Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 33(3): 1231–1266.
DOI: 10.3934/era.2025055
Received: 27 December 2024
Revised: 01 February 2025
Accepted: 14 February 2025
Published: 03 March 2025

Research article

Multi-convolutional neural network brain image denoising study based on feature distillation learning and dense residual attention

Huimin Qu, Haiyan Xie* and Qianying Wang

School of Science, Dalian Maritime University, Dalian 116026, China

* **Correspondence:** Email: xiehy@dlmu.edu.cn.

Abstract: Medical image denoising is particularly important in brain image processing. Noise in acquisition and transmission degrades image quality and affects the reliability of diagnosis and research. Due to the complexity of the brain's structure and minor density differences, noise can increase diagnosis difficulty, so high-quality images are essential for disease detection, prognosis assessment, and treatment plan development. This paper proposes a multi-convolutional neural network based on feature distillation learning and dense residual attention to enhance the quality of brain images and improve denoising performance. The overall network structure contains four parts: a global sparse network (GSN), a dense residual attention network (DRAN), a feature distiller network (FDN), and a feature processing block (FPB). Before feeding the brain images into the denoising network model, they are preprocessed using a modified watershed algorithm based on a combination of a morphological gradient, Sobel's operator, and Canny's operator. The GSN is used to extract global features and increase the sensory field, and the DRAN efficiently extracts key features by combining improved channel attention and spatial attention mechanisms. The FDN extracts useful features through two feature distillation blocks, suppresses redundant information, and reduces computational complexity. The FPB performs feature fusion. Experimental results on brain image datasets and ground-based open datasets show that the proposed model outperforms existing methods in several metrics, and helps to improve the accuracy of brain disease diagnosis and treatment.

Keywords: watershed algorithm; dense residual attention network; feature distillation network; multi-convolutional neural network; brain image denoising

1. Introduction

Rapid advances in medical imaging technology have made brain image denoising particularly important for disease diagnosis [1]. First, the diversity of noise types, including Gaussian noise and salt-and-pepper noise, requires denoising algorithms to have strong adaptability to different noise characteristics. Second, brain images possess complex anatomical structures, such as gray matter, white matter, and cerebral vessels. Noise interferes with the identification of lesion areas and reduces diagnostic accuracy. By balancing noise suppression and feature retention, denoising technology improves image quality and preserves key pathological information to support the diagnosis and treatment of brain diseases. Positron emission tomography (PET) [2] and magnetic resonance imaging (MRI) [3] each have unique imaging characteristics. MRI is known for its high-resolution images of anatomical structures, which are effective in differentiating between normal and abnormal tissues and provide detailed information on internal structures [4]. In contrast, PET mainly provides information on the metabolic or biochemical function of tissues and organs, and is highly sensitive, enabling the detection of changes at the molecular level in the early stages of disease [2]. The brain images used in this paper are publicly available PET images. Due to the different noise characteristics of brain images, traditional denoising methods are difficult to apply directly [5]. The central challenge in image denoising is accurately distinguishing the original content from noise and artifacts. Effective denoising methods must suppress noise while preserving edge information, maintaining global contrast, preserving texture detail integrity, and avoiding introducing new artifacts. All of these factors together determine the effectiveness of denoising [6]. Pan et al. [7] first introduced the hypergraph for brain network decoupling, proposing a decoupling module with an adversarial strategy, a sparse capacity loss to enhance performance, and an analysis module to identify Alzheimer's-related neural circuits.

In recent years, researchers have proposed various techniques to solve the PET [2] denoising challenges, and the denoising techniques are classified into traditional denoising methods and deep learning-based denoising methods. Traditional denoising methods can be broadly classified into several categories, including techniques based on statistical models [8], transform domain methods [4], filtering, gradient models, multiscale techniques, non-local self-similarity (NSS) models, and non-local mean algorithms [9], have been developed for medical image denoising. Despite the progress of the existing methods in medical image processing, the complex optimization process and cumbersome parameter adjustments limit their usefulness. In summary, traditional denoising methods are limited in preserving image details and practical information.

With the advancement of deep learning, image denoising research based on convolutional neural networks has become an important direction in image processing due to its efficiency and robustness [10]. Zhang et al. [11] proposed feed-forward denoising convolutional neural networks (DnCNN), which combine deep architecture, residual learning, and batch normalization techniques to improve denoising performance and speed up training. Unlike traditional models that only target specific noise levels, DnCNN can handle the Gaussian denoising of unknown noise levels, implicitly removing the noise through residual learning and recovering clean images. The enhanced convolutional neural denoising network (ECNDNet) proposed by Tian et al. [12] combined residual learning, batch normalization, and dilation convolution to improve denoising performance and convergence speed. However, the method does not fully consider the local and overall features of the image. Zhang et al. [13] proposed a fast and flexible denoising convolutional neural network (FFDNet), an efficient and flexible denoising network, which balances performance and speed through adjustable noise level maps, can handle a wide range of noises, and outperforms BM3D on CPUs. Jifara et al. [14] designed a deep convolutional neural network for denoising a small sample of medical images using residual learning to learn the noise from the noise-containing images and then subtract the residuals to get the denoised image. Combining batch normalization and residual learning improved the model's accuracy and the training's efficiency. Tian et al. [15] proposed a novel deep convolutional neural network for image denoising that accelerates convergence by increasing the width and a batch-renormalization denoising network (BRDNet) for low-configuration hardware. BRDNet expands the sensory field using dilated convolution to reduce the computational cost and improves denoising by residual learning. Hema et al. [16] proposed a scheme combining a convolutional neural network (CNN) with an autoencoder, which is used to extract key features and suppress noise. In contrast, CNNs are used to process these features efficiently, achieving high accuracy and robustness in tumor prediction. Tian et al. [17] proposed a multi-stage image denoising CNN with the wavelet transform (MWDCNN), combining a dynamic convolutional block (DCB), two cascaded wavelet transforms and enhancement blocks (WEBs), and a residual block (RB), where the DCB dynamically adjusts the convolutional parameters to balance the performance and cost, and the WEBs suppress noise. The RB removes the redundant information and reconstructs a clear image. Li et al. [18] proposed an anchor-free object detector based on CNN, which integrates transformer-based global and local feature extraction (GLFT) to enhance the extraction of semantic information from images. Although these CNNs effectively reduce fixed-pattern noise, their performance still needs to be improved when facing more complex noise distributions, and these models lack flexibility in dealing with spatial variations and real noise.

CNN can extract features from image pixels, but they suffer from limited feature extraction, single information transfer, and gradient vanishing in denoising tasks. Convolutional operations favor local features and lead to smooth processing, making it difficult to focus on key areas and underperforming, especially in complex details and non-uniform noise. Channel attention enhances key information and feature delivery by dynamically adjusting the feature map weights, which helps to improve the denoising effect, especially in dense RBs. The essence of the spatial attention mechanism lies in highlighting the relative importance of the locations of the feature map in the spatial dimension by learning a weight map. To address the trade-off between performance and complexity, Wang et al. [19] proposed an efficient channel attention module that avoids dimensionality reduction, captures channel interactions lightly, and significantly improves CNNs' performance but neglects attention to spatially essential locations and has insufficient detail extraction capability. The feature fusion and attention network (FFA-DMRI) was proposed by Hong et al. [20] to separate noise from observed MRI, which combines feature fusion and spatial attention mechanisms to locate MRI regions of interest accurately. It performs well on the Alzheimer's disease neuroimaging initiative (ADNI) dataset. However, the method did not consider channel weight assignment and could not highlight key features. To solve the problem that deep learning image drawing algorithms are prone to losing deep features, resulting in missing texture and irrational structure, Chen et al. [21] proposed an improved image drawing network that combines a multi-scale feature module and an enhanced attention module. Multi-scale fusion through dilated convolution reduces information loss, and the attention module enhances semantic mapping to ensure that precise texture results are generated. Deng and Hu [22] proposed a progressive multi-scale denoising network (PMSDNet), which enhances noise extraction by a progressive multi-scale fusion block (PMSFB) and focuses on multi-noise pixels and high-frequency regions with a pixel attention block (PAB), but does not pay enough attention to low-frequency features and smoothing the background. Duong et al. [23] proposed a self-encoder multi-branch network combining a pyramid context module (PCM) and a residual bottleneck attention module (RBAM), where the PCM handles global information and the RBAM focuses on the key features and is computationally low. However, the articles above do not consider the issue of gradient reduction and the repeated use of multi-layered features, and the use of the attention mechanism alone has limitations in capturing multi-layered features and complex structures, especially dealing with high-frequency details.

Although the attention mechanism can focus on the key regions of an image, it still falls short in capturing multi-scale features and detailed structures when used alone and is prone to gradient vanishing. Therefore, in recent years, some studies have combined the dense RB with the attention mechanism to take advantage of the feature reuse of the dense RB, which not only enhances the extraction of multi-scale features but also effectively mitigates the problem of gradient vanishing, thus significantly improving the denoising effect. Zhang et al. [24] proposed a residual dense network (RDN) to adaptively integrate features through a continuous memory (CM) mechanism and local feature fusion (LFF). RDN achieves global feature fusion in low-resolution (LR) space and extracts globally dense features from low-quality (LQ) images through global residual learning. Gao et al. [25] proposed a hierarchical feature fusion (HFF) framework combining multi-scale dense blocks (MSDBs) and a self-attention (SA) module for extracting multi-scale features and global information to enhance the image reconstruction accuracy, which performed well in super-resolution reconstruction. Li et al. [26] proposed a parallel multi-scale feature CNN (PMSF-CNN) for image denoising using parallel multi-scale feature extraction to enhance detail capture, adaptive dense residuals to enhance information flow, and a channel attention mechanism with a spatial attention mechanism to improve noise suppression. In a no-reference image quality assessment (NR-IQA), Zhang et al. [27] proposed the attention-driven residual dense network (ADRDN), which enhances the feature extraction by a multi-scale feature extraction block (MSFEB) and a dual-pooling squeeze and excitation block (DPSEB), removes the redundant information using 1×1 convolution, and ultimately regresses the image quality scores through the fully connected layer, which enhances the efficiency and accuracy of the evaluation. However, that paper ignores the problem that dense RBs have certain limitations in the feature extraction process, and the issue of computational complexity is challenging to solve, thus affecting the denoising performance.

General CNNs have high computational complexity, and, in recent years, more and more research has been devoted to developing more efficient lightweight models. As a research hotspot for lightweight models, a feature distillation network maintains performance while reducing the computational costs by compressing the model and retaining the key information. Hui et al. [28] proposed a deep compact CNN for the efficient reconstruction of high-resolution images from low-resolution images through feature extraction blocks, information distillation blocks, and reconstruction blocks, using a small number of filters and group convolution for fast execution. Liu et al. [29] proposed a lightweight and efficient feature distillation connection (FDC) and a residual feature distillation network (RFDN) to learn more discriminative feature representations through a shallow residual block (SRB) and multiple distillation connections. RFDN strikes a better balance between performance and model complexity. Zhang et al. [30] proposed a lightweight fusion distillation network (LFDN) to remove noise through an encoder–decoder architecture and multiscale information fusion efficiently. LFDN utilizes feature distillation and attention mechanisms to filter the key information, reduce redundant parameters, and maintain the image details. Zong et al. [31] designed an asymmetric information distillation block (AIDB) with information reuse and asymmetric feature extraction capability. Parameters are reduced by horizontal and vertical extraction, and information reuse supplements the high-dimensional information. The asymmetric information distillation network (AIDN) outperformed the state-of-the-art (SOTA) model in terms of performance and complexity. Yu et al. [32] proposed a single-image super-resolution network based on the residual feature distillation mechanism, which improves the image reconstruction quality through the residual feature distillation mechanism and the spatial attention module, and reduces the model complexity and the number of parameters while maintaining high reconstruction performance. Jiang et al. [33] proposed a fast and lightweight denoising method based on deep learning by retraining the model with natural images and migrating the learning with simulated noisy ultrasound images. However, none of these studies explored performance enhancement by widening the network model.

Many researchers have found that increasing the width of the network is more effective in improving model performance than simply deepening the network. The dual residual attention network (DRANet) proposed by Wu et al. [34] captured complementary features by widening the network. Its residual attention block (RAB) and a novel hybrid dilated residual attention block (HDRAB) used jump connections to extract local features and dilation convolution to enhance contextual information capture. The flexible and effective U-shaped network (FEUNet) proposed by Wu et al. [35] adapted to different noise levels through adjustable noise level maps, enhanced denoising using U-Net architecture, and enhanced multiscale feature capture by combining with upper and lower dual convolutional networks through direct passes during intermediate processing of images. The network proposed by Thakur and Maji [36] consisted of dual convolutional networks. The first network extracted the inverted features of the input image, and the second focused on removing the original features. The dual denoising network (DudeNet) [37] used a dual convolutional neural network to extract image characteristics, achieving complementary local and global characteristics of the image and enhancing the denoising performance of the model. The underwater dual residual convolutional neural network (UDRN) [38] introduced RBs in the DudeNet and improved the loss function to enhance the denoising performance. These methods perform feature extraction and image processing using dual CNNs. We further improved the UDRN denoising model by upgrading the RB to a dense RB fusing channel attention and spatial attention, introducing a feature distiller network (FDN) to enhance feature extraction, and optimizing the dual convolutional networks to a multi-convolutional network denoising model.

In image processing, edge detection tools such as the Canny operator, Sobel operator, and Laplace operator extract edge information. Region-based segmentation methods divide regions by analyzing pixels' similarity, and standard techniques include watershed algorithms [39], split and merge methods, and region-growing methods. Neural networks use the learning mechanism of neurons to improve segmentation accuracy. In image denoising, the watershed algorithms [40] are used to segment the foreground and background to enhance the denoising effect so that the algorithms focus on the important regions and reduce the mishandling of the background. The foreground usually contains critical details, and the watershed algorithms [41] protect this information from being lost during the denoising process. In addition, distinguishing between the foreground and background helps to reduce computational complexity, improve denoising efficiency, and provide cleaner images for subsequent feature extraction or object detection. The improved algorithm proposed by Wu and Li [42] removed noise through grayscale conversion and morphological open and close reconstruction, preserved the main contours, extracted the markers using the maximum interclass variance method, and finally applied the watershed algorithms to improve segmentation accuracy and preserve edge information. Yesmin et al. [43] proposed an improved marker-based watershed approach to achieve denoised image segmentation by first removing noise through preprocessing and then combining the Otsu thresholding method [44] and marker-controlled watersheds, but difficulties in marker selection and automatic generation may lead to over segmentation or under segmentation. These articles use watershed algorithms for image segmentation. They improve the segmentation effect by denoising, while these papers considered the watershed algorithms as image preprocessing tools, mainly for extracting foreground information. The watershed algorithms not only effectively separate the central regions in the image by removing the background interference and irrelevant noise but also enhance the structural clarity, which provides a reliable basis for subsequent network training and feature extraction to optimize the models' performance further.

In the field of image denoising, most of the existing models focus on a single deep learning architecture and rely only on specific deep learning models, which have limitations in accurately capturing image boundaries and details. By systematically analyzing studies related to attention mechanisms, dense RBs, feature extraction blocks, dual CNNs, and watershed algorithms, we have found that these approaches have potential advantages in image denoising tasks. In light of this, this paper proposes a novel multi-convolutional neural network based on feature distillation learning and dense residual attention (MFDRAN) model, which is innovative in that it employs a multi-CNN structure. The images are first preprocessed using the watershed algorithm (WA), followed by a combination of feature distillation learning as well as dense residual attention blocks. This multi-level and multi-method fusion makes the MFDRAN model break through the limitations of traditional models and achieves an improvement in image denoising performance.

The main contributions of this paper are as follows.

- We propose the MFDRAN, which effectively improves the performance of the brain image denoising task. This task operates in a networked information stream using the global sparse network (GSN), the dense residual attention network (DRAN), the FDN and the feature processing block (FPB).
- Before the brain image enters the denoising network, the brain image is preprocessed with the watershed algorithms, which preserves the foreground portion of the valid information and reduces the background noise.
- The DRAN is proposed. The design accurately extracts key features by adaptively adjusting the channel and spatial location weights to enhance the model's feature selection and expression ability. In addition, 1×1 convolution is used for dimensionality reduction to remove redundant information and strengthen the salient features, thus optimizing the model's performance.
- Two feature distillation RBs form the FDN that suppress redundant information, reduce the network's computational complexity, and preserve essential image details.
- We conducted qualitative and quantitative experiments on brain images, ground-based open datasets, and real noisy image datasets, and verified that the proposed model has good denoising performance. This provides strong support for accurate diagnosis and analysis in brain image processing.

The remainder of this paper is organized as follows: Section 2 briefly describes the related work. Section 3 describes the proposed multi-CNN based on feature distillation learning and dense residual attention in detail. Section 4 shows the experimental results and analyses, and compares the performance of the proposed network with other methods. Finally, the conclusions are given in Section 5.

2. Basic theory

2.1. The WA

Watershed segmentation is an image segmentation algorithm that combines geomorphology and area growth, which treats the grayscale image as a topographic map, with regions of high gray values representing mountainous terrain and regions of low gray values representing low-lying areas. Water flows along the hillside to the low-lying area, forming a "catchment area", and the boundary of the catchment area is the watershed line. By constructing a dam at the junction of the catchment area, the water can be prevented from overflowing, thus determining the boundary location of image segmentation and making watershed segmentation an effective image segmentation method. The WA is an image segmentation algorithm based on mathematical morphology, terrain, and the region's growth [45]. The algorithm can compute and localize the target accurately and approximate the contour of the target [42]. Marker-based controlled watershed segmentation divides the image into connected components using labels. Foreground markers correspond to essential objects, and background markers are associated with background regions. Foreground labels satisfy the following conditions: 1) a region surrounded by points with a higher "elevation"; 2) the points within the region form a connected component; and 3) all points within the component have the same gray value. The background labels are defined by the watershed lines [41]. WAs has the limitations of false labeling in medical image segmentation and are prone to over segmentation and image contamination by noise during acquisition. Liang and Fu [41] proposed an improved WA based on morphological processing and complete variance modeling for medical image segmentation. The method starts with morphological gradient preprocessing and full variance denoising of brain injury MRI images to preserve the edges and suppress noise. Then, inner and outer markers are generated by the forced minimum technique, and watershed segmentation is performed after correcting the gradient map.

The morphological gradient calculates the difference between expansion and corrosion by performing morphological operations on the image. The morphological gradient mathematical expression is

$$G_{morph}(x, y) = D\left(I(x, y)\right) - E(I(x, y)), \tag{2.1}$$

where D and E denote the dilation and erosion operations, respectively, and $I(x, y)$ is the gray value of the image at position (x, y). Sivakumar and Janakiraman [40] proposed a modified watershed segmentation (MWS) method for segmenting tumor regions from MRI images. Firstly, MRI images are extracted from the database, denoised, and enhanced by a high-pass filter. Then the edges are extracted with enhanced Canny edge detection to improve the input quality of the WA. The Canny operator is formulated as follows:

$$G_{canny}(x, y) = Canny(I(x, y)), \tag{2.2}$$

where $I(x, y)$ is the gray value of the image at position (x, y) and $Canny(x, y)$ denotes the application of the Canny edge detection algorithm on the image I. The Sobel algorithm [45] is a fast and effective edge detection method that can smooth and suppress noise. The Sobel gradient [46] is computed separately in the x and y directions. The mathematical expression for the Sobel gradient is

$$G_{sobel}(x, y) = \sqrt{S_x(x, y)^2 + S_y(x, y)^2}, \tag{2.3}$$

where $S_x(x, y)$ and $S_y(x, y)$ are the gradients in the x and y directions computed using the Sobel operator.

2.2. Attention mechanisms and dense RBs

Neuroscientific studies have shown that the brain's visual cortex focuses on specific regions in a visual scene through selective attention mechanisms, effectively processing relevant information and ignoring irrelevant information. In neural networks, the attentional mechanisms [47] filter out important information by assigning weights to different parts of the input data. Attentional mechanisms [48] improve a model's performance by directing the model's attention to important content and locations, and channel attention can adaptively realign features by considering interdependencies between channels [49]. Attentional mechanisms are widely used in computer vision to enhance image generation by extracting intact features and reconstructing damaged features [21]. For the brain image, the pixels in the background are primarily black. Brain regions' pixel values are more significant than zero. Average pooling retains the background information and is suitable for cases where all pixels contribute. Average pooling alone is not ideal for brain image denoising. In contrast, maximum pooling is better at extracting texture information and helps the network focus on brain regions. The following describes three methods for aggregating local areas of the feature map in the attention mechanism: average pooling [50], maximum pooling [51], and standard deviation pooling.

Average pooling:

$$avg_{out(X)} = \frac{1}{C}\sum_{i=1}^{c} X_i, \quad \text{where } i = 1, \cdots, c, \tag{2.4}$$

where X_i denotes the feature map of the i-th channel.

Maximum pooling:

$$max_{out(X)} = \max X_i, \quad \text{where } i = 1, \cdots, c. \tag{2.5}$$

Standard deviation pooling:

$$std_{out(X)} = \sqrt{\frac{1}{C}\sum_{i=1}^{c}(X_i - avg_{out(X)})^2}, \quad \text{where } i = 1, \cdots, c. \tag{2.6}$$

Increasing the network depth extends the model's representation capability to fit the training data more adequately, which theoretically can reduce the training error. However, the training error may increase in practice due to gradient disappearance or explosion. Figure 1 shows the schematic structure of the RB, dense block, and dense RB.

To address this problem, He et al. [52] proposed a residual learning framework to simplify the training process for deeper networks than ever before. Figure 1(a) shows the RB. The RB is a network structure that contains a "skip connection", which passes the input directly to the output layer and adds it to the output of the convolutional layer to form the so-called "residual". This structure helps alleviate the gradient vanishing problem in deep networks, improving training efficiency and model performance. Based on the RB, Huang et al. [53] proposed a dense convolutional network (DenseNet) connecting each layer to all other layers, strengthening feature transfer, mitigating gradient vanishing, and promoting feature fusion and reuse. The dense blocks are shown in Figure 1(b), where each layer receives the feature mappings of all previous layers as input. This dense connectivity effectively enhances feature reuse, helps capture richer feature representations, reduces the gradient vanishing problem, and improves the parameter utilization efficiency of the network at the same time. Subsequently, Zhang et al. [24] proposed a RDN, which uses a residual dense block (RDB) to extract local features, combined with CM mechanisms and regional and global feature fusion to achieve efficient training and global feature learning. Figure 1(c) shows the dense RB. The input X_0 is passed through the H_1 convolution layer to get the output X_1, and X_1 is spliced with X_0 on the channel axis to get the spliced feature tensor, which is fed into the H_2 convolution layer to get the output X_2. In the same way, X_2 is spliced with the feature information of all the previous layers before it is fed into the next layer. This cycle continues until the output of the last layer, X_3, is spliced with the feature information of all previous layers and the input information, X_0, to obtain the final output of the module.

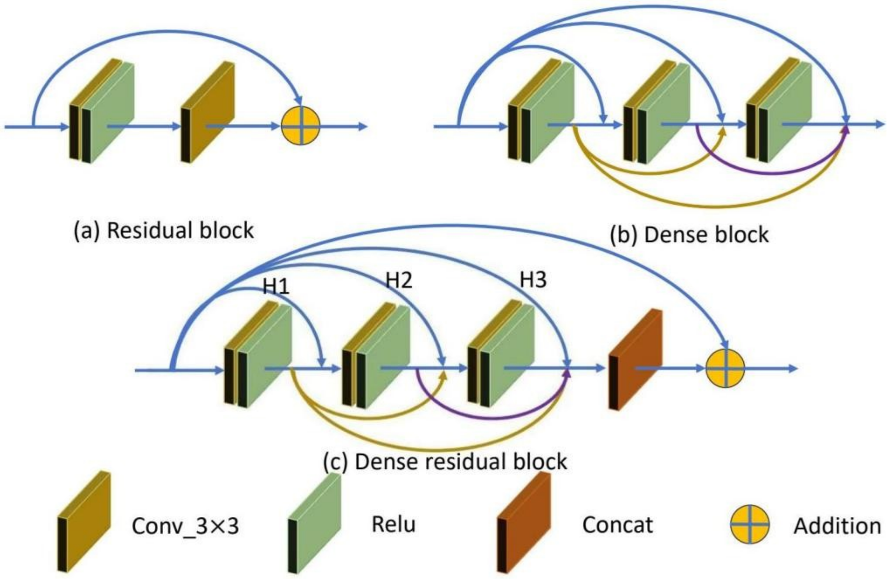


Figure 1. Schematic structure of the RB, dense block, and dense RB.

2.3. The feature distillation RB

Feature distillation is a key step in knowledge distillation, which enables even lightweight models to achieve close performance by efficiently migrating the essential features from the complex teacher network (T-Net) to the student network (S-Net). Knowledge distillation aims to compress and transfer features and parameters from the teacher model, allowing the student network to reduce the number of parameters while significantly maintaining efficient performance. T-Net [54] effectively guides the learning process of S-Net through rich feature representations for efficient knowledge transfer. S-Net [55] employs an encoder–decoder architecture to avoid the limitations of traditional convolutional stacking, and also utilizes bottleneck separable convolution (BSConv) to decompose convolutional operations into depth and pointwise convolutions to significantly reduce the number of parameters and the computational overhead. S-Net introduces a middle distillation block (MDB) between the encoder and decoder to optimize feature extraction through three stages: feature distillation, compression, and enhancement. Combining the BSConv and MDB modules, S-Net improves the feature extraction efficiency while reducing the parameters, achieving a balance between performance and efficiency [55].

Many scholars have adopted lightweight and optimization strategies to improve image reconstruction quality and make the model more lightweight and efficient. By extracting and streamlining valuable features, the distillation module can suppress redundant information, reduce the computational complexity of the network, and preserve important image details. This feature distillation strategy enables residual feature dense blocks (RFDBs) to achieve efficient image reconstruction with less computational overhead. RFDBs avoid the problem of gradient vanishing in deep networks by introducing residual connectivity. Residual connections allow the network to learn the residuals between inputs and outputs rather than understanding the input-to-output mapping directly. This design speeds up the convergence of the network and makes training more stable. Other scholars [56] processed the input features using a 3 × 3 convolutional layer and then performed multiple subsequent distillation steps. In each step, a channel segmentation operation is performed on the previous features to divide the features into the selected and the remaining parts. The remaining portion is further processed and transferred to the next computational unit. The selective part can be considered as the refined feature. In short, a stepwise channel separation operation preserves some features and extracts information from other parts.

2.4. The GSN

The core of the GSN is implemented through a sparse mechanism in collaboration with 3 convolutional layers. Layers 2, 5, 9, and 12 use a 3 × 3 convolutional kernel with an expansion factor of 2. The rest of the convolutional layers use a 3 × 3 convolutional kernel. In the GSN, the first convolutional layer changes the number of input image channels from 3 to 64. Each convolutional layer contains convolution, batch normalization (BN), and ReLU activation and has the same structure except for the convolutional kernel size. Therefore, the final output of the GSN is

$$T_{GSN} = C(S(X)), \tag{2.7}$$

where $C(\cdot)$ is the final convolutional output, $C(X)$ is the sparse convolutional output, and X is the sparse convolutional input.

3. Network construction

In this paper, the GSN and FPB proposed by Yang et al. [38] were applied to optimize the model's feature extraction and processing process. On this basis, this paper proposes a multi-CNN (MFDRAN) based on feature distillation learning and dense residual attention for brain image denoising, proposes a dense residual network based on an improved channel attention mechanism and a spatial attention mechanism for parallel processing, innovatively introduces the FDN, and applies the combination of these four modules to the task of brain image denoising. This

MFDRAN brain image denoising model fully uses the multi-convolutional structure to effectively process complex information in brain images. In the preprocessing stage, important information in the image's foreground is extracted using an improved WA to provide an accurate feature base. The model can efficiently extract and retain critical features by incorporating dense residual attention blocks (DRABs) into the feature distillation framework, achieving noise suppression with reduced computational complexity. This structure enables the proposed denoising model to perform well in the brain image denoising task with high performance and robustness. The overall network structure is shown in Figure 2.

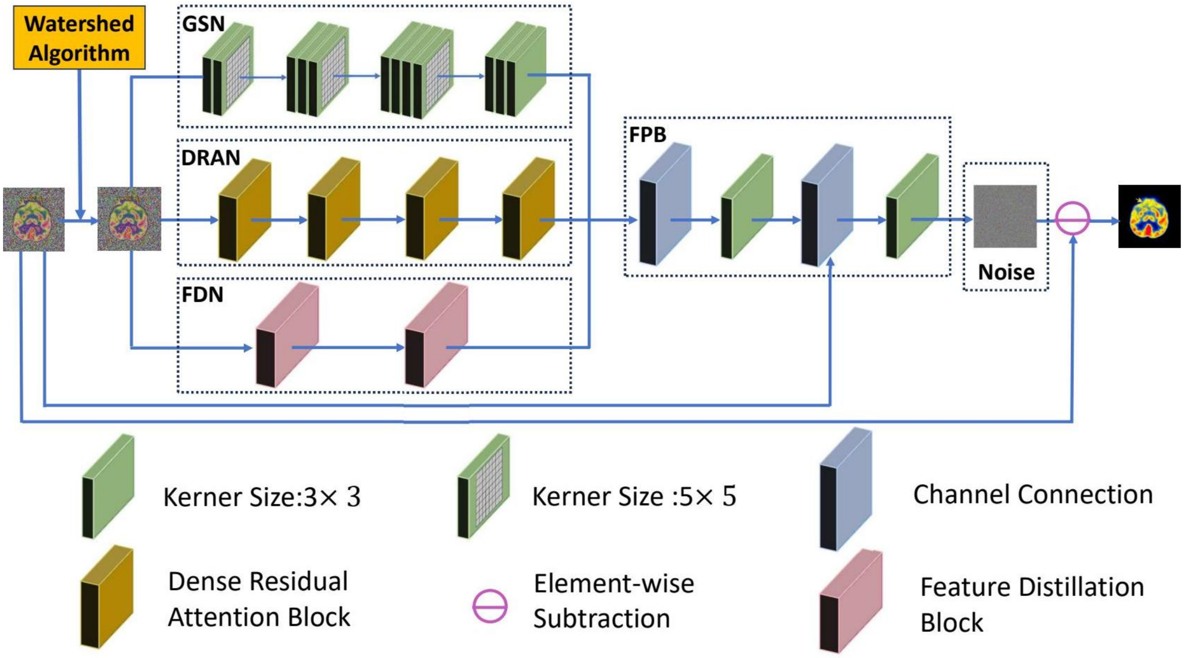


Figure 2. Diagram of the MFDRAN network's architecture.

The MFDRAN brain image denoising model adopts a hierarchical architecture composed of five key components: the WA, the GSN, the DRAN, the FDN, and the FPB. In Figure 2, the brain image is first taken as the network input. The important foreground information in the image is extracted by the WA. The preprocessed image is input into the subsequent network. The GSN and DRAN extract the brain image's overall and local features. The FDN adopts two feature distillation blocks (FDBs), which mainly extract the critical and representative features in the brain image and reduce computational complexity at the same time to retain the image's details and structure, especially the efficient features after removing noise to achieve the task of shallow denoising of the brain image. Then the different features are fused by the FPB, and the first feature extraction result is further processed, after which, the final brain image noise feature map is obtained. Finally, the input image and the second output noise feature map are subtracted to get a potentially clear brain image.

Since the GSN and the FPB partially adopt the structure in the UDRN [38] model, they are not described in detail in the introduction section of the network model in this paper. Next, the other components of the network will be described in detail, including the WA, the DRAN, and the FDN.

3.1. The WA

In the image preprocessing stage, the image's edge information is first extracted using the morphological gradient, the Sobel operator, and the Canny operator, where the morphological gradient highlights the edges by the difference between erosion and dilation operations, the Sobel operator calculates the gradient by using discrete first-order differentiation, and the Canny operator determines the edges after Gaussian filtering, gradient computation, non-maximum value suppression, and double threshold detection steps. After several ablation experiments, the weights are set to 0.7, 0.2, and 0.1, thus generating high-quality gradient maps and providing a reliable input base for the subsequent WA. The WA combines this gradient information and performs image segmentation by simulating the water flow process in the terrain, which can accurately extract the foreground region and effectively remove background interference and noise. Next, the extracted foreground region is input into the denoising network, which models and suppresses the noise through convolution, residual joining, and attention mechanisms to generate a clear denoised image. According to the evaluation indicators, namely the peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and normalized mean square error (NMSE), the process effectively combines gradient feature extraction, segmentation, and denoising, making it particularly suitable for medical image processing scenarios and able to efficiently extract and optimize the target region to provide high-quality data support for subsequent diagnosis and analysis.

3.2. The DRAN

Our proposed DRAN consists of four dense RBs, as well as improved channel attention mechanisms and spatial attention mechanisms. It adopts dense connectivity so that the output of each layer is passed not only to the next layer but also to all subsequent layers. Through dense connectivity, LFF, attention mechanisms, and local residual learning, a CM mechanism is constructed, which is used to fully fuse the feature information of each convolutional layer and enhance the richness and diversity of feature expression. The network progressively extracts features through multi-layer convolution in dense blocks, and the output of each layer is spliced with the input feature maps to preserve the rich feature information. To avoid the redundancy and computational overhead of directly integrating the feature maps of all convolutional layers, the network introduces a feature fusion strategy, which effectively combines the local features first and then gradually passes the optimized features to the subsequent fusion stage. In the feature extraction process, the improved channel attention mechanism and spatial attention mechanism are applied to the dense block to improve the accuracy of feature extraction, focus on essential features, and capture the salient features of the image at the channel level and spatial level, respectively. Among them, the improved channel attention mechanism dynamically adjusts the weights of different channels to focus on more representative feature channels. In contrast, the enhanced spatial attention mechanism helps the network to locate key regions in the image and improve the information representation in the spatial dimension. The feature representation is optimized by weighted fusion, and the last convolution layer subsequently processes the fused features. After a series of dense RBs, the initial input image is summed with the processed feature map, which preserves the original features, effectively enhances the image's detailed information, improves the brain image's denoising effect, and avoids the problem of vanishing gradients. This architecture allows the network to be trained more stably. It does not significantly increase the computational complexity when increasing the depth, thus capturing multi-dimensional feature information more effectively when processing complex images.

This paper's improved channel attention mechanism generates weights for each channel through mean pooling, maximum pooling, and standard deviation pooling, thus generating weights for each channel. First, the global information of the channel is extracted using mean pooling and maximum pooling, and the standard deviation is obtained through standard deviation pooling, reflecting the discrete nature of the channel features. Then, after two convolutional layers, the feature dimensions are reduced from the number of input channels to the ratio of the number of input channels to the ratio of the reduction in dimensionality, and then restored to the original number of channels. The final generated weights are restricted to between 0 and 1 by a sigmoid function to form the final channel weights. This weight is multiplied by the essential input feature map to highlight essential channels and suppress information from unimportant ones. The flowchart of the channel attention mechanism is shown in Figure 3.

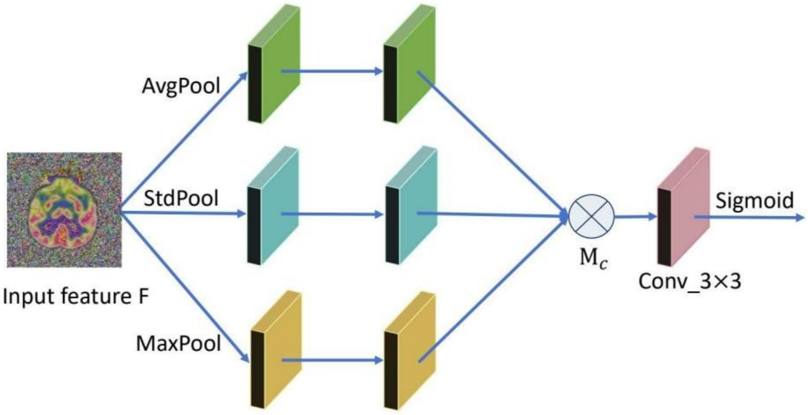


Figure 3. Flowchart of the channel attention mechanism.

In Figure 3, the input feature F of the channel attention module is of size H × W × C. First, the feature map is pooled into feature vectors of size 1 × 1 × C by mean pooling F_{avg}, maximum pooling F_{max}, and standard deviation pooling F_{std}, respectively. Then the weighted pooling result F_{pool} can be expressed as

$$F_{pool} = 0.1 \cdot F_{avg} + 0.8 \cdot F_{max} + 0.1 \cdot F_{std}. \tag{3.1}$$

Next, F_{pool} is processed through a fully connected dual-layer neural network. The number of neurons in the first layer is C/r, the number of neurons in the second layer is C, and the ReLU activation function is applied after the first layer. The output result F_{fc} is

$$F_{fc} = FC_2(ReLU(FC_1(F_{pool}))), \tag{3.2}$$

where FC_1 and FC_2 are the first and second fully connected layers, respectively; FC_1 is the downscaled convolutional weight matrix for reducing the number of channels from C to C/r (where r is the reduction ratio); and FC_2 is the upscaled convolutional weight matrix for restoring the number of channels from C/r to C. The outputs F_{fc} are then passed through a sigmoid activation function to obtain the channel weight coefficients M_c. M_c is calculated as follows:

$$M_c = Sigmoid(F_{fc}). \tag{3.3}$$

Finally, the weighting coefficients M_c are multiplied by the initial feature F to obtain the weighted new feature F_1. F_1 is calculated as follows:

$$F_1 = M_c \cdot F. \tag{3.4}$$

This gives the feature map after applying the channel attention mechanism.

This paper's improved spatial attention mechanism calculates aggregated spatial features by mean pooling, maximum pooling, and standard deviation pooling. For each pixel location, the weighting coefficients of average pooling, maximum pooling, and standard deviation pooling are determined after several ablation experiments, which are 0.35, 0.45, and 0.2, respectively, and the pooling results are weighted and combined to generate an integrated feature map, which can more accurately characterize the spatial properties. Next, spatial attention weights are computed through a convolutional layer, and a sigmoid function is used to limit the range of weights. Finally, these spatial weights are applied to the input feature map to adjust the influence of each pixel. The flowchart of the spatial attention mechanism is shown in Figure 4.

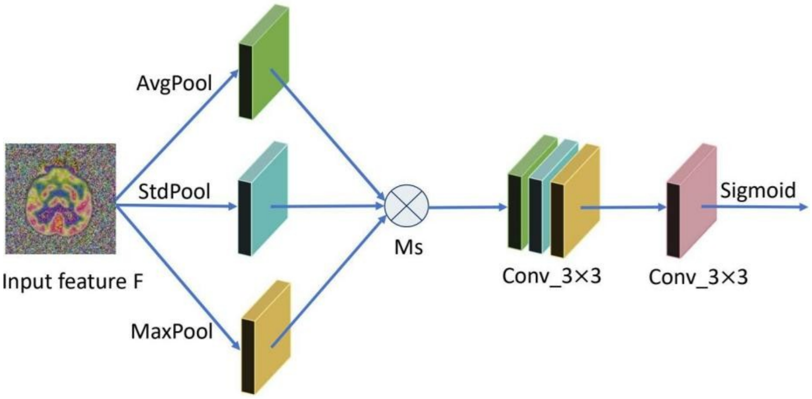


Figure 4. Flowchart of the spatial attention mechanism.

First, for the input feature map $F \in \mathbb{R}^{B\times C\times H\times W}$, the average pooling, maximum pooling, and standard deviation pooling operations are performed. Weighted combinations are performed according to the set weights, assuming that the average pooling result is F_{avg} and the maximum pooling result is F_{max}. The standard deviation pooling result is F_{std}. Then the weighted pooling result, F_{pool}, can be expressed as

$$F_{pool}(F) = w_{avg} \cdot F_{avg} + w_{max} \cdot F_{max} + w_{std} \cdot F_{std}, \tag{3.5}$$

where w_{avg}, w_{max}, and w_{std} are the weights for average pooling, maximum pooling, and standard deviation pooling, respectively. Next, the generative spatial attention weight Y is activated by the convolutional layer Conv_3×3 , and Y can be expressed as

$$Y = \sigma(\text{Conv\_3×3}(F_{\text{pool}}(F))). \tag{3.6}$$

Finally, the spatial attention weight Y is applied to the feature map F to obtain the weighted output $F \cdot Y$, where σ denotes the sigmoid activation function.

The dense residual attention block's flowchart is shown in Figure 5. The dense residual attention block (DRAB) consists of densely connected convolutional layers and an attention module. A dense block enables feature reuse by densely connecting different convolutional layers, which enhances the transfer and utilization of features. In a dense block, the output from each layer is passed to the subsequent layers and spliced with the outputs of all previous layers. This design can mitigate the gradient vanishing problem and maximize the information flow.

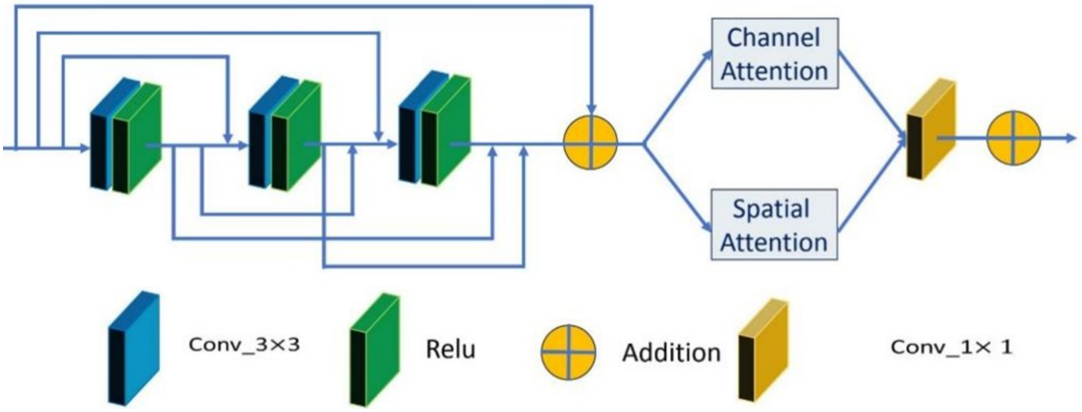


Figure 5. Flowchart of the DRAB.

The following is the detailed computation process of the dense block, assuming that the input feature map is F. The dimension is $C\times H\times W$, where C is the number of channels, and $H\times W$ are the height and width of the feature map, respectively. The first layer of convolution receives the input feature map F and performs the convolution operation $Conv^{k_1\times k_1}$, where the size of the convolution kernel is k_1. After the PReLU activation function, we get the output feature map F_1 from the first layer of convolution. F_1 is calculated as follows:

$$F_1 = PReLU(Conv^{k_1\times k_1}(F)). \tag{3.7}$$

The output feature map F_1 from the first convolutional layer is spliced with the original input feature map F to form the first fused feature map F_{concat_1}. F_{concat_1} is calculated as follows:

$$F_{concat_1} = Concat(F, F_1), \tag{3.8}$$

where $Concat(\cdot)$ represents “spliced”. The second layer of convolution receives the spliced feature map F_{concat_1}; performs the convolution operation $Conv^{k_2\times k_2}$, where the size of the convolution kernel is k_2; and undergoes the PReLU activation function to get the feature map F_2 output from the second layer convolution. F_2 is calculated as follows:

$$F_2 = PReLU(Conv^{k_2\times k_2}(F_{concat_1})). \tag{3.9}$$

The second-layer convolutional output feature map F_2 is spliced with the original input feature map F and the first-layer convolutional output feature map F_1 to form the second fusion feature map F_{concat_2}. F_{concat_2} is calculated as follows:

$$F_{concat_2} = Concat(F, F_1, F_2). \tag{3.10}$$

This operation makes the output of each layer participate in the feature extraction of the subsequent convolutional layer, which ensures the efficient flow of information. The second fused feature map

F_{concat_2} is subjected to the convolution operation $Conv^{k_3\times k_3}$, where the size of convolution kernel is k_3. It undergoes the *PReLU* activation function to obtain the third-layer convolutional output of the feature map F_3. F_3 is calculated as follows:

$$F_3 = PReLU(Conv^{k_3\times k_3}(F_{concat_2})). \tag{3.11}$$

This convolutional layer continues to extract high-level features from the spliced feature map. The feature map F_3 from the third convolutional layer is spliced with the original input feature map F and the feature map F_1 from the first convolutional layer and the feature map F_2 from the second convolutional layer to form the third fused feature map F_{concat_3}. F_{concat_3} is calculated as follows:

$$F_{concat_3} = Concat(F, F_1, F_2, F_3). \tag{3.12}$$

After the convolutional layer, DenseBlock introduces the channel attention mechanism to enhance the selection of important features. The channel attention mechanism weights the features according to the importance of each channel to get the feature map CA_{out} after reweighting by the channel attention mechanism. CA_{out} is calculated as follows:

$$CA_{out} = ChannelAttention(F_{concat_3}), \tag{3.13}$$

where $ChannelAttention(\cdot)$ represents the result of processing the features through the channel attention mechanism. After the channel attention mechanism, DenseBlock further applies the spatial attention mechanism to enhance the selection of important spatial locations and obtains the feature map SA_{out} after reweighting by the spatial attention mechanism. SA_{out} is calculated as follows:

$$SA_{out} = SpatialAttention(F_{concat_3}), \tag{3.14}$$

where $SpatialAttention(\cdot)$ is the result of processing the features through the channel attention mechanism. The spatial attention mechanism weights individual pixel positions according to the importance of the feature map at different locations. The outputs of channel and spatial attention represent the importance of the feature map in the channel dimension and spatial dimension, respectively. DenseBlock weights and fuses both outputs to combine these two attention mechanisms to obtain the new feature map F_{fuesd}. F_{fuesd} is calculated as follows:

$$F_{fuesd} = 0.55 \times CA_{out} + 0.45 \times SA_{out}. \tag{3.15}$$

In this method, after several experimental optimizations, the weighting coefficients of the feature maps for the spatial attention mechanism and the channel attention mechanism were found to be 0.55 and 0.45, respectively, aiming to obtain the best feature fusion effect. The fused feature is F_{out}. F_{out} is calculated as follows:

$$F_{out} = Conv^{k_4\times k_4}(F_{fuesd}). \tag{3.16}$$

The fused feature F_{out} is subjected to the convolution operation $Conv^{k_4\times k_4}$, where the size of convolution kernel is k_4. This step performs further feature extraction on the weighted fused feature map to generate the final output. Finally, DenseBlock introduces residual concatenation, which adds the input feature map F and the convolutional output F_{out} to get the final output F_{final}. F_{final} is calculated as follows:

$$F_{final} = F + F_{out}. \tag{3.17}$$

Residual concatenation effectively mitigates the gradient vanishing problem and maintains the flow of input features. DenseBlock maximizes feature reuse through dense concatenation and feature splicing operations with three-layer convolution and enhances attention to essential features through the channel attention and spatial attention mechanisms. The final residual concatenation ensures efficient delivery of the input features and promotes stable gradient updating.

3.3. The FDN

The FDN of this paper contains two FDBs, each of which performs three feature distillation processes, and the flowchart of the FDB is shown in Figure 6. In the FDB of this paper, a total of three feature distillation processes are performed. After several ablation experiments, we set the distillation rate of each distillation step to 25% to strike a balance between the extraction of key features and the retention of residual information. This setting ensures that the network effectively focuses on key features while avoiding information loss due to overcompression, thus improving the overall performance. As shown in Figure 6, specifically, we divide each distillation operation into the distillation layer (DL) and the refinement layer (RL) to process the input features. The DL generates the initial refined features, and the RL further refines these features. As a result, the features are divided into two parts: one is retained by the DL, and the other is refined by the RL and passed to the next computational unit.

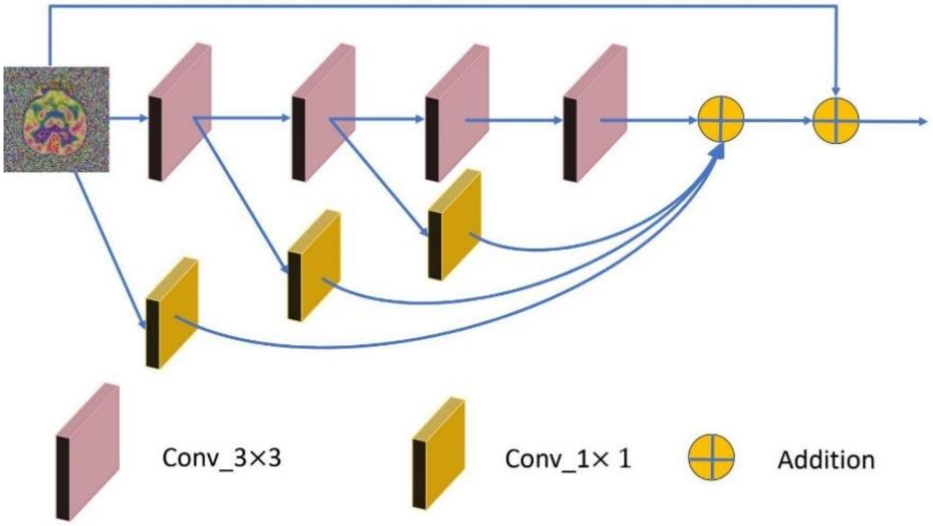


Figure 6. Flowchart of the feature distiller block.

Given the output F^{i-1}_{RFDB} of the $(i-1)$th FDB, the computational process in the output F^{i}_{RFDB} of the ith FDB can be described as follows: first, the input feature is the output F^{i-1}_{RFDB} of the $(i-1)$th FDB, 25% of the channels of F^{i-1}_{RFDB} are passed through a 1×1 convolutional layer to get the first level of refinement feature $F_{distilled_1}$, and 75% of the channels of F^{i-1}_{RFDB} are passed through a 3×3 convolutional layer to get the first-level of the to-be-refined feature F_{coarse_1}. The first-level refined feature $F_{distilled_1}$ and the first-level to-be-refined feature F_{coarse_1} are given as:

$$F_{distilled_1} = Conv^{1\times1}(F^{i-1}_{RFDB}), \tag{3.18}$$

$$F_{coarse_1}{=}Conv^{3\times3}(F^{i-1}_{RFDB}). \tag{3.19}$$

The first level of features to be refined, F_{coarse_1}, goes to the next distillation unit to continue refining the features. Finally, 25% of the channels of these refined features are passed through the DL to obtain the second level of refined features, $F_{distilled_2}$, and 75% of the channels are passed through the second level of features to be refined in the RL, F_{coarse_2}, given as, respectively:

$$F_{distilled_2} = Conv^{1\times1}(F_{coarse_1}), \tag{3.20}$$

$$F_{coarse_2} = Conv^{3\times3}(F_{coarse_1}). \tag{3.21}$$

Similarly, the third-stage distillation feature $F_{distilled_3}$ and the third-stage to-be-refined feature F_{coarse_3} can be obtained, and this process can be expressed as:

$$F_{distilled_3} = Conv^{1\times1}(F_{coarse_2}), \tag{3.22}$$

$$F_{coarse_3} = Conv^{3\times3}. \tag{3.23}$$

It is worth noting that we perform a refinement operation on the third level of features to be refined F_{coarse_3} to obtain the fourth level of refined features F_{coarse_4}. This process can be expressed as

$$F_{coarse_4} = Conv^{3\times3}(F_{coarse_3}). \tag{3.24}$$

All distillation features and the last refinement feature are spliced in the channel dimension to obtain a multi-scale feature map F_{concat} containing detailed and global information for each layer. F_{concat} is calculated as follows:

$$F_{concat} = Concat(F_{distilled_1}, F_{distilled_2}, F_{distilled_3}, F_{coarse_4}). \tag{3.25}$$

The merged multi-scale feature map F_{concat} is compressed and fused by 1×1 convolution $Conv^{1\times1}$ to obtain the second fused feature map F_{concat_2}. F_{concat_2} is calculated as follows:

$$F_{concat_2} = Conv^{1\times1}(F_{concat}), \tag{3.26}$$

where 1×1 convolution reduces the feature dimensions and integrates the multi-channel information, reducing redundancy and highlighting key features. Finally, residual concatenation is added to sum the fused output F_{concat_2} with the original input F^{i-1}_{RFDB} element by element to form the final output. The final output F_{out} is calculated as

$$F_{out} = F^{i-1}_{RFDB} + F_{concat_2}. \tag{3.27}$$

4. Numerical experiments

In this section, we provide an overview of the datasets used in the experiments and discuss the details of implementing the network's training. At the same time, we analyze the contributions of the framework's various components, present the model's experiments on the image test set, and explore the experimental setup and results quantitatively and qualitatively.

4.1. Experimental preparation

4.1.1. Experimental setup

Experiments were conducted using Adam's algorithm [57] to optimize the training of the network. Specifically, this was based on using the Pytorch framework to train the network. After several experiments, the hyperparameters β_1 and β_2 controlling the decay rate of the first-order momentum estimation and second-order momentum estimation were set to β_1=0.9 and β_2=0.9, respectively.

Before feeding the training and test images into the network, all images were converted to tensors and normalized to [0,1]. The learning rate in deep learning determines how much the model parameters are updated. Higher learning rates speed up convergence but may trigger oscillations and affect stability. Therefore, gradually decreasing the learning rate is often used during training to balance speed and stability. In this paper, the learning rate was dynamically adjusted according to different stages in the training process. Expressly, in various components, the learning rate during network training was set as follows: at epoch $\leq$ 40, the learning rate was set to 1×10^{-3}; at $40 <$ epoch < 60, the learning rate was set to 1×10^{-4}; at epoch $\geq$ 60, the learning rate was set to 1×10^{-5}.

The noise simulation in this study was confined to applying artificial Gaussian noise at three specific levels, with standard deviations of 15, 25, and 35.

4.1.2. Experimental data

For the medical image denoising task, we chose the brain image dataset AANLIB [58] from the Harvard Medical Dataset, which contains a wide range of brain images and is suitable for evaluating the denoising performance of the MFDRAN model on medical imaging noise. To validate the model's generality on different color images, the publicly available datasets CBSD68 [59] and Kodak24 [60] were also introduced to provide standardized benchmarks for color image denoising performance comparison. In addition, to evaluate the performance of the model in a real noise environment, the real dataset SIDD [61] and an actual authentic noise dataset RNI15 [62] were also used in this study to validate the denoising capability of the MFDRAN model. In summary, the experimental dataset selected in this study covers medical images, standard images of natural scenes, and authentic noise images containing various noise types and image contents, which provides a scientific basis for evaluating the denoising effect of the model under different noise conditions.

Before data preprocessing, 40 clearer brain images were screened from the dataset, of which 20 were used for the test set and 20 for the training set. After preprocessing the 20 images used for the training set, 3872 colorful synthetic noise images with a size of 41 × 41 were finally generated as the final training dataset for training the designed network model. The preprocessing steps for the 20 training set images were as follows: first, during data preprocessing, the sliding window size was set to 41 × 41, and the step size was 10; the images were cropped step by step to generate the actual training samples, and all brain images were normalized to the range of [0, 1]. To ensure the fairness of the experiments, all methods were trained on the same training and test sets, and the number of model training rounds was kept the same. To expand the diversity of the training dataset, each cropped image was randomly transformed four times according to the following operations: 1) horizontal flipping, 2) 90° counterclockwise rotation, 3) 180° counterclockwise rotation, 4) 270° counterclockwise rotation, 5) the original cropped image, 6) horizontal flipping after 90° counterclockwise rotation, 7) horizontal flipping after 180° counterclockwise rotation, and 8) horizontal flipping after 270° counterclockwise rotation. The study also obtained 3872 images. Finally, 3872 41 × 41 color synthetic noisy brain images were obtained in this paper and used as the training dataset.

This experiment consists of two parts: the ablation experiments and the comparison experiments. The ablation experiments are described below.

4.2. Ablation experiments

This paper designed three ablation experiments to validate the effectiveness of each module. The experiments evaluated the contribution of the WA, the FDN, and the DRAN to the model's performance. The same dataset was used and trained the same number of times under a noise level of 25 to ensure the fairness and reliability of the evaluation results.

4.2.1. WA ablation experiments

We performed ablation experiments on the WA to evaluate the impact of the WA in a brain image preprocessing step on the performance of denoising models. Specifically, this experiment used two scenarios for comparison. The first experiment aimed to directly denoise brain images containing additive noise using the denoising model without any preprocessing step. The second experiment applied the WA to the original brain image for preprocessing. It separated and extracted the foreground regions, added noise to the extracted foreground regions, and finally input them into the denoising model for further processing. In both experiments, we used three quantitative metrics, PSNR, SSIM, and NMSE, to assess the difference in denoising effectiveness. The experimental results are shown in Table 1, where the bolded data represent the best performance. From Table 1, it can be seen that the network structure preprocessed is 0.6154 dB higher than the "No preprocessing" network structure in terms of the PSNR value, which exhibits a relative enhancement of about 1.52%, and this is a relative enhancement of about 0.48% in terms of the SSIM value. In addition, the network structure preprocessed by the WA is slightly higher than the "No preprocessing" network structure in terms of the NMSE metrics. It is shown that the model has some improvement in structural similarity. In summary, it is shown that the WA helps reduce noise's interference with the model based on accurate foreground extraction, thus enhancing the denoising effect.

Table 1. Comparison of the results of watershed preprocessing.

Network architecture	PSNR	SSIM	NMSE
No preprocessing	40.3819	0.9851	0.0013
Preprocessed	**40.9973**	**0.9898**	**0.0012**

4.2.2. FDN ablation experiments

We conducted ablation experiments on the FDN to obtain average test results, shown in Table 2, to evaluate the effect of FDBs on the denoising performance of the CNN. First, the original CNN denoising model was used as a benchmark to evaluate the effectiveness of the denoising task. Subsequently, one, two, and three FDBs were added sequentially to this model, and three experimental models were constructed to analyze the specific effects of the number of FDBs on the denoising performance of the model. The experimental results show that the model denoising performance was improved after adding the FDBs. The test results in Table 2 show that the model containing two FDBs performs well in terms of the PSNR, SSIM, and NMSE metrics, significantly outperforming the base model and other configurations. Specifically, the network model containing two FDBs improved the PSNR value by 0.5781 dB, which is about 1.43%, and the SSIM value by about 0.48% in terms of the denoising effect, compared with the model without FDBs. The configuration of two FDBs demonstrates more substantial advantages in detail recovery and noise suppression compared with the model containing only one FDB. However, with a third FDB, the model's performance did not show a significant improvement but a slight decrease. This result indicates an optimal value for the number of FDBs in the model performance. Too many distillation blocks may lead to information redundancy and increased computational complexity. In summary, two FDBs achieved the optimal balance between the denoising effect and computational efficiency.

Table 2. Comparison of the results of different FDBs.

Network architecture	PSNR	SSIM	NMSE
No RFDB	40.3819	0.9851	0.0013
An RFDB	40.8847	0.9888	0.0012
Two RFDBs	**40.9600**	**0.9898**	**0.0011**
Three RFDBs	40.9309	0.9893	0.0012

4.2.3. DRAN ablation experiments

We conducted ablation experiments on the DRAN to obtain average test results, shown in Table 3. The impact of DRNs based on improved channel and spatial attention mechanisms on brain image denoising performance was verified by ablation experiments using DRNs. The experimental design included two comparison scenarios: the first scenario was the network model without a dense residual attention block (WDRAB) for denoising brain images; the second scenario was the network model incorporating a DRAB based on the improved channel attention mechanism and spatial attention mechanism processed in parallel for denoising brain images.

Table 3. The comparison results of DRAN.

Network architecture	PSNR	SSIM	NMSE
WDRAB	40.3819	0.9851	0.0013
DRAB	**40.9792**	**0.9893**	**0.0011**

Table 3 shows the average test results of the two sets of experiments. Our experiments used the PSNR, SSIM, and NMSE as quantitative metrics to measure the signal-to-noise ratio, structural similarity, and error level of the denoised images, respectively. The data analysis shows that the network containing dense residual attention blocks significantly outperformed the network without dense residual attention blocks in the image-denoising task. Specifically, as can be seen in Table 3, the PSNR of the DRAN reaches 40.9792 dB, which is improved compared with 40.3819 dB for the network without the DRAB, an improvement of about 1.48%, indicating that it is more advantageous in restoring the details of the image. The SSIM improved from 0.9851 to 0.9893, which suggests that the network with DRAN performs better in maintaining the image structure similarity. The NMSE reduced to 0.0011, which further reduces the reconstruction error compared with 0.0013 for the network without dense residual attention blocks. In summary, the improved density residual attention network has been introduced.

4.3. *Comparative experiments*

The comparison experiments section is divided into two parts: brain image comparison experiments and ground-based open dataset comparison experiments.

4.3.1. Brain image comparison experiments

The experimental part was designed to verify the effectiveness of the MFDRAN in the denoising of brain images proposed in this paper. Based on 20 brain images from the Harvard Medical dataset, the experiments were set up with synthetic noisy images at noise levels of 15, 25, and 35 as the test set. The experiment started with watershed preprocessing of brain images to preserve and enhance the foreground region (the effective information). On this basis, a multi-convolution denoising neural network consisting of two FDBs and a DRAN was built to enhance the feature extraction capability of the network, and comparative experiments were conducted. In the quantitative experiments on the synthetic noise dataset, PSNR, SSIM, and NMSE metrics were used to evaluate the denoising performance. The results show that this paper's model significantly outperforms the other comparative denoising models regarding noise suppression and detail retention. It also has a good visual effect in image detail restoration, demonstrating good practicality and robustness.

In this study, we proposed and validated the performance of our denoising model, MFDRAN, for denoising additive Gaussian noise in brain images with a noise level of 15. The experiments used brain images from the Harvard Medical dataset as the test set. The evaluation metric for the experiment was PSNR. The MFDRAN model was trained in the training set for 70 iterations and subsequently evaluated on the test set. The experimental results are shown in Table 4, where the bolded data represent the best performance. The experimental results show that the MFDRAN model significantly outperformed the comparative models in the denoising effect, including ECNDNet, DnCNN, DudeNet, and UDRN. The average PSNR value of MFDRAN reaches 43.4619 dB, which is an improvement of about 9.29 dB and 1.92 dB compared with the DnCNN and DudeNet models, respectively, and also an improvement of about 0.73 dB compared with UDRN. These data thoroughly verify the advantages of MFDRAN in denoising performance. Further analysis shows that MFDRAN has higher PSNR values in all test images, demonstrating strong denoising ability and image quality enhancement. In addition, the statistical analysis ($p < 0.05$) shows that the denoising performance enhancement of MFDRAN is significant. This indicates that the MFDRAN model outperforms other models regarding the denoising effect. It reflects its consistency, stability, and generalization ability in denoising tasks, and it is particularly suitable for denoising brain images at low noise levels.

Table 4. Results of different methods for brain images with a noise level of 15.

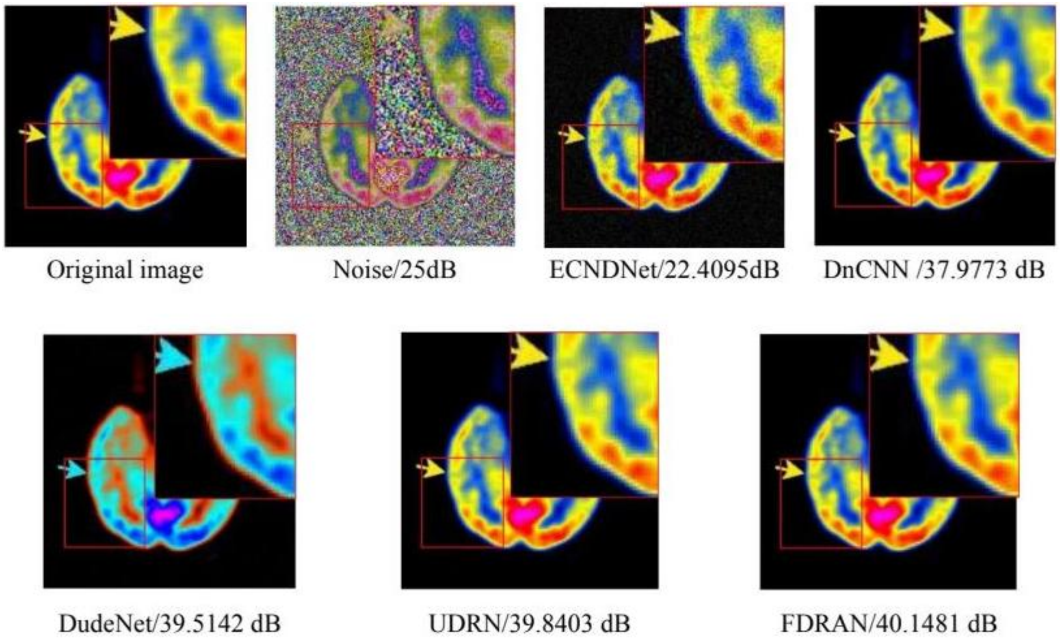
Image	ECNDNet	DnCNN	DudeNet	UDRN	MFDRAN
Image 1	28.4412	36.7754	42.8468	44.8510	**45.5732**
Image 2	28.4377	36.5052	42.7507	44.4998	**45.1592**
Image 3	28.4173	36.1027	42.5177	44.2240	**44.8368**
Image 4	28.3872	35.8521	42.3344	44.0132	**44.6403**
Image 5	28.3563	35.5895	42.1590	43.6382	**44.3233**
Image 6	28.3280	35.3740	41.9391	43.3480	**43.9758**
Image 7	28.3169	35.0834	41.8996	43.1651	**43.8347**
Image 8	28.3216	34.9632	41.9413	43.1350	**43.8283**
Image 9	28.3258	34.6409	41.9274	43.0917	**43.8036**
Image 10	28.3161	34.3814	41.8624	43.0721	**43.7992**
Image 11	28.2823	34.0516	41.7378	42.9175	**43.5711**
Image 12	28.2193	33.0453	40.7179	41.6658	**41.9438**
Image 13	28.1728	33.0216	41.0571	41.8494	**42.7239**
Image 14	28.1197	32.9514	41.0563	42.1193	**42.9966**
Image 15	28.0702	32.9173	40.8885	41.9185	**42.8534**
Image 16	28.0318	32.7365	40.6749	41.4084	**42.4563**
Image 17	28.0230	32.8701	40.8092	41.7762	**42.6045**
Image 18	28.0199	32.5721	40.7413	41.6556	**42.3935**
Image 19	28.0260	32.2054	40.4815	41.0596	**41.8056**
Image 20	28.0134	31.8451	40.5480	41.2084	**42.1147**
Average	28.2313	34.1742	41.5445	42.7309	**43.4619**

With a noise level of 25, Table 5 demonstrates the denoising effect of five different methods on the AANLIB brain dataset. The bolded data in the table represent the best performance. The evaluation metric for the experiment is PSNR. This experiment evaluated the performance of the newly proposed MFDRAN denoising model against the comparison model in processing image noise. Specifically, we calculated the PSNR values of each model on the same test set at a noise level of 25 to quantify the denoising effect. The experimental results show that the PSNR values of the new model, MFDRAN, range from 39.6180 dB to 42.6261 dB, while the best-performing PSNR values of the comparison model range from 38.8715 dB to 42.1468 dB. The comparison shows that the MFDRAN model exhibits higher PSNR values in most test samples, indicating that it is significantly better than the original model in denoising. In addition, the PSNR values of 17 out of 20 test set images exceeded 40 dB, proving that the MFDRAN model can achieve excellent denoising results in most cases. This indicates that the performance of the MFDRAN model in the denoising task is stable and can effectively improve the image quality. It can be seen that the average denoising performance of the MFDRAN model is improved by about 1.606% compared with the UDRN. The enhancement is significant according to the statistical analysis ($p < 0.05$), which indicates that the MFDRAN denoising model exhibits a more stable performance and stronger denoising ability in different scenarios. This enhancement not only verifies the positive impact of the structural improvement of the MFDRAN model on image quality but also indicates the superior reliability and adaptability of the model in practical applications.

Table 5. Results of different methods for brain images with a noise level of 25.

Image	ECNDNet	DnCNN	DudeNet	UDRN	MFDRAN
Image 1	22.9458	41.3676	41.6487	42.1468	**42.6261**
Image 2	22.9368	41.5018	41.6142	41.9843	**42.5447**
Image 3	22.9089	41.2898	41.3201	41.7332	**42.2945**
Image 4	22.8679	40.8700	41.2038	41.4941	**41.9846**
Image 5	22.8318	40.5338	40.9140	41.1757	**41.7416**
Image 6	22.7997	40.2966	40.6730	40.8990	**41.4467**
Image 7	22.7799	40.0882	40.5312	40.7417	**41.2466**
Image 8	22.7730	39.9893	40.6445	40.8250	**41.3685**
Image 9	22.7775	39.7859	40.5822	40.6830	**41.3144**
Image 10	22.7771	39.8075	40.5324	40.7228	**41.4097**
Image 11	22.7348	39.5453	40.4306	40.5869	**41.3118**
Image 12	22.6643	37.4394	39.0277	39.0619	**39.6431**
Image 13	22.5949	38.5461	39.7078	39.5001	**40.4633**
Image 14	22.5176	38.7659	39.9242	39.9429	**40.6431**
Image 15	22.4542	38.5646	39.7725	39.7896	**40.4994**
Image 16	22.4095	37.9773	39.5142	39.2908	**40.1481**
Image 17	22.3919	38.0750	39.7174	39.6761	**40.3079**
Image 18	22.3921	37.8021	39.5144	39.4951	**40.1496**
Image 19	22.3968	37.1777	39.0273	38.8715	**39.6180**
Image 20	22.3773	37.0677	39.1398	39.0170	**39.8345**
Average	22.6666	39.3246	40.2720	40.3819	**41.0298**

A detailed comparison of the brain image denoising results for the AANLIB dataset was performed under a noise level of 25, and the results are shown in Figure 7. The local zoom in the figure's upper right corner demonstrates each method's performance regarding detail recovery. As seen from Figure 7, the MFDRAN method proposed in this paper shows significant denoising effect and color preservation advantages. It excels in edge detail reconstruction of brain images and the color consistency of color images, respectively. MFDRAN can more accurately reconstruct the edge details and the image structure during denoising of the brain images, and compared with the other methods, it was more accurate. Compared with different techniques, MFDRAN is more effective in suppressing artifacts and preserving accurate details. Compared with ECNDNet, MFDRAN removes the noise more thoroughly. Compared with UDRN and DudeNet, it performs better in the reconstruction of edge texture, which significantly reduces the blurring of edges and the loss of texture and results in a clear and natural image, especially in the denoising task of complex brain images. On the other hand, the MFDRAN model outperforms DudeNet in color consistency in denoising color images. The MFDRAN model's processed images with stable colors maintained the original hue and saturation naturally and accurately. In contrast, DudeNet's processed images showed color shifts, which led to some color distortion. This indicates that the MFDRAN model can better retain the image's color information during denoising, while DudeNet still has room for optimization in color information retention. Overall, the MFDRAN model demonstrates superior performance in the denoising task, outperforming other methods in detail restoration and color preservation of complex images, resulting in more transparent, more natural, and visually faithful image results.



Original image Noise/25dB ECNDNet/22.4095dB DnCNN /37.9773 dB

DudeNet/39.5142 dB UDRN/39.8403 dB FDRAN/40.1481 dB

Figure 7. Visual comparison of brain images by different methods at a noise level of 25.

Table 6 compares the effectiveness of the five denoising methods on the AANLIB brain dataset at a noise level of 35. The best-performing data are labeled in bold font in the table to highlight the best performance. The evaluation metric for the experiment is PSNR. This table demonstrates the differences between noise suppression and image detail retention methods, which helps to visually analyze the denoising effect of different techniques on this dataset. Under the test conditions with a noise level of 35, our proposed MFDRAN model significantly outperforms the other compared models regarding the denoising effect. The test results in the table show that the average PSNR value of MFDRAN reaches 38.8955 dB, which is significantly improved compared with all other models. For example, the average PSNR of the comparison models DnCNN and DudeNet are 35.7648 dB and 38.7003 dB, respectively, which fail to achieve the denoising performance of MFDRAN. Compared with the UDRN model, the average PSNR value of UDRN is 38.7212 dB, and the PSNR value of MFDRAN is improved by about 0.45% compared with UDRN. In addition, in 20 test images, the MFDRAN model's denoising results exceeded 40 dB in the PSNR value part of the sample, further proving its superior denoising capability. According to the statistical analysis ($p < 0.05$), the denoising enhancement of MFDRAN is statistically significant. This suggests that at a noise level of 35, MFDRAN demonstrates superior performance in improving image quality and stronger generalization ability in terms of the consistency and stability of noise suppression.

Table 6. Results of different methods for brain images with a noise level of 35.

Image	ECNDNet	DnCNN	DudeNet	UDRN	MFDRAN
Image 1	21.6690	39.4936	40.4237	40.2731	**40.4410**
Image 2	21.6655	39.4007	40.3312	40.2481	**40.3935**
Image 3	21.6391	39.0510	39.9121	40.0118	**40.1265**
Image 4	21.6057	38.2241	39.8032	39.6742	**39.8019**
Image 5	21.5696	37.8699	39.3969	39.4052	**39.4067**
Image 6	21.5397	38.3380	39.1945	39.1782	**39.2154**
Image 7	21.5225	37.0960	39.0124	38.9465	**39.1093**
Image 8	21.5182	36.7240	39.1624	39.0521	**39.1497**
Image 9	21.5241	38.0336	39.1158	38.9940	**39.1688**
Image 10	21.5190	36.6233	39.0518	39.0971	**39.2376**
Image 11	21.4774	35.1597	38.8826	38.9561	**39.0515**
Image 12	21.4044	33.2355	37.1475	37.2345	**37.5672**
Image 13	21.3417	32.8824	37.9772	38.1361	**38.3786**
Image 14	21.2768	29.4202	38.1779	38.3204	**38.5903**
Image 15	21.2158	30.9225	38.0231	38.1840	**38.4428**
Image 16	21.1677	30.3132	37.7971	37.7674	**38.0799**
Image 17	21.1555	35.2424	38.0409	38.0425	**38.2263**
Image 18	21.1543	35.2248	37.7989	37.9100	**38.0830**
Image 19	21.1548	36.0785	37.3693	37.4680	**37.5985**
Image 20	21.1362	35.9614	37.3870	37.5240	**37.8414**
Average	21.4129	35.7648	38.7003	38.7212	**38.8955**

The denoising test results for the test and training sets of brain images are shown in Table 7, where the best-performing data are shown in bold font to highlight the best performance. For the test set, the denoising performance of MFDRAN outperforms that of the comparison models at different noise levels (σ = 15, σ = 25, σ = 35). MFDRAN achieves the highest PSNR at all noise levels, such as 43.4619 dB at σ=15, which is an enhancement of about 0.731 dB over the UDRN, demonstrating more substantial denoising capability. Its SSIM reaches 0.9905 at σ = 25 and σ = 35, which is similar to that of UDRN, maintaining the integrity of the image structure. Regarding NMSE, MFDRAN has a minor error under different noise levels, such as 0.0007 at σ = 15, which is slightly better than UDRN, indicating stable error control. ECNDNet is significantly lower than the other models in each metric, especially under high noise, and it is insufficient to meet the demand for high-quality denoising.

For the training set, we test ECNDNet, DnCNN, DudeNet, UDRN, and MFDRAN models under noise levels of 15, 25, and 35 to compare the denoising performance of each model. The results shown in Table 7 indicate that in terms of the PSNR metrics, the MFDRAN model performs well at all noise levels; in particular, at low to medium noise levels (σ = 15 and σ = 25), its PSNR reaches 43.6248 dB and 41.2177 dB, respectively, which demonstrates good noise immunity performance. Regarding SSIM, MFDRAN maintains a high value close to 0.9905 under different noise levels, which ensures the fidelity of the structural details of the brain images and slightly outperforms the performance of the UDRN model under medium and high noise conditions. In summary, the MFDRAN model improves denoising, structure preservation, and error control capabilities, and performs well in all metrics of the brain image denoising task, especially under high-noise conditions, which verifies its versatility and stability under different noise levels, and has potential clinical applications.

Table 7. Performance comparison with other methods on the brain images.

Dataset	Indicator	Noise	ECNDNet	DnCNN	DudeNet	UDRN	MFDRAN
Test set	PSNR	15	28.2313	34.1742	41.5445	42.7309	**43.4619**
		25	22.6666	39.3246	40.2720	40.8265	**41.0298**
		35	21.4129	35.7648	38.7003	38.7212	**38.8955**
	SSIM	15	0.2902	0.9598	0.6767	**0.9921**	0.9904
		25	0.1749	0.9879	0.7090	0.9870	**0.9905**
		35	0.1570	0.9792	0.7332	0.9840	**0.9905**
	NMSE	15	0.0098	0.0032	0.0609	0.0008	**0.0007**
		25	0.0490	**0.0009**	0.0612	0.0012	0.0012
		35	0.0498	**0.0018**	0.0624	0.0019	0.0019
Training set	PSNR	15	28.2428	34.3502	42.5156	42.8301	**43.6248**
		25	22.6803	39.5662	40.4391	40.7007	**41.2177**
		35	21.4268	35.8128	37.9619	38.9037	**39.2229**
	SSIM	15	0.2852	0.9598	0.7100	**0.9923**	0.9905
		25	0.1706	0.9882	0.7151	0.9872	**0.9905**
		35	0.1531	0.9794	0.6958	0.9842	**0.9905**
	NMSE	15	0.0103	0.0031	0.0603	**0.0007**	**0.0007**
		25	0.0515	**0.0009**	0.0611	0.0012	0.0012
		35	0.0526	0.0018	0.0634	**0.0019**	**0.0019**

4.3.2. Ground-based open dataset comparison experiments

To evaluate the denoising performance of the proposed MFDRAN model on different types of images, this study applied it to the task of additive noise denoising on the CBSD68 dataset. It evaluated its performance under various noise levels. The test results are shown in Table 8, where the best-performing data are shown in bold font to highlight the best performance. It can be seen from Table 8 that the performance of the MFDRAN model in image denoising and its generality is systematically evaluated by comparing several denoising models with different structures under different noise levels. On the CBSD68 dataset, the MFDRAN model significantly outperforms the other models in all the metrics, especially at a noise level of 35. For the PSNR metric, MFDRAN reaches 28.9878 dB, and for the SSIM metric, MFDRAN outperforms UDRN and DudeNet in denoising, with an improvement of 22.38% and 34.08%, respectively. Compared with DnCNN's SSIM value of 0.7601, MFDRAN has improved by 30.15%; compared with ECNDNet's SSIM value of 0.2951, MFDRAN has improved by 235.27%. Compared with DnCNN's SSIM value of 0.7601, MFDRAN is 30.14% higher; compared with ECNDNet's SSIM value of 0.2951, MFDRAN is 235.27% higher. In addition, in terms of NMSE, MFDRAN's 0.0077 is 41.67% and 19.79% lower than DudeNet and DnCNN, respectively, and 54.71% lower than ECNDNet. Under the conditions of 15 and 25 noise levels, MFDRAN outperforms other models in PSNR, SSIM, and NMSE, and always shows better denoising performance under each noise level. These results fully demonstrate that MFDRAN has significant advantages in denoising performance, image structure information retention, and reconstruction accuracy, and especially shows excellent robustness and practical application value in high-noise scenarios.

Table 8. Performance comparison with other methods on the CBSD68 dataset.

Dataset	Indicator	Noise	ECNDNet	DnCNN	DudeNet	UDRN	MFDRAN
CBSD68	**PSNR**	15	25.8158	28.897	33.3771	33.4277	**33.4404**
	SSIM	15	0.6034	0.8615	0.8221	0.9212	**0.9893**
	NMSE	15	0.0041	0.0053	0.0091	**0.0025**	**0.0025**
	PSNR	25	20.5214	28.0764	30.2138	30.6765	**30.7292**
	SSIM	25	0.4069	0.8148	0.7756	0.8641	**0.9893**
	NMSE	25	0.0208	0.0075	0.0121	0.0052	**0.0049**
	PSNR	35	18.9731	27.0477	28.7108	28.8788	**28.9878**
	SSIM	35	0.2951	0.7601	0.7379	0.8084	**0.9893**
	NMSE	35	0.0170	0.0096	0.0132	**0.0076**	0.0077

To evaluate the denoising performance of the MFDRAN model proposed in this paper on different types of images, this study applied it to the task of additive noise denoising on the Kodak24 dataset. It evaluated its performance under various noise levels. The test results are shown in Table 9, where the best-performing data are shown in bold font to highlight the best performance. As can be seen in Table 9, at a noise level of 15, MFDRAN shows significant advantages in all three evaluation metrics. Specifically, the PSNR of MFDRAN reaches 34.0285 dB, which is 0.80% and 0.82% better than UDRN (33.7573 dB) and DudeNet (33.7534 dB), respectively, and 15.96% better than DnCNN (29.3545 dB), and a significant 36.80% better than ECNDNet (24.8419 dB). MFDRAN also outperforms the other compared models in the SSIM and NMSE metrics, where SSIM reaches 0.9893, which is significantly higher than UDRN (0.9147) and DudeNet (0.8457), and NMSE is reduced to 0.0017, which outperforms all the compared models. These results indicate the MFDRAN model excels in denoising accuracy, and is better at recovering image details and reducing redundant information. Meanwhile, MFDRAN still improves its denoising ability compared with other models at noise levels of 25 and 35, which further validates its robustness and superiority in different noise scenarios.

To verify the denoising performance of the MFDRAN model on authentic noisy images, this study conducted in-depth experimental analyses based on the SIDD and RNI15 datasets. The SIDD dataset contains nearly noiseless photos, making it suitable for quantitative and qualitative evaluation of the MFDRAN model and other different structural models, and demonstrates the denoising effect of the model from multiple perspectives. The RNI15 dataset, on the other hand, contains authentic noisy images, which helps to compare the difference in denoising performance between the MFDRAN model and other models more intuitively to validate the superior performance of the MFDRAN model in complex noise environments.

The denoising visual effects of different models on the actual image SIDD dataset are shown in Figure 8. The top left corner of the figure shows the local zoomed-in area, which is convenient for observing the details of the denoising effect. According to the denoising results, the DudeNet model shows some color deviation after denoising, and there are still more apparent artifacts in the results of ECNDNet. In contrast, the color of the denoised image of the DnCNN model is deepened compared with the original image. In contrast, the denoising results of the MFDRAN and UDRN models are visually more similar. Still, according to the PSNR values, our proposed model performs better in denoising performance, further indicating that it is more capable of detail preservation and noise suppression.

Table 9. Performance comparison with other methods on the Kodak24 dataset.

Dataset	Indicator	Noise	ECNDNet	DnCNN	DudeNet	UDRN	MFDRAN
Kodak24	**PSNR**	15	24.8419	29.3545	33.7534	33.7573	**34.0285**
	SSIM	15	0.5732	0.882	0.8457	0.9147	**0.9893**
	NMSE	15	0.0055	0.003	0.0052	0.0019	**0.0017**
	PSNR	25	21.7449	20.5143	31.2911	31.046	**31.2216**
	SSIM	25	0.4072	0.7919	0.8129	0.8638	**0.9893**
	NMSE	25	0.0067	0.0758	0.0063	0.0036	**0.0032**
	PSNR	35	19.0184	26.7120	29.4296	29.7051	**29.8735**
	SSIM	35	0.3002	0.7895	0.7710	0.8248	**0.9893**
	NMSE	35	0.0120	0.0060	0.0075	0.0050	**0.0045**

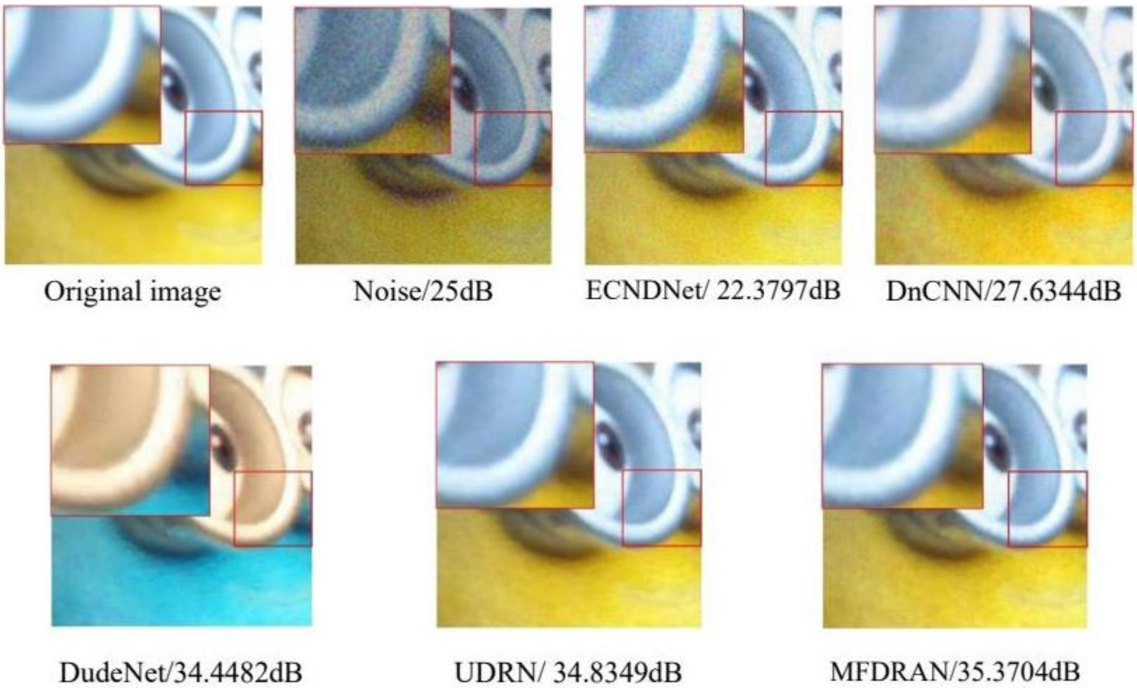


Figure 8. Visual comparison of the SIDD dataset obtained by different methods at a noise level of 25.

The denoising visual effects of different models on the actual noisy image RNI15 dataset are shown in Figure 9. The bottom right corner of the figure shows the local zoomed-in area, which makes it easy to observe the details of the denoising effect. Compared with the DnCNN and DudeNet models, our MFDRAN model performs more accurately in color reproduction, effectively avoiding the problem of color deviation and ensuring that the color of the denoised image is closer to the actual scene. Compared with the ECNDNet and UDRN models, the MFDRAN model can significantly preserve the details and texture features of the image after denoising the real noisy image RNI15 dataset, resulting in the denoised image presenting higher detail clarity and visual fidelity. This indicates that the MFDRAN model has an advantage in denoising performance and outperforms the comparison model regarding the comprehensive ability of detail retention and color reproduction.

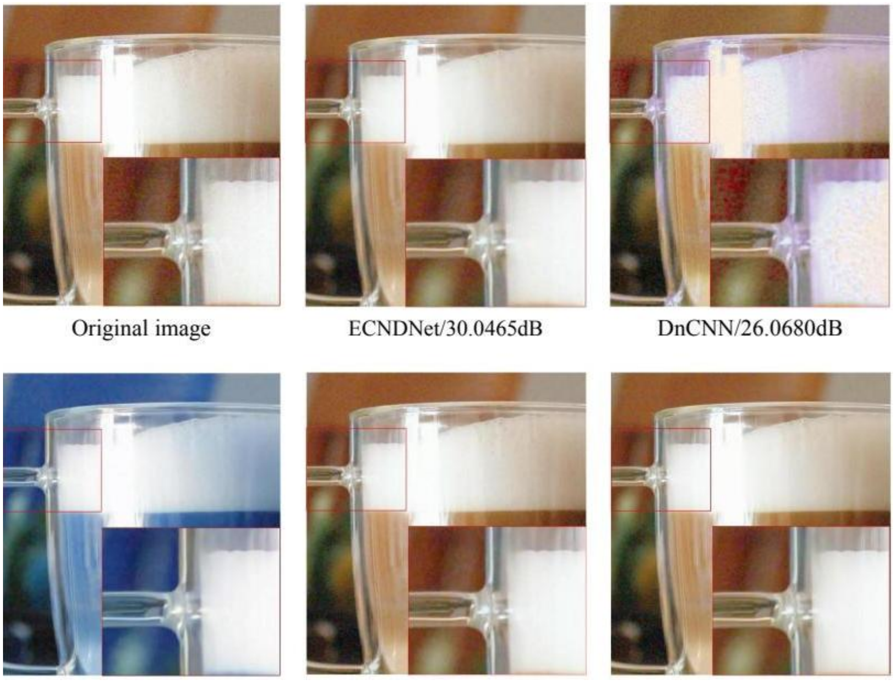


Figure 9. Visual comparison of the RNI15 dataset obtained by different methods.

To fully evaluate the complexity of the proposed MFDRAN models, we conducted experiments in the Pytorch framework using a hardware environment consisting of an Intel(R) Xeon(R) Silver processor, 64 GB of RAM, and an NVIDIA Quadro P2200 graphics card. In our experiments, we evaluated the runtime and number of parameters for each model.

Table 10 compares the training time and number of parameters required by the MFDRAN model with those of the ECNDNet, DnCNN, DudeNet, and UDRN models for the brain image denoising task. It is worth noting that ECNDNet and DnCNN are single-channel networks, and DudeNet and UDRN are two-channel networks, while our proposed MFDRAN is a multi-CNN. The data show that although our MFDRAN model has a higher number of parameters due to its multi-CNN structure, its running time is significantly better than that of the DudeNet model in the dual-channel network. In addition, according to the experimental results in Table 7, MFDRAN outperforms both the single-channel and dual-channel networks in terms of image denoising performance. Compared with the other compared models, MFDRAN not only shows an advantage in denoising quality, but also demonstrates a better balance in terms of running efficiency, which gives it a significant advantage in terms of comprehensive performance and further highlights its potential for application in complex tasks.

Table 10. Running time and parameters of different models in a single training session.

Models	**Training time (s)**	**Parameters (M)**
ECNDNet	225.6290	0.520M
DnCNN	197.7364	0.558M
DudeNet	832.2255	1.080M
UDRN	439.2483	0.820M
MFDRAN	760.2237	1.301M

5. Conclusions

In this paper, we propose a brain image denoising model called MFDRAN, which incorporates GSN, DRAN, FDN, and FPB to form a multi-CNN to improve the accuracy and effectiveness of brain image denoising. The preprocessing part of the model adopts an improved WA, which combines the multiple edge extraction method with the morphological gradient, Sobel's operator, and Canny's operator to effectively extract the foreground information and gradient features of the image, which lays a reliable foundation for the subsequent denoising and, at the same time, shows excellent performance in terms of segmentation accuracy and noise suppression. The DRAN internally combines the improved channel attention mechanism and spatial attention mechanism, which retains the key features of the image through multi-layer feature fusion and adaptive information filtering, and significantly enhances the clarity and diagnostic details of the brain image. The experimental results show that the MFDRAN model has significant advantages in suppressing redundant image information and enhancing high-frequency details, providing an effective way to improve brain image quality. The application potential in medical image denoising was further validated, especially in enhancing brain image clarity and diagnostic accuracy. Although the MFDRAN model performs well in denoising and image quality enhancement, the current experiments mainly focus on Gaussian noise models, and the types and properties of noise in real applications are more complex. Therefore, future research should consider introducing more real noise models or using medical image datasets containing real noise to verify the effectiveness and robustness of the proposed method in complex noise environments. To improve the generalization ability of the model, combining deep learning with traditional image processing techniques can be explored. With these optimizations, the MFDRAN model is expected to be more widely used in real medical environments.

Use of AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the China Postdoctoral Science Foundation under grant 2020M68924.

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. P. S. Chaitanya, S. K. Satpathy, A multilevel de-noising approach for precision edge-based fragmentation in MRI brain tumor segmentation, *Trait. Signal*, **40** (2023), 1715–1722. https://doi.org/10.18280/ts.400440

2. K. Gong, K. Johnson, G. El Fakhri, Q. Li, T. Pan, PET image denoising based on denoising diffusion probabilistic model, *Eur. J. Nucl. Med. Mol. Imaging*, **51** (2024), 358–368. https://doi.org/10.1007/s00259-023-06417-8

3. H. Aetesam, S. K. Maji, Deep variational magnetic resonance image denoising via network conditioning, *Biomed. Signal Process. Control*, **95** (2024), 106452. https://doi.org/10.1016/j.bspc.2024.106452

4. A. Sharma, V. Chaurasia, MRI denoising using advanced NLM filtering with non-subsampled shearlet transform, *Signal Image Video Process.*, **15** (2021), 1331–1339. https://doi.org/10.1007/s11760-021-01864-y

5. S. V. M. Sagheer, S. N. George, A review on medical image denoising algorithms, *Biomed. Signal Process. Control*, **61** (2020), 102036. https://doi.org/10.1016/j.bspc.2020.102036

6. J. Zhang, W. Gong, L. Ye, F. Wang, Z. Shangguan, Y. Cheng, A review of deep learning methods for denoising of medical low-dose CT images, *Comput. Biol. Med.*, **171** (2024), 108112. https://doi.org/10.1016/j.compbiomed.2024.108112

7. J. Pan, Q. Zuo, B. Wang, C. L. P. Chen, B. Lei, S. Wang, DecGAN: Decoupling generative adversarial network for detecting abnormal neural circuits in Alzheimer's disease, *IEEE Trans. Artif. Intell.*, **5** (2024), 5050–5063. https://doi.org/10.1109/TAI.2024.3416420

8. M. Li, R. Idoughi, B. Choudhury, W. Heidrich, Statistical model for OCT image denoising, *Biomed. Opt. Express*, **8** (2017), 3903–3917. https://doi.org/10.1364/BOE.8.003903

9. S. Dolui, A. Kuurstra, I. C. S. Patarroyo, O.V. Michailovich, A new similarity measure for non-local means filtering of MRI images, *J. Visual Commun. Image Representation*, **24** (2013), 1040–1054. https://doi.org/10.1016/j.jvcir.2013.06.011

10. C. Swarup, K. U. Singh, A. Kumar, S. K. Pandey, N. Varshney, T. Singh, Brain tumor detection using CNN, AlexNet & GoogLeNet ensembling learning approaches, *Electron. Res. Arch.*, **31** (2023), 2900–2924. https://doi.org/10.3934/era.2023146

11. K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising, *IEEE Trans. Image Process.*, **26** (2017), 3142–3155. https://doi.org/10.1109/TIP.2017.2662206

12. C. Tian, Y. Xu, L. Fei, J. Wang, J. Wen, N. Luo, Enhanced CNN for image denoising, *CAAI Trans. Intell. Technol.*, **4** (2019), 17–23. https://doi.org/10.1049/trit.2018.1054

13. K. Zhang, W. Zuo, L. Zhang, FFDNet: Toward a fast and flexible solution for CNN-based image denoising, *IEEE Trans. Image Process.*, **27** (2018), 4608–4622. https://doi.org/10.1109/TIP.2018.2839891

14. W. Jifara, F. Jiang, S. Rho, M. Cheng, S. Liu, Medical image denoising using convolutional neural network: A residual learning approach, *J. Supercomput.*, **75** (2019), 704–718. https://doi.org/10.1007/s11227-017-2080-0

15. C. Tian, Y. Xu, W. Zuo, Image denoising using deep CNN with batch renormalization, *Neural Networks*, **121** (2020), 461–473. https://doi.org/10.1016/j.neunet.2019.08.022

16. M. S. Hema, Sowjanya, N. Sharma, G. Abhishek, G. Shivani, P. P. Kumar, Identification and classification of brain tumor using convolutional neural network with autoencoder feature selection, in *International Conference on Emerging Technologies in Computer Engineering*, **1591** (2022), 251–258. https://doi.org/10.1007/978-3-031-07012-9_22

17. C. Tian, M. Zheng, W. Zuo, B. Zhang, Y. Zhang, D. Zhang, Multi-stage image denoising with the wavelet transform, *Pattern Recognit.*, **134** (2023), 109050. https://doi.org/10.1016/j.patcog.2022.109050

18. T. Li, Z. Zhang, M. Zhu, Z. Cui, D. Wei, Combining transformer global and local feature extraction for object detection, *Complex Intell. Syst.*, **10** (2024), 4897–4920. https://doi.org/10.1007/s40747-024-01409-z

19. Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-Net: Efficient channel attention for deep convolutional neural networks, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 11531–11539. https://doi.org/10.1109/CVPR42600.2020.01155

20. D. Hong, C. Huang, C. Yang, J. Li, Y. Qian, C. Cai, FFA-DMRI: A network based on feature fusion and attention mechanism for brain MRI denoising, *Front. Neurosci.*, **14** (2020), 577937. https://doi.org/10.3389/fnins.2020.577937

21. Y. Chen, R. Xia, K. Yang, K. Zou, MFMAM: Image inpainting via multi-scale feature module with attention module, *Comput. Vision Image Understanding*, **238** (2024), 103883. https://doi.org/10.1016/j.cviu.2023.103883

22. J. Deng, C. Hu, Recovering a clean background: A new progressive multi-scale CNN for image denoising, *Signal Image Video Process.*, **18** (2024), 4541–4552. https://doi.org/10.1007/s11760-024-03093-5

23. M. Duong, B. N. Thi, S. Lee, M. Hong, Multi-branch network for color image denoising using dilated convolution and attention mechanisms, *Sensors*, **24** (2024), 3608. https://doi.org/10.3390/s24113608

24. Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image restoration, *IEEE Trans. Pattern Anal. Mach. Intell.*, **43** (2021), 2480–2495. https://doi.org/10.1109/TPAMI.2020.2968521

25. F. Gao, Y. Wang, Z. Yang, Y. Ma, Q. Zhang, Single image super-resolution based on multi-scale dense attention network, *Soft Comput.*, **27** (2023), 2981–2992. https://doi.org/10.1007/s00500-022-07456-3

26. Y. Li, T. Xie, D. Mei, Application of convolutional neural networks for parallel multi-scale feature extraction in noise image denoising, *IEEE Access*, **12** (2024), 98599–98610. https://doi.org/10.1109/ACCESS.2024.3427143

27. Y. Zhang, C. Wang, X. Lv, Y. Song, Attention-driven residual-dense network for no-reference image quality assessment, *Signal Image Video Process.*, **18** (2024), 537–551. https://doi.org/10.1007/s11760-024-03172-7

28. Z. Hui, X. Wang, X. Gao, Fast and accurate single image super-resolution via information distillation network, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 723–731. https://doi.org/10.1109/CVPR.2018.00082

29. J. Liu, J. Tang, G. Wu, Residual feature distillation network for lightweight image super-resolution, in *European Conference on Computer Vision*, **12537** (2020), 41–55. https://doi.org/10.1007/978-3-030-67070-2_2

30. Y. Zhang, Y. Liu, Q. Li, J. Wang, M. Qi, H. Sun, et al., A lightweight fusion distillation network for image deblurring and deraining, *Sensors*, **21** (2021), 5312. https://doi.org/10.3390/s21165312

31. Z. Zong, L. Zha, J. Jiang, X. Liu, Asymmetric information distillation network for lightweight super resolution, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, (2022), 1248–1257. https://doi.org/10.1109/CVPRW56347.2022.00131

32. Z. Yu, K. Xie, C. Wen, J. He, W. Zhang, A lightweight image super-resolution reconstruction algorithm based on the residual feature distillation mechanism, *Sensors*, **24** (2024), 1049. https://doi.org/10.3390/s24041049

33. M. Jiang, C. You, M. Wang, H. Zhang, Z. Gao, D. Wu, et al., Controllable deep learning denoising model for ultrasound images using synthetic noisy image, in *Computer Graphics International Conference*, **14495** (2024), 297–308. https://doi.org/10.1007/978-3-031-50069-5_25

34. W. Wu, S. Liu, Y. Xia, Y. Zhang, Dual residual attention network for image denoising, *Pattern Recognit.*, **149** (2024), 110291. https://doi.org/10.1016/j.patcog.2024.110291

35. W. Wu, G. Lv, S. Liao, Y. Zhang, FEUNet: A flexible and effective U-shaped network for image denoising, *Signal Image Video Process.*, **17** (2023), 2545–2553. https://doi.org/10.1007/s11760-022-02471-1

36. R. K. Thakur, S. K. Maji, Multi scale pixel attention and feature extraction based neural network for image denoising, *Pattern Recognit.*, **141** (2023), 109603. https://doi.org/10.1016/j.patcog.2023.109603

37. C. Tian, Y. Xu, W. Zuo, B. Du, C. Lin, D. Zhang, Designing and training of a dual CNN for image denoising, *Knowl. Based Syst.*, **226** (2021), 106949. https://doi.org/10.1016/j.knosys.2021.106949

38. J. Yang, H. Xie, N. Xue, A. Zhang, Research on underwater image denoising based on dual-channel residual network, *Comput. Eng.*, **49** (2023), 188–198. https://doi.org/10.19678/j.issn.1000-3428.0064662

39. S. Ghaderi, S. Mohammadi, K. Ghaderi, F. Kiasat, M. Mohammadi, Marker-controlled watershed algorithm and fuzzy C-means clustering machine learning: automated segmentation of glioblastoma from MRI images in a case series, *Annal. Med. Surg.*, **86** (2024), 1460–1475. https://doi.org/10.1097/MS9.0000000000001756

40. V. Sivakumar, N. Janakiraman, A novel method for segmenting brain tumor using modified watershed algorithm in MRI image with FPGA, *Biosystems*, **198** (2020), 104226. https://doi.org/10.1016/j.biosystems.2020.104226

41. Y. Liang, J. Fu, Watershed algorithm for medical image segmentation based on morphology and total variation model, *Int. J. Pattern Recognit Artif Intell.*, **33** (2019), 1954019. https://doi.org/10.1142/S0218001419540193

42. Y. Wu, Q. Li, The algorithm of watershed color image segmentation based on morphological gradient, *Sensors*, **22** (2022), 8202. https://doi.org/10.3390/s22218202

43. T. Yesmin, H. Lohiya, P. P. Acharjya, Detection and segmentation of brain tumor by using modified watershed algorithm and thresholding to reduce over-segmentation, in *2023 IEEE International Conference on Contemporary Computing and Communications (InC4)*, (2023), 1–6. https://doi.org/10.1109/InC457730.2023.10262891

44. X. Xu, S. Xu, L. Jin, E. Song, Characteristic analysis of Otsu threshold and its applications, *Pattern Recognit. Lett.*, **32** (2011), 956–961. https://doi.org/10.1016/j.patrec.2011.01.021

45. C. Kumari, A. Mustafi, A novel radial kernel watershed basis segmentation algorithm for color image segmentation, *Wireless Pers. Commun.*, **133** (2023), 2105–2124. https://doi.org/10.1007/s11277-023-10831-4

46. C. Zhang, J. Fang, Edge detection based on improved sobel operator, in *Proceedings of the 2016 International Conference on Computer Engineering and Information Systems*, (2016), 129–132. https://doi.org/10.2991/ceis-16.2016.25

47. B. Chen, W. Wu, Z. Li, T. Han, Z. Chen, W. Zhang, Attention-guided cross-modal multiple feature aggregation network for RGB-D salient object detection, *Electron. Res. Arch.*, **32** (2024), 643–669. https://doi.org/10.3934/era.2024031

48. Z. Cai, L. Xu, J. Zhang, Y. Feng, L. Zhu, F. Liu, ViT-DualAtt: An efficient pornographic image classification method based on Vision Transformer with dual attention, *Electron. Res. Arch.*, **32** (2024), 6698–6716. https://doi.org/10.3934/era.2024313

49. Y. Chen, R. Xia, K. Yang, K. Zou, DARGS: Image inpainting algorithm via deep attention residuals group and semantics, *J. King Saud Univ. Comput. Inf. Sci.*, **35** (2023), 101567. https://doi.org/10.1016/j.jksuci.2023.101567

50. J. Yang, F. Xie, H. Fan, Z. Jiang, J. Liu, Classification for dermoscopy images using convolutional neural networks based on region average pooling, *IEEE Access*, **6** (2018), 65130–65138. https://doi.org/10.1109/ACCESS.2018.2877587

51. H. Wu, X. Gu, Max-pooling dropout for regularization of convolutional neural networks, in *International Conference on Neural Information Processing*, **9489** (2015), 46–54. https://doi.org/10.1007/978-3-319-26532-2_6

52. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90

53. G. Huang, Z. Liu, L. V. D. Maaten, K. Q. Weinberger, Densely connected convolutional networks, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 2261–2269. https://doi.org/10.1109/CVPR.2017.243

54. Z. Feng, Y. Chen, L. Xie, Unsupervised anomaly detection via knowledge distillation with non-directly-coupled student block fusion, *Mach. Vision Appl.*, **34** (2023), 104. https://doi.org/10.1007/s00138-023-01454-7

55. X. Zhao, X. Cai, Y. Xue, Y. Liao, L. Lin, T. Zhao, UKD-Net: efficient image enhancement with knowledge distillation, *J. Electron. Imaging*, **33** (2024), 023024. https://doi.org/10.1117/1.JEI.33.2.023024

56. Y. Li, J. Cao, Z. Li, S. Oh, N. Komuro, Lightweight single image super-resolution with dense connection distillation network, *ACM Trans. Multimedia Comput. Commun. Appl.*, **17** (2021), 1–17. https://doi.org/10.1145/3414838

57. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, preprint, arXiv:1412.6980.

58. K. A. Johnson, J. A. Becker, The Whole Brain Atlas, 2005. Available from: https://www.med.harvard.edu/aanlib/home.html.

59. S. Roth, M. J. Black, Fields of experts: A framework for learning image priors, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, **2** (2005), 860–867. https://doi.org/10.1109/CVPR.2005.160

60. True Color Kodak Images, Kodak Lossless True Color Image Suite: PhotoCD PCD0992, 2013. Available from: https://r0k.us/graphics/kodak/.

61. A. Abdelhamed, S. Lin, M. S. Brown, A high-quality denoising dataset for smartphone cameras, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 1692–1700. https://doi.org/10.1109/CVPR.2018.00182

62. M. Lebrun, M. Colom, J. Morel, The noise clinic: A blind image denoising algorithm, *Image Process. On Line*, **5** (2015), 1–54. https://doi.org/10.5201/ipol.2015.125

AIMS Press