increases system complexity, cost, and deployment difficulty to some extent. Jeon et al.[3] proposed a dual-discriminator conditional generative adversarial network to efficiently fuse infrared and visible light images of different resolutions through the adversarial interaction between the generator and two discriminators. However, this method requires the simultaneous training of two discriminator networks and one generator network, leading to increased training complexity. Xu et al. [4]addressed the problem of high computational complexity that may arise from pixel-level implementation of image fusion methods, which can introduce artifacts and/or inconsistencies. They proposed the SEDRFuse network, which combines the characteristics of autoencoders and generative adversarial networks. However, due to the collaborative training of multiple components, the network's complexity results in shortcomings in training stability and generalization capability.

Based on the aforementioned research, this paper introduces an optimized network, NestFuse small, which focuses on lightweight and channel pruning of the NestFuse network. Expanding upon the original network's utilization of convolutional layers for feature extraction to reduce model complexity, this approach further incorporates a quantization pruning model. This model quantizes the floating-point parameters of the network to 8 bits, rendering them more compatible with embedded devices, thereby reducing storage during the inference process. Additionally, it eliminates surplus network structures by reducing the number of network channels, thus lowering network complexity. The model is optimized and trained leveraging the self-learning capability of convolutional neural networks, achieving commendable compression effects while upholding performance. Experimental results demonstrate that, in comparison to the original model, NestFuse small attains a superior trade-off between performance and computational complexity, rendering it more suitable for embedded and mobile devices.

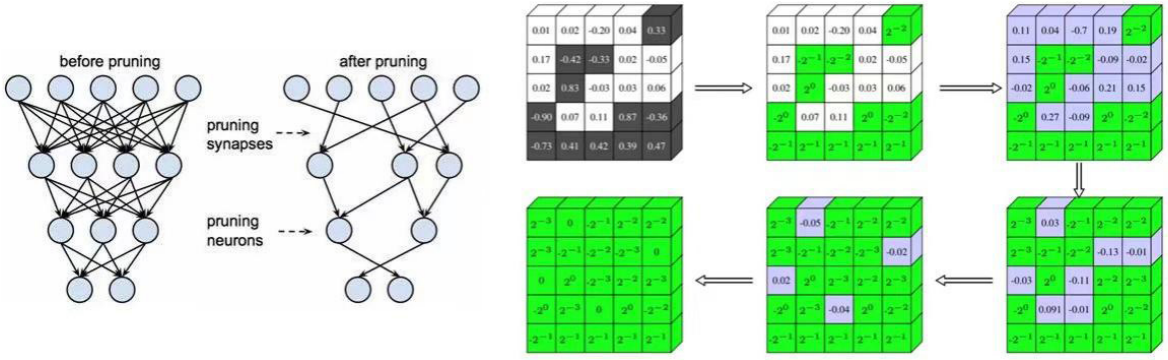


**Figure 1 Pruning and Quantization Illustration**

## 3 Preparation of the Study

This study utilized three datasets provided by the team from Microsoft Research Asia: the COCO dataset, IV images, and the TNO image fusion dataset. The COCO dataset[5] contains a large number of visual images that have been collected from a wide variety of life situations, it has detailed labeling information for each image, including object bounding boxes, instance segmentation, and keypoint locations, commonly used for various computer vision tasks, for example: object detection segmentation, pose estimation. Due to its diversity and richness, the authors employed it as the training image set, with image sizes set at 256×256.IV images consist of a collection of hyperspectral images containing different scenes and objects, widely referenced in various aspects of image visualization. The authors obtained 42 hyperspectral images with a bit depth of 8 from this collection. The TNO image fusion dataset[6] comprises multi-band images recorded by different multi-band camera systems, with each folder's reference section containing information about registration conditions and the corresponding camera systems. The authors used IV images and the TNO image fusion dataset as the test set.

Based on the image dataset used in this experiment, The authors used both pixel loss (pixel loss) and structural similarity loss (SSIM loss) loss functions in training the network. Pixel loss fully represents the pixel-level difference between the model-generated image and the real image. By minimizing this loss function, the model can learn to generate results that are closer to the real image. The pixel loss is calculated using the following formula:

**\text{Pixel Loss} = \frac{1}{N} \sum_{i=1}^{N} (I_i -**

**\hat{I}_i)^2**

SSIM loss measures the similarity between two images by comparing their brightness, contrast, and structural similarity. During training, minimizing the SSIM loss allows the model to learn to generate results that are structurally more similar to the real images. The SSIM index is calculated using the following formula:

**\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}**

The loss values are stored in a .mat file for subsequent analysis and visualization. Subsequently, the model is switched to evaluation mode (eval) and moved to the CPU for preservation. The model parameters are then saved to a file for post-training model loading and utilization. The combined consideration of these two loss functions comprehensively integrates pixel-level disparities and image structural similarity, thereby providing comprehensive guidance for the model's training to learn to generate images of higher quality and greater structural similarity.

NestFuse[7], developed by Tariq Durrani et al., integrates nested connections and spatial/channel attention models to perform fusion processing on input images through multi-level feature fusion and reconstruction. In comparison to traditional image fusion models, NestFuse demonstrates improved utilization of hierarchical feature information, focusing on crucial regions and channels within the images, Therefore image information from different modalities can be better fused in this network, so it can produce fused images with more information richness and quality. As an efficient deep learning model, it exhibits remarkable performance and holds significant potential in domains such as image recognition and computer vision. However, due to the complexity of its network, further optimization is necessary to reduce its framework size and spatial footprint for deployment on resource-constrained devices. This paper addresses this challenge by focusing on lightweight the network.

Structurally, three important parts, the encoder, fusion, and decoder, control the operation of NestFuse. The encoder initially processes the input through a convolutional layer and then conducts feature extraction using four dense blocks, ultimately yielding four feature tensors. The fusion module combines the feature tensors from the two encoders to produce a fused feature tensor. The decoder, based on the fused feature tensor, generates the final output through a series of operations such as upsampling and dense block processing. To streamline the network structure, NestFuse introduces two types of convolutional layers. The ConvLayer is a custom convolutional layer utilizing 1x1 convolutions, incorporating reflection padding, 2D convolution (conv2d), 2D dropout (dropout) operations, and a boolean variable, is_last. The autoencoder employs 3x3 convolutions with a stride and input/output channel number of 1. It encompasses multiple instances of the ConvLayer class, facilitating the construction of convolutional operations within the encoder and decoder sections.

## 4 Experimental Component

### (1) Experimental environment

The experimental setup consisted of a system powered by an Intel Core i9-13900K processor and an NVIDIA GeForce RTX 3060 GPU. Data preprocessing and post-processing tasks were conducted on this configuration. Python programming language, along with the PyTorch framework, was employed for deep learning model development and training.

The COCO dataset is used as the training set, while the IV images and TNO image fusion datasets are used as the test sets. The training was conducted using a lightweight neural network, NestFuse, with the learning rate set to 0.0001, using 4 photos per batch and 2 iterations.

### (2) Experimental Results and Analysis

In the experiment, the authors systematically manipulated the parameter λ within the loss function and employed the modified structural similarity for no-reference image($ssim_\alpha$), visual information fidelity(VIF), and entropy(En) metrics as quantitative benchmarks to ascertain the model's optimal performance.

**\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}**

**VIF = \frac{\sum_{i=1}^{N} \mu_i \cdot \phi_i}{\sigma_{\phi}^2}**

**En = -\sum_{i=1}^{N} p_i \cdot \log(p_i)**

The author conducted iterative experiments with λ values set at 1, 10, 100, and 1000, finding that the optimal network weights were obtained after exceeding 600 iterations. Comparative analysis of the three quality metrics under different λ values revealed that, while the maximum SSIMa value is lower when λ is set to 100 (1e2) compared to 1000 (1e3), the maximum values of VIF and En are optimal for the selected λ value. Considering the comprehensive fusion performance, the NestFuse network demonstrates superior performance with λ set to 100 (1e2) compared to other λ values: Input image with more information, the fused image retains more features of the source image and is visually more natural. On the other hand, the loss of the NestFuse network model is approximately 0.001, indicating commendable performance in deep feature extraction. Consequently, λ was set to 100. To better cater to resource-constrained devices for both training and inference, compared to the original NestFuse network that operates with floating-point numbers, NestFuse

small conducts computations using 8-bit binary numbers. Additionally, exploiting the inherent redundancy in spectral images, the authors halved the number of convolution channels in the intermediate layers of convolution blocks while maintaining the input and output channel numbers of feature maps unchanged, thereby further reducing the network's parameter count while ensuring certain performance metrics.

Based on these findings, the author conducted a comparative analysis of the three quality metrics of the NestFuse small model under different λ values with the original model. Additionally, they contrasted the average levels of image precision, spatial occupancy, and computational time before and after optimization. It was determined that when λ is set to 100, signifying the model's optimal performance, the optimized model's performance only exhibits a marginal decrease compared to the original model in terms of precision, with $ssim_{\alpha}$, VIF, and En decreasing by1.613 %,1.564 %, and1.311 %, the number of parameters of the pictures decreasing to 79.35% of the original network, while the computational speed of the network is significantly improved to 164.8% of the pre-optimization one. This suggests that the optimized model has traded memory release and computational speedup for a very small loss of performance.

**Table.1 The maximum values of SSIMa, VIF, and En for the NestFuse and NestFuse Small networks under varying λ values.**

| | NestFuse | | | NestFuse small | | |
|---|---|---|---|---|---|---|
| λ | $ssim_{\alpha}$ | VIF | En | $ssim_{\alpha}$ | VIF | En |
| 1 | 0.73512 | 0.74792 | 6.88973 | 0.72128 | 0.73634 | 6.74901 |
| 10 | 0.73516 | 0.74733 | 6.88281 | 0.72231 | 0.72983 | 6.72192 |
| 100 | 0.73532 | 0.75204 | 6.89421 | 0.72346 | 0.74028 | 6.80381 |
| 1000 | 0.73547 | 0.74746 | 6.88939 | 0.72463 | 0.73241 | 6.73867 |

From Table 1, it is evident that the NestFuse performs optimally when λ is set to 100. The model undergoes only a slight degradation in performance

| | NestFuse | NestFuse small |
|---|---|---|
| params | 10.931044M | 8.674205M |
| FPS | 0.75467 | 1.24386 |

From Table 2, the memory usage of the optimized network is dramatically reduced and the computational speed is significantly improved.

**5 Conclusion**

In conclusion, this paper focuses on the lightweight transformation of the NestFuse network to address the challenges of high computational workload and low efficiency in hyperspectral image fusion. The proposed optimized version, NestFuse Small, implements 8-bit quantization and also reduces the number of parameters in the model by pruning, optimizing image structure, and enhancing model computational speed. After over 600 iterations with λ set to 100, the original model's performance peaked. At this stage, the optimized network, while incurring minimal performance loss, significantly reduces image footprint and increases model computational speed, making it more conducive for deployment in resource-constrained devices such as embedded systems.

# References

[1]Yang Kai. Research on Hyperspectral Image Classification Based on Attention Networks [D]. University of Chinese Academy of Sciences (Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences), 2023. DOI: 10.27605/d.cnki.gkxgs.2023.000013.

[2]Rao Weiqiang. Research on Deep Learning Target Detection Method for Hyperspectral Image [D]. University of Chinese Academy of Sciences (Aerospace Information Research Institute, Chinese Academy of Sciences), 2022. DOI: 10.44231/d.cnki.gktxc.2022.000074.

[3]L. Jian, X. Yang, Z. Liu, G. Jeon, M. Gao, and D. Chisholm, "SEDRFuse: A Symmetric Encoder–Decoder With Residual Block Network for Infrared and Visible Image Fusion," in IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-15, 2021, Art no. 5002215, doi: 10.1109/TIM.2020.3022438.

[4]Xu H, Liang P, Yu W, et al. Learning a Generative Model for Fusing Infrared and Visible Images via Conditional Generative Adversarial Network with Dual Discriminators[C]//IJCAI. 2019: 3954-3960.

[5]Lin T Y, Maire M, Belongie S, et al. Microsoft coco: Common objects in context[C]//Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer International Publishing, 2014: 740-755.

[6]Toet A. The TNO multiband image data collection[J]. Data in

brief, 2017, 15: 249-251.
[7]Li H, Wu X J, Durrani T. NestFuse: An infrared and visible image fusion architecture based on nest connection and spatial/channel attention models[J]. IEEE Transactions on Instrumentation and Measurement, 2020, 69(12): 9645-9656