Artificial Intelligence Review (2024) 57:23
https://doi.org/10.1007/s10462-023-10648-4

DWSR: an architecture optimization framework for adaptive super-resolution neural networks based on meta-heuristics

Shu-Chuan Chu[1] · Zhi-Chao Dou[1] · Jeng-Shyang Pan[1,2] · Lingping Kong[3] · Václav Snášel[3] · Junzo Watada[4]

Published online: 30 January 2024

Abstract

Despite recent advancements in super-resolution neural network optimization, a fundamental challenge remains unresolved: as the number of parameters is reduced, the network's performance significantly deteriorates. This paper presents a novel framework called the Depthwise Separable Convolution Super-Resolution Neural Network Framework (DWSR) for optimizing super-resolution neural network architectures. The depthwise separable convolutions are introduced to reduce the number of parameters and minimize the impact on the performance of the super-resolution neural network. The proposed framework uses the RUNge Kutta optimizer (RUN) variant (MoBRUN) as the search method. MoBRUN is a multi-objective binary version of RUN, which balances multiple objectives when optimizing the neural network architecture. Experimental results on publicly available datasets indicate that the DWSR framework can reduce the number of parameters of the Residual Dense Network (RDN) model by 22.17% while suffering only a minor decrease of 0.018 in Peak Signal-to-Noise Ratio (PSNR), the framework can reduce the number of parameters of the Enhanced SRGAN (ESRGAN) model by 31.45% while losing only 0.08 PSNR. Additionally, the framework can reduce the number of parameters of the HAT model by 5.38% while losing only 0.02 PSNR.

Keywords Neural architecture search · Super-resolution · Swarm intelligence · Multi-objective · Runge Kutta optimizer

1 Introduction

Super-resolution reconstruction of images is the technique of restoring a low-resolution image to a high-resolution image that is true, clear, and with as few human traces as possible Hou and Andrews (1978). Compared with low-resolution images, high-resolution images usually contain greater pixel density, richer texture details, and higher trustworthiness. However, we usually cannot directly obtain high-resolution images with sharpened edges and no block blur due to the limitations of recording devices and image degradation models Bulat et al. (2018). There are many image super-resolution methods, such as interpolation-based, degradation model-based, and deep learning-based methods Keys (1981);

Extended author information available on the last page of the article

Schermelleh et al. (2019). Dong et al. first proposed using convolutional neural networks to deal with the image super-resolution problem in 2014 Dong et al. (2014). A three-layer convolutional neural network (SRCNN) is designed to learn the mapping relationship between low-resolution and high-resolution images directly in this paper. In 2016, Shi et al. considered the Efficient Sub-Pixel Convolutional Neural Network (ESPCN) from a low-resolution image and learned how to scale the image from a sample Shi et al. (2016). In 2017, Christian Ledig et al. proposed a super-resolution image reconstruction by adversarial networks from a photo-aware perspective Ledig et al. (2017). In recent years, researchers have been committed to getting higher accuracy of the network and enhancing the credibility of the generated images Liu et al. (2022). At the same time, they also hope to reduce the number of network parameters and improve the confidence of image generation.

Most of the network architectures currently in use have been carefully designed by researchers. The number of parameters usually increases when designing to improve network performance, increasing the generation time of super-resolution images Zhang et al. (2018); Lim et al. (2017); Wang et al. (2018). There is a pressing requirement to design lightweight networks based on how to reduce the number of network parameters effectively. Neural Architecture Search (NAS) Elsken et al. (2019) has made breakthroughs in various applications in recent years. Examples include image recognition, image segmentation, and super-resolution. In the super-resolution domain, chu et al. were the earliest to suggest the NAS technique with the Multi-objective reinforced evolution in mobile neural architecture search (MoreMNAS) to search for super-resolution neural network architecture Chu et al. (2019). Song et al. proposed using different super-resolution network sub-block combinations to enhance the network performance and reduce the network parameters Song et al. (2020). However, these methods are also just a mix-and-match combination of previous methods with limited optimization of the number of parameters. Depthwise separable convolution (DW Conv) has made notable achievements in the research of lightweight neural networks. However, using DW Conv fully in super-resolution neural networks leads to significant performance degradation. In this paper, we propose a framework that aims to be able to automatically insert DW Conv at appropriate locations in the network while minimizing the impact on performance.

Since the search space is enormous, finding a method with high searchability is necessary to improve the search speed and quickly find a better network architecture Morales-Hernández et al. (2022); Mishra and Kane (2022). Metaheuristic algorithms do not require problem-specific knowledge or information, which makes them suitable for complex problems where the problem structure and properties may not be easily understood or modeled. The meta-heuristic algorithm can perform a global search to find the approximate solution of the optimal solution Rodríguez-Molina et al. (2020); Akhand et al. (2020). During the meta-heuristic algorithm search, the exploration phase explores the search space as much as possible to find the areas where the optimal solution may exist Chu et al. (2006); Meng et al. (2019). Since the optimal solution may exist at any location throughout the search space, a detailed search of the areas near the current optimal solution is performed in the development phase. In most cases, there are some correlations between solutions. The meta-heuristic algorithm uses these correlations to adjust the solution process Chu et al. (2005); Wang et al. (2022). The mathematically based RUNge Kutta optimizer (RUN) algorithm Ahmadianfar et al. (2021) is a typical example. The RUN Optimizer balances the exploration and development phases by designing the Runge Kutta Search Mechanism (RKM) for exploration and the Enhanced Solution Quality (ESQ) mechanism for exploitation. However, the RUN optimizer is designed for continuous optimization problems and cannot be applied to combinatorial optimization problems. In this paper, we propose a transfer function that maps the continuous solution space to the discrete solution space, enabling it to solve combinatorial optimization problems. In addition, in order to avoid performance degradation due to excessive use of DW Conv (or excessive number of parameters due to excessive use of DW Conv) during the optimization process, we also propose a multi-objective optimization strategy for balancing the relationship between PSNR and the number of parameters during the search process.

In this paper, an efficient and straightforward method for super-resolution network optimization is proposed. The search advantage of the meta-heuristic algorithm is implemented for NAS. The main contributions of this study are as follows:

- MoBRUN is employed to balance the PSNR and the number of parameters.
- A grid mechanism is established for the non-dominated solution in archives.
- New leader selection schemes are presented to improve the position updating method of population individuals in the binary multi-objective meta-heuristic algorithm.
- A novel framework is proposed to apply the MoBRUN algorithm to NAS to optimize super-resolution neural networks.

The remaining sections of this manuscript are organized as follows: Sect. 2 briefly describes the development of the meta-heuristic algorithm and its application in NAS. Section 3 introduces the original RUNge algorithm in detail. Section 4 presents the improved MoBRUN algorithm. Section 5 provides the proposed framework. Section 6 discussed the experiment and its results. Section 7 is the conclusion of this paper.

2 Related works

Research into meta-heuristics has a long history. In the past, most research on meta-heuristic algorithms has emphasized their use in problems such as engineering optimization. An example is the Particle Swarm Optimization (PSO) algorithm Marini and Walczak (2015), initially based on the stochastic optimization technique of populations. Simulated Annealing (SA) algorithm Delahaye et al. (2019) for simulated metal annealing design. The Ant Colony Optimization (ACO) Zhou et al. (2022) algorithm is designed to abstract ants searching for food and record their paths. Meta-heuristic algorithms have demonstrated their usefulness in several fields Wang et al. (2014); Chu et al. (2022).

In recent years, some researchers have tried to apply meta-heuristic algorithms to solve NAS problems. Wang et al. combined the PSO algorithm with Convolutional Neural Network (CNN) and proposed the cPSO-CNN algorithm Wang et al. (2019), which can automatically search the CNN architecture. Lu et al. explored a multi-objective genetic algorithm for neural network search Lu et al. (2020), which is better in terms of interactivity and structural design.

Together, these studies outline the critical role of meta-heuristics in NAS. However, the focus of such studies remains narrow and only deals with applying meta-heuristics to NAS. Once the meta-heuristic algorithm is converted to the binary version, the position is only selected between 0 and 1 Beheshti (2020); Akay et al. (2021). When the binary meta-heuristic optimization algorithm performs multi-objective optimization, the leader selection mechanism enables individuals to converge to the current Pareto frontier. This phenomenon reduces the diversity of individuals, and the algorithm is likely to fall into the optimal local solution Tian et al. (2021); Liu et al. (2020); Zhang et al. (2020). Therefore, the MoBRUN algorithm is proposed for solving the problem and finding the best solution for the multi-objective NAS problem.

3 RUNge Kutta optimizer

The RUN algorithm proposed by Iman et al. is based on the specific slope calculation of the Runge Kutta method Butcher (1987). It is an effective global optimization search strategy. RUN consists of two main parts: RKM for exploration and ESQ mechanism for exploitation.

3.1 Initialization step

The meta-heuristic algorithm is a method that uses N individuals to optimize D dimensions. For the enhancement of increase the randomness and diversity of individuals in the initial stage, the initial positions of individuals in the RUN optimizer are generated using Eq.(1).

$$x_{n,d} = L_d + rand.(U_d - L_d) \tag{1}$$

where $x_{n,d}$ in Eq.(1) is the location of the individual and the solution of the optimization problem of dimension D. L_d and U_l are the upper and lower bounds of the d-th variable of the problem to be optimized ($d = 1, 2, ..., D$). The $rand$ is a random number within [0, 1].

The dominant search mechanism in RUN is an RK4 based approach. This method searches the decision space with the aid of three randomly selected solutions. The mechanism can be modeled as:

$$SM = \frac{1}{6}(x_{RK})\Delta x \tag{2}$$

in which

$$x_{RK} = k_1 + 2 \times k_2 + 2 \times k_3 + k_4 \tag{3}$$

RUN performs random global (exploration) and local (exploitation) searches in each iteration. When $rand < 0.5$, perform the global search method; otherwise, perform the local search method. The search method is designed using the RK method. The new solution is determined by Eq.(4).

$$x_{n+1} = \begin{cases} (x_c + SF \times X_c \times h \times r) + SM \times SF + \mu \times randn.(x_m - x_c) & \textbf{\textit{if}}\ rand < 0.5 \\ (x_m + SF \times X_m \times h \times r) + SM \times SF + \mu \times randn.(x_{r1} - x_{r2}) & \textbf{\textit{else}} \end{cases} \tag{4}$$

in which

$$\mu = 0.5 + 0.1 \times randn \tag{5}$$

where r is used to change the search direction and add diversity, and is an integer taking the value of 1 or -1. SF is an adaptive factor. Parameter μ is a random number, and $randn$ is a normally distributed random number. Parameter h is a random number taking values in the range [0,2]. Parameters x_m and x_c are calculated by the following equation:

$$x_c = x_n \times rand + (1 - rand) \times x_{r1} \tag{6}$$

$$x_m = x_{best} \times rand + (1 - rand) \times x_{lbest} \tag{7}$$

where x_{best} is the optimal solution achieved, the x_{lbet} is the best solution obtained for current iteration, and x_{r1} is the position of an individual randomly selected in the population.

3.2 Enhanced solution quality mechanism

The ESQ mechanism is used to improve the quality of the solution and avoid getting trapped in a local optimum in each iteration. When *rand* > 0.5, the ESQ mechanism performs the following scheme to create the solution:

$$x_{new2} = \begin{cases} x_{new1} + \mid (randn - x_{avg} + x_{new1}) \mid \times w.r. & \textit{if } w < 1 \\ (x_{new1} - x_{avg}) + \mid randn - x_{avg} + u.x_{new1} \mid \times w.r & \textit{if } w > 1 \end{cases} \tag{8}$$

where w is a random number that decreases as the algorithm progresses, and parameter x_{avg} is the average of three randomly selected solutions. The x_{new1} is the best solution, and x_{avg} is the random number determined.

The solution calculated in Eq.(8) may not be better than the current one. In order to obtain a better solution, When $rand < w$, take the following steps to generate a new solution.

$$x_{new3} = x_{new2} \times (1 - rand) + SF.(v.x_b + rand.x_{RK} - x_{new2}) \tag{9}$$

where v is a random number taking values in the range [0,2].

The pseudo-code of the RUN algorithm is given in Algorithm 1.

Algorithm 1 The pseudo-code of RUN algorithm

```
Initialization: Generate the population size $N$, dimension $D$, and maximum number of iterations $Max_{iter}$.
Determine the solution $X_{best}$.
while i < $Max_{iter}$ do
    while n < N do
        Calculate the position of each individual by Eq.(4).
        if 0.5 > $rand$ then
            Calculate position $x_{new2}$ by Eq.(8)
            if $f(x_{new2}) > f(x_n)$ then
                if $w > rand$ then
                    Calculate position $x_{new3}$ by Eq.(9)
        Update positions $x_w$ and $x_b$
        n=n+1
    Update position $x_{best}$
    i=i+1
Output: The best solution
```

4 The proposed MoBRUN method

Because the RUN algorithm was initially designed for solving problems with continuous space, we proposed MoBRUN to solve the NAS problem of super-resolution networks.

4.1 Binary conversion

The original RUN algorithm is designed to solve continuous problems, so it is required to convert to a binary version to solve the NAS problem. Numerous studies show that the values in continuous space can be converted into binary space after normalization by transfer function. The common transfer functions are S-, V-, and U-shaped transfer functions Mirjalili and Lewis (2013); Mirjalili et al. (2020); He et al. (2022), and here we choose to use the V-shaped transfer function to convert the RUN algorithm. The V-shaped transfer function is shown in Eq.(10).

$$V_{x_{n,d}} = | \, erf(\frac{\pi}{2} \times x_{n,d}) \, | \tag{10}$$

where $x_{n,d}$ is the position of the n-th individual in the d-th dimension, and erf is the Gaussian error function. The V-shaped transfer function converts the individual solution space from continuous to 0-1 space by Eq.(11) after normalization.

$$x_{n,d}^{t+1} = \begin{cases} \neg x_{n,d}^{t} & \textbf{\textit{if}} \ rand < V_{x_{n,d}} \\ x_{n,d}^{t} & \textbf{\textit{if}} \ rand \geq V_{x_{n,d}} \end{cases} \tag{11}$$

The image of the V-shaped function is shown in Fig. 1.

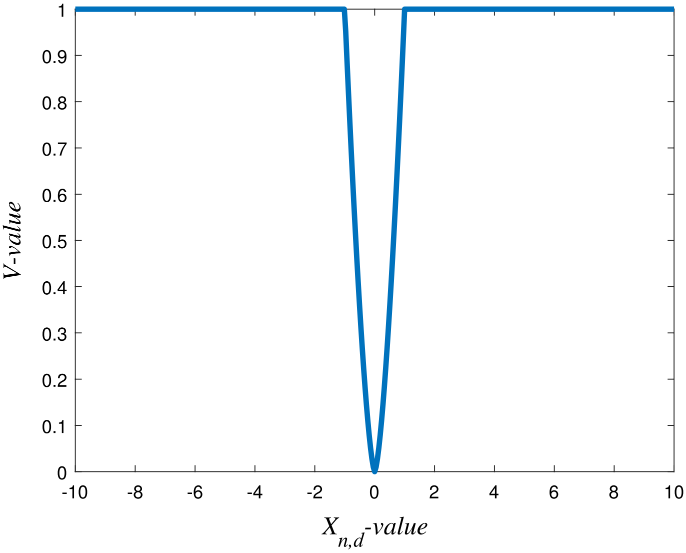


Fig. 1 V-shaped transfer function

4.2 Multi-objective strategy

This subsection applies two components to improve the RUN algorithm so that it is able to execute multi-objective optimization. One is the archive component responsible for storing the Pareto optimal solution, and another one is the leader selection component that selects leaders from the archive. The leader selection component assists the RUN algorithm in selecting the optimal solution for position updating.

An archive is a storage unit with a fixed size for storing Pareto optimal solutions. In an optimization problem with m objective functions, the solution vector $x = (x_1, x_2, ..., x_n)$ is assumed to minimize each objective function $f_i(x)$. In this context, a non-dominated solution is defined as follows:

$$\forall i \in 1, 2, ..., m, f_i(x_{new}) \leq f_i(x) \tag{12}$$

The non-dominated solutions $f_i(x_{new})$ obtained during the iterative process are compared with all solutions in the archive. Since the archive has a fixed size, new solutions entering the archive need to run the grid mechanism to redistribute the archive when it is full: the most crowded part of the current archive is found, and one of the solutions is omitted. The new solution is then inserted into the most sparse part to multiply the diversity of the Pareto optimal frontier of the final approximation. It should be noted that in the process of using a binary meta-heuristic optimization algorithm to solve problems, a large number of identical solution spaces will be generated in the archive. Therefore, it is necessary to prioritize removing identical solutions when deciding the dominance relation.

When the RUN algorithm conducts a search work, we hope to find an optimal solution to guide the next step. Therefore, a leader selection mechanism is introduced to handle this problem. The leader selection mechanism in which the best solution is selected from the archive using a roulette wheel method. The advantage of this mechanism is expressed as follows:

$$P_t = \frac{q}{N_s} \tag{13}$$

where q is a constant greater than 1 and N_s is the best solutions number in the archive for the current iteration.

However, in the RUN algorithm, using the Pareto dominance relation for location update will lead to the problem of a slow location update. Therefore, to increase location diversity and speed up searches, we use the replacement strategy shown in Algorithm 2 to update the location.

Algorithm 2 The pseudo-code of replacement strategy

```
Input: The fitness value obtained for the t-th iteration of the n-th individual $F_n^t$, the fitness value obtained for the t + 1-th iteration of the n-th individual $F_n^{t+1}$.
if $F_n^t$ dominated $F_n^{t+1}$ then
    Update the position.
    if any $F_n^t > F_n^{t+1}$ then
        if rand()>0.7 then
            Update the position.
Output: The final position
```

5 DWSR Framework

Super-resolution neural networks usually have three phases: feature extraction, nonlinear mapping, and reconstruction. Compared with neural networks dealing with classification problems, super-resolution networks require more computational resources and are unsuitable for mobile devices. Researchers have recently preferred to design lightweight super-resolution neural networks Kim et al. (2021). However, designing new neural network architectures is time-consuming and laborious. So we can optimize based on previous neural network architectures to reduce the cost.

Andrew et al. proposed using a depthwise separable filter to reduce the number of neural network parameters for operation on mobile devices in 2017 Howard et al. (2017). For example, in a standard convolution operation, the number of parameters required is proportional to the product of the input and output channels and the kernel size. Specifically, if there are C_I input channels, C_O output channels, and a convolution kernel of size $D_K \times D_K$, then $C_I \times C_O \times {D_K}^2$ parameters are needed. By contrast, using a depthwise separable convolution reduces the computational cost by the following equation.

$$\frac{D_F \times D_F \times C_O \times C_I \times D_K \times D_K + C_I \times D_F \times D_F}{D_F \times D_F \times C_I \times D_K \times D_K \times C_O} = \frac{1}{C_O} + \frac{1}{{D_K}^2} \tag{14}$$

where D_F is the input feature map resolution.

First, as shown in Fig. 2, the deep convolution operation uses only a single convolution kernel for each input channel, which reduces the number of convolution kernels to $C_I \times {D_K}^2$. Second, as shown in Fig. 3, point-by-point convolution uses a 1 × 1 convolution kernel to map the result of deep convolution from C_I channels to C_O channels, requiring only $C_I \times C_O$ parameters. Deeply separable convolution reduces the computational complexity of the convolution operation by splitting it into two steps and using fewer parameters, which results in a lightweight and efficient model. In addition, it also works better with fewer data because the depth-separable convolution reduces the possibility of overfitting.

However, it should be noted that the performance of the super resolution neural network will decline sharply if all convolutions are replaced by depth separable filters. Therefore, we propose the DWSR framework: the MoBRUN algorithm combined with a depthwise separable filter is used to optimize the super-resolution neural network, and the number of

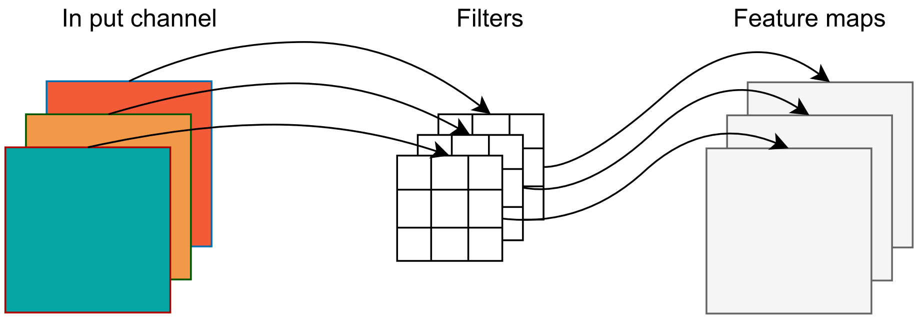


Fig. 2 The depthwise convolution operation

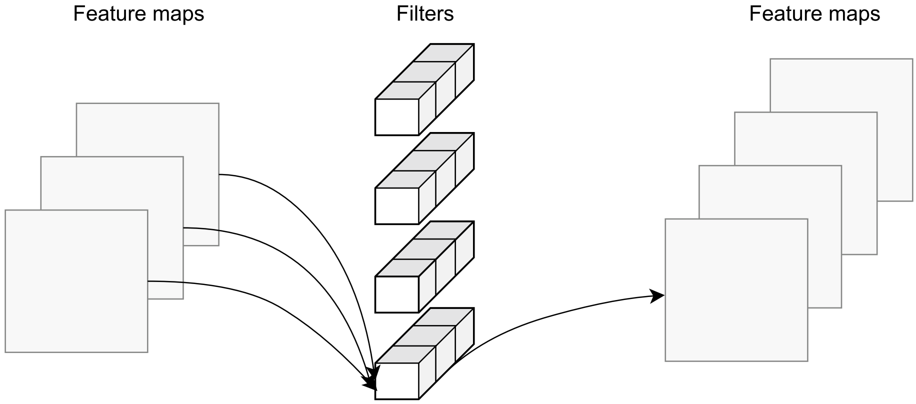


Fig. 3 The pointwise convolution operation

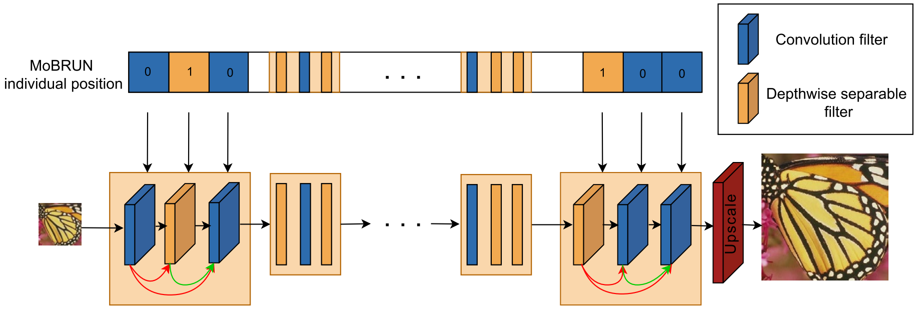


Fig. 4 The proposed DWSR framework

parameters and PSNR are evaluated to obtain the optimized super-resolution neural network architecture. Figure 4 shows the architecture of the DWSR framework.

The DWSR uses the MoBRUN algorithm to determine the position corresponding to the DW convolution in the neural network. MoBRUN uses 0 to indicate the corresponding position using regular convolution and 1 to indicate the corresponding position using the depthwise separable filter. This approach allows adaptive solution space exploration to obtain the best neural network architecture. The DWSR framework flowchart is shown in Fig. 5.

6 Experiments

In this section, RDN Zhang et al. (2018), ESRGAN Wang et al. (2018), and HAT Chen et al. (2023) are selected to test the effectiveness of the DWSR framework. When the DWSR framework is used for network architecture search, the PSNR and the number of parameters are used as decision indicators. Finally, we select three levels of network architectures for each network based on the number of parameters and compare the results with the actual model results.

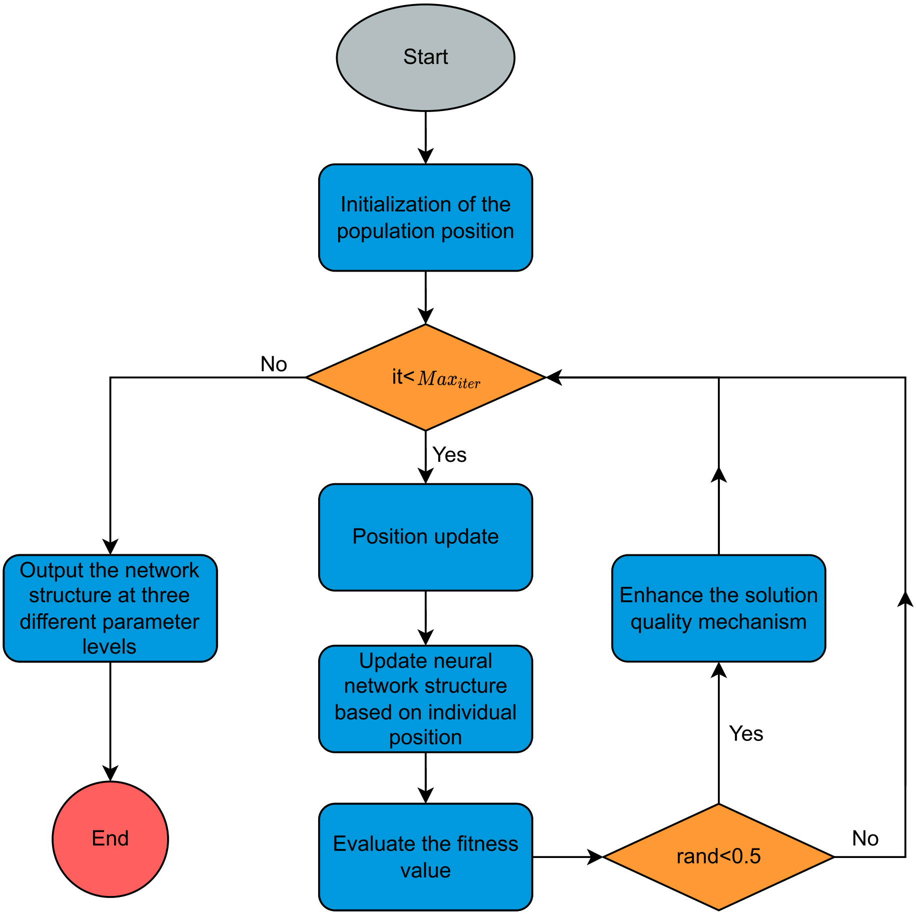

Start
Initialization of the population position
it<Max_{iter}
No
Yes
Position update
Update neural network structure based on individual position
Evaluate the fitness value
rand<0.5
Yes
No
Enhance the solution quality mechanism
Output the network structure at three different parameter levels
End

Fig. 5 The proposed DWSR framework flowchart

Table 1 Parameter settings of RDN, RRDB, and HAT in DWSR framework

	DWSR-RDN	DWSR-RRDB	DWSR-HAT
Dataset	DIV2K Agustsson and Timofte (2017)	DIV2K Agustsson and Timofte (2017)	DF2K Agustsson and Timofte (2017), Timofte et al. (2017)
Patchsize	16 × 16	64 × 64	32 × 32
Number of iterations	2500	2500	2500
Scale	× 4	× 4	× 4
Validation Dataset	Set5 Bevilacqua et al. (2012)	Set5 Bevilacqua et al. (2012)	Set5 Bevilacqua et al. (2012)
Downsampling method	bicubic(BI)	bicubic(BI)	bicubic(BI)
Learning rate	2×10^{-4}	1×10^{-4}	2×10^{-4}
Loss	L1 loss	L1 loss	L1 loss

6.1 Network choice

For the purpose of proving the effectiveness of DWSR, three typical neural networks are used for verification: RDN, ESRGAN, and HAT. RDN combines the advantages of ResNet and DenseNet to maximize the extraction of all feature information at the LR level. It can create a deeper super-resolution network through the GFF module. On the other hand, ESRGAN employs the GAN method to generate super-resolution images with more realistic details. ESRGAN also introduces the Residual-in-Residual Dense Block (RRDB) to construct generators that use perceptual loss to enrich the texture features of the generated images. HAT not only enhances the representation capability of the network by introducing a Hybrid Attention Block (HAB) but also establishes cross-window connections to activate more pixels by Overlapping the Cross-Attention Block (OCAB). The selected network architectures are verified using a baseline network structure: the RDN network uses 16 RDB blocks, the ESRGAN network uses 23 RRDB blocks and the HAT uses 6 residual hybrid attention groups (RHAG) with 6 HABs in each RHAG.

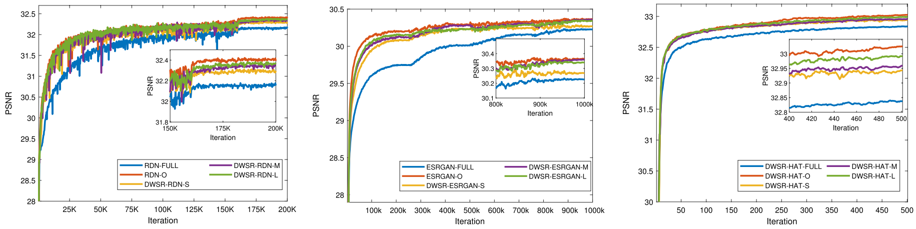
6.2 DWSR training details

When using DWSR framework to search network architecture, the initial training time of the network accounts for the majority. In order to reduce time consumption, the DWSR framework uses a smaller patch size for training. In the architecture search phase of ESRGAN, the PSNR-oriented model (RRDB Net) is used to initialize of the generator. Table 1 shows the hyperparameter settings for DWSR-RDN, DWSR-RRDB, and DWSR-HAT.

Table 2 shows the hyperparameter settings in the MoBRUN algorithm, where α, β and *ArchiveSize* are the parameters used to control the multi-objective strategy.

Table 2 Parameter values of the MoBRUN algorithm

Parameter	Value
α	0.1
β	4
Archive size	15
Population size	10
Number of iterations	40



(a) DWSR-RDN (b) DWSR-ESRGAN (c) DWSR-HAT

Fig. 6 Training curves for different network architectures (PSNR is evaluated on Set5 with Y channels)

Table 3 The archive for DWSR-RDN

Achieve	DWSR-RDN				
	Parameters(M)	Madds(G)	Flops(G)	PSNR	DW blocks
NO.1	7.55	16.39	8.22	29.85	97
NO.2	8.58	18.50	9.27	29.93	97
NO.3	8.97	19.29	9.67	30.06	88
NO.4	9.48	20.35	10.20	30.09	87
NO.5	9.93	21.27	10.66	30.12	83
NO.6	10.38	22.20	11.13	30.13	81
NO.7	10.71	22.86	11.45	30.17	80
NO.8	11.06	23.59	11.82	30.23	73
NO.9	12.45	26.43	13.24	30.30	63
NO.10	13.67	28.93	14.49	30.34	62
NO.11	15.15	31.97	16.01	30.36	45
NO.12	15.25	32.17	16.11	30.42	42
NO.13	17.31	36.39	18.22	30.47	30
NO.14	19.15	40.16	20.10	30.48	22
NO.15	20.05	42.00	21.02	30.49	19

6.3 Implementation details

The archives obtained by RDN are shown in Table 3, the archives obtained by RRDB are shown in Table 4, and the archives obtained by HAT are shown in Table 5. Three models of different levels of models are selected from three archives according to the size of

Table 4 The archive for DWSR-RRDB

Achieve	DWSR-RRDB				
	Parameters(M)	Madds(G)	Flops(G)	PSNR	DW blocks
NO.1	3.95	169.76	85.34	26.76	246
NO.2	4.01	171.58	86.25	27.06	244
NO.3	4.20	177.81	89.36	27.13	238
NO.4	5.58	223.06	111.94	27.17	166
NO.5	5.64	225.16	112.99	27.36	168
NO.6	6.30	246.70	123.74	27.44	140
NO.7	6.05	238.66	119.73	27.40	155

Table 5 The archive for DWSR-HAT

Achieve	DWSR-HAT				
	Parameters(M)	Madds(G)	Flops(G)	PSNR	DW blocks
NO.1	15.68	62.10	441.44	31.85	64
NO.2	16.16	66.00	443.36	31.89	58
NO.3	16.24	66.65	443.68	31.91	57
NO.4	16.48	68.60	444.64	31.92	54
NO.5	16.88	71.85	446.24	31.94	49
NO.6	17.12	73.80	447.20	31.94	46
NO.7	18.08	81.60	451.04	31.96	34
NO.8	18.64	86.15	453.28	31.99	27
NO.9	18.88	88.10	454.24	32.00	24
NO.10	19.68	94.60	457.44	32.03	14
NO.11	19.84	95.90	458.08	32.03	12

parameters: (DWSR-RDN-S: NO.4, DWSR-RDN-M: No.10, and DWSR-RDN-L: No.13. DWSR-RRDB-S: NO.3, DWSR-RRDB-M: NO.4, DWSR-RRDB-L: NO.6. DWSR-HAT-S: NO.2, DWSR-HAT-M: NO.6, DWSR-HAT-L: NO.10). Moreover, the three obtained network models were compared with the original network model and the fully used DW convolutional model. This paper trained three types of networks using the parameters shown in Table 6. The learning rate in RDN is halved at every 50K iterations. To obtain more realistic texture effects, the RRDB is also trained using GAN loss with the learning rate set to 1×10^{-4} and halved at [50k, 100k, 200k, 300k] iterations.

Furthermore, the convergence process of the different networks is visualized in Fig. 6. The convergence curves show that the network performance will be significantly affected when DW convolution is used exclusively instead of standard convolution. Not only does the DWSR further guarantee the training process, but also it will not compromise the effect markedly. These visual and quantitative analyses demonstrate the advantages and effectiveness of our proposed framework.

Table 6 Parameter setting of RDN, ESRGAN, and HAT in training

	DWSR-RDN	DWSR-ESRGAN	DWSR-HAT
Dataset	DIV2K Agustsson and Timofte (2017)	DIV2K Agustsson and Timofte (2017)	DF2K Agustsson and Timofte (2017); Timofte et al. (2017)
Patchsize	32 × 32	192 × 192	64 × 64
Iterations	200000	1000000	500000
Scale	× 4	× 4	× 4
Validation Dataset	Set5 Bevilacqua et al. (2012)	Set5 Bevilacqua et al. (2012)	Set5 Bevilacqua et al. (2012)
Downsampling method	bicubic(BI)	bicubic(BI)	bicubic(BI)
Learning rate	2×10^{-4}	1×10^{-4}	2×10^{-4}
Loss	L1 loss	L1 loss	L1 loss

6.4 Results with BI degradation model

The paper used a common BI degradation model in SR to generate LR images. The optimized networks were compared with six state-of-the-art image SR methods: SRCNN Dong et al. (2014), SRDenseNet Tong et al. (2017), LapSRN Lai et al. (2017), CARN Ahn et al. (2018), DRRN Tai et al. (2017), VDSR Kim et al. (2015), SwinIR Liang et al. (2021), and EDT Li et al. (2021) on five standard benchmark datasets. These datasets are Set5 Bevilacqua et al. (2012), Set14 Zeyde et al. (2012), BSD100 Martin et al. (2001), and Urban100 Huang et al. (2015). SR results were evaluated using PSNR and SSIM on the Y channel (i.e., luminance) of the transformed YCbCr space.

Table 7 (×4 scale) provides a quantitative comparison of the performance of the benchmark dataset. The best performance of DWSR-RDN, DWSR-ESRGAN, and DWSR-HAT optimization are bolded. The table shows that the proposed methods (DWSR-RDN-L, DWSR-ESRGAN-L, and DWSR-HAT-L) achieve better performance on the ×4 scale than other prominent methods on all benchmark datasets with essentially no loss of performance. DWSR-RDN-L lost 0.018 PSNR on average over the five datasets, DWSR-RRDB-L lost 0.08 PSNR on average over the five datasets, and DWSR-HAT-L lost 0.02 PSNR on average over the five datasets.

Figure 7 intuitively illustrates the qualitative comparison of the ×4 scale on different images. It is clear that the images reconstructed by other methods contain significant artefacts and blurred edges. In contrast, optimized super-resolution network provides equally more realistic images with sharp edges. Optimized network architecture is able to recover images better than other prominent models while substantially reducing the number of parameters. DWSR-RDN-L has a 22.17% reduction in the number of parameters compared to the original architecture, DWSR-ESRGAN-L reduces the number of parameters by 31.45% compared to the original architecture, and the DWSR-HAT-L reduces the number of parameters by 5.76% compared to the original architecture. However, they obtains comparable performance to the original network in terms of artifact removal and texture reconstruction.

Table 7 Comparison with state-of-the-art methods based on ×4 super-resolution tasks

Method	Set5	Set14	Urban100	Manga109	B100
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	28.42/0.8104	26.00/0.7027	23.14/0.6577	24.89/0.7866	25.96/0.6675
SRCNN Dong et al. (2014)	30.48/0.8628	27.50/0.7513	24.52/0.7221	27.58/0.8555	26.90/0.7101
SRDenseNet Tong et al. (2017)	32.02/0.8934	28.50/0.7782	26.05/0.7819	-/-	27.53/0.7337
LapSRN Lai et al. (2017)	31.54/0.8855	28.19/0.7720	25.21/0.7553	29.09/0.8893	27.32/0.7280
CARN Ahn et al. (2018)	32.13/0.8937	28.60/0.7806	26.07/0.7837	-/-	27.58/0.7349
DRRN Tai et al. (2017)	31.68/0.8888	28.21/0.7721	25.44/0.7638	29.18/0.8914	27.38/0.7284
VDSR Kim et al. (2015)	31.35/0.8838	28.01/0.7674	25.18/0.7524	-/-	27.29/0.7251
SwinIR Liang et al. (2021)	32.92/0.9044	29.09/0.7950	27.92/0.7489	32.03/0.9260	27.45/0.8254
EDT Li et al. (2021)	32.82/0.9031	29.09/0.7939	27.91/0.7483	32.05/0.9254	27.46/0.8246
RDN-O	32.34/0.8973	28.74/0.7855	26.38/0.7958	30.82/0.9125	27.65/0.7391
RDN-F	32.10/0.8940	28.55/0.7807	25.97/0.7828	30.38/0.9070	27.54/0.7353
DWSR-RDN-S	32.23/0.8960	28.69/0.7841	26.23/0.7911	30.69/0.9108	27.61/0.7377
DWSR-RDN-M	32.29/0.8964	28.68/0.7838	26.28/0.7928	30.79/0.9122	27.63/0.7383
DWSR-RDN-L	**32.30/0.8971**	**28.71/0.7850**	**26.37/0.7951**	**30.82/0.9128**	**27.64/0.7388**
RRDB-O	32.24/0.8962	28.66/0.7838	26.20/0.7904	30.55/0.9098	27.60/0.7386
RRDB-F	31.84/0.8916	28.39/0.7782	25.67/0.7728	29.75/0.9006	27.44/0.7334
DWSR-RRDB-S	32.10/0.8949	28.54/0.7811	25.97/0.7825	30.23/0.9063	27.54/0.7364
DWSR-RRDB-M	**32.15/0.8955**	28.58/0.7819	26.08/0.7863	30.38/0.9080	27.57/0.7374
DWSR-RRDB-L	32.11/0.8952	**28.59/0.7823**	**26.13/0.7879**	**30.44/0.9088**	**27.58/0.7375**
HAT-O	33.04/0.9056	29.23/0.7973	28.00/0.7517	32.48/0.9292	27.97/0.8368
HAT-F	32.84/0.9032	29.08/0.7947	27.94/0.7493	32.18/0.9280	27.76/0.8326
DWSR-HAT-S	32.94/0.9049	29.14/0.7958	27.97/0.7509	32.38/0.9287	27.90/0.8352
DWSR-HAT-M	32.96/0.9052	29.17/0.7958	27.98/0.7513	32.41/0.9289	27.92/0.8356
DWSR-HAT-L	**33.00/0.9055**	**29.21/0.7958**	**27.98/0.7514**	**32.47/0.9290**	**27.95/0.8367**

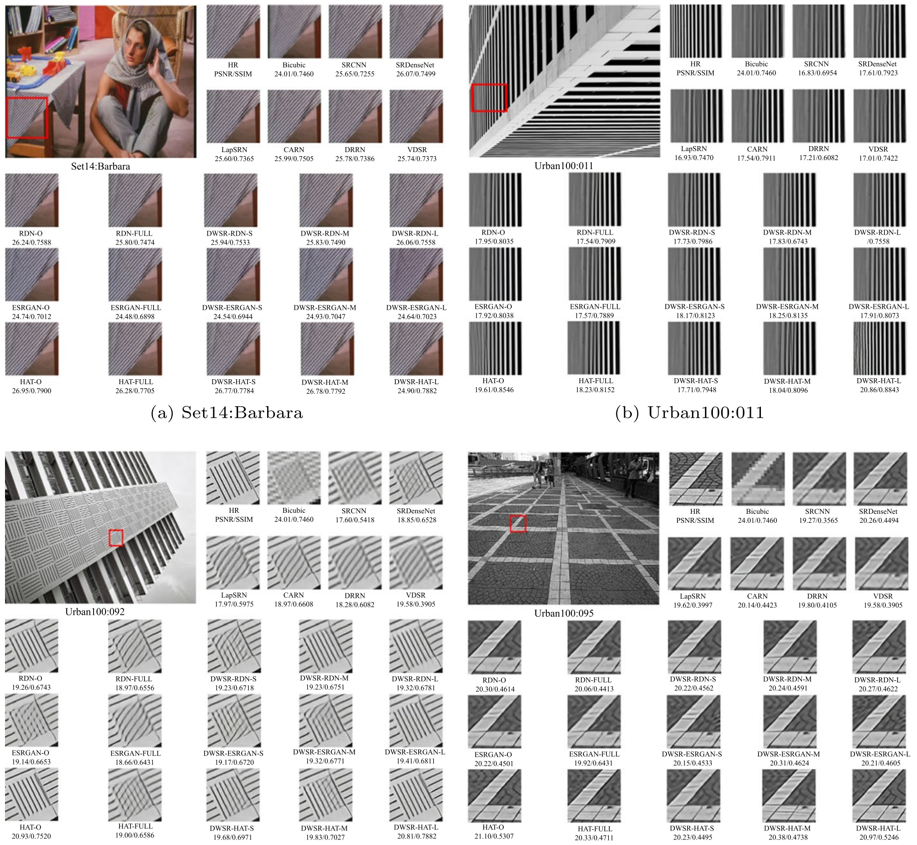
6.5 Ablation atudy

For ablation experiments, we train our framework for image super-resolution (×4) based on DIV2K Agustsson and Timofte (2017) datasets and RDN Zhang et al. (2018) model. The results are evaluated on Set5 benchmark dataset.

6.5.1 Design choices for grid size

We conducted an ablation study to demonstrate the importance of different archive sizes. Specifically, we evaluated three different archive sizes: 5, 15, and 30, while keeping all other experimental settings constant. The results of these experiments are presented in Table 8.

Table 8 shows the effect of different archive sizes on the results. It is apparent that there is a strong correlation between archive size and the resulting network structure. When the archive size is set to 5, the obtained network structure scheme has a low PSNR and dense distribution. Conversely, when the archive size is set to 30, the archive cannot be fully



(a) Set14:Barbara

(b) Urban100:011

(c) Urban100:092

(d) Urban100:095

Fig. 7 Visual results of the BI degradation model using a scale factor of ×4

utilized. When the archive size is set to 15, the distribution scheme is more even and has a richer PSNR distribution than when the archive size is set to 30. In order to strike a balance between performance and efficiency, we select an archive size of 15 for the remainder of the experiments.

Table 8 Ablation study on Archive size design

Archive Size	Number of archives retained	Max PSNR	Min PSNR	Average PSNR
5	5	30.11	29.92	29.98
15	15	30.49	29.85	30.23
30	18	30.50	29.91	30.25

6.5.2 Design choices for iteration number

Table 9 presents the impact of the number of iterative searches on the final performance of the prediction model. There is a positive correlation between the number of iterations and the model's final performance. When the number of iterations is small, the final performance of the model cannot be accurately evaluated, even though the search time is reduced accordingly. As the number of iterations increases beyond 2500, the improvement in evaluation gain gradually becomes saturated. To strike a balance between search time and evaluation accuracy, we set the number of search iterations to 2500 for the remainder of the experiments to obtain an accurate evaluation within a relatively short search time.

6.5.3 Design choices for meta-heuristic algorithm

To ensure a fair comparison, we evaluated the MoBRUN algorithm against several classical meta-heuristics over 40 iterations. The results of this comparison, in terms of PSNR and the number of archives, are presented in Table 10. It is evident that MoBPSO produces only 9 archives, with a resulting network architecture that has a low PSNR. MoBDE experiences similar difficulties. While MoBMPA, MoBGWO, and MoBSMA select a larger number of archives, they suffer from a lack of diversity and tend to be densely concentrated in certain ranges. By contrast, the MoBRUN algorithm is able to deliver better solutions.

7 Conclusion

Fine-tuning DW convolution has always been a challenge for obtaining satisfactory CNN network architectures. This is primarily due to the high cost of trial and error involved in the process. To overcome this obstacle, it is necessary to speed up the network search and reduce the cost required to evaluate the network. In this paper, we propose the DWSR framework, which introduces a meta-heuristic algorithm to accelerate the network architecture search. The multi-objective mechanism provides multiple network structure choices, and the network architecture search is accelerated by changing the patch size rather than reducing the number of iterations. The DWSR framework with these mechanisms minimizes the impact on the network performance while obtaining fewer parameters, and there is a substantial improvement in network search speed. Our work suggests: Developing suitable variants of meta-heuristic algorithms is a potential direction for optimizing super-resolution networks.

Table 9 Ablation study on iteration number design

DW Blocks	Iterations						
	500	1000	1500	2000	2500	3000	200k
97	7.86	18.09	20.87	23.55	23.99	24.95	29.85
87	7.01	18.56	21.56	23.15	24.35	25.04	30.09
80	9.23	18.58	21.42	23.98	24.36	25.28	30.17
62	8.15	18.02	22.41	23.66	24.5	25.34	30.34
30	8.16	19.31	22.38	24.47	24.62	25.4	30.47

Table 10 Ablation study on meta-heuristic algorithm choices

Meta-heuristic Algorithm	Number of archives retained	Max PSNR	Min PSNR	Average PSNR
MoBPSO Kalaiarasu and Anitha (2020)	9	28.12	26.73	27.15
MoBDE Wang et al. (2023)	11	29.42	28.57	29.11
MoBMPA Zhong et al. (2021)	12	30.21	30.03	30.15
MoBGWO Moldovan and Slowik (2021)	15	29.75	28.54	29.12
MoBSMA Yacoubi et al. (2022)	15	30.50	28.21	29.99
MoBRUN	15	30.49	29.91	30.25

Author contributions S-CC, Z-CD, and J-SP wrote the main manuscript text and LK, VS, and JW revised the content critically. All authors reviewed the manuscript.

Funding N/A. No funding to declare.

Declarations

Conflict of interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agustsson, E, Timofte, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2017)

Ahmadianfar I, Heidari AA, Gandomi AH, Chu X, Chen H (2021) Run beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method. Expert Syst Appl 181:115079

Ahn N, Kang B, Sohn K-A (2018) Fast, accurate, and lightweight super-resolution with cascading residual network. In: Proceedings of the European Conference on Computer Vision (ECCV)

Akay B, Karaboga D, Gorkemli B, Kaya E (2021) A survey on the artificial bee colony algorithm variants for binary, integer and mixed integer programming problems. Appl Soft Comput 106:107351

Akhand MAH, Ayon SI, Shahriyar SA, Siddique N, Adeli H (2020) Discrete spider monkey optimization for travelling salesman problem. Appl Soft Comput 86:105887

Beheshti Z (2020) A time-varying mirrored s-shaped transfer function for binary particle swarm optimization. Info Sci 512:1503–1542

Bevilacqua, M, Roumy, A, Guillemot, C, Alberi-Morel, M.L. Low-complexity single-image super-resolution based on nonnegative neighbor embedding (2012)

Bulat A, Yang J, Tzimiropoulos, G (2018) To learn image super-resolution, use a gan to learn how to do image degradation first. In: Proceedings of the European Conference on Computer Vision (ECCV)

Butcher JC (1987) The numerical analysis of ordinary differential equations: Runge-Kutta and General Linear Methods. Wiley-Interscience, Hoboken

Chen, X, Wang, X, Zhou, J, Qiao, Y, Dong, C. Activating more pixels in image super-resolution transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 22367–22377 (2023)

Chu S-C, Roddick JF, Pan J-S (2005) A parallel particle swarm optimization algorithm with communication strategies. J Info Sci Eng 21(4):9

Chu S-C, Tsai P-w, Pan J-S (2006) Cat swarm optimization. In: Yang Q, Webb G (eds) PRICAI 2006: trends in artificial intelligence. Springer, Berlin, pp 854–858

Chu S-C, Xu X-W, Yang S-Y, Pan J-S (2022) Parallel fish migration optimization with compact technology based on memory principle for wireless sensor networks. Knowledge-Based Syst 241:108124

Chu X, Zhang B, Xu R, Ma H (2019) Multi-objective reinforced evolution in mobile neural architecture search. arXiv preprint arXiv:1901.01074

Delahaye D, Chaimatanan S (2019) Mongeau, M. In: Gendreau M, Potvin J-Y (eds) Simulated annealing: from basics to applications. Springer, Cham, pp 1–35

Dong C, Loy CC, He K, Tang X (2014) Learning a deep convolutional network for image super-resolution. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer Vision - ECCV 2014. Springer, Cham, pp 184–199

Elsken T, Metzen JH, Hutter F (2019) Neural architecture search: a survey. J Mach Learn Res 20(1):1997–2017

He Y, Zhang F, Mirjalili S, Zhang T (2022) Novel binary differential evolution algorithm based on taper-shaped transfer functions for binary optimization problems. Swarm Evol Comput 69:101022

Hou H, Andrews H (1978) Cubic splines for image interpolation and digital filtering. IEEE Trans Acoust Speech Signal Process 26(6):508–517

Howard, A.G, Zhu, M, Chen, B, Kalenichenko, D, Wang, W, Weyand, T, Andreetto, M, Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR **abs/1704.04861** (2017)

Huang J-B, Singh A, Ahuja N (2015) Single image super-resolution from transformed self-exemplars. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Kalaiarasu M, Anitha J (2020) Analysis of micro array gene expression data using multi-objective binary particle swarm optimization with fuzzy weighted clustering (mobpso-fwc) technique. J Medical Imaging Health Info 10(5):1049–1056

Keys R (1981) Cubic convolution interpolation for digital image processing. IEEE Trans Acoustics Speech Signal Process 29(6):1153–1160

Kim S, Jun D, Kim B-G, Lee H, Rhee E (2021) Single image super-resolution method using CNN-based lightweight neural networks. Appl Sci 11(3):1092

Kim J, Lee JK, Lee KM (2015) Accurate image super-resolution using very deep convolutional networks. CoRR **abs/1511.04587**

Lai, W.-S, Huang, J.-B, Ahuja, N, Yang, M.-H. Deep laplacian pyramid networks for fast and accurate super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)

Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, Shi W (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Li W, Lu X, Qian S, Lu J, Zhang X, Jia J (2021) On efficient transformer-based image pre-training for low-level vision. arXiv preprint arXiv:2112.10175

Liang J, Cao J, Sun G, Zhang K, Van Gool L, Timofte R (2021) Swinir: Image restoration using swin transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pp. 1833–1844

Lim B, Son S, Kim H, Nah S, Mu Lee K (2017) Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops

Liu H, Li Y, Duan Z, Chen C (2020) A review on multi-objective optimization framework in wind energy forecasting techniques and applications. Energy Convers Manag 224:113324

Liu H, Ruan Z, Zhao P, Dong C, Shang F, Liu Y, Yang L, Timofte R (2022) Video super-resolution based on deep learning: a comprehensive survey. Artif Intell Rev 55(8):5981–6035

Lu Z, Deb K, Goodman E, Banzhaf W, Boddeti VN (2020) Nsganetv2: evolutionary multi-objective surrogate-assisted neural architecture search. In: Vedaldi A, Bischof H, Brox T, Frahm J-M (eds) Computer Vision - ECCV 2020. Springer, Cham, pp 35–51

Marini F, Walczak B (2015) Particle swarm optimization (pso) a tutorial. Chemom Intell Lab Syst 149:153–165

Martin D, Fowlkes C, Tal D, Malik J (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, vol. 2, pp. 416–4232

Meng Z, Pan J-S, Tseng K-K (2019) Pade: an enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization. Knowledge-Based Syst 168:80–99

Mirjalili S, Lewis A (2013) S-shaped versus V-shaped transfer functions for binary particle swarm optimization. Swarm Evol Comput 9:1–14

Mirjalili S, Zhang H, Mirjalili S, Chalup S, Noman N (2020) A novel u-shaped transfer function for binary particle swarm optimisation. In: Nagar AK, Deep K, Bansal JC, Das KN (eds) Soft computing for problem solving 2019. Springer, Singapore, pp 241–259

Mishra V, Kane L (2022) A survey of designing convolutional neural network using evolutionary algorithms. Artif Intell Rev 224:1–38

Moldovan D, Slowik A (2021) Energy consumption prediction of appliances using machine learning and multi-objective binary grey wolf optimization for feature selection. Appl Soft Comput 111:107745

Morales-Hernández A, Van Nieuwenhuyse I, Rojas Gonzalez S (2022) A survey on multi-objective hyperparameter optimization algorithms for machine learning. Artif Intell Rev, 1–51

Rodríguez-Molina A, Mezura-Montes E, Villarreal-Cervantes MG, Aldape-Pérez M (2020) Multi-objective meta-heuristic optimization in intelligent control: a survey on the controller tuning problem. Appl Soft Comput 93:106342

Schermelleh L, Ferrand A, Huser T, Eggeling C, Sauer M, Biehlmaier O, Drummen GP (2019) Super-resolution microscopy demystified. Nat Cell Biol 21(1):72–84

Shi W, Caballero J, Huszár F, Totz J, Aitken AP, Bishop R, Rueckert D, Wang Z (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1874–1883

Song D, Xu C, Jia X, Chen Y, Xu C, Wang Y (2020) Efficient residual dense block search for image super-resolution. Proc AAAI Conf Artif Intell 34(07):12007–12014

Tai Y, Yang J, Liu X (2017) Image super-resolution via deep recursive residual network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Tian Y, Si L, Zhang X, Cheng R, He C, Tan KC, Jin Y (2021) Evolutionary large-scale multi-objective optimization: a survey. ACM Comput Surv 54(8):1–34

Timofte, R, Agustsson, E, Van Gool, L, Yang, M.-H, Zhang, L. Ntire 2017 challenge on single image super-resolution: Methods and results. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2017)

Tong, T, Li, G, Liu, X, Gao, Q. Image super-resolution using dense skip connections. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)

Wang GL, Chu SC, Tian AQ, Liu T, Pan JS (2022) Improved binary grasshopper optimization algorithm for feature selection problem. Entropy 24(6):777

Wang H, Wu Z, Rahnamayan S, Sun H, Liu Y, Pan J-S (2014) Multi-strategy ensemble artificial bee colony algorithm. Info Sci 279:587–603

Wang P, Xue B, Liang J, Zhang M (2023) Feature selection using diversity-based multi-objective binary differential evolution. Info Sci 626:586–606

Wang Y, Zhang H, Zhang G (2019) CPSO-CNN: an efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks. Swarm Evolut Comput 49:114–123

Wang X, Yu K, Wu S, Gu J, Liu Y, Dong C, Qiao Y, Change Loy C (2018) Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops

Yacoubi S, Manita G, Amdouni H, Mirjalili S, Korbaa O (2022) A modified multi-objective slime mould algorithm with orthogonal learning for numerical association rules mining. Neural Comput Appl. https://doi.org/10.1007/s00521-022-07985-w

Zeyde R, Elad M, Protter M (2012) On single image scale-up using sparse-representations. In: Boissonnat J-D, Chenin P, Cohen A, Gout C, Lyche T, Mazure M-L, Schumaker L (eds) Curves Surfaces. Springer, Berlin, Heidelberg, pp 711–730

Zhang X, Liu H, Tu L (2020) A modified particle swarm optimization for multimodal multi-objective optimization. Eng Appl Artif Intell 95:103905

Zhang Y, Tian Y, Kong Y, Zhong B, Fu Y (2018) Residual dense network for image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Zhong K, Zhou G, Deng W, Zhou Y, Luo Q (2021) MOMPA: multi-objective marine predator algorithm. Comput Methods Appl Mech Eng 385:114029

Zhou X, Ma H, Gu J, Chen H, Deng W (2022) Parameter adaptation-based ant colony optimization with dynamic hybrid mechanism. Eng Appl Artif Intell 114:105139

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Shu-Chuan Chu[1] · Zhi-Chao Dou[1] · Jeng-Shyang Pan[1,2] · Lingping Kong[3] · Václav Snášel[3] · Junzo Watada[4]

✉ Jeng-Shyang Pan
jengshyangpan@gmail.com

Shu-Chuan Chu
scchu0803@gmail.com

Zhi-Chao Dou
douzhichao2021@163.com

Lingping Kong
lingping_kong@yahoo.com

Václav Snášel
vaclav.snasel@vsb.cz

Junzo Watada
junzuo.watada@gmail.com

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, Shandong, China

[2] Department of Information Managementy, Chaoyang University of Technology, Taichung 41349, Taiwan

[3] Faculty of Electrical Engineering and Computer Science, VšB-Technical University of Ostrava, Ostrava, Czech Republic

[4] Graduate School of Information, Production and Systems, Waseda University, Kitakyushu 808-0135, Japan