CPP-Net: Embracing Multi-Scale Feature Fusion into Deep Unfolding CP-PPA Network for Compressive Sensing

Zhen Guo
Northwestern Polytechnical University
Xi'an 710072, China
guozhen2022@mail.nwpu.edu.cn

Hongping Gan*
Northwestern Polytechnical University
Xi'an 710072, China
ganhongping@nwpu.edu.cn

Abstract

In the domain of compressive sensing (CS), deep unfolding networks (DUNs) have garnered attention for their good performance and certain degree of interpretability rooted in CS domain, achieved by marrying traditional optimization solvers with deep networks. However, current DUNs are ill-suited for the intricate task of capturing fine-grained image details, leading to perceptible distortions and blurriness in reconstructed images, particularly at low CS ratios, e.g., 0.10 and below. In this paper, we propose CPP-Net, a novel deep unfolding CS framework, inspired by the primal-dual hybrid strategy of the Chambolle and Pock Proximal Point Algorithm (CP-PPA). First, we derive three iteration submodules, $\mathbf{X}^{(k)}$, $\mathbf{V}^{(k)}$ and $\mathbf{Y}^{(k)}$, by incorporating customized deep learning modules to solve the sparse basis related proximal operator within CP-PPA. Second, we design the Dual Path Fusion Block (DPFB) to adeptly extract and fuse multi-scale feature information, enhancing sensitivity to feature information at different scales and improving detail reconstruction. Third, we introduce the Iteration Fusion Strategy (IFS) to effectively weight the fusion of outputs from diverse reconstruction stages, maximizing the utilization of feature information and mitigating the information loss during reconstruction stages. Extensive experiments demonstrate that CPP-Net effectively reduces distortion and blurriness while preserving richer image details, outperforming current state-of-the-art methods. Codes are available at https://github.com/ICSResearch/CPP-Net.

1. Introduction

Compressive Sensing (CS) stands as a pivotal signal processing technique, facilitating the recovery of sparse or compressible signals from significantly fewer measurements than required by the Shannon-Nyquist sampling the-

*Corresponding author.
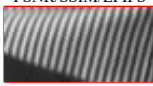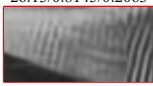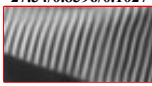
Figure 1. The PSNR (dB), SSIM and LPIPS [45] performance comparison of DGUNet$^+$ [24], OCTUF [31] and our CPP-Net at a CS ratio of 0.10. Our proposed CPP-Net outperforms other methods with higher image quality and improved human perception quality.

orem [5]. This capability accelerates signal acquisition, leading to a substantial reduction in sampling costs and data storage requirements. Furthermore, its advent has spurred advancements in various research domains, encompassing signal processing [4] in diverse areas such as applied mathematics [39], computational imaging [9, 10, 23, 34], computer science [49], *etc.* These explorations have profoundly propelled the development of CS theory, giving rising to various practical applications, including but not limited to natural image CS [20, 40, 41] and CS magnetic resonance imaging (CS-MRI) [22, 33, 36].

The CS sampling process can be expressed as $\mathbf{b} = \mathbf{\Phi x}$, where $\mathbf{x} \in \mathbb{R}^N$ denotes the original signal, $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ ($M \ll N$) signifies the sampling matrix, $\mathbf{b} \in \mathbb{R}^M$ represents the measurements derived from $\mathbf{x}$, and the CS ratio τ is defined as $\frac{M}{N}$. As $M \ll N$, the primary objective of CS is to recover $\mathbf{x}$ from this underdetermined system. Conventional CS methods are designed to address the Basis Pursuit De-Noising (BPDN) problem for signal recovery, mathematically represented as follows:

$$\min_{\mathbf{x}} \lambda \|\mathbf{\Psi x}\|_{\ell_1} + \frac{1}{2}\|\mathbf{\Phi x} - \mathbf{b}\|_{\ell_2}^2, \qquad (1)$$

Figure 2. Overall architecture of our proposed CPP-Net, which consists of K reconstruction stages. $\overline{\mathbf{x}}$ is the original image, the function $\mathcal{F}_{vec}(\cdot)$ divides the images into non-overlapping blocks and then flattens them into ve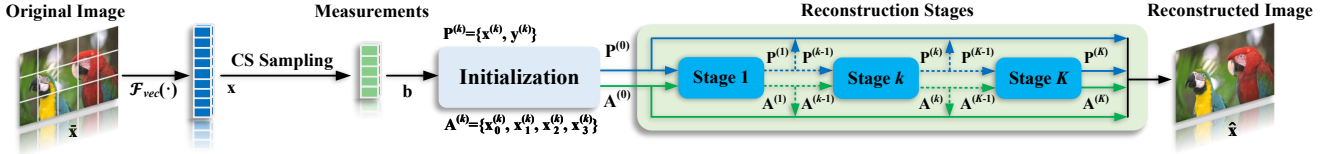ctors, $\mathbf{x}$ represents the obtained vectorized image blocks, $\mathbf{b}$ is the measurements, $\mathbf{P}^{(k)}$ and $\mathbf{A}^{(k)}$ represent the set of iteration auxiliary variables, and $\hat{\mathbf{x}}$ is the output reconstructed image.

where λ is the regularization parameter, $\mathbf{\Psi} \in \mathbb{R}^{N \times N}$ denotes the orthogonal sparse basis, $\| \cdot \|_{\ell_1}$ denotes the ℓ_1-norm, and $\| \cdot \|_{\ell_2}$ denotes the Euclidean norm.

Traditional optimization algorithms [2, 3, 12, 15] aim to discover solutions that not only align with the measurements but also encourage sparsity in the sparse domain. These approaches excels at handling problems with sparse constraints and offers robust theoretical foundations. However, they often demand high computational complexity and require manual parameter selection and tuning for various datasets. In recent years, deep learning (DL) has been introduced into the field of image CS, giving rise to pure DL-based CS methods [13, 14, 21, 25, 28, 32, 37, 48], which employ Convolutional Neural Networks (CNNs) or Transformers to learn the mapping relationship between measurements and original images. However, these pure DL-based methods remain a black box with limited transparency and interpretability, particularly within the CS domain, constraining their broader utility and further development. To bridge the gap between DL and traditional optimization algorithms, deep unfolding networks (DUNs) [29, 31, 36, 41, 43, 46] incorporate DL architectures into the iterative steps of optimization algorithms. This combination not only brings the benefits of rapid reconstruction and adaptive learning of parameters, but also introduces a degree of interpretability rooted in the CS domain.

However, existing DUNs struggle to capture fine-grained image details, resulting in noticeable degradation, including distortions and blurriness, especially at low CS ratios, *e.g.*, 0.10 or lower. This is primarily due to two key challenges:

- **Inadequate extraction and fusion of multi-scale feature information**: Current DUNs often focus on optimizing images at high CS ratios, and are inadequate in extracting and fusing feature information at various scales from images, resulting in a lack of sensitivity to fine-grained image features. This deficiency compromises their ability to process fine-grained details accurately, such as subtle textures, edges, and intricate details when measurements are limited.
- **Information fusion deficiency of different reconstruction stages**: DUNs typically consist of multiple iterative reconstruction stages responsible for extracting and processing specific image information. However, existing

methods often struggle to balance and fuse this information effectively, resulting in information loss and confusion. This flawed information fusion strategy can impact DUNs' performance, as different stages of information may have varying importance and contributions to the final reconstruction.

Furthermore, image details are pivotal for image content recognition and analysis. Particularly at low CS ratios, enhancing image details becomes especially vital to fully exploit the advantages of CS, such as resource-efficient transmission and storage. Consequently, the reconstruction of image details at low CS ratios is a vital and challenging task to ensure high-quality image perception.

To address these challenges, we propose CPP-Net, a novel deep unfolding framework inspired by the Chambolle and Pock Proximal Point Algorithm (CP-PPA), as illustrated in Fig. 2. By integrating a customized DL module into the iterative optimization solvers of CP-PPA to address the problem of sparse basis related to proximal operator, we derive and establish the deep unfolding CP-PPA framework, comprising three iteration submodules: $\mathbf{X}^{(k)}$, $\mathbf{V}^{(k)}$, and $\mathbf{Y}^{(k)}$. Subsequently, building upon the proposed framework, we design the Dual Path Fusion Block (DPFB), which employs up and down sampling as well as Multi-Scale Transformer Block (MSTB) to proficiently extract and fuse multi-scale feature information. Additionally, we introduce the Iteration Fusion Strategy (IFS) by concatenating outputs from different reconstruction stages in the channel dimension and applying attentional weighting for adaptive fusion. The main contributions are summarized as follows:

- We propose CPP-Net, a novel deep unfolding framework for CS, drawing inspiration from the primal-dual hybrid strategy of CP-PPA, achieving superior reconstructed image and human perception quality.
- We design the Dual Path Fusion Block (DPFB) as an extension of the sparse basis structure associated with the proximal operator within CPP-Net to adeptly extract and fuse multi-scale feature information, enhancing sensitivity at different scales and improving detail reconstruction.
- We introduce the Iteration Fusion Strategy (IFS) to effectively weight the fusion of information from different reconstruction stages in CPP-Net, maximizing feature uti-
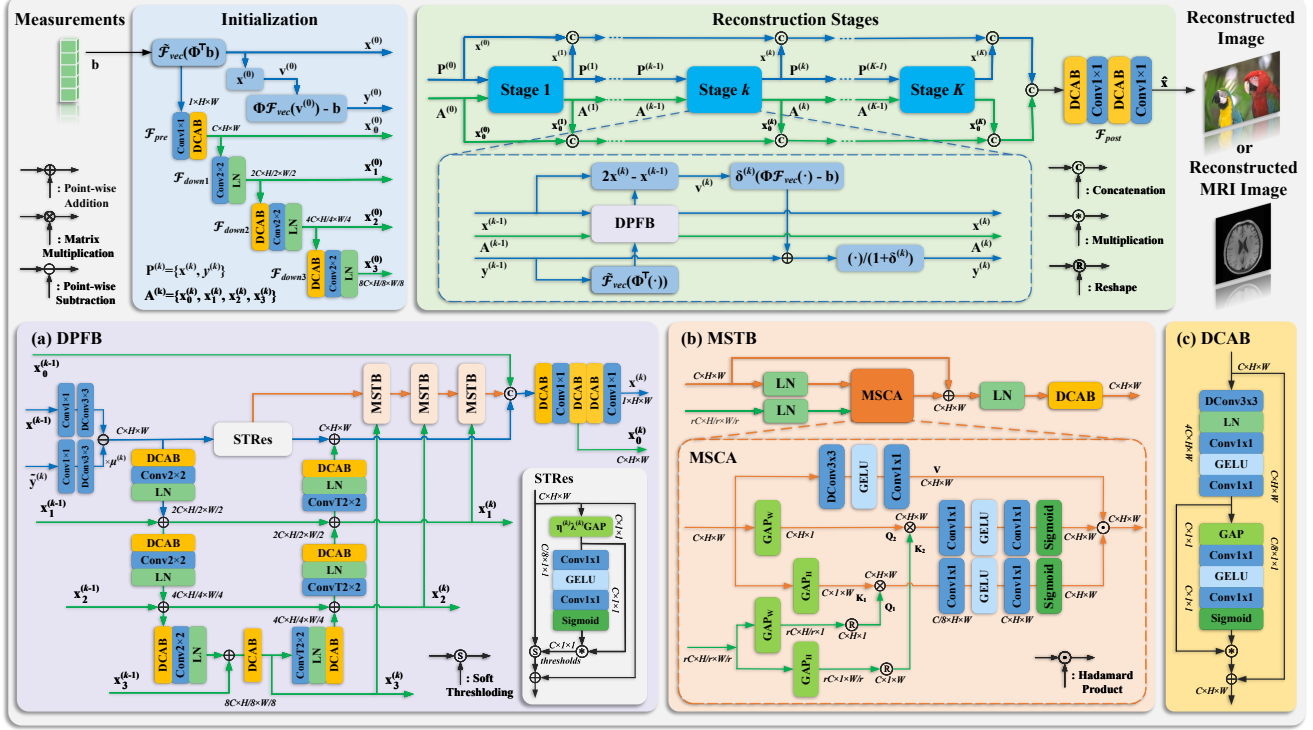
Figure 3. Details of our proposed CPP-Net. We present the details of initialization and reconstruction stages in the first row. The second row demonstrates the details of components in CPP-Net, *i.e.*, illustrations of (a) Dual Path Fusion Block (DPFB) and Soft Thresholding Residual connection (STRes); (b) Multi-Scale Transformer Block (MSTB) and Multi-Scale Cross Attention (MSCA); (c) Depth-wise Channel Attention Block (DCAB).

lization and enhancing image detail reconstruction while reducing information loss during reconstruction stages. Extensive experiments demonstrate that our proposed CPP-Net outperforms current state-of-the-art (SOTA) methods by preserving richer image details, reducing distortion and blurring, leading to higher image quality and improved human perception quality.

2. Related Works

2.1. Traditional Chambolle and Pock Proximal Point Algorithm

The Chambolle and Pock Proximal Point Algorithm (CP-PPA) [6, 16, 17] is a straightforward yet powerful optimization algorithm that utilizes an enhanced primal-dual hybrid approach [50], offering convergence and efficiency. Consider the following saddle-point problem:

$$\min_{\mathbf{x} \in \mathbb{R}^N} \max_{\mathbf{y} \in \mathbb{R}^M} \Theta(\mathbf{x}, \mathbf{y}) := \theta_1(\mathbf{x}) - \mathbf{y}^\top \Phi \mathbf{x} - \theta_2(\mathbf{y}), \quad (2)$$

where $\Phi \in \mathbb{R}^{M \times N}$. The CP-PPA solves this problem through following iterative steps:

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} \Theta(\mathbf{x}, \mathbf{y}^{(k)}) + \frac{\alpha}{2} \|\mathbf{x} - \mathbf{x}^{(k)}\|_{\ell_2}^2, \quad (3)$$

$$\mathbf{v}^{(k+1)} = 2\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}, \quad (4)$$

$$\mathbf{y}^{(k+1)} = \arg\max_{\mathbf{y}} \Theta(\mathbf{v}^{(k+1)}, \mathbf{y}) - \frac{\beta}{2} \|\mathbf{y} - \mathbf{y}^{(k)}\|_{\ell_2}^2, \quad (5)$$

where α and β are regularization parameters.

CP-PPA addresses both the primal and dual problems alternately and iteratively, and achieves faster convergence compared to the widely-used Iterative Shrinkage/Thresholding Algorithm (ISTA) [2, 6], which solely optimizes the primal problem. Furthermore, CP-PPA provides stronger convergence guarantees, increasing its potential to theoretically attain globally optimal solutions [17]. However, as a traditional iterative optimization algorithm, CP-PPA has its limitations. Its performance heavily relies on parameter settings, particularly the choice of α and β. Inaccurate parameter choices can lead to slow convergence or system instability [17]. Additionally, for the problem outlined in Eq. (1), a persistent challenge remains in the selection of an appropriate sparse basis Ψ due to the diverse content and characteristics of images [1].

2.2. Deep Unfolding Networks

The core idea of DUNs is to unfold the traditional iterative optimization algorithms into the form of deep networks,

making it an end-to-end trainable model. DUNs offer the combined advantages of rapid reconstruction and adaptive learning of parameters, while introducing interpretability rooted in the CS domain.

Since Zhang and Ghanem proposed ISTA-Net [41], based on the famous ISTA algorithm (a special case of Proximal Gradient Descent (PGD) algorithm for ℓ_1-regularization term), these are many ISTA-based network variants have been gradually proposed for CS, including OPINE-Net$^+$ [42], TransCS [27], OCTUF [31], DGUNet$^+$ [24], and others [7, 8, 24, 29, 30]. Similarly, Yang *et al.* proposed ADMM-CSNet [36] based on the Alternating Direction Method of Multipliers (ADMM) [3]; Zhang *et al.* proposed AMP-Net [46] inspired by the Approximate Message Passing (AMP) [12] for CS, and so on. Nevertheless, current DUNs usually neglect the enhancement of details at different scales in images at low CS ratios, and at the same time are insufficient for the transfer and utilization of information from different stages, resulting in significant degradation, *e.g.*, distortions and blurriness, in reconstructed images. For example, OCTUF [31] utilizes cross-attention mechanisms to optimize the image but disregards the finer image details of various scales, leading to substantial areas of blurring. Similarly, while DGUNet$^+$ [24] utilizes downsampling for encoding and up-sampling for decoding to unfold PGD, it lacks effective fusion of feature information across different scales. This deficiency results in noticeable distortions in the reconstructed images. In this paper, we propose CPP-Net, a novel deep unfolding framework inspired by the CP-PPA with efficient multi-scale feature fusion to achieve precise image detail reconstruction.

3. Proposed Method

The details of our CPP-Net are illustrated in Fig. 3. CPP-Net includes one initialization module and K reconstruction stages, with the default setting of eight stages. Each stage represents one iteration in the CP-PPA, and consists of three iteration submodules ($\mathbf{X}^{(k)}$, $\mathbf{V}^{(k)}$ and $\mathbf{Y}^{(k)}$) and one DPFB.

3.1. Initialization Module

As shown in Fig. 3, initialization module responds to an initial estimate $\mathbf{P}^{(0)}$ from the measurements and initialize the iteration parameters $\mathbf{A}^{(0)}$. $\mathbf{P}^{(k)}$ contains the iterative auxiliary variables ($\mathbf{x}^{(k)}$ and $\mathbf{y}^{(k)}$), while $\mathbf{A}^{(k)}$ contains the downsampling results at different scales ($\mathbf{x}_0^{(k)}$, $\mathbf{x}_1^{(k)}$, $\mathbf{x}_2^{(k)}$ and $\mathbf{x}_3^{(k)}$) in the k-th reconstruction stages. In order to facilitate convolutional operations and iterative computations, we use $\mathcal{F}_{vec}(\cdot)$ and $\widetilde{\mathcal{F}}_{vec}(\cdot)$ to denote the chunking and flattening functions as well as their inverse processes, respectively.

3.2. Reconstruction Stages

Iteration Submodules. We introduce a DL-based operator, denoted as $\mathcal{D}(\cdot)$, to replace $\boldsymbol{\Psi}$, which offers enhanced adaptability to various images and a high degree of self-adjustment without the need for manual parameter tuning. We define two functions: $f(\mathbf{x}) = \lambda \|\mathcal{D}(\mathbf{x})\|_{\ell_1}$ and $h(\mathbf{z}) = \frac{1}{2}\|\mathbf{z} - \mathbf{b}\|_{\ell_2}^2$. Therefore, we reformulate the Eq. (1) as a saddle-point problem, expressed as:

$$\min_{\mathbf{x}\in\mathbb{R}^N} \max_{\mathbf{y}\in\mathbb{R}^M} \Theta(\mathbf{x},\mathbf{y}) := f(\mathbf{x}) + \mathbf{y}^\top \boldsymbol{\Phi}\mathbf{x} - h^*(\mathbf{y}). \quad (6)$$

Here, h^* is the convex conjugate of the convex lower-semicontinuous function h:

$$h^*(\mathbf{y}) = \sup_{\mathbf{z}\in\mathbb{R}^M} \mathbf{z}^\top \mathbf{y} - \frac{1}{2}\|\mathbf{z} - \mathbf{b}\|_{\ell_2}^2. \quad (7)$$

By alternately iterating and optimizing the primal and dual problems, and independently learning regularization terms α, β and λ for each iteration, iterative steps can be derived as follows:

$$\mathbf{x}^{(k)} = \arg\min_{\mathbf{x}} \Theta(\mathbf{x},\mathbf{y}^{(k-1)}) + \frac{\alpha^{(k)}}{2}\|\mathbf{x} - \mathbf{x}^{(k-1)}\|_{\ell_2}^2, \quad (8)$$

$$\mathbf{v}^{(k)} = 2\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}, \quad (9)$$

$$\mathbf{y}^{(k)} = \arg\max_{\mathbf{y}} \Theta(\mathbf{v}^{(k)},\mathbf{y}) - \frac{\beta^{(k)}}{2}\|\mathbf{y} - \mathbf{y}^{(k-1)}\|_{\ell_2}^2. \quad (10)$$

Since the constant term is independent of the solution, the solution for Eq. (8) can be expressed as:

$$\mathbf{x}^{(k)} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + (\mathbf{y}^{(k-1)})^\top \boldsymbol{\Phi}\mathbf{x} + \frac{1}{2\eta^{(k)}}\|\mathbf{x} - \mathbf{x}^{(k-1)}\|_{\ell_2}^2$$
$$= \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{1}{2\eta^{(k)}}\|\mathbf{x} - \mathbf{x}^{(k-1)} + \eta^{(k)}\boldsymbol{\Phi}^\top \mathbf{y}^{(k-1)}\|_{\ell_2}^2, \quad (11)$$

where $\eta^{(k)} = 1/\alpha^{(k)}$. $\mathbf{x}^{(k)}$ can be obtained using the soft thresholding function denoted as $\mathcal{S}(\cdot)$, a special proximal operator for ℓ_1-norm:

$$\mathbf{x}^{(k)} = \widetilde{\mathcal{D}}(\mathcal{S}_{\lambda^{(k)}\eta^{(k)}}(\mathcal{D}(\mathbf{x}^{(k-1)} - \eta^{(k)}\boldsymbol{\Phi}^\top \mathbf{y}^{(k-1)}))), \quad (12)$$

where

$$\mathcal{S}_\epsilon(\mathbf{x}) = \text{sgn}(\mathbf{x}) * \max(|\mathbf{x}| - \epsilon, 0), \quad (13)$$

and $\widetilde{\mathcal{D}}(\cdot)$ represents the left inverse of $\mathcal{D}(\cdot)$. Similarly, solving for $\mathbf{y}^{(k)}$ in Eq. (10) is equivalent to:

$$\mathbf{y}^{(k)} = \arg\min_{\mathbf{y}} h^*(\mathbf{y}) - \mathbf{y}^\top \boldsymbol{\Phi}\mathbf{v}^{(k)} + \frac{1}{2\delta^{(k)}}\|\mathbf{y} - \mathbf{y}^{(k-1)}\|_{\ell_2}^2$$
$$= \arg\min_{\mathbf{y}} h^*(\mathbf{y}) + \frac{1}{2\delta^{(k)}}\|\mathbf{y} - \mathbf{y}^{(k-1)} - \delta^{(k)}\boldsymbol{\Phi}\mathbf{v}^{(k)}\|_{\ell_2}^2$$
$$= \text{prox}_{\delta^{(k)}h^*}(\mathbf{y}^{(k-1)} + \delta^{(k)}\boldsymbol{\Phi}\mathbf{v}^{(k)})$$
$$= \frac{1}{1 + \delta^{(k)}}(\mathbf{y}^{(k-1)} + \delta^{(k)}(\boldsymbol{\Phi}\mathbf{v}^{(k)} - \mathbf{b})), \quad (14)$$

where $\mathrm{prox}(\cdot)$ denotes the proximal operator and $\delta^{(k)} = 1/\beta^{(k)}$.

In summary, for the sake of convenience in iterative and convolutional operations, we can express the explicit solutions for the iteration submodules $\mathbf{X}^{(k)}$, $\mathbf{V}^{(k)}$, and $\mathbf{Y}^{(k)}$ in the k-th reconstruction stages as follows:

$$
\begin{cases}
\mathbf{X}^{(k)} : \mathbf{x}^{(k)} = \widetilde{\mathcal{D}}(\mathcal{S}_{\lambda^{(k)}\eta^{(k)}}(\mathcal{D}(\\
\qquad\qquad \mathbf{x}^{(k-1)} - \eta^{(k)}\widetilde{\mathcal{F}}_{vec}(\mathbf{\Phi}^\top \mathbf{y}^{(k-1)})))), \\
\mathbf{V}^{(k)} : \mathbf{v}^{(k)} = 2\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}, \\
\mathbf{Y}^{(k)} : \mathbf{y}^{(k)} = \dfrac{1}{1+\delta^{(k)}}(\mathbf{y}^{(k-1)} + \delta^{(k)}(\mathbf{\Phi}\mathcal{F}_{vec}(\mathbf{v}^{(k)}) - \mathbf{b})).
\end{cases}
\tag{15}
$$

The convergence analysis of the proposed framework is presented in the *Supplementary Material* (Sec. 2). The essence of our CPP-Net resides in the structure $\widetilde{\mathcal{D}}(\mathcal{S}(\mathcal{D}(\cdot)))$. Therefore, we design the DPFB as an extension of $\widetilde{\mathcal{D}}(\mathcal{S}(\mathcal{D}(\cdot)))$ to enhance CPP-Net's capability for characterizing information and to achieve effective extraction and fusion of multiscale features. DPFB consists mainly of DCABs, MSTBs and STRes. The specific details are presented below.

Depth-wise Channel Attention Block (DCAB). We introduce the DCAB as the fundamental building block to enhance the feature extraction and attentional weighting capabilities of our CPP-Net. As illustrated in Fig. 3(c), DCAB begins with a 3×3 depth-wise convolution (DConv_3) to extract spatial information, followed by a layer normalization (LN) to enhance stability. It then employs two 1×1 convolutions (Conv_1) to expand and reduce the channel count, with a Gaussian error linear units (GELU) activation in between. Additionally, DCAB incorporates channel attention block inspired by [18] to highlight informative features while suppressing less relevant ones across channels to achieve adaptive weighting.

Multi-Scale Transformer Block (MSTB). We design the MSTB, a composition of LN, Multi-Scale Cross Attention (MSCA) and DCAB, to improve the aggregation and fusion of feature information across various scales. As shown in Fig. 3(b), MSTB takes two inputs of varying scales, applies LN to each, and then processes them through the MSCA. Following the residual connection, the feature undergoes further refinement via the DCAB. Assuming that $\mathbf{x}^{(k)} \in \mathbb{R}^{C \times H \times W}$ and $\mathbf{x}_i^{(k)} \in \mathbb{R}^{rC \times H/r \times W/r}$ represent the inputs to the MSCA, the process unfolds as follows: Initially, $\mathbf{x}^{(k)}$ undergoes Global Average Pooling (GAP) along both the width ($\mathrm{GAP}_\mathrm{W} : \mathbb{R}^{C \times H \times W} \to \mathbb{R}^{C \times H \times 1}$) and height ($\mathrm{GAP}_\mathrm{H} : \mathbb{R}^{C \times H \times W} \to \mathbb{R}^{C \times 1 \times W}$) dimensions, producing Query $\mathbf{Q}_2$ and Key $\mathbf{K}_1$, respectively. Similarly, $\mathbf{x}_i^{(k)}$ generates the corresponding $\mathbf{Q}_1$ and $\mathbf{K}_2$ after reshaping. Moreover, $\mathbf{V}$ is derived from $\mathbf{x}^{(k)}$ by applying a depth-wise 3×3 convolution, followed by a GELU activation, and finally a 1×1 convolution. The process can be represented

---

Algorithm 1 CPP-Net

Input: Original image $\overline{\mathbf{x}}$, sampling matrix $\mathbf{\Phi}$, block size B, the number of reconstruction stages K.

Output: Final reconstructed image $\hat{\mathbf{x}}$.

1: **Sampling:** $\mathbf{x} = \mathcal{F}_{vec}(\overline{\mathbf{x}})$, $\mathbf{b} = \mathbf{\Phi}\mathbf{x}$.
2: **Initialization:** $k = 0$, $\mathbf{x}^{(0)} = \widetilde{\mathcal{F}}_{vec}(\mathbf{\Phi}^\top \mathbf{b})$, $\mathbf{v}^{(0)} = \mathbf{x}^{(0)}$, $\mathbf{y}^{(0)} = \mathbf{\Phi}\mathcal{F}_{vec}(\mathbf{v}^{(0)}) - \mathbf{b}$, $\mathbf{x}_0^{(0)} = \mathcal{F}_{pre}(\mathbf{x}^{(0)})$, $\mathbf{x}_1^{(0)} = \mathcal{F}_{down1}(\mathbf{x}_0^{(0)})$, $\mathbf{x}_2^{(0)} = \mathcal{F}_{down2}(\mathbf{x}_1^{(0)})$, $\mathbf{x}_3^{(0)} = \mathcal{F}_{down3}(\mathbf{x}_2^{(0)})$, $\mathbf{A}^{(0)} = \{\mathbf{x}_0^{(0)}, \mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \mathbf{x}_3^{(0)}\}$.
3: **Reconstruction:**
4: **while** $k < K$ **do**
5: $\quad k \leftarrow k + 1$;
6: $\quad \widetilde{\mathbf{y}}^{(k-1)} = \widetilde{\mathcal{F}}_{vec}(\mathbf{\Phi}^\top \mathbf{y}^{(k-1)})$;
7: $\quad \mathbf{x}^{(k)}, \mathbf{A}^{(k)} \leftarrow \mathrm{DPFB}^{(k)}(\mathbf{x}^{(k-1)}, \widetilde{\mathbf{y}}^{(k-1)}, \mathbf{A}^{(k-1)})$;
8: $\quad \mathbf{v}^{(k)} = 2\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$;
9: $\quad \mathbf{y}^{(k)} = \dfrac{1}{1+\delta^{(k)}}(\mathbf{y}^{(k-1)} + \delta^{(k)}(\mathbf{\Phi}\mathcal{F}_{vec}(\mathbf{v}^{(k)}) - \mathbf{b}))$;
10: **end while**
11: $\hat{\mathbf{x}} = \mathcal{F}_{post}(\mathrm{Concat}(\{\mathbf{x}^{(k)}\}_{k=0}^K, \{\mathbf{x}_0^{(k)}\}_{k=0}^K))$.

---

as follows:

$$
\begin{cases}
\mathbf{Q}_2 = \mathrm{GAP}_\mathrm{W}(\mathbf{x}^{(k)}), \mathbf{K}_1 = \mathrm{GAP}_\mathrm{H}(\mathbf{x}^{(k)}), \\
\mathbf{Q}_1 = \mathrm{R}(\mathrm{GAP}_\mathrm{W}(\mathbf{x}_i^{(k)})), \mathbf{K}_2 = \mathrm{R}(\mathrm{GAP}_\mathrm{H}(\mathbf{x}_i^{(k)})), \\
\mathbf{V} = \mathrm{Conv}_1(\mathrm{GELU}(\mathrm{DConv}_3(\mathbf{x}^{(k)}))).
\end{cases}
\tag{16}
$$

Here, $\mathrm{R}(\cdot)$ denotes the reshape function. Subsequently, the Queries and Keys generated by these two inputs undergo the matrix multiplication, resulting in two matrices in $\mathbb{R}^{C \times H \times W}$. These matrices are then subjected to channel downscaling and upscaling through two 1×1 convolutions, followed by the Sigmoid function to obtain the corresponding attention weights. The MSCA output, denoted as $\widetilde{\mathbf{x}}^{(k)}$, is computed by the Hadamard product of $\mathbf{V}$ with the obtained attention weights:

$$
\widetilde{\mathbf{x}}^{(k)} = \mathrm{SG}(\mathbf{Q}_1 \otimes \mathbf{K}_1) \odot \mathrm{SG}(\mathbf{Q}_2 \otimes \mathbf{K}_2) \odot \mathbf{V},
\tag{17}
$$

where $\mathrm{SG}(\cdot) = \mathrm{Sigmoid}(\mathrm{Conv}_1(\mathrm{GELU}(\mathrm{Conv}_1(\cdot))))$.

Dual Path Fusion Block (DPFB). Our approach prioritizes the efficient extraction and fusion of multi-scale features to enhance performance of details reconstruction. To achieve this, we design the DPFB as the cornerstone of our CPP-Net framework, as illustrated in Fig. 3(a). The DPFB begins with an initial step that includes a 1×1 convolution and a 3×3 depth-wise convolution to increase channel count. It then downsamples the feature maps using 2×2 convolutions with a stride of 2 (Conv_2), which reduces the feature maps' size by half while doubling the channel count, facilitating efficient capture of multi-scale features. Furthermore, the DPFB effectively fuses multi-scale information

Table 1. Average PSNR (dB)/SSIM performance comparisons of recent CS methods on various datasets at different CS ratios.

Datasets	CS Ratios	ISTA-Net+ (CVPR 2018)	AMP-Net (TIP 2021)	CASNet (TIP 2022)	DGUNet+ (CVPR 2022)	FSOINet (ICASSP 2022)	TransCS (TIP 2022)	AutoBCS (TCYB 2023)	CSformer (TIP 2023)	DPC-DUN (TIP 2023)	OCTUF (CVPR 2023)	TCS-Net (TCI 2023)	CPP-Net (Our Method)
General100 [11]	0.01	19.00/0.4700	22.71/0.6282	23.48/0.6480	22.86/0.6190	23.27/0.6363	21.66/0.5415	22.24/0.6164	23.35/0.6394	19.95/0.5363	23.31/0.6346	22.58/0.5978	24.17/0.6593
	0.04	23.76/0.6549	26.96/0.7695	28.50/0.8171	27.92/0.8078	28.39/0.8135	27.25/0.7843	27.10/0.7964	27.81/0.7986	26.61/0.7531	28.35/0.8122	26.57/0.7712	29.04/0.8284
	0.10	28.54/0.8104	30.82/0.8722	32.78/0.9099	32.41/0.9073	32.70/0.9085	31.39/0.8918	30.76/0.8927	31.60/0.8880	31.17/0.8716	32.77/0.9084	29.90/0.8748	33.16/0.9159
	0.25	34.32/0.9250	36.01/0.9508	38.07/0.9657	37.55/0.9645	38.13/0.9660	37.08/0.9604	35.92/0.9581	36.51/0.9558	36.50/0.9481	38.26/0.9666	34.63/0.9504	38.32/0.9674
	Avg.	26.41/0.7151	29.13/0.8052	30.71/0.8352	30.19/0.8247	30.62/0.8311	29.35/0.7945	29.01/0.8159	29.82/0.8205	28.56/0.7773	30.67/0.8305	28.42/0.7986	31.17/0.8428
LIVE29 [26]	0.01	19.15/0.4279	22.07/0.5448	22.31/0.5523	22.27/0.5503	22.37/0.5502	21.48/0.4968	21.46/0.5347	22.33/0.5516	19.81/0.4764	22.31/0.5458	22.01/0.5245	22.60/0.5567
	0.04	22.02/0.5625	24.89/0.6857	25.27/0.7136	25.32/0.7168	25.34/0.7150	24.87/0.6942	24.55/0.6991	25.27/0.7048	23.70/0.6470	25.26/0.7113	24.71/0.6904	25.52/0.7211
	0.10	25.05/0.7104	27.69/0.8067	28.16/0.8338	28.27/0.8379	28.23/0.8359	27.65/0.8195	27.07/0.8184	27.80/0.8148	26.83/0.7812	28.23/0.8345	27.21/0.8165	28.46/0.8414
	0.25	29.42/0.8648	32.01/0.9183	32.55/0.9303	32.56/0.9323	32.66/0.9326	32.05/0.9251	31.21/0.9227	31.82/0.9196	31.17/0.9008	32.69/0.9327	31.10/0.9208	32.82/0.9343
	Avg.	23.91/0.6414	26.67/0.7389	27.07/0.7575	27.11/0.7593	27.15/0.7584	26.51/0.7339	26.07/0.7437	26.81/0.7477	25.38/0.7014	27.12/0.7561	26.26/0.7381	27.35/0.7634
McM18 [44]	0.01	19.99/0.4942	23.78/0.6426	24.23/0.6538	23.05/0.6267	24.10/0.6464	22.81/0.5736	23.26/0.6248	23.66/0.6526	21.10/0.5553	23.87/0.6409	23.63/0.6144	24.57/0.6614
	0.04	24.27/0.6577	27.90/0.7879	28.48/0.8166	28.16/0.8091	28.50/0.8157	27.94/0.7976	27.54/0.8002	28.12/0.8030	26.51/0.7539	28.33/0.8120	27.54/0.7907	28.97/0.8277
	0.10	28.54/0.8104	31.68/0.8860	32.47/0.9100	32.32/0.9070	32.47/0.9097	31.88/0.8998	31.13/0.8973	31.69/0.8907	30.67/0.8701	32.49/0.9093	30.97/0.8913	32.97/0.9159
	0.25	33.99/0.9237	36.88/0.9560	37.77/0.9659	37.74/0.9655	37.85/0.9663	37.27/0.9627	36.25/0.9608	36.60/0.9570	35.86/0.9462	37.93/0.9667	35.89/0.9579	38.08/0.9676
	Avg.	26.70/0.7215	30.06/0.8181	30.74/0.8366	30.32/0.8271	30.73/0.8345	29.98/0.8084	29.55/0.8208	30.02/0.8258	28.54/0.7814	30.66/0.8322	29.51/0.8136	31.15/0.8432
OST300 [35]	0.01	19.36/0.4208	22.31/0.5288	22.47/0.5338	22.36/0.5306	22.49/0.5335	21.67/0.4826	21.65/0.5176	22.48/0.5299	20.12/0.4645	22.46/0.5298	22.28/0.5127	22.76/0.5400
	0.04	22.06/0.5475	24.92/0.6651	25.15/0.6911	25.24/0.6973	25.25/0.6953	24.86/0.6756	24.50/0.6756	25.19/0.6843	23.61/0.6249	25.19/0.6910	24.74/0.6728	25.39/0.7001
	0.10	24.78/0.6896	27.35/0.7859	27.66/0.8124	27.84/0.8187	27.75/0.8159	27.31/0.8018	26.87/0.7991	27.53/0.7950	26.25/0.7561	27.77/0.8148	27.04/0.8000	27.93/0.8207
	0.25	28.53/0.8433	31.06/0.9009	31.35/0.9135	31.53/0.9170	31.55/0.9171	31.07/0.9096	30.52/0.9083	31.05/0.9038	29.93/0.8792	31.60/0.9175	30.55/0.9084	31.67/0.9185
	Avg.	23.68/0.6253	26.41/0.7202	26.66/0.7377	26.74/0.7409	26.76/0.7405	26.23/0.7174	25.89/0.7255	26.56/0.7283	24.98/0.6812	26.76/0.7383	26.15/0.7235	26.94/0.7448
Set11 [21]	0.01	17.45/0.4131	20.20/0.5581	21.76/0.6019	22.15/0.6114	21.73/0.5937	20.15/0.5066	19.63/0.5605	21.86/0.6071	18.03/0.4601	21.75/0.5934	21.09/0.5505	22.19/0.6135
	0.04	21.56/0.6240	25.26/0.7722	26.25/0.8118	26.52/0.8230	26.37/0.8119	25.41/0.7883	24.73/0.7871	26.41/0.8058	24.38/0.7498	26.45/0.8126	25.46/0.7863	27.23/0.8337
	0.10	26.49/0.8036	29.40/0.8779	30.29/0.9005	30.93/0.9088	30.44/0.9018	29.54/0.8877	28.44/0.8827	30.09/0.8925	29.42/0.8801	30.70/0.9030	29.04/0.8834	31.27/0.9135
	0.25	32.44/0.9237	34.63/0.9481	35.65/0.9592	36.18/0.9616	35.80/0.9595	35.06/0.9548	33.56/0.9481	34.99/0.9534	34.75/0.9483	36.10/0.9604	33.94/0.9508	36.35/0.9631
	Avg.	24.49/0.6911	27.37/0.7891	28.49/0.8184	29.02/0.8262	28.59/0.8167	27.54/0.7844	26.59/0.7946	28.34/0.8147	26.65/0.7596	28.75/0.8174	27.38/0.7928	29.26/0.8310
Set14 [38]	0.01	18.22/0.4014	21.64/0.5433	22.03/0.5600	21.86/0.5409	22.00/0.5538	20.91/0.4853	20.93/0.5343	22.07/0.5493	19.04/0.4551	21.94/0.5500	21.64/0.5219	22.52/0.5694
	0.04	22.08/0.5708	25.50/0.7007	26.04/0.7330	25.88/0.7250	26.08/0.7324	25.50/0.7133	25.07/0.7153	25.87/0.7160	24.32/0.6630	26.04/0.7302	25.25/0.7073	26.54/0.7440
	0.10	26.00/0.7289	28.77/0.8183	29.37/0.8467	29.34/0.8455	29.35/0.8451	28.81/0.8343	28.00/0.8286	28.79/0.8214	28.03/0.7950	29.47/0.8454	28.19/0.8283	29.93/0.8537
	0.25	30.62/0.8700	33.21/0.9144	33.95/0.9308	33.70/0.9294	34.05/0.9309	33.37/0.9244	32.14/0.9203	32.95/0.9174	32.78/0.9023	34.18/0.9312	32.23/0.9206	34.41/0.9336
	Avg.	24.23/0.6428	27.28/0.7442	27.85/0.7676	27.70/0.7602	27.87/0.7656	27.15/0.7393	26.54/0.7496	27.42/0.7510	26.04/0.7039	27.91/0.7642	26.83/0.7445	28.35/0.7752
Urban100 [19]	0.01	16.67/0.3734	19.62/0.5025	20.08/0.5366	20.15/0.5335	19.87/0.5223	18.98/0.4398	19.23/0.4991	20.14/0.5298	17.31/0.4216	19.88/0.5167	19.61/0.4945	20.55/0.5554
	0.04	19.66/0.5370	22.81/0.6825	23.73/0.7412	24.05/0.7478	23.69/0.7376	23.27/0.7117	22.50/0.7029	24.03/0.7377	22.36/0.6768	23.68/0.7329	22.93/0.7036	24.66/0.7691
	0.10	23.51/0.7201	26.04/0.8151	27.40/0.8606	28.01/0.8709	27.53/0.8627	26.77/0.8418	25.36/0.8242	27.30/0.8483	26.96/0.8361	27.79/0.8621	25.87/0.8291	28.49/0.8801
	0.25	28.91/0.8834	30.89/0.9202	32.19/0.9396	32.77/0.9452	32.62/0.9430	31.77/0.9332	29.60/0.9187	31.83/0.9347	32.36/0.9323	32.99/0.9445	30.13/0.9241	33.38/0.9485
	Avg.	22.19/0.6285	24.84/0.7301	25.85/0.7695	26.25/0.7744	25.93/0.7664	25.20/0.7316	24.17/0.7362	25.83/0.7626	24.75/0.7167	26.09/0.7641	24.64/0.7378	26.77/0.7883

through two pathways: one involves up-sampling the acquired multi-scale features using a 2×2 transposed convolution (ConvT_2) with a stride of 2 and residual connections, while the other employs three MSTBs for weighted feature fusion at different scales. The outputs from these two pathways, along with the feature maps from the previous stage ($\mathbf{x}_0^{(k-1)}$), are concatenated along the channel dimension. Subsequently, the DPFB employs DCABs and 1×1 convolutions to achieve effective attentional weighted fusion, resulting in two output feature maps: $\mathbf{x}_0^{(k)}$ and $\mathbf{x}^{(k)}$. We also introduce the Soft Thresholding Residual connection (STRes), inspired by [47], as an adaptive method for threshold adjustment.

Iteration Fusion Strategy (IFS). Upon completing K reconstruction stages, in order to maximize the utilization of output features at each stage while reducing information loss across stages, we perform the following steps: First, we concatenate the outputs $\mathbf{x}^{(k)}$ and $\mathbf{x}_0^{(k)}$ from each stage along the channel dimension. Then we employ the DCABs and 1×1 convolutions to achieve attentional weighted feature fusion across iterative reconstruction stages, contributing to the final output image denoted as $\hat{\mathbf{x}}$:

$$\hat{\mathbf{x}} = \mathcal{F}_{post}(\mathrm{Concat}(\{\mathbf{x}^{(k)}\}_{k=0}^{K}, \{\mathbf{x}_0^{(k)}\}_{k=0}^{K})), \quad (18)$$

where Concat denotes the concatenation along the channel dimension. In summary, the proposed CPP-Net is described in Algorithm 1.

3.3. Loss Function

We define the training set of images as $\{\overline{\mathbf{x}}_i\}_{i=1}^{n}$, where n is the number of training images. To quantify the dissimilarity between the reconstructed image $\hat{\mathbf{x}}_i$ and the corresponding ground truth image $\overline{\mathbf{x}}_i$, we employ the ℓ_2-loss. The sampling matrix $\mathbf{\Phi}$ is jointly learned through the successive reconstruction stages. Consequently, the loss function is formulated as follows:

$$\mathcal{L}(\Theta) = \frac{1}{n} \sum_{i=1}^{n} \|\hat{\mathbf{x}}_i - \overline{\mathbf{x}}_i\|_{\ell_2}^2, \quad (19)$$

where Θ represents the set of learnable parameters for our proposed CPP-Net.

4. Experiments

4.1. Qualitative Evaluation

The best and second-best results in the tables are highlighted in red and blue colors, respectively. As detailed in Tab. 1, our CPP-Net consistently outperforms all competing methods across various cases, excelling in both PSNR and SSIM. For example, on General100 at a CS ratio of 0.04, CPP-Net surpasses AutoBCS [13], CSformer [37], DPC-DUN [30], OCTUF [31], and TCS-Net [14] by approximately 1.94 dB (7.16%), 1.23 dB (4.42%), 2.43 dB (9.13%), 0.69 dB (2.43%), and 2.47 dB (9.30%) in terms of PSNR, respectively. Similarly, when considering SSIM, CPP-Net
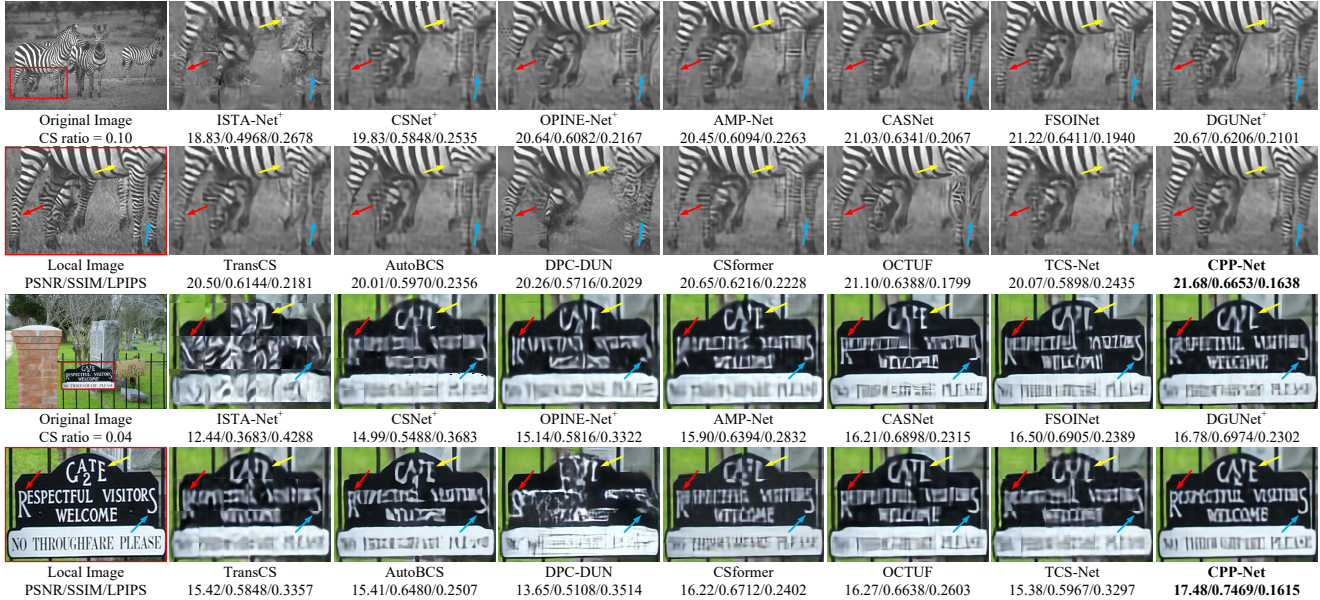
Figure 4. Comparisons of visual and corresponding PSNR (dB)/SSIM/LPIPS performance at CS ratios of 0.10 and 0.04. The arrows point to details in the reconstructed image for better comparison.

leads by around 0.0320 (4.02%), 0.0298 (3.73%), 0.0753 (10.00%), 0.0162 (1.99%), and 0.0572 (7.42%). Furthermore, visual comparisons of the reconstructed image details are presented in Fig. 4. These comparisons reveal that images reconstructed by CPP-Net exhibit a notable reduction in block artifacts, a common issue observed in ISTA-Net$^+$ [41], CSNet$^+$ [28], DGUNet$^+$ [24], TransCS [27], CSformer, and TCS-Net. Moreover, these competing methods, *e.g*., AMP-Net [46], CASNet [7], FSOINet [8], DGUNet$^+$, DPC-DUN, CSformer and OCTUF, exhibit susceptibility to blurring and the omission of fine image details, leading to compromised visual perception and overall image quality. Conversely, CPP-Net's reconstructed images showcase a superior preservation of fine image details, thereby delivering an enhanced visual experience, higher human perception quality and overall image quality. Furthermore, detailed experimental settings and more comparisons can be found in *Supplementary Material* (Sec. 3.1 and Sec. 3.2).

4.2. Ablation Studies

In this part, we conducted the ablation studies to investigate the impact of the number of stages and the effectiveness of the different components of our proposed CPP-Net.

Table 2. The impact of different number of iterative reconstruction stages in our CPP-Net on Set14 at a CS ratio of 0.25.

Stages	3	5	7	8 (default)	9
PSNR/SSIM	33.65/0.9301	34.21/0.9326	34.27/0.9332	34.41/0.9336	34.46/0.9338

Number of stages. We investigated the impact of the

number of reconstruction stages in CPP-Net, specifically 3, 5, 7, 8 (the default setting) and 9 stages, to explore the benefits associated with different stage counts. As shown in Tab. 2, we observe a direct correlation between the number of stages and the performance, highlighting the efficacy of our iterative network design. As a result, we choose eight stages as the default configuration, as it strikes a balanced compromise between model complexity and performance.

Table 3. Ablation studies of different components in CPP-Net. ✔ and ✘ represent the inclusion and exclusion of the component, respectively. DS represents the downsampling operations, US is the upsampling operations, RB denotes the residual block with two 3×3 convolutions and GELU in between, and Shared denotes the parameters sharing in the reconstruction stages.

Cases	STRes	DS	US	DCAB	RB	MSTB	IFS	Shared
Net-1	✘	✘	✘	✘	✘	✘	✔	✘
Net-2	✔	✘	✘	✘	✘	✘	✔	✘
Net-3	✔	✔	✔	✘	✔	✘	✔	✘
Net-4	✔	✔	✔	✔	✘	✘	✔	✘
Net-5	✔	✔	✔	✔	✘	✔	✘	✘
Net-6	✔	✔	✔	✔	✘	✔	✔	✔
CPP-Net	✔	✔	✔	✔	✘	✔	✔	✘

Different components. We conducted ablation experiments to assess the individual components within our CPP-Net. The experimental cases are detailed in Tab. 3, and the corresponding results are summarized in Tab. 4. Our results show that the inclusion of DPFB (*i.e*., STRes, DS, US, DCAB, MSTB) and IFS leads to progressive improvements of our method's performance. The default CPP-Net out-

performs other cases and alternatives, achieving the highest PSNR and SSIM, while Net-5 ranks second except on Urban100.

Table 4. Comparisons of average PSNR (dB)/SSIM, parameter counts and FLOPs for different methods at a CS ratio of 0.10.

Methods	Set11	Set14	LIVE29	General100	Urban100	Param.	FLOPs
Net-1	30.15/0.8955	28.95/0.8382	27.66/0.8228	32.10/0.9041	26.39/0.8373	1.31	76.17
Net-2	30.21/0.8969	28.98/0.8388	27.70/0.8236	32.16/0.9052	26.45/0.8390	1.34	76.19
Net-3	30.74/0.9063	29.44/0.8480	28.12/0.8349	32.55/0.9113	27.39/0.8621	21.61	243.52
Net-4	31.21/0.9132	29.83/0.8529	28.43/0.8404	33.12/0.9155	28.42/0.8784	11.69	114.82
Net-5	31.24/0.9134	29.88/0.8533	28.44/0.8409	33.13/0.9158	28.36/0.8779	9.13	114.88
Net-6	30.75/0.9064	29.50/0.8486	28.14/0.8353	32.57/0.9119	27.36/0.8623	2.60	153.47
CPP-Net	31.27/0.9135	29.93/0.8537	28.46/0.8414	33.16/0.9159	28.49/0.8801	12.31	153.47
CASNet	30.29/0.9005	29.37/0.8467	28.16/0.8338	32.78/0.9099	27.40/0.8606	16.90	205.24
DGUNet$^+$	30.93/0.9088	29.34/0.8455	28.27/0.8379	32.41/0.9073	28.01/0.8709	6.81	97.79
FSOINet	30.44/0.9018	29.35/0.8451	28.23/0.8359	32.70/0.9085	27.53/0.8627	0.64	17.19
TransCS	29.54/0.8877	28.81/0.8343	27.65/0.8195	31.39/0.8918	26.77/0.8418	1.49	25.86
AutoBCS	28.44/0.8827	28.00/0.8286	27.07/0.8184	30.76/0.8927	25.36/0.8242	2.01	20.11
CSformer	30.09/0.8925	28.79/0.8214	27.80/0.8148	31.60/0.8880	27.30/0.8483	6.65	21.04
DPC-DUN	29.42/0.8801	28.03/0.7950	26.83/0.7812	31.17/0.8716	26.96/0.8361	1.64	65.54
OCTUF	30.70/0.9030	29.47/0.8454	28.23/0.8345	32.77/0.9084	27.79/0.8621	0.40	21.51
TCS-Net	29.04/0.8834	28.19/0.8283	27.21/0.8165	29.90/0.8748	25.87/0.8291	0.52	7.03

4.3. Performance under Noise

We evaluated the performance of our CPP-Net under different levels of noise to show the robustness of our method. We first added Gaussian noise to the images of Set11 and then evaluated the performance of our CPP-Net under noisy images. As shown in Fig. 5, the performance of each method decreases as the variance increases, but CPP-Net still outperforms the other methods at all tested Gaussian noise levels. Besides, performance under salt-and-pepper noise is presented in the *Supplementary Material* (Sec. 3.3).

5. Extension to CS-MRI

Table 5. Average PSNR (dB)/SSIM performance comparisons of recent CS-MRI methods on Brain dataset using Pseudo Radial masks at different CS ratios.

Methods	0.05	0.10	0.20	Avg.
Zero-filled	24.20/0.5417	26.81/0.6030	30.41/0.7229	27.14/0.6225
DC-CNN [25]	30.81/0.8370	34.33/0.8957	38.43/0.9467	34.52/0.8931
RDN [32]	30.95/0.8421	34.38/0.8998	38.47/0.9474	34.60/0.8964
ISTA-Net$^+$ [41]	31.28/0.8547	34.62/0.9035	38.57/0.9478	34.82/0.9020
CDDN [48]	31.58/0.8513	34.67/0.9014	38.65/0.9476	34.97/0.9001
ADMM-CSNet [36]	31.37/0.8608	34.45/0.8985	38.52/0.9471	34.78/0.9021
HiTDUN [43]	32.72/0.8770	35.71/0.9179	39.27/0.9529	35.90/0.9159
CPP-Net	34.87/0.9176	36.61/0.9318	39.67/0.9559	37.05/0.9351

We expanded CPP-Net into the domain of CS-MRI, aiming to reconstruct magnetic resonance images from partial Fourier data. In this context, the sampling matrix $\mathbf{\Phi}$ is defined by a partial Fourier transform matrix, specifically $\mathbf{\Phi} = \mathbf{SF}$. Here, $\mathbf{S}$ represents a sub-sampling mask, and $\mathbf{F}$ corresponds to the Discrete Fourier Transform (DFT). To assess the performance, we utilized the same set of training and testing brain images as used in previous works [36, 41], using Pseudo Radial masks as $\mathbf{S}$. The results are summarized in Tab. 5. Notably, our CPP-Net consistently outper-
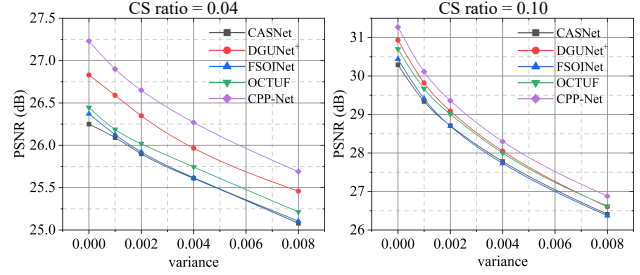

Figure 5. Comparisons of PSNR (dB) performance on Set11 at CS ratios of 0.04 and 0.10 under different Gaussian noise levels.
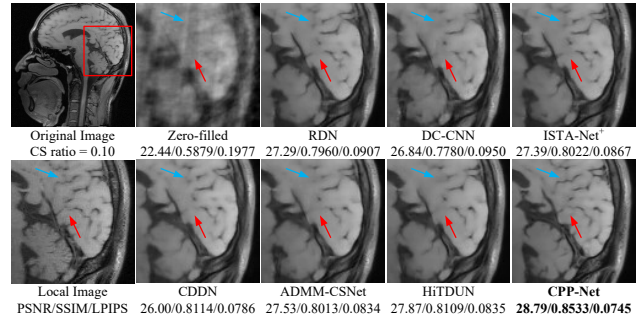

Figure 6. Comparisons of visual and corresponding PSNR (dB)/SSIM/LPIPS performance on Brain dataset using Pseudo Radial masks at a CS ratio of 0.10. The arrows point to details in the reconstructed image for better comparison.

forms other methods across all tested CS ratios. This superior performance is further illustrated in Fig. 6, emphasizing the CPP-Net's capacity to capture richer details and provide superior image quality compared to alternative methods. Besides, additional CS-MRI experiments are presented in the *Supplementary Material* (Sec. 3.4).

6. Conclusion

In this paper, we propose CPP-Net, a novel deep unfolding framework inspired by CP-PPA. We then enhance the framework in two key aspects: with DPFB for multi-scale feature extraction and fusion, and IFS for information-weighted fusion across iterative reconstruction stages. Extensive experiments validate CPP-Net's superior performance in preserving image details beyond current SOTA methods. Future work aims to extend CPP-Net to diverse tasks, *e.g.*, image denoising, deraining and super-resolution.

Acknowledgement

References

[1] Arya Bangun, Arash Behboodi, and Rudolf Mathar. Sensing matrix design and sparse recovery on the sphere and the rotation group. *IEEE Trans. Signal Process.*, 68:1439–1454, 2020. 3

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009. 2, 3

[3] Stephen Boyd. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2010. 2, 4

[4] Emmanuel J Candès and Yaniv Plan. A probabilistic and RIPless theory of compressed sensing. *IEEE Trans. Intell. Transp. Syst.*, 57(11):7235–7254, 2011. 1

[5] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE Signal Process. Mag.*, 25(2): 21–30, 2008. 1

[6] Antonin Chambolle and Thomas Pock. A First-order primal-dual algorithm for convex problems with applications to imaging. *J. Math Imaging Vis.*, 40(1):120–145, 2011. 3

[7] Bin Chen and Jian Zhang. Content-aware scalable deep compressed sensing. *IEEE Trans. Image Process.*, 31:5412–5426, 2022. 4, 7

[8] Wenjun Chen, Chunling Yang, and Xin Yang. FSOINET: Feature-space optimization-inspired network for image compressive sensing. In *IEEE Int. Conf. Acoust. Speech Signal Process.*, pages 2460–2464, 2022. 4, 7

[9] Ziheng Cheng, Bo Chen, Ruiying Lu, Zhengjue Wang, Hao Zhang, Ziyi Meng, and Xin Yuan. Recurrent neural networks for snapshot compressive imaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(2):2264–2281, 2022. 1

[10] Chao Deng, Yuanlong Zhang, Yifeng Mao, Jingtao Fan, Jinli Suo, Zhili Zhang, and Qionghai Dai. Sinusoidal sampling enhanced compressive camera for high speed imaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(4):1380–1393, 2019. 1

[11] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *Eur. Conf. Comput. Vis.*, pages 391–407. 2016. 6

[12] David L. Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. *Natl. Acad. Sci.*, 106(45):18914–18919, 2009. 2, 4

[13] Hongping Gan, Yang Gao, Chunyi Liu, Haiwei Chen, Tao Zhang, and Feng Liu. AutoBCS: Block-based image compressive sensing with data-driven acquisition and noniterative reconstruction. *IEEE Trans. Cybern.*, 53(4):2558–2571, 2023. 2, 6

[14] Hongping Gan, Minghe Shen, Yi Hua, Chunyan Ma, and Tao Zhang. From patch to pixel: A transformer-based hierarchical framework for compressive image sensing. *IEEE Trans. Comput. Imaging*, 9:133–146, 2023. 2, 6

[15] Tom Goldstein and Stanley Osher. The split bregman method for ℓ_1-regularized problems. *SIAM J. Imaging Sci.*, 2(2): 323–343, 2009. 2

[16] Guoyong Gu, Bingsheng He, and Xiaoming Yuan. Customized proximal point algorithms for linearly constrained convex minimization and saddle-point problems: A unified approach. *Comput. Optim. Appl.*, 59(1):135–161, 2014. 3

[17] Bingsheng He and Xiaoming Yuan. Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective. *SIAM J. Imaging Sci.*, 5(1):119–149, 2012. 3

[18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 7132–7141, 2018. 5

[19] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 5197–5206, 2015. 6

[20] Kuldeep Kulkarni and Pavan Turaga. Reconstruction-free action inference from compressive imagers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(4):772–784, 2015. 1

[21] Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok. ReconNet: Non-iterative reconstruction of images from compressively sensed measurements. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 449–458, 2016. 2, 6

[22] Dong Liang, Jing Cheng, Ziwen Ke, and Leslie Ying. Deep magnetic resonance image reconstruction: Inverse problems meet neural networks. *IEEE Signal Process. Mag.*, 37(1): 141–151, 2020. 1

[23] Yang Liu, Xin Yuan, Jinli Suo, David J Brady, and Qionghai Dai. Rank minimization for snapshot compressive imaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(12):2990–3006, 2018. 1

[24] Chong Mou, Qian Wang, and Jian Zhang. Deep generalized unfolding networks for image restoration. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pages 17378–17389, 2022. 1, 4, 7

[25] Jo Schlemper, Jose Caballero, Joseph V. Hajnal, Anthony N. Price, and Daniel Rueckert. A deep cascade of convolutional neural networks for dynamic MR image reconstruction. *IEEE Trans. Med. Imag.*, 37(2):491–503, 2018. 2, 8

[26] H.R. Sheikh, M.F. Sabir, and A.C. Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Trans. Image Process.*, 15(11):3440–3451, 2006. 6

[27] Minghe Shen, Hongping Gan, Chao Ning, Yi Hua, and Tao Zhang. TransCS: A transformer-based hybrid architecture for image compressed sensing. *IEEE Trans. Image Process.*, 31:6991–7005, 2022. 4, 7

[28] Wuzhen Shi, Feng Jiang, Shaohui Liu, and Debin Zhao. Image compressed sensing using convolutional neural network. *IEEE Trans. Image Process.*, 29:375–388, 2020. 2, 7

[29] Jiechong Song, Bin Chen, and Jian Zhang. Deep memory-augmented proximal unrolling network for compressive sensing. *Int. J. Comput. Vis.*, 131(6):1477–1496, 2023. 2, 4

[30] Jiechong Song, Bin Chen, and Jian Zhang. Dynamic path-controllable deep unfolding network for compressive sensing. *IEEE Trans. Image Process.*, 32:2202–2214, 2023. 4, 6

[31] Jiechong Song, Chong Mou, Shiqi Wang, Siwei Ma, and Jian Zhang. Optimization-inspired cross-attention transformer for compressive sensing. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023. 1, 2, 4, 6

[32] Liyan Sun, Zhiwen Fan, Yue Huang, Xinghao Ding, and John Paisley. Compressed sensing MRI using a recursive dilated network. In *AAAI Conf. Artif. Intell.*, 2018. 2, 8

[33] Jonathan I Tamir, Frank Ong, Suma Anand, Ekin Karasan, Ke Wang, and Michael Lustig. Computational MRI with physics-based constraints: Application to multicontrast and quantitative imaging. *IEEE Signal Process. Mag.*, 37(1):94–104, 2020. 1

[34] Lizhi Wang, Zhiwei Xiong, Guangming Shi, Feng Wu, and Wenjun Zeng. Adaptive nonlocal sparse representation for dual-camera compressive hyperspectral imaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(10):2104–2111, 2016. 1

[35] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pages 606–615, 2018. 6

[36] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. ADMM-CSNet: A deep learning approach for image compressive sensing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(3):521–538, 2020. 1, 2, 4, 8

[37] Dongjie Ye, Zhangkai Ni, Hanli Wang, Jian Zhang, Shiqi Wang, and Sam Kwong. CSformer: Bridging convolution and transformer for compressive sensing. *IEEE Trans. Image Process.*, 32:2827–2842, 2023. 2, 6

[38] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Proc. Int. Conf. Curves Surfaces*, pages 711–730, 2012. 6

[39] Zhiyuan Zha, Xin Yuan, Bihan Wen, Jiantao Zhou, Jiachao Zhang, and Ce Zhu. A benchmark for sparse coding: When group sparsity meets rank minimization. *IEEE Trans. Image Process.*, 29:5094–5109, 2020. 1

[40] Zhiyuan Zha, Bihan Wen, Xin Yuan, Saiprasad Ravishankar, Jiantao Zhou, and Ce Zhu. Learning nonlocal sparse and low-rank models for image compressive sensing: Nonlocal sparse and low-rank modeling. *IEEE Signal Process. Mag.*, 40(1):32–44, 2023. 1

[41] Jian Zhang and Bernard Ghanem. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1828–1837, 2018. 1, 2, 4, 7, 8

[42] Jian Zhang, Chen Zhao, and Wen Gao. Optimization-inspired compact deep compressive sensing. *IEEE J. Sel. Top. Signal Process.*, 14(4):765–774, 2020. 4

[43] Jian Zhang, Zhenyu Zhang, Jingfen Xie, and Yongbing Zhang. High-throughput deep unfolding network for compressive sensing MRI. *IEEE J. Sel. Top. Signal Process.*, 16(4):750–761, 2022. 2, 8

[44] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *J. Electron. Imaging*, 20(2):023016, 2011. 6

[45] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 586–595, 2018. 1

[46] Zhonghao Zhang, Yipeng Liu, Jiani Liu, Fei Wen, and Ce Zhu. AMP-Net: Denoising-based deep unfolding for compressive image sensing. *IEEE Trans. Image Process.*, 30:1487–1500, 2021. 2, 4, 7

[47] Minghang Zhao, Shisheng Zhong, Xuyun Fu, Baoping Tang, and Michael Pecht. Deep residual shrinkage networks for fault diagnosis. *IEEE Trans. Ind. Inform.*, 16(7):4681–4690, 2020. 6

[48] Hao Zheng, Faming Fang, and Guixu Zhang. Cascaded dilated dense network with two-step data consistency for MRI reconstruction. In *Adv. Neural Inform. Process. Syst.*, 2019. 2, 8

[49] Ziyang Zheng, Wenrui Dai, Duoduo Xue, Chenglin Li, Junni Zou, and Hongkai Xiong. Hybrid ISTA: Unfolding ISTA with convergence guarantees using free-form deep neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(3):3226–3244, 2023. 1

[50] Mingqiang Zhu and Tony Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *Ucla Cam Report*, 34:8–34, 2008. 3