IEICE TRANSACTIONS

on Information and Systems

DOI:10.1587/transinf.2024EDP7009

Publicized:2024/04/23

This advance publication article will be replaced by the finalized version after proofreading.

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

A Channel Contrastive Attention-based Local-Nonlocal Mutual block on Super-Resolution

Yuhao LIU[†], ***Member***, **Zhenzhong CHU**[††], ***and*** **Lifei WEI**[†a)], ***Nonmembers***

SUMMARY In the realm of Single Image Super-Resolution (SISR), the meticulously crafted Nonlocal Sparse Attention-based block demonstrates its efficacy in noise reduction and computational cost reduction for nonlocal (global) features. However, it neglect the traditional Convolutional-based block, which proficient in handling local features. Thus, merging both the Nonlocal Sparse Attention-based block and the Convolutional-based block to concurrently manage local and nonlocal features poses a significant challenge. To tackle the aforementioned issues, this paper introduces the Channel Contrastive Attention-based Local-Nonlocal Mutual block (CCLN) for Super-Resolution (SR). (1) We introduce the CCLN block, encompassing the Local Sparse Convolutional-based block for local features and the Nonlocal Sparse Attention-based network block for nonlocal features. (2) We introduce Channel Contrastive Attention (CCA) blocks, incorporating Sparse Aggregation into Convolutional-based blocks. Additionally, we introduce a robust framework to fuse these two blocks, ensuring that each branch operates according to its respective strengths. (3) The CCLN block can seamlessly integrate into established network backbones like the Enhanced Deep Super-Resolution network (EDSR), achieving in the Channel Attention based Local-Nonlocal Mutual Network (CCLNN). Experimental results show that our CCLNN effectively leverages both local and nonlocal features, outperforming other state-of-the-art algorithms.

key words: *Single-Image Super-Resolution, Self-Attention, Channel Contrastive Attention, Local and Nonlocal features, Contrastive Learning*

1. Introduction

SISR aims to reconstruct a high-resolution (HR) image from a single low-resolution (LR) input image, presenting a non-bijective mapping between LR and HR images. This results in a challenging and ill-posed problem, complicating the generation of high-quality HR details. Recent advances in deep convolutional neural networks for SISR, including Convolution[1]–[3], Attention[4]–[9], Sparse Aggregation[10], [11], Contrastive Learning[11], [12], and Local-Nonlocal[11], [12] Mutual-based approaches, have achieved notable success[13].

However, existing methods focus on extracting either local or nonlocal features, each with its unique advantages. Convolution-based SR[2], [3] excels at extracting local features but is constrained by limited receptive fields. Non-Local Attention-based SR[7] effectively captures nonlocal features, while they are limited to specific LR image features (local or nonlocal features).

To overcome these limitations, we propose a novel approach that combines the strengths of Convolution-based SR and Nonlocal Attention-based SR. The Nonlocal Attention-based block emphasizes nonlocal feature extraction, while the Convolution-based block focuses on local features, synergistically enhancing their benefits and mitigating drawbacks. Challenges in this approach include: (1) Quadratic computational cost (to the input size) in traditional Nonlocal Attention-based SR[7], making parallel connection challenging. (2) The Nonlocal Sparse Attention-based network incorporates Sparse Aggregation, but simply placing two blocks in parallel is not viable. This is because noise passes through another branch, causing the Sparse Aggregation to fail for the Nonlocal Sparse Attention-based network. (3) Generating appropriate input feature maps for both parallel blocks is crucial for optimal performance.

To address these challenges, we introduce the CCLN block for SR, incorporating it into EDSR[3] to achieve CCLNN. (1) Utilizing NLSA from Mei et al.[10] as our attention-based block reduces computational cost and noise. (2) Contrastive Learning-based Channel Attention for our Convolutional-based block efficiently extracts desired feature maps, suppressing noise. (3) A framework facilitates the fusion of two blocks: CCA ensures noise-free input for Convolutional-based blocks, and NLSA can extract nonlocal feature maps with a simple PA block beforehand. This comprehensive approach aims to overcome current limitations and enhance SISR performance.

In summary, the main contributions of this paper are three fold:

(1) Dual Block in Parallel for CCLN: We present a simple yet effective block (CCLN) encompassing the Local Sparse Convolutional-based block for local features and the Nonlocal Sparse Attention-based network block for nonlocal features. Each branch optimally functions within its specialized domain.
(2) Contrastive Learning and a robust framework: We introduce the CCA blocks to generate Sparse Aggregation for Convolutional-based block, and we also introduce a robust framework designed to facilitate the fusion of two blocks. This ensures a harmonized and noise-free output for both branches.
(3) Seamless Integration and Enhanced Performance: We seamlessly integrate the CCLN block into the backbone of the Enhanced Deep Super-Resolution network

[†]The author is with College of Information Engineering, Shanghai Maritime University, Shanghai,201306, China

[††]The author is with School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai,200093, China

a) E-mail: lfwei@shmtu.edu.cn

(EDSR), achieving the Channel Attention based Local-Nonlocal Mutual Network (CCLNN). Experimental results demonstrate that our CCLNN effectively leverages both local and nonlocal features, surpassing other state-of-the-art algorithms.

2. Related Works

In this section, we will briefly introduce some of the famous algorithms related to our work, including: Convolution[1]–[3], Attention[4]–[9], Sparse Aggregation[10], [11], Contrastive Learning[11], [12] and Local-Nonlocal Mutual[14]–[16] based SISR.

Convolutional-based SR, which is the typical classical deep neural network algorithm for SISR, can extract the local features efficiently, but the receptive field is limited by the size of the convolutional kernel (typically 3×3). The Super-Resolution Convolutional Neural Network (SRCNN) [1] is known as the first Convolutional-based SR, which has only 3 convolutional layers. The Very Deep Convolutional Networks for SR (VDSR)[2] is with 20 convolutional layers, so VDSR can extract deeper features, which will improve the SR performance, but difficult to converge. EDSR[3], another notable Convolutional-based SR, can achieve great depth but still stable for the training procedure by removing unnecessary modules. The SRCNN, VDSR, and EDSR are well known classical SISR algorithms, the performance is not superior, but they are still an inspiration for other algorithms.

Attention, which has been categorized into classical attention and nonlocal attention (self-attention) models, has been extensively studied in previous research. The classical attention-based SR is straightforward and efficient, such as: Channel Attention[4], Spatial Attention[5], and Pixel Attention (PA)[6]. These attention-based models can extract feature weights efficiently under different dimensions. Channel Attention can extract weights under the channel dimension (1D), Spatial Attention can extract weights under the spatial dimension (2D), PA can extract weights under the all-pixel dimension (3D). However, none of the three classical attention models took into account the weight values from long-range (Nonlocal) features.

Many Self-Attention (also called Nonlocal Attention) models have been introduced to account for long-range (Nonlocal) features. Mei et al.[7] proposed the Cross-Scale Non-Local (CSNL) attention module, which can deal with different scale nonlocal features. By introducing a down-sample scaling factor, CSNL can deal with different scale nonlocal features. Niu et. al.[8] propose the Holistic Attention Network (HAN), consisting of a Layer Attention Module (LAM) and a Channel-Spatial Attention Module (CSAM) to model the interdependencies among layers, channels, and positions. These Self-Attention-based models perform well, but their computational cost is quadratic of the input size, and failed to remove noise[11].

Sparse Aggregation is introduced to reduce the cost of Self-Attention-based (Nonlocal Attention) models and to get rid of noise. Mei et. al.[10] proposed NLSA, which introduced Sparse Aggregation into Nonlocal Attention. NLSA rectified Nonlocal Attention with spherical Locality Sensitive Hashing (LSH), which divides the input space into hash buckets of related features, and computes the attention only within the bucket to realize Sparse Aggregation. The NLSA can reduce the computational cost from quadratic to asymptotically linear with respect to the spatial size. However, it only considered the nonlocal features and don't consider the local features, so there is room for performance improvement.

Contrastive Learning is used to distinguish relevant and irrelevant features. Wang et al.[12] introduced Contrastive Learning to BlindSR by proposing a Degradation-Aware SR (DASR) network. They proposed a Contrastive loss for unsupervised degradation representation learning by contrasting positive pairs. As for SISR, Xia et al.[11] proposed a novel Efficient Non-Local Contrastive Attention (ENLCA) in SISR, which is considered as the first to introduce Contrastive Learning into the Nonlocal Attention to improve sparsity by pulling relevant features closer and pushing irrelevant features away in the representation space. The Contrastive Learning in ENLCA is a great motivation for us to introduce sparseness in the local blocks.

Local-Nonlocal Mutual method is a complex challenge. Behjati et al.[16] proposed a novel procedure called Residual Attention Feature Group (RAFG), in which both Parallelizing Attention and Residual Block are linearly fused. They also proposed a Directional Variance Attention Network (DiVANet), which is a computationally efficient yet accurate network for SISR. Our previous work[14] proposed a Local and Non-Local Features Based Feedback Network (LNFSR) on SR which introduced three different blocks, also proposed an Up-Fusion-Delivery layer to hold three blocks. Although LNFSR achieved acceptable performance, the network is too large and the fusion method is crude and needs further optimization. Our another previous work[15] proposed a Dynamic Fusion of Local and Non-Local Features-Based Feedback Network (DLNFN), which introduce two different blocks, also proposed a dynamic weight block for fusion two different blocks' outputs. The DLNFN achieved acceptable performance, but the network is too computation cost, can't get rid of the noise due to without Sparse Aggregation.

Building upon the previously analyzed works, we have introduced the CCLN block for SISR. This design anticipates the specialized roles of the Nonlocal Sparse Attention-based block (for nonlocal features) and the Convolutional block (for local features), leveraging their respective strengths. To extract nonlocal features with Sparse Aggregation, we adopted the NLSA approach developed by Mei[10]. Additionally, Contrastive Learning has been incorporated into the Convolutional block for Sparse Aggregation, drawing inspiration from Xia's research[11].

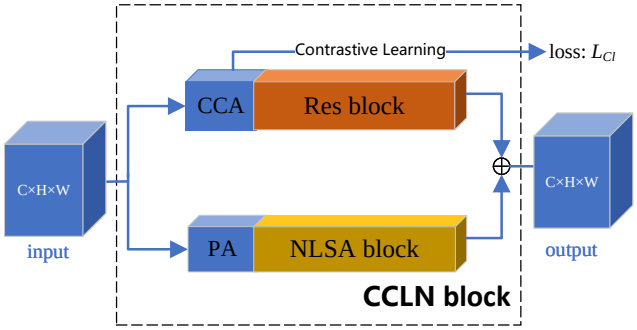

Fig. 1: The overall architecture of our CCLN block.

3. The Channel Contrastive Attention-based Local-Nonlocal Mutual (CCLN) block for SR

In this section, we introduce our CCLN block for SR. Firstly, we outline the overall architecture of the CCLN block in Section 3.1, followed by a detailed discussion of its components in Section 3.2. Lastly, we present the integration of our CCLN block into EDSR, introducing the CCLNN, in Section 3.3.

3.1 the overall architecture of our CCLN block

In this section, we provide a comprehensive introduction to the architecture of our CCLN block. Figure 1 illustrates the overall structure of CCLN, comprising two fundamental blocks (Res block and NLSA block) and a framework (the Channel Contrastive and Pixel Attention framework) to seamlessly integrate them. Our CCLN block proposal is both straightforward and efficient. Generally, $X \in \mathfrak{R}^{C\times H\times W}$ refers to the input feature maps for a CCLN block, and $Y \in \mathfrak{R}^{C\times H\times W}$ refers to the output feature maps, where C is the channels for the output maps, H and W are the height and width of each input feature map. Thus, the computation of Y output is derived by Equation 1.

$$Y = CCLN(X) \tag{1}$$

The fundamental building blocks of our CCLN include the Res block and the NLSA block, as depicted in Figure 1. The Res block consists of 2 convolutional layers with a ReLU activation layer between them, and a skip connection linking the head to the tail. This structure is akin to the basic block in EDSR, emphasizing the extraction of local features. On the other hand, the NLSA block, as intricately designed in Mei's work [10], is Transformer-based and focuses on extracting nonlocal features. Leveraging the Channel Contrastive and Pixel Attention framework within our CCLN block, both blocks synergistically operate at their optimal capacity.

The Channel Contrastive and Pixel Attention framework aims to integrate two fundamental blocks into our CCLN, optimizing the performance of both blocks, as indicated by the blue lines and blocks in Figure 1. The generation of desired inputs for these two basic blocks is pivotal for the effectiveness of our framework. Consequently, we introduced PA before the NLSA block, facilitating the extraction of desired nonlocal feature inputs for the subsequent NLSA block. Additionally, we introduce CCA before the Res block, enabling the extraction of desired local features for the Res block.

Data flow and connections in our CCLN follow this sequence: The input feature X is directed into both the PA and CCA blocks to generate desired feature maps for subsequent processing. The flow and connections proceed as follows: Firstly, the output of PA is labeled as $PA(X)$, and the output of CCA is labeled as $CCA(X)$. The feature map of $PA(X)$ is channeled into the NLSA block, which excels in extracting nonlocal features even from distant features. The output of the NLSA block is denoted as $NLSA(PA(X))$. Secondly, the feature map of $CCA(X)$ is fed into the Res block, capable of extracting local features within the receptive field limited by the kernel size (set to 3×3 for our CCLN). The output of the Res block is denoted as $Res(CCA(X))$, and the Contrastive Learning Loss is expressed as ℓ_{cl}. Lastly, the ultimate output of our CCLN is simply the summation of NLSA's output and Res's output, as denoted by equation 2:

$$Y = CCLN(X) = NLSA(PA(X)) + Res(CCA(X)) \tag{2}$$

The architecture of our CCLN is straightforward and efficient, avoiding excessive computational costs. The PA achieves attention across all pixels using just one convolutional layer with a 1×1 kernel. Consequently, the PA introduces $C \times C$ parameters to our CCLN block. On the other hand, the CCA introduces $2 \times C \times C/r$ parameters to our CCLN block. With C representing the channels for the output and r being the reduction parameter for CCA (set to 16 in our CCLN). Therefore, our CCLN block only introduces $(1 + 2/r) \times C^2$ parameters to accommodate both base blocks, the proposed structure of our CCLN block minimally impacts computational costs.

3.2 Channel Contrastive Attention (CCA) of our CCLN block

In this section, we provide a comprehensive introduction to our CCA, illustrated in Figure 2. Our CCA signifies an improvement upon the traditional CA structure, with detailed steps outlined below. The key difference between our CCA and the traditional CA is in step (2):

(1) First, the input features applies a 2D adaptive average pooling, generating $C \times 1 \times 1$ dimensions' output, then applies 2 full connection layer, the output is a vector as C dimensional vector.
(2) Next, different from traditional CA, the C-dimensional vector are applied Contrastive Learning to introduce the sparse aggregation.
(3) Finally, the C-dimensional vector with Sparse Aggregation are fed into the original input features on each

channel.

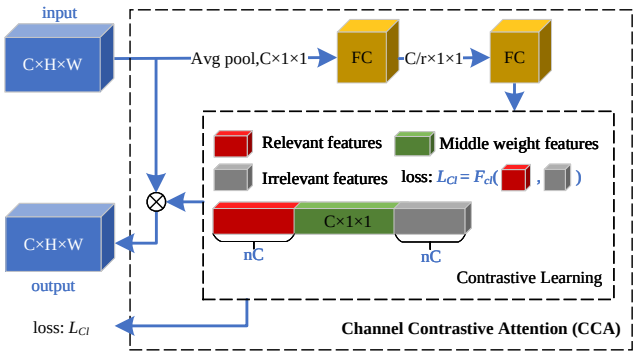

Fig. 2: The structure of our CCA.

In step (2), we employ Contrastive Learning on the C-dimensional vector to introduce Sparse Aggregation into CA block. The application of Contrastive Learning on CA serves to balance the sparse constraint on outputs of two branches (the outputs of the Res block and NLSA block) to eliminate noise. Intuitively, within our CCA, increasing the channel weight emphasizes relevant features, and decreasing the channel weight (approaching 0) to filter out irrelevant features, while leaving the middle weight features unconstrained. So Contrastive Learning functions to filter out irrelevant features while amplifying the weight of relevant features, without affecting the middle weight features. As depicted in Figure 2, the Contrastive Learning Loss ℓ_{cl} for our CCA can be formulated using Equations 3 and 4:

$$T_i' = sort \downarrow (W_i^{CA}) \tag{3}$$

$$\ell_{cl} = \sum_{i=1}^{C} -log(\frac{\sum_{j=1}^{nC} exp(T_i')/nC}{\sum_{j=(1-n)C}^{C} exp(T_i')/nC}) + b \tag{4}$$

where $W^{CA} \in \Re^C$ is a C-dimensional vector, W_i^{CA} is the i-th value of vector W^{CA}, $sort \downarrow (\cdot)$ means descending sort the input, C is the channel of feature maps, n is the percentage of channels of relevant and irrelevant features, b is a margin constant. The input weight values W^{CA} are first sorted using Equation 3, following which Contrastive Learning is applied to the relevant (previous nC weight values) and irrelevant (last nC weight values) features to introduce Sparse Aggregation.

Our CCA is motivated by Xia's work[11], but the Contrastive Learning loss ℓ_{cl} in our CCA is simpler since our Contrastive Learning works on the channel dimension. We will discuss the n parameter in our ablation study section (in Section 4.2), while other parameter values are the same as in Xia's work.

3.3 Channel Contrastive Attention-based Local-Nonlocal Mutual Network (CCLNN)

Our CCLN block seamlessly integrates into ResNet backbone algorithms, such as EDSR, achieving the CCLNN. Therefore, in this paper, we leverage the CCLNN to demonstrate the efficacy of our CCLN block. As depicted in Figure 3, the CCLNN model utilizes the EDSR backbone with 32 residual blocks, incorporating 5 CCLN blocks with one insertion after every 8 residual blocks. The losses from all five CCLN blocks ($\ell_{cl}[1], \cdots, \ell_{cl}[5]$) are aggregated to form the final loss function.

The loss function: The overall loss function ℓ of our CCLNN is designed as Equations 5 and 6:

$$\ell_{rec} = \left\| I^{HR} - I^{SR} \right\|_1 \tag{5}$$

$$\ell = \ell_{rec} + \lambda \cdot \sum_{i=1}^{B} \ell_{cl}[i]/B \tag{6}$$

Where ℓ_{rec} represents the Mean Absolute Error (MAE) aiming to reduce the distortion between the predicted SR image I^{SR} and the target HR image I^{HR}, and λ is the weight between ℓ_{rec} and ℓ_{cl}, the total block number B is 5 in our CCLNN. We will discuss the λ parameter in our ablation study section (in Section 4.2).

Other implementation details: Here are other implementation details not mentioned above:

(1) We employed the ReLU as the activation function, and the feature-map channel was set to 256. For all convolutional layers (except those mentioned above) in the network, the kernel size is 3×3. In Equation 4, we set n for Contrastive Learning to 15% for scales $\times2$ and $\times3$, and 10% for scale $\times4$. The margin b in Equation 4 is set to 1, and λ set to $4e-3$ for the weight between ℓ_{rec} and ℓ_{cl} in Equation 6.

(2) We utilize MAE loss (L_1 loss) for optimizing our CCLNN, employing the Adam optimizer to optimize network parameters with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and an initial learning rate of $1e-4$. The learning rate is reduced by half every 200 epochs, and the training process set to a total of 1000 epochs.

(3) We initiate the training process with a warm-up phase, training the network for the first 150 epochs solely with the loss function ℓ_{rec} as Equations 5. Subsequently, we proceed to train with the full loss function ℓ as Equations 6. The network is implemented using the PyTorch framework.

4. Experimental Results

4.1 Datasets and Evaluation Metrics

We conducted all our experiments using the DIV2k database for training, utilizing the entire train set (800 HR images) to train all models. For augmenting the train images, we implemented the following image reuse strategy: firstly, each image is randomly cropped into a small patch. Secondly, all patches perform random rotations of 0°, 90°, 180°, 270°, and horizontal flipping. Finally, LR image patches are generated

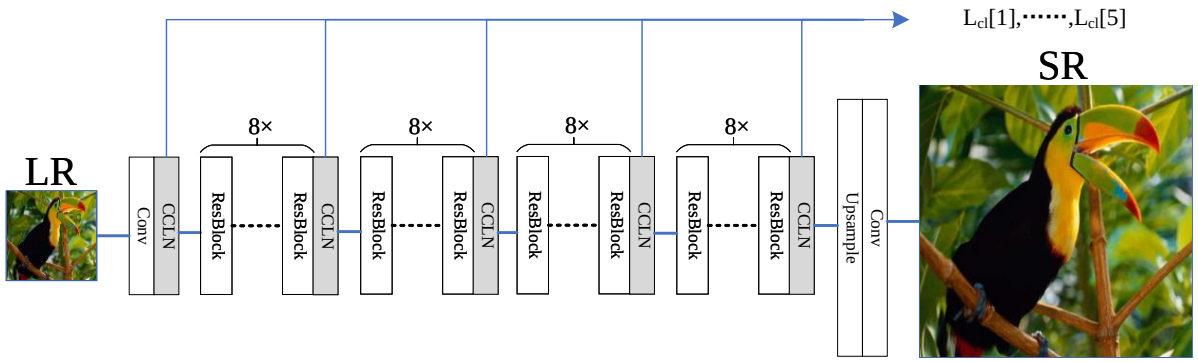

Fig. 3: The proposed CCLNN network. Five CCLN blocks are embedded after every eight residual blocks.

from HR image patches using the BiCubic method. During each epoch, all training images perform this data augmentation process 20 times. We set the input patch size (LR image patch) to 48×48 for our CCLNN to strike a balance between performance and computational cost.

4.2 Ablation Study

In this section, we perform ablation study to validate the effectiveness of our CCLNN. To reduce training costs, we halve some critical parameters, specifically setting the feature-map channel to 128, using 16 residual blocks, and 3 CCLN blocks. This modified version is referred to as CCLNN-L, designed for lightweight applications. We have also adjusted the training strategy to suit the CCLNN-L model, specifying a total of 500 epochs, a warm-up epoch of 75 and adjust the learning rate by multiple 0.5 for every 150 epochs. For a fair comparison, the algorithms compared in this section set to the same parameters and training strategy.

Whether the Local-Nonlocal Mutual strategy enhances performance: We conducted an ablation study to assess whether the Local-Nonlocal Mutual Network enhances performance, surpassing both single local feature-based and single nonlocal feature-based networks. We compared four algorithms, including two single local-feature-based networks and two single nonlocal-feature-based networks: (1) Activation of only the Res block within our CCLN block, denoted as CCLNN-local, to evaluate performance on a single Nonlocal feature-based Network. (2) Application of the well-known EDSR (with feature-map channel= 128, 16 residual blocks, denoted as EDSR-L) to eliminate the influence of our framework. (3) Activation of only the NLSA block in our CCLN block, denoted as CCLNN-Non-L, to evaluate performance on a single nonlocal-feature-based network. (4) Application of the well-known NLSN[10] (with feature-map channel= 128, 16 residual blocks, denoted as NLSN-L) to eliminate the influence of our framework. The performances (PSNR) are detailed in Table 1.

In Table 1, our CCLNN-L demonstrated the best performance, showcasing the efficiency of the Local-Nonlocal Mutual Network. The traditional lightweight NLSN outperforms the lightweight EDSR, aligning with findings in their previous work [3], [10]. Our CCLNN enhances the performance of the NLSA-based structure but diminishes the performance of the EDSR-based structure. This highlights that different blocks possess distinct characteristics, emphasizing the need for a sophisticated structure design tailored to each block.

Table 1: the ablation study on the Local-Nonlocal Mutual strategy at ×2 scale on Set5

Algorithm	Local-feature-based		Nonlocal-feature-based		Local-Nonlocal Mutual
	EDSR-L	CCLNN-Local	NLSN-L	CCLNN-Non-L	CCLNN-L (our)
PSNR	38.07	38.05	38.09	38.13	38.19

Whether the Contrastive Learning enhances performance: We conducted an ablation study to assess the influence of Contrastive Learning within our CCA block on performance. For comparison, we developed two distinct algorithms, namely CCLNN-TSparse and CCLNN-NSparse. The details of these two comparison algorithms are as follows:

(1) The CCLNN-TSparse: We employ traditional Sparse-based Channel Attention as a comparison algorithm. We define the Sparse learning loss ℓ_{cl} under L_1 norm for CCLNN-TSparse as Equation 7, utilizing the same notations as Equations 3.

$$\ell_{sparse} = \frac{1}{C} \cdot \sum_{i=1}^{C} |W_i^{CA}| \tag{7}$$

The overall loss function ℓ of CCLNN-TSparse is formulated as Equation 8, which is similar to Equation 6, except that ℓ_{cl} is replaced by ℓ_{sparse}.

$$\ell = \ell_{rec} + \lambda \cdot \sum_{i=1}^{B} \ell_{sparse}[i]/B \tag{8}$$

(2) The CCLNN-NSparse: We have omitted the Contrastive Learning step for the CCA block, causing it to regress to traditional Channel Attention. The performance (PSNR) is detailed in Table 2.

In Table 2, our CCLNN-L demonstrated the best performance, showcasing the efficiency of Contrastive Learning.

Table 2: the ablation study on the Channel Contrastive Attention at ×2 scale on Set5

Algorithm	CCLNN -NSparse	CCLNN -TSparse	CCLNN -L (our)
PSNR	38.17	38.14	38.19

The CCLNN without Sparse (CCLNN-NSparse) achieved the second-best performance with a slight margin compared to CCLNN-L, while the CCLNN with traditional Sparse (CCLNN-TSparse) performed the worst, exhibiting a significant margin compared to CCLNN-L. We suspect that the middle weight features (green in Figure 2) play a crucial role in further refining the SR image. Contrastive Learning's ability to filter out irrelevant features and magnify the weight of relevant features, while leaving medium features unconstrained, contributes to its effectiveness.

The percentage of channels (parameter: n) in Equation 4: As mentioned in Equation 4, the percentage of channels with relevant and irrelevant features, denoted as n, influences the performance of our CCLNN. To identify the optimal value for n, we conducted an ablation study, setting n to $[5\%, 15\%, 25\%, 35\%, 45\%]$ (ensuring $n < 50\%$ to prevent overlap). The performances (PSNR) are detailed in Table 3, where $n = 15\%$ performs the best, so we choose the parameter $n = 15\%$ for scale ×2.

Table 3: the ablation study on the value of n in Equation 4 at ×2 scale on Set5

n	5%	15%	25%	35%	45%
PSNR	38.18	38.19	38.18	38.18	38.17

the weight (parameter: λ) between ℓ_{ref} and ℓ_{cl} in Equation 6: As mentioned in Equation 6, the weight between ℓ_{rec} and ℓ_{cl}, denoted as λ, significantly influences the performance of our CCLNN. Therefore, we conducted an ablation study to determine the appropriate value for λ. Specifically, we set λ to $[2, 4, 6, 8] \times e^{-3}$. The performance (PSNR) is detailed in Table 4, where $\lambda = 2 \times e^{-3}$ yields the best performance, so we choose the parameter $\lambda = 2 \times e^{-3}$. Additionally, we observe that the performance degrades rapidly if λ is too large.

Table 4: the ablation study on the value of λ in Equation 6 at ×2 scale on Set5

$\lambda(\times e^{-3})$	2	4	6	8
PSNR	38.18	38.19	38.16	38.16

4.3 Comparisons with state-of-the-arts

This section provides a quantitative comparison of our CCLNN with other well-known state-of-the-art SR algorithms. The selected state-of-the-art algorithms for our experiment include SRCNN[1], VDSR[2], EDSR[3], RCAN[4], RNAN+[17], A2N[18], DiVANet+[16], and NLSN[10]. SRCNN, being the first proposed convolutional-based SR algorithm, is considered the baseline for SR methods. SRCNN, VDSR, EDSR, and A2N are representative convolutional-based algorithms, while RCAN, RNAN+, and NLSN are representative attention-based algorithms. Additionally, A2N and DiVANet+ are representative fusion-block-based algorithms. We utilized official PyTorch-based models for RCAN, A2N, DiVANet+, and NLSN algorithms in our experiments, and retrained SRCNN, VDSR, and EDSR algorithms as the official PyTorch-based models were not available. To ensure a fair comparison, we excluded all training tricks such as noise and Gaussian blurring.

We conducted experiments with upscale factors in the range of $[\times 2, \times 3, \times 4]$ for all state-of-the-art SR algorithms. The performance is reported on five well-known standard benchmark datasets: Set5[19], Set14[20], B100[21], Urban100[22], and Manga109[23]. We evaluated the SR results based on PSNR and SSIM. The comprehensive results are detailed in Table 5, where our CCLNN demonstrated the best performance across all experiments.

At the ×2 scale, our CCLNN demonstrated the highest performance. It secured the top position in three datasets (Set5, B100, and Urban100), but ranked second (according to the PSNR metric) and third (according to the SSIM metric) in the Manga109 dataset. In the B100 dataset, our CCLNN obtained a lower score in the SSIM metric compared to NLSN, but outperformed it in the PSNR metric. Moving to the ×3 scale, our CCLNN demonstrated the highest performance, surpassing the second-place algorithm with a slight margin. It ranked best in two datasets (Set5 and B100), second in two datasets (Urban100 and Manga109), and in the Set14 dataset, our CCLNN outperformed NLSN in the SSIM metric and tied in the PSNR metric. At the ×4 scale, our CCLNN demonstrated the highest performance. It ranked best in two datasets (Set5 and Set14) but second in the Urban100 dataset. In the B100 dataset, CCLNN underperformed compared to NLSN in the SSIM metric but outperformed in the PSNR metric. In the Manga109 dataset, CCLNN underperformed compared to RNAN+ in the PSNR metric but best in the SSIM metric.

We evaluated the resection field of our CCLNN and NLSN (which performed second best) at scale ×2 using the newly proposed Diffusion Index (DI) evaluation metric[24]. A higher DI value indicates greater pixel involvement. Our CCLNN had an average DI of 21.11, while NLSN's DI was 19.66, indicating that our CCLNN can encompass more pixels with the help of local features.

However, it is important to acknowledge that the improvements in PSNR and SSIM of our CCLNN compared to NLSN (the second-place algorithm), especially in the ×3 scale, are not substantial. We aim for a balance between performance and cost, and our proposed CCLNN achieves the best performance with an acceptable margin without sig-

Table 5: The performance (PSNR/SSIM) of the considered state-of-the-art algorithms. (the best performance is shown in **red** and the second-best performance is shown in **blue**)

Algorithm	Scale	Params	Set5		Set14		B100		Urban100		Manga109	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SRCNN	×2	0.02M	36.45	0.9527	32.32	0.9043	31.04	0.8838	29.11	0.8887	34.82	0.9627
VDSR	×2	0.67M	37.58	0.9587	33.16	0.9133	31.94	0.8964	31.05	0.9164	37.44	0.9737
RCAN	×2	15.44M	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384	39.44	0.9786
EDSR	×2	40.73M	38.25	0.9613	33.97	0.9205	32.36	0.9020	32.98	0.9361	39.17	0.9781
A2N	×2	1.04M	38.06	0.9608	33.75	0.9194	32.22	0.9002	32.43	0.9311	38.87	0.9769
DiVANet+	×2	0.9M	38.23	0.9618	33.88	0.9201	32.36	0.9018	32.67	0.9330	39.15	0.9780
RNAN+	×2	9.11M	38.22	0.9613	33.97	0.9216	32.36	0.9018	32.9	0.9351	39.41	0.9789
NLSN	×2	41.80M	38.34	0.9618	34.08	0.9231	32.43	0.9027	33.42	0.9394	39.59	0.9789
CCLNN (our)	×2	48.07M	38.35	0.9618	34.15	0.9230	32.44	0.9030	33.46	0.9398	39.48	0.9785
SRCNN	×3	0.02M	32.52	0.9052	29.09	0.8160	28.10	0.7781	25.84	0.7869	29.62	0.8999
VDSR	×3	0.67M	33.76	0.9225	29.96	0.8347	28.85	0.7986	27.32	0.8324	32.41	0.9356
RCAN	×3	15.63M	34.74	0.9299	30.65	0.8482	29.32	0.8111	29.09	0.8702	34.44	0.9499
EDSR	×3	43.68M	34.74	0.9297	30.50	0.8461	29.24	0.8095	28.76	0.8651	34.01	0.9481
A2N	×3	1.04M	34.47	0.9279	30.44	0.8437	29.14	0.8059	28.41	0.8570	33.78	0.9458
DiVANet+	×3	0.9M	34.66	0.9289	30.53	0.8452	29.26	0.8077	28.66	0.8610	34.02	0.9473
NLSN	×3	44.75M	34.85	0.9306	30.70	0.8485	29.34	0.8117	29.25	0.8726	34.57	0.9508
CCLNN (our)	×3	51.02M	34.88	0.9308	30.71	0.8485	29.35	0.8118	29.18	0.8723	34.46	0.9507
SRCNN	×4	0.02M	30.15	0.8531	27.20	0.7415	26.55	0.6985	24.05	0.7005	26.79	0.8331
VDSR	×4	0.67M	31.35	0.8825	28.16	0.7703	27.26	0.7244	25.28	0.7554	29.14	0.8886
RCAN	×4	15.59M	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087	31.22	0.9173
EDSR	×4	43.09M	32.49	0.8985	28.81	0.7872	27.71	0.7409	26.58	0.8015	30.98	0.9148
A2N	×4	1.05M	32.30	0.8966	28.71	0.7842	27.61	0.7374	26.27	0.7920	30.67	0.9110
DiVANet+	×4	0.9M	32.48	0.8978	28.78	0.7848	27.73	0.7395	26.49	0.7963	30.78	0.9124
RNAN+	×4	9.26M	32.56	0.8992	28.90	0.7883	27.77	0.7424	26.75	0.8052	31.37	0.9175
NLSN	×4	44.16M	32.59	0.9000	28.87	0.7891	27.78	0.7444	26.96	0.8109	31.27	0.9184
CCLNN (our)	×4	50.43M	32.64	0.9003	28.91	0.7892	27.79	0.7439	26.93	0.8104	31.36	0.9191

nificantly increasing the number of parameters (15%, 14%, and 14% more than NLSN's in the ×2, ×3, and ×4 scale, respectively).

4.4 Visualized Analysis on our CCLNN

In this section, we provide a visualized analysis of our CCLNN along with five comparison algorithms. The comparison algorithms are as follows: SRCNN[1], EDSR[3], RCAN[4], A2N[18], and NLSN[10]. Additionally, we include HR and LR images as benchmarks, and our CCLNN is positioned at the bottom right. We selected three representative sections (textures, letters, and figures) under scales ×2, ×3, and ×4 in Figure 4.

The first image in Figure 4 is 'img060' from the Urban100 database at scale ×2. The floor exhibits linear textures that are uniform and typically long-distance features. The lines in HR have a top-right to bottom-left direction, while LR lacks directional features. Therefore, the direction must be learned from long-distance features. Only our CCLNN generated the correct SR image, especially for the direction of the floor's textures. The SRCNN generates a completely wrong direction textures, and the other comparison algorithms generate partially wrong dictation textures.

The second image in Figure 4 is the 'ppt3' image from the Set14 database at scale ×3. The letters 'way', which are typical English letters, are too small to be identified in the ×3 LR image, causing all algorithms to struggle in generating clear letters. Only our CCLNN can produce recognizable letters ('way'), particularly the letters 'a' and 'y'. For the comparison algorithms, they fail to generate a clear letter 'a', and only RCAN generates a plausible letter 'y', still not as accurate as our CCLNN's. The convolutional-based algorithms (SRCNN, EDSR, and A2N) fail to generate the letter 'w', while the attention-based algorithms (RCAN and NLSN) successfully generate a clear word 'w', showcasing

Fig. 4: Visualized comparison on our CCLNN with other comparing Algorithms

the effectiveness of attention-based algorithms.

The third image in Figure 4 is image '253027' from the B100 database under scale ×4. The stripes of the zebra are blurred and difficult to identify in the ×4 LR image. Only our CCLNN generates correct stripes, while all the comparison algorithms produce imperfect stripes.

Based on the visualized analysis above, our CCLNN consistently produces superior SR images compared to the comparison algorithms. This is evident across three selected typical sections (textures, letters, and figures), highlighting that our CCLNN outperforms other state-of-the-art algo-

rithms in the visualized comparison.

5. Conclusions

In this paper, we introduce a novel CCLN block for Super-Resolution (SR). The CCLN seamlessly integrates a Convolutional-based block for local features and a Nonlocal Sparse Attention-based network block for nonlocal features. Additionally, we introduce a framework to combine these two blocks, allowing each branch to leverage its strengths. We introduce the CCA block also employs Contrastive Learning to generate Sparse Aggregation in the local features, aimed at eliminating noise. Our CCLN block can be seamlessly integrated into the ResNet backbone, such as EDSR, to achieve the CCLNN. Experimental results demonstrate that our CCLNN effectively utilizes both local and nonlocal features, outperforming other state-of-the-art algorithms. However, we observe that achieving the ideal feature distribution for the inputs of the two branches is critical for the performance of the Local-Nonlocal Mutual block. Therefore, our future work will focus on introducing a feature distribution method, such as the Clustering method[25], before the Local-Nonlocal Mutual block to further enhance performance.

Acknowledgments

This work was supported by the National Natural Science Foundation of China, under Grant No. U2006228 and No. 61972241. This work was supported by Shanghai Soft Science Research Project No. 23692106700. This work was supported by the Natural Science Foundation of Shanghai under Grant No. 22ZR1427100.

References

[1] C. Dong, C.C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," Computer Vision – ECCV 2014, pp.184–199, Springer, Springer International Publishing, 2014.

[2] J. Kim, J.K. Lee, and K.M. Lee, "Accurate image super-resolution using very deep convolutional networks," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.1646–1654, 2016.

[3] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.136–144, 2017.

[4] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," Proceedings of the European Conference on Computer Vision (ECCV), pp.286–301, 2018.

[5] S. Woo, J. Park, J.Y. Lee, and I.S. Kweon, "Cbam: Convolutional block attention module," Proceedings of the European conference on computer vision (ECCV), pp.3–19, 2018.

[6] H. Zhao, X. Kong, J. He, Y. Qiao, and C. Dong, "Efficient image super-resolution using pixel attention," Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, pp.56–72, Springer, 2020.

[7] Y. Mei, Y. Fan, Y. Zhou, L. Huang, T.S. Huang, and H. Shi, "Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.5690–5699, 2020.

[8] B. Niu, W. Wen, W. Ren, X. Zhang, L. Yang, S. Wang, K. Zhang, X. Cao, and H. Shen, "Single image super-resolution via a holistic attention network," Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16, pp.191–207, Springer, 2020.

[9] B. Li, Y. Lu, W. Pang, and H. Xu, "Image colorization using cyclegan with semantic and spatial rationality," Multimedia Tools and Applications, pp.1–15, 2023.

[10] Y. Mei, Y. Fan, and Y. Zhou, "Image super-resolution with non-local sparse attention," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.3517–3526, 2021.

[11] B. Xia, Y. Hang, Y. Tian, W. Yang, Q. Liao, and J. Zhou, "Efficient non-local contrastive attention for image super-resolution," Proceedings of the AAAI Conference on Artificial Intelligence, pp.2759–2767, 2022.

[12] L. Wang, Y. Wang, X. Dong, Q. Xu, J. Yang, W. An, and Y. Guo, "Unsupervised degradation representation learning for blind super-resolution," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.10581–10590, 2021.

[13] Z. Wang, J. Chen, and S.C. Hoi, "Deep learning for image super-resolution: A survey," IEEE Trans. Pattern Anal. Mach. Intell., pp.1–1, 2020.

[14] Y. Liu, Z. Chu, and B. Li, "A local and non-local features based feedback network on super-resolution," Sensors, vol.22, no.24, 2022.

[15] Y. Liu and Z. Chu, "A dynamic fusion of local and non-local features-based feedback network on super-resolution," Symmetry, vol.15, no.4, p.885, 2023.

[16] P. Behjati, P. Rodriguez, C. Fernández, I. Hupont, A. Mehri, and J. Gonzàlez, "Single image super-resolution based on directional variance attention network," Pattern Recognition, vol.133, p.108997, 2023.

[17] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," ICLR, 2019.

[18] H. Chen, J. Gu, and Z. Zhang, "Attention in attention network for image super-resolution," arXiv preprint arXiv:2104.09497, 2021.

[19] M. Bevilacqua, A. Roumy, C. Guillemot, and M.L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," In Proceedings of the 23rd British Machine Vision Conference (BMVC), 2012.

[20] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representation," International conference on curves and surfaces, pp.711–730, Springer, 2010.

[21] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, pp.416–423 vol.2, 2001.

[22] J.B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.5197–5206, 2015.

[23] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, "Sketch-based manga retrieval using manga109 dataset," Multimedia Tools and Applications, vol.76, no.20, pp.21811–21838, 2017.

[24] J. Gu and C. Dong, "Interpreting super-resolution networks with local attribution maps," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.9199–9208, 2021.

[25] S. Xiong, B. Li, and S. Zhu, "Dcgnn: a single-stage 3d object detection network based on density clustering and graph neural network," Complex & Intelligent Systems, pp.1–10, 2022.

Yuhao Liu received the Ph.D. degrees in College of Computer Science and Technology from JiLin University in 2014. He is currently a lecturer with College of Information Engineering, Shanghai Maritime University. His current research interest is Single-Image Super-Resolution.

Zhenzhong Chu received the BSc. degree in mechanical design, manufacturing and automation in 2007, from Harbin Engineering University, and the Ph.D. degree in mechanical electronic engineering in 2013, from Harbin Engineering University. He is an associate professor in the School of Mechanical Engineering, University of Shanghai for Science and Technology. His current research interests include Image Processing, Path Planning.

Lifei Wei received the BSc and MSc degrees in applied mathematics from the University of Science and Technology Beijing, in 2005 and 2007, respectively and the Ph.D. degree in computer science from Shanghai Jiao Tong University, in 2013. He is currently a Professor with College of Information Engineering, Shanghai Maritime University. His main research interests include image processing, information security, AI security.