Applied Intelligence
https://doi.org/10.1007/s10489-024-05445-7

Supporting ANFIS interpolation for image super resolution with fuzzy rough feature selection

Muhammad Ismail[1,2] · **Changjing Shang**[1] · **Jing Yang**[3] · **Qiang Shen**[1]

Accepted: 2 April 2024

Abstract

Image Super-Resolution (ISR) is utilised to generate a high-resolution image from a low-resolution one. However, most current techniques for ISR confront three main constraints: i) the assumption that there is sufficient data available for training, ii) the presumption that areas of the images concerned do not involve missing data, and iii) the development of a computationally efficient model that does not compromise performance. In addressing these issues, this study proposes a novel lightweight approach termed Fuzzy Rough Feature Selection-based ANFIS Interpolation (FRFS-ANFISI) for ISR. Popular feature extraction algorithms are employed to extract the potentially significant features from images, and population-based search mechanisms are utilised to implement effective FRFS methods that assist in selecting the most important features among them. Subsequently, the processed data is entered into the ANFIS interpolation model to execute the ISR operation. To tackle the sparse data challenge, two adjacent ANFIS models are trained with sufficient data where appropriate, intending to position the ANFIS model of sparse data in the middle. This enables the two neighbouring ANFIS models to be interpolated to produce the otherwise missing knowledge or rules for the model in between, thereby estimating the corresponding outcomes. Conducted on standard ISR benchmark datasets while considering both sufficient and sparse data scenarios, the experimental studies demonstrate the efficacy of the proposed approach in helping deal with the aforementioned challenges facing ISR.

Keywords ANFIS · Sparse data · ANFIS interpolation · Image super resolution · Fuzzy rough feature selection

1 Introduction

Image super-resolution (ISR) is a subfield of computer vision and image processing that transforms low-resolution (LR) images into high-resolution (HR) images through enhancing the resolution of the former. HR images contain more pixels and hence, show more detailed information, whereas LR images show less detail and may involve considerable missing information due to a lower pixel count or lower pixel density. In certain situations, images obtained from different cameras may not be clear, and detailed information may be difficult to extract due to blurring or noise effects. Besides, high-quality cameras can be expensive, making it difficult for everyone to afford them and hence to gain direct access to HR images. Indeed, LR images or images with areas of sparse data are common problems encountered when capturing images with conventional cameras.

The problem of ISR has garnered significant attention from the image processing and computer vision community [1]. Given its importance, ISR is widely exploited in various application areas, such as medical diagnosis [2], security surveillance [3], remote sensing for earth observation [4, 5], astronomical observation [6], face recognition [7–9], reconstruction [10], vehicle identification [11], and Martian rock type classification [12] among others. High-quality HR images are simply desired in daily life with

✉ Changjing Shang
cns@aber.ac.uk

Muhammad Ismail
isi@aber.ac.uk; ismail@iba-suk.edu.pk

Jing Yang
yangjing199002@sxu.edu.cn

Qiang Shen
qqs@aber.ac.uk

[1] Department of Computer Science, Faculty of Business and Physical Sciences, Aberystwyth University, Aberystwyth, Wales, UK

[2] Department of Computer Science, Sukkur IBA University, Sukkur, Sindh, Pakistan

[3] School of Automation and Software Engineering, Shanxi University, Taiyuan, Shanxi, China

Published online: 20 April 2024

almost every application problem. Fortunately, companies and organisations can acquire such images without always needing to purchase expensive image-capturing equipment, as ISR techniques can be used to enhance images from a software perspective.

Previously, the main focus of ISR techniques was on the frequency domain, with wavelet or Fourier transform being the major techniques used to convert LR images into HR ones. However, such techniques typically ignored both the degradation process and the prior spatial information about the images concerned [1], restricting their ability to cope with complex situations. To address these limitations, contemporary ISR techniques based on the spatial domain have been developed, including learning-based and reconstruction-based methods [13]. The working mechanism of learning-based approaches is similar to data-driven machine learning. For instance, fuzzy rules learning-based techniques have been used to solve ISR problems, wherein the learned fuzzy rules are exploited to derive non-linear relationship mappings between LR and corresponding HR images [14]. On the other hand, reconstruction-based ISR methods utilise appropriately designed priors, such as edges, to recover missed details during the image reconstruction process [15, 16].

To aid in the successful addressing of challenging ISR problems, this paper proposes a novel lightweight FRFS-based ANFIS [17] interpolation approach that substantially extends the most recently developed ANFIS interpolation technique [13]. A pre-processing step is introduced that consists of feature extraction (FE) and fuzzy rough set-based feature selection (FRFS) mechanisms [18–20]. In particular, FRFS is adopted to facilitate the overall ISR system to work in uncertain and imprecise domains while maintaining interpretability. It offers an overall computationally efficient technique without compromising performance, inspired by the observation that many algorithms work on meta-level features of the images rather than their pixels, as pixel-based processing generally consumes more time. The existing ANFIS interpolation solution forms the core to deal with problems where certain images or regions of images are missing and hence, there are insufficient training data to develop a functioning ANFIS with existing ANFIS-learning-based techniques. This tackles a common yet important problem in data-driven ISR.

Following such an approach, in implementation, two adjacent ANFIS models are trained with sufficient data if and when such data is available, with the aim of placing a developing ANFIS model that addresses a certain region of sparse data in between them. As a result, a novel FRFS-ANFISI approach is established here, which works with features instead of pixels and employs the most important features extracted from the images, making the learned model lightweight. To evaluate the working efficacy of the proposed approach, a number of performance metrics are employed, including PSNR and SSIM, assessing the results on the following SR benchmark datasets: Set5 [21], Set14 [22], BSD100 [23], and Urban100 [24]. Promising experimental outcomes are achieved, reflecting the performance of FRFS-ANFISI both quantitatively and qualitatively

The remainder of the paper is organised as follows. Section 2 discusses the system architecture of the proposed approach. Section 3 contains the implementation specifications in principle. Section 4 presents an experimental evaluation of the proposed techniques. Finally, section 5 concludes the paper with interesting further work pointed out.

2 System architecture

The proposed FRFS-supported ANFIS interpolation approach for addressing the ISR problem is depicted in Fig. 1. Its implementation process consists of three phases: feature selection, training, and testing.

2.1 Selection of extracted features

In computer vision, one of the popular techniques of image processing is to break down an image into local regions of interest and then, describe the image using certain characteristics or features regarding the individual regions. Such a technique helps to reduce the complexity of actual image data while using their existing characteristics for further processing. Typically, *Keypoint detector* is used to detect salient regions of an image consistently irrespective of various transformations (e.g., scale, rotation) applied to the image, while *Keypoint descriptor* is used to extract the most distinctive and useful information from the already identified salient regions of that image [25]. There are many mature techniques for building feature detectors and descriptors in the literature (e.g., [25–30]). In this paper, *BRISK* [25] and *KAZE* [28] are adopted for the proposed approach owing to their popularity and observed suitability. As these methods are standard in the literature, only a brief introduction is provided here for completion; more details about them can be found in the literature (as referenced).

> BRISK features: Extraction of binary robust invariant scalable keypoints (BRISK) is a technique to detect (typically corner) point objects from a given image. It is both scale-invariant and rotation-invariant, different from many other (corner) feature detection techniques (e.g., FAST [31] and ORB [32]). A comprehensive evaluation of benchmark datasets reveals its adaptiveness and high-quality performance as with state-of-the-art feature extraction algorithms such as SURF [33] and SIFT [29].
> KAZE features: The extraction of this type of features is aimed to reduce noise, retain object boundaries, and ulti-

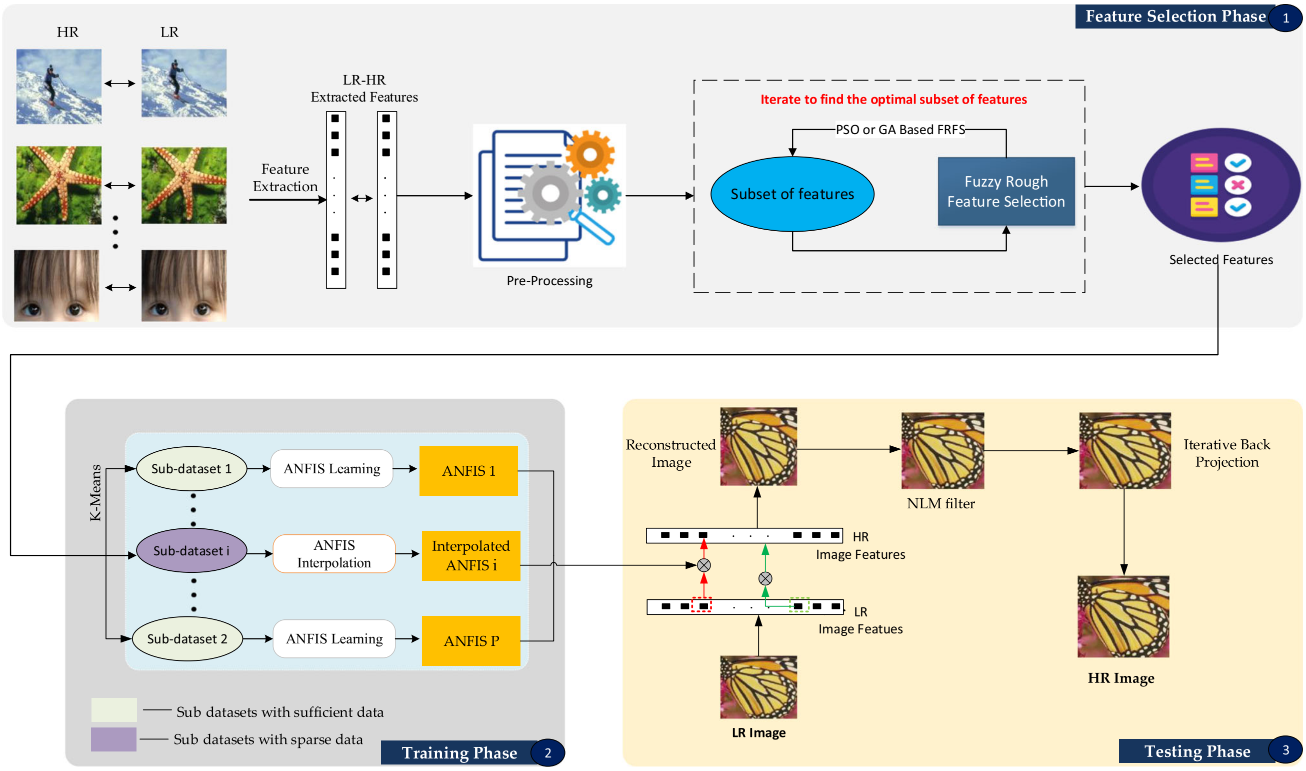


Fig. 1 Framework of proposed approach, involving three phases of feature selection, training and testing

mately achieve superior localisation accuracy [28]. It is a two dimensional multi-scale technique used to automatically detect features of interest at different scale levels or resolutions. KAZE outperforms state-of-the-art techniques such as SURF and SIFT, although it may be less computationally efficient.

Not all features extracted from a given image are important, and some may even be irrelevant or contain noise and be misleading, which will adversely affect the results of subsequent processing. Thus, it is crucial to identify unnecessary features and remove them from the input data. Various feature selection (FS) techniques have been proposed [34], but in many cases, it is not feasible to acquire an optimal feature subset through the required exhaustive search. When dealing with uncertain and ambiguous information, fuzzy rough-set (FR) theory has been shown to be a suitable approach [35]. Therefore, in the proposed approach, FR feature selection (FRFS) techniques [18] are employed, having recognised that fuzzy sets can handle vagueness, while rough sets can handle indiscernibility.

FRFS analyses only the patterns hidden in the data and requires no additional information, such as external expert knowledge or threshold value, for analysis. It works relying upon minimal knowledge representation for the underlying data. While there are many nature-inspired optimisation algorithms that work as subset search mechanisms in FRFS, the approach proposed herein uses particle swarm optimisation (PSO) and genetic algorithm (GA) due to their demonstrated effectiveness [19, 20]. Details on both algorithms and their corresponding encoding in FRFS are summarised in the following.

2.1.1 PSO-based fuzzy rough feature selection (*PSO-FRFS*)

Being a stochastic algorithm, PSO is commonly used in resolving optimisation problems [36], where a group of particles (representing potential solutions with each having a position and a velocity) move through the search space to seek the optimal solution. The movement is influenced by the particle's own experience and the experience of the group. At each iteration, the position of each particle is updated with respect to its velocity and the best position visited so far. The velocity of each particle is updated on the basis of: the velocity itself, the difference between its position and the best position it has visited, and the difference between its position and the best position of the group. The process continues until a stopping criterion is met or a maximum number of iterations is reached.

The PSO can be employed in FRFS to optimise the search for the optimal subset of features from the given feature space [19]. Without losing generality, consider the feature space derivable from a given image described in the form of a set of feature subsets. Then, each feature subset can be regarded as a position or a point in this space. There can be 2^N possible types of subsets for N features, and each type may be different in terms of both the subset size (number of elements) and the elements contained. The ideal feature subset would be the one with the lowest number of features without compromising the performance of subsequent processing with the selected subset.

Introducing the particle swarm into the feature space with each particle occupying one position, each particle moves in the space having the aim of exploring and finding the optimal position. To do so, each particle tries to find its best local and global position by iteratively changing its current position and communicating with others. During the iterative process, it is directed to reach the desired optimal solution, thereby identifying the best subset of features within the given feature space [19]. This ability of particles exploring the problem space significantly enhances computational efficiency, on required time and memory, for implementing the process of selecting an optimal subset of features in FRFS. Of course, the underlying fuzzy-rough feature dependency-measures are not altered, retaining the mathematical rigour of the generic FRFS approach.

2.1.2 GA search-based fuzzy rough feature selection (*GA-FRFS*)

Genetic Algorithm (GA) is another commonly used population-based computational method for solving search and optimisation problems. It implements a meta-heuristic process inspired by natural selection, following biological operators such as chromosome representation, selection, crossover, mutation, and fitness evaluation [20, 37]. More specifically, the GA algorithm begins with the random generation of an *initial population* P_0, where each individual I represents a potential solution to the problem at hand. A fitness function f is defined to quantify the quality of each individual. Parents are selected from the previous generation based on a *selection* probability $P(I)$, usually proportional to $f(I)$. The *crossover* operator C is then applied to pairs of parents, producing offsprings that inherit genetic information. A *mutation* operator M introduces small, random changes to some individuals. The resulting offsprings, along with surviving individuals from the previous generation, form the new generation P_t. This process iterates for a predetermined number of generations T. The algorithm concludes by identifying the best individual(s) in the final population P_T based on the fitness function.

The utilisation of GA to aid in implementing FRFS involves the random initialisation of the chromosome population, followed by the calculation of the fitness of each chromosome (which encodes a feature subset) and the selection of the best solution, regarding both individual chromosomes and on the entire population, based on their fitness values. Then, a crossover probability is applied to these local and global bests to generate offspring, which are subject to a mutation probability. The resulting offspring are stored as a new population, and this process iterates until a termination criterion is reached [37]. Once again, this application of a GA does not change the underlying measurements of features or feature subsets, but help accelerate the optimal feature subset search process.

2.2 ANFIS interpolation

Recall Fig. 1, the proposed FRFS-ANFISI approach involves a development process consisting of three learning phases: feature selection, training, and testing. The feature selection phase is performed first, followed by the training and testing phases. Algorithms 1 and 2 provide step-by-step instructions for the training and testing phases, respectively. The common techniques facilitating these two phases are detailed below.

Algorithm 1 Training phase of proposed approach.

```
Input:
  1. $D_{lr}$ features from LR training images $\{Z^l\}$;
  2. $D_{hr}$ features from HR training images $\{Z^h\}$;
  3. Scale factor $s$;
Training
  1. Create feature vector pairs $\mathbf{Q} = (\mathbf{x}, \mathbf{z})$ by combining LR-HR feature
     pairs of $\{\mathbf{X}^l, \mathbf{Z}^h\}$ images;
  2. Employ K-Means clustering algorithm to divide feature pairs $Q$ into
     $K$ subsets: $\{Q_i \,|\, \sum_i i = K\}$;

  3. Calculate each cluster centre $C_i$ for each subset $Q$;
  4. For those subsets $Q_i$ having sufficient data:
     a) Employ standard learning method of ANFIS to train ANFIS
        $\mathcal{A}_\mathbf{i}$;
  5. For those $Q_i$ having sparse data:
     a) For source ANFISs, take two closest neighbouring ANFISs;
     b) For target or Interpolated ANFIS $\mathcal{A}_\mathbf{m}$, apply ANFIS
        interpolation;
     c) Fine tune intermediate ANFIS $\mathcal{A}_\mathbf{m}$;
Output:
  Trained ANFIS models $\{\mathcal{A}_i\}$ and their cluster centres $\{C_i\}$.
```

The ANFIS interpolation technique, whose original ideas are described in [15], aims to overcome the issue of sparse data in the target domain (where less familiar image regions lie). It facilitates the construction of an effective target ANFIS model using rule interpolation, based on two neighbouring source ANFIS models that represent nearby regions to the target domain whose nature is substantially better known. For completeness, a brief introduction to the underlying ANFIS system is first given in the following. The general ANFIS

Algorithm 2 Testing phase of proposed approach.

```
Input:
  1. Testing low resolution image $\mathbf{Y^l}$;
  2. $E_{lr}$ features from LR image $\mathbf{Y^l}$;
  3. Trained ANFISs $\{\mathcal{A}_i\}$ and their cluster centres $\{C_i\}$;
Testing phase:
  1. Divide $E_{lr}$ features into $Q_i$ subsets regarding each $C_i$ cluster;
  2. For each feature $q_i \in Q_i$:
     a) Identify and select appropriate ANFIS model $\mathcal{A}_i$;
     b) Perform inference by employing trained ANFISs $\mathcal{A}_i$;
  3. Combine HR features to generate corresponding HR image $Y$;
  4. Post-processing:
     Employ NLM filter to remove noise;
     Employ IBP algorithm to refine image further;
Output:
  Estimated high resolution image $\mathbf{Y}$.
```

interpolation process is outlined, which can be divided into three stages as described subsequently.

2.2.1 Outline of ANFIS

ANFIS [17] stands for a type of fuzzy inference system that is implemented within the framework of an adaptive network. The information content of an ANFIS includes the network structure and the corresponding parameter learning mechanism. In general, there are totally five layers in a general ANFIS structure. An example of two-input and one-output system is illustrated here, with each input variable of the system described by two fuzzy sets. In this case, the system's rule base contains 4 fuzzy if-then rules of the so-called TSK type [38], as expressed below:

Rule1: If x_1 is A_1 and x_2 is A_3, then $y_1 = p_1x_1 + q_1x_2 + r_1$
Rule2: If x_1 is A_1 and x_2 is A_4, then $y_2 = p_2x_1 + q_2x_2 + r_2$
Rule3: If x_1 is A_2 and x_2 is A_3, then $y_3 = p_3x_1 + q_3x_2 + r_3$
Rule4: If x_1 is A_2 and x_2 is A_4, then $y_4 = p_4x_1 + q_4x_2 + r_4$

Structurally, such an ANFIS can be illustrated as shown in Fig. 2 [15], where square nodes in the first and fourth layer are the adaptive nodes with modifiable parameters, and circle nodes in the remaining layers involve fixed operations without parameters. In the first layer, each square node is defined with a membership function $\mu_{A_i}(x)$, where $i \in \{1, \cdots, 4\}$, x is the input variable and A_i is a fuzzy set defining an imprecise value of x. The most popularly applied function is triangular shaped due to its simplicity, which is defined by

$$\mu_A(x) = \begin{cases} k_1x + b_1 & a_0 \leq x \leq a_1 \\ k_2x + b_2 & a_1 \leq x \leq a_2 \\ 0 & otherwise \end{cases} \tag{1}$$

where $k_1 = 1/(a_1 - a_0)$, $b_1 = -a_0/(a_1 - a_0)$, $k_2 = 1/(a_1 - a_2)$, and $b_2 = -a_2/(a_1 - a_2)$ are named premise parameters, with $\{a_0, a_1, a_2\}$ being the three vertexes of the triangular membership function.

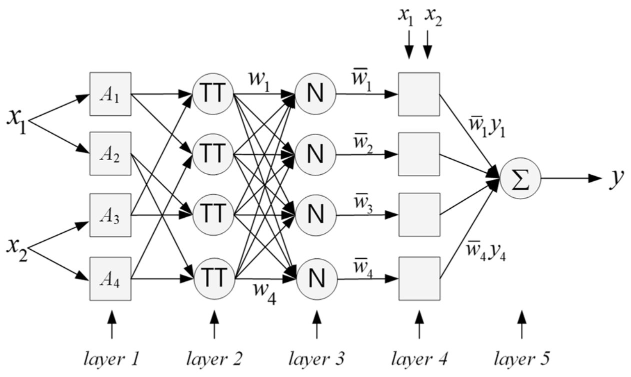


Fig. 2 Example of ANFIS encoding four rules

The second layer, *Layer 2* multiplies the incoming membership value of each variable and outputs the product: $w_i = \mu_{A_i}(x_1) \times \mu_{A_j}(x_2)$ $(i \in \{1, 2\},\ j \in \{3, 4\})$. The output of this layer w_i stands for the firing strength of a certain rule. Then *Layer 3* normalises each rule's firing strength by computing the relative proportion of a given rule's firing strength to the total of N rules' firing strengths $\bar{w_i} = w_i / \sum_{j=1}^{N} w_j$ (here $N = 4$, $i \in \{1, \cdots, 4\}$). In the fourth layer, each node is a square node with a linear function $\bar{w_i} y_i = \bar{w_i}(p_i x_1 + q_i x_2 + r_i)$, where p_i, q_i, r_i are referred to as consequent parameters. Finally, *Layer 5* is the output layer, computing the overall output in response to the current input, i.e., $y = \sum_i \bar{w_i} y_i$.

2.2.2 Rule dictionary construction

Suppose that for a given cluster of data, the two source domain ANFIS models and the target domain model are denoted as $\mathcal{A}_{s1}$, $\mathcal{A}_{s2}$, and $\mathcal{A}_t$, respectively. The two source ANFISs each contain fuzzy rules which are learned by following the conventional ANFIS training process, given the relevant regions are well known (or with sufficient training data). The task of constructing a rule dictionary is to store rule antecedent parts and rule consequent parts of the corresponding fuzzy rules. Then, the interpolation process is applied to exploit such information of the rule dictionary to learn $\mathcal{A}_t$ in the target domain.

LR images can be represented by the features extracted from them, which can be used to represent the rule antecedent parts with multiple attributes that correspond to specific features of the image. For the present ISR application fuzzy rules R_i, $i \in \{1, 2, \ldots, N\}$, are of the first-order TSK form [38], such that

$$R_i : if\ \ y\ is\ A_i\ then\ z_i = p_i y + r_i \tag{2}$$

where y is an input feature selected by FRFS (using either PSO or GA to optimise the search), representing an LR image element at a particular location, and its fuzzy set value is represented by A_i; z_i is the consequent of this rule (signifying the corresponding HR element), which is calculated as a linear combination of two parameters p_i and r_i together with the fuzzy value of y. Thus, ISR is a regression problem, and p_i and r_i are regression coefficients which enable the regression operation by transforming an LR image into an HR one.

To construct a rule dictionary, the antecedent and consequent parts are extracted from the fuzzy rules retrieved from $\mathcal{A}_{s1}$ and $\mathcal{A}_{s2}$. The resulting antecedents and consequent parts are then separated and arranged for easy indexing, such that

$$D = \{D_a, D_c\} \tag{3}$$

where D_a and D_c represent the antecedent parts and the consequent parts within the rule dictionary, respectively. More specifically, D_a containing all antecedents of the fuzzy rules is represented by

$$D_a = \{A_1 \ A_2 \ \cdots \ A_N\} \tag{4}$$

and D_c containing consequent parts of the fuzzy rules is represented by:

$$D_c = \begin{bmatrix} p_1 & p_2 & \cdots & p_N \\ r_1 & r_2 & \cdots & r_N \end{bmatrix} \tag{5}$$

where each column represents the linear coefficients in the consequent part of a certain rule.

2.2.3 Intermediate ANFIS creation

After constructing the rule dictionary, the second stage involves creating an intermediate ANFIS. The initial step in this stage involves the use of the K-means algorithm (although a more advanced clustering method, e.g., one of those described in [39, 40], may be used if preferred) to divide the sparse training data into C clusters.

Generally, let the set of training data (sparse or not) be denoted as (x, z). By generating clusters in the initial stage, it becomes possible to create fuzzy rules that are interpolated from the centre of distinct clusters. The intermediate rules are aggregated to construct an intermediate ANFIS, following the conventional interpretation of a set of TSK rules as an ANFIS. During implementation, the first procedure is to calculate the centre $c^{(k)}$ of each cluster C_k, where $k \in \{1, \ldots, C\}$. To construct an intermediate ANFIS, the L nearest rule antecedents $A_i \in D_a, i = 1, \ldots, L$ are selected, on the basis of their proximity to $c^{(k)}$. These rule antecedents are extracted from D_a of the rule dictionary generated in the previous stage. To achieve this, a distance metric is employed, which for the sake of simplicity, may be implemented with $d^i = d(A_i, c^{(k)}) = |Rep(A_i) - c^{(k)}|$. Here, $Rep(A_i)$ represents the representative value of the fuzzy set A_i [41]. The antecedents A_i of those L rules with the smallest distance d^i are chosen. To simplify the computational process, the value of L can be set to the minimum, namely two [42], unless specified otherwise.

The next step of intermediate ANFIS generation is to identify the best reconstruction weights for the closest rules that were selected in the previous step. This is an optimisation problem, which can be solved by running the following:

$$w^{(k)} = \min_{w^{(k)}} ||c^{(k)} - \sum_{i \in \mathcal{L}} Rep(A_i) w_i^{(k)}||^2, \ s.t. \ \sum_{i \in \mathcal{L}} w_i^{(k)} = 1 \tag{6}$$

where $w_i^{(k)}$ implies the relative weighting of R_i. Such an optimisation problem is subject to the constraint that the sum of all weights is equal to one. It can be shown that the solution to this problem is as follows:

$$w^{(k)} = \frac{G^{-1}\mathbf{1}}{\mathbf{1}^T G^{-1}\mathbf{1}} \tag{7}$$

where $\mathbf{1}$ indicates a column vector of ones, $G = (c^{(k)}\mathbf{1}^T - Y)^T(c^{(k)}\mathbf{1}^T - Y)$ denotes the well-defined Gram matrix, and the chosen rule antecedents are shown in the columns of Y.

In order to emphasise on the information contained within the k^{th} cluster, the corresponding weight $w^{(k)}$ is employed to modify both the antecedent and consequent parts concerned. This approach follows the general methodology adopted by typical FRI methods (e.g., [41, 43]), and can be expressed as follows:

$$R_k : if \ y \ is \ A_k, \ then \ z_k = p_k y + r_k \tag{8}$$

where $k = 1, 2, \cdots, C$, with the parameters that describe the intermediate interpolated ANFIS being computed by

$$A_k = \sum_{i \in \mathcal{L}} w_i^{(k)} A_i, \ p_k = \sum_{i \in \mathcal{L}} w_i^{(k)} p_i, \ r_k = \sum_{i \in \mathcal{L}} w_i^{(k)} r_i \tag{9}$$

2.2.4 ANFIS fine-tuning

The final stage of the ANFIS generation process involves fine-tuning the intermediate ANFIS that was generated through interpolation. At this stage, the interpolated ANFIS is used as the initial input for the fine-tuning process, which ultimately produces the final ANFIS on the target domain. The process is implemented by utilising the standard ANFIS training technique [17]. This approach has an important advantage over directly learning an ANFIS model from data, helping overcome the challenge of obtaining an working ANFIS with limited training data. This is because the interpolated intermediate ANFIS is exploited to serve as the initial setup for the expected network.

3 Implementational specifications

This section specifies the implementation of the proposed approach, for each of the feature selection, training, and testing phases.

3.1 Feature selection phase

To enable informative feature selection, supervised learning strategy is adopted here. A given HR image dataset $I = \{1, 2, 3, ..., n\}$ is first resized to a consistent standard size of say, $m \times m$ resolution. Then, scale down those HR images with a scale factor s, using bicubic interpolation to generate their corresponding LR images. After this, resize the LR image back to the size of $m \times m$, which will make the LR image in its enlarged version containing missing detailed information due to low resolution. As such, two pairs of image datasets result, one for LR images and the other for HR images. Following the common practice in ISR, in preparation for subsequent processing, the resulting LR and HR images are transformed from the RGB space to $YCbCr$. In the resultant $YCbCr$ space, each image now involves three components of Y, Cb and Cr.

Once the data are prepared as above, BRISK and KAZE features are extracted from the Y components of each image. Based on these features two different implementations can be made depending on which optimisation procedure is preferred to support the FRFS process. Nevertheless, independent of whether PSO-FRFS or GA-FRFS is utilised, there are a number of parameters that must be initialised with respect to the ISR application. The generic ones are specified as follows:

- T-Norm: *Łukasiewicz* $(max(x + y - 1, 0))$;
- Implicator: *Łukasiewicz* $(min(1 - x + y, 1))$;
- Dependency measure: *Weak Gamma*.

Further detailed parameters required in order to run the PSO-FRFS and GA-FRFS algorithms are specified below. These are set by following common practice in the literature, that is, through empirical studies, while considering the practical applications of the algorithms.

> Parameters for PSO-FRFS: These include the individual optimal solution $c_1 = 1$, and the global optimal solution that is shared amongst all particles $c_2 = 2$. The maximum number of generations to evaluate is set such that $Maxgeneration = 50$, and the number of particles in the swarm is set to 100. The report frequency is set to $Maxgeneration$ which means that the results will be generated after every 50 generations.
>
> Parameters for GA-FRFS: In this algorithm, the population size which indicates the number of attribute sets is set to 20 and the maximum generation is also set to 20. The crossover probability C is set to 0.6, whereas the mutation probability M is set to 0.033. The report frequency is set to 20.

3.2 Training phase

The important features selected by FRFS are denoted as D_{lr} and D_{hr}, respectively for those extracted from the LR and HR images. They act as input in the training phase. Suppose that the K-means algorithm is employed on the pre-processed dataset 'D_{lr}' to divide it into Q clusters. For instance, consider $Q = 3$, thus D_{lr} is divided into three categories of feature values, say, 'Small' (D^s_{lr}), 'Large' (D^l_{lr}) and 'Medium' (D^m_{lr}) for source domain 1, source domain 2 and target domain, respectively. From these, the source models $ANFIS_1$ and $ANFIS_2$ can then be trained with D^s_{lr} and D^l_{lr} datasets, assuming that sufficient training data is available to enable the use of the conventional ANFIS learning procedure.

Given the source models, the target model $ANFIS_t$ can subsequently be interpolated using these two source ones, guided by the information contained within (D^m_{lr}). Of course, an implicit assumption is herein made such that there is no sufficient data to perform conventional training of $ANFIS_t$ directly, unlike the situations for the learning of $ANFIS_1$ and $ANFIS_2$. The step by step training process has been provided in Alg. 1. Otherwise, the ANFIS interpolation procedure is unnecessary.

3.3 Test phase

In this final phase of the proposed $FRFS\_ANFISI$ approach, a test image (Y^t) is taken as an input to be converted into its corresponding HR image (Y). There are several steps to implement this phase as follows. First, following the common practice of the ISR literature, transform the given target RGB image Y^t into its counterpart in the $YCbCr$ space. Then, its illuminance form, namely, the Y component is extracted from the $YCbCr$ space. Next, transform Y into its corresponding LR image Y^l using a specified scale factor s (with the original serving as the standard for comparison in the test). From this, apply either PSO-FRFS or GA-FRFS to the resulting LR image to yield selected features, using the same specification as performed during the training phase. Finally, load all trained ANFIS models $\{A_i\}$ together with their corresponding cluster centres $\{C_i\}$ that have been returned by the training phase.

Continuing with the illustration of the training phase, suppose that source domain 1 is for small-valued features, the target domain is for medium-valued features, and source domain 2 is for large-valued features. Then, in the test phase, 'Small', 'Medium' and 'Large' feature values of the test image (Y^t) are fed to the source domain 1, target domain, and source domain 3, respectively. Before reconstructing the eventual output image, post-processing is performed. For this purpose, non-local means filter (NLM) is employed to remove certain noise effects whereas the iterative back propagation (IBP) algorithm is utilised to refine the fine details of the image. After that, the final HR image is reconstructed for qualitative and quantitative analyses. Details about the above steps of the test phase are summarised in Alg. 2.

4 Experimental evaluation

The above specification has been realised in the following experimental investigations, in an effort to assess and reflect on the potential of the present approach.

4.1 Datasets

The experimental studies are divided into training and test phases. As with the common practice that existing research involves, the datasets employed for training in this study contain the standard set of images regarding people, animals, flowers, and other natural objects [13]. In addition, the standard ISR benchmark BSD200 [23, 44] images are also used together with natural images to create a variety of training datasets to train the ANFIS models, as done in the most recent work [13]. In total, the training data for the experimental analysis contains 275 images. Amongst them, 200 are from the *BSD200* benchmark dataset, whilst the remaining 75 are natural images. For illustrative purposes, examples of the training images are shown in Fig. 3a.

To evaluate the performance of the proposed approach, standard benchmark datasets for ISR such as **Set5** [21], **Set14** [22], **BSD100** [23] and **Urban100** [24] are employed and visually depicted in Fig. 3b-e. As can be seen, the test images are generally rather different from those used for training.

4.2 Experimental setup

The experiments were performed on a Dell laptop having Intel(R) Core(TM) i5-10300H CPU @ 2.5GHz (8 CPUs) and memory (RAM) of 8 GB, running the Windows 10 operating system. Other tools and software used throughout the experiments include MATLAB version R2020a and WEKA version 3.7.2.

In the experimental investigation, for comparative evaluation, three different models are implemented and run. Particularly, Model 1 (reference model) is trained by employing ANFIS with sufficient target data (D_{lr}^m). To train Model 2 (ablation model), ANFIS without interpolation is employed on sparse data. For Model 3 (proposed model), the ANFIS interpolation is employed on the sparse data. To examine the impact of sparseness in the target domain, training data in the target domain are created by randomly deleting 95% of those from D_{lr}^m.

4.3 Performance evaluation metrics

The popular performance evaluation metrics used for the ISR problems are the peak signal to noise (PSNR) and the structure similarity (SSIM). These are adopted here, details of which are summarised below for completeness. Generally, the larger the values of PSNR and SSIM, the better the performance in terms of the reconstructed HR image. Furthermore, an experimental analysis of computational complexity is also reported later.

4.3.1 Peak signal to noise ratio (PSNR)

PSNR computes the peak signal to noise ratio between a given pair of images, derived from mean square error (MSE) between the ground truth (original HR) image $\mathbf{Y}$ and the reconstructed (or estimated) HR image $\hat{\mathbf{Y}}$, as follows:

$$MSE = \frac{\parallel \mathbf{Y} - \hat{\mathbf{Y}} \parallel_F^2}{MN} \tag{10}$$

where $\parallel . \parallel_F$ denotes the Frobenius norm [45] of a matrix, M and N represent respectively the length and width of an image. The raw values of PSNR can be quite wide-ranging. To limit its values within a range that facilitates performance comparison, the following logarithmic decibel is used instead:

$$PSNR = 10\log_{10}(\frac{V_{max}^2}{MSE}) \tag{11}$$

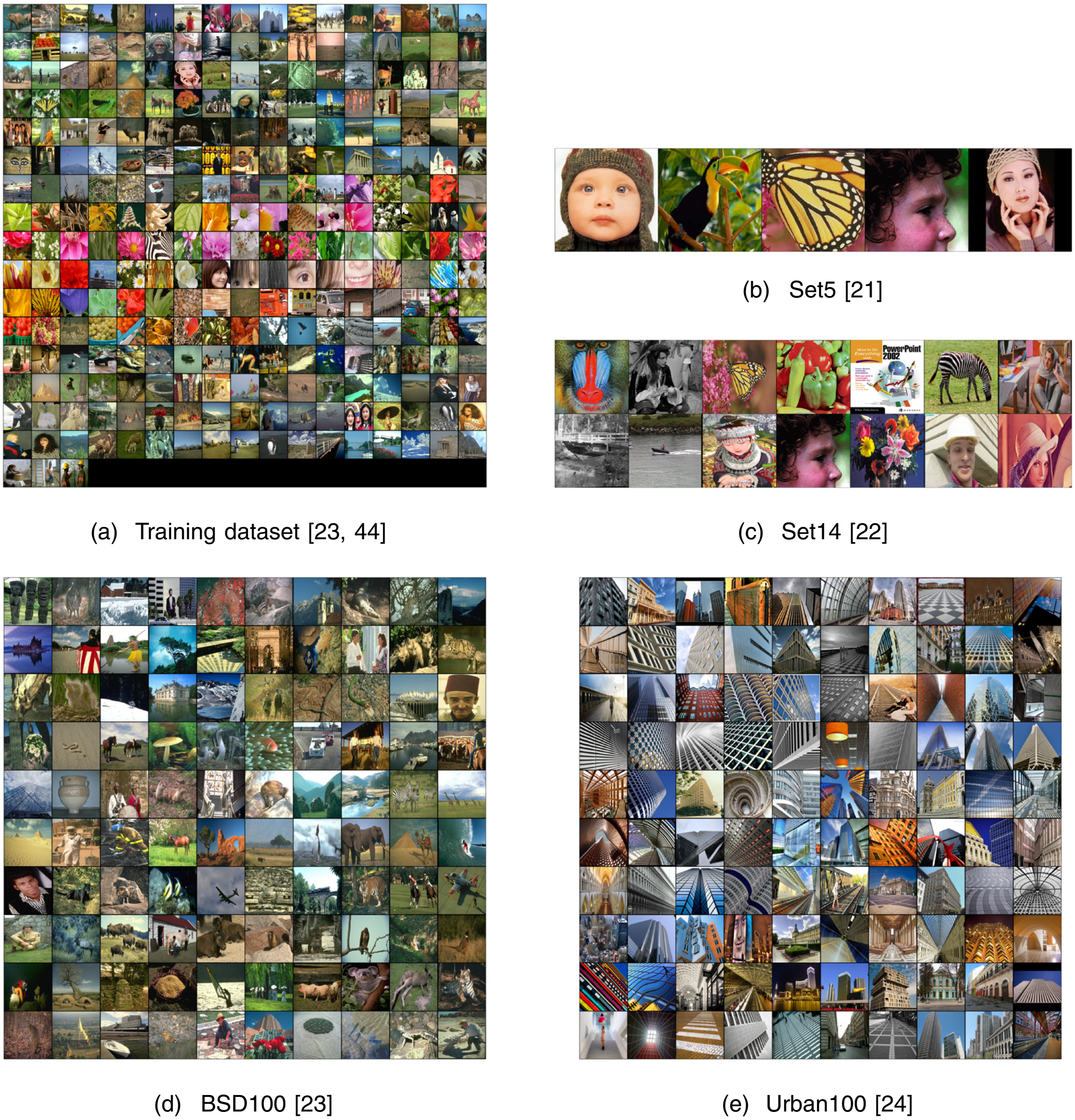
where V_{max} denotes the maximal feature value of any image.

4.3.2 Structure similarity (SSIM)

This metric computes the similarity between the ground truth image $\mathbf{Y}$ and the reconstructed HR image $\hat{\mathbf{Y}}$, as follows:

$$SSIM = \frac{4\mu_{\mathbf{Y}}\mu_{\hat{\mathbf{Y}}}\sigma_{\mathbf{Y}\hat{\mathbf{Y}}}}{(\mu_{\mathbf{Y}}^2 + \mu_{\hat{\mathbf{Y}}}^2)(\sigma_{\mathbf{Y}}^2 + \sigma_{\hat{\mathbf{Y}}}^2)} \tag{12}$$

where $\mu_{\mathbf{Y}}$ and $\mu_{\hat{\mathbf{Y}}}$ are the mean values; $\sigma_{\mathbf{Y}}$ and $\sigma_{\hat{\mathbf{Y}}}$ are the corresponding standard deviations of the pixel values within images $\mathbf{Y}$ and $\hat{\mathbf{Y}}$, respectively; and $\sigma_{\mathbf{Y}\hat{\mathbf{Y}}}$ is the covariance of $\mathbf{Y}$ and $\hat{\mathbf{Y}}$. The values of SSIM range between -1 and 1, with a value of 1 indicating perfect similarity and a value of -1 indicating completely dissimilarity.



(a) Training dataset [23, 44]

(b) Set5 [21]

(c) Set14 [22]

(d) BSD100 [23]

(e) Urban100 [24]

Fig. 3 Visualisation of benchmark super-resolution datasets for training and test phases

4.4 Experimental results

The experimental investigations are focussed on the evaluation of model performance, both quantitatively and qualitatively, supported with specific examinations of learning epoch variation and computational complexity.

4.4.1 Quantitative evaluation

Table 1 shows the PSNR and SSIM measures. Note that there are 16 sub-experiments in total, four per dataset. The naming schemes used are basically self-explanatory. For example, regarding experiment number 6, the name ‘BRISK_GA_401’ indicates that features are extracted by BRISK and selected by *GA-FRFS* algorithm, with the last 3 digits signifying that 401 (out of 1600) most important features are selected.

Overall, it can be observed from the average PSNR and SSIM results (in Table 1) that the outcomes achieved by (proposed) Model 3 (which is associated with sparse data but supported with ANFIS interpolation) are either the same or almost equivalent to those obtained by Model 1 (which is the reference model trained with full data). Whilst it is expected that good performance can be attained using ANFIS interpolation. It is a positive surprise and extremely promising outcome that Model 3 can meet the performance of Model 1.

Particularly, *PSO-FRFS* offers more promising results than *GA-FRFS*. Indeed, using features extracted by BRISK,

Table 1 Average PSNR and SSIM results on Set5, Set14, BSD100 and Urban100 datasets

Exp. #	Experiment Name		Bicubic		Model 1		Model 2		Model 3	
		Dataset	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Sparse Data in Target Domain					✗	✗	✓	✓	✓	✓
ANFIS Interpolation					✗	✗	✗	✗	✓	✓
1	BRISK_PSO_62	Set5	33.66	0.9299	33.74	0.9301	32.84	0.8970	33.70	0.9291
2	BRISK_GA_401				28.94	0.8737	28.44	0.8293	28.90	0.8718
3	KAZE_PSO_41				35.15	0.9519	33.22	0.9081	35.12	0.9516
4	KAZE_GA_623				35.15	0.9519	33.91	0.9186	35.10	0.9505
5	BRISK_PSO_62	Set14	30.24	0.8688	29.66	0.8797	29.17	0.8497	29.64	0.8793
6	BRISK_GA_401				29.11	0.8798	28.57	0.8402	29.11	0.8793
7	KAZE_PSO_41				30.82	0.9063	29.93	0.8764	30.81	0.9061
8	KAZE_GA_623				30.82	0.9063	30.28	0.8807	30.79	0.9052
9	BRISK_PSO_62	BSD100	26.88	0.8403	30.25	0.8720	29.87	0.8493	30.24	0.8715
10	BRISK_GA_401				27.52	0.8677	27.21	0.8448	27.50	0.8671
11	KAZE_PSO_41				30.86	0.8864	29.99	0.8499	30.84	0.8860
12	KAZE_GA_623				30.86	0.8864	30.33	0.8583	30.84	0.8853
13	BRISK_PSO_62	Urban100	29.56	0.8431	28.28	0.8764	27.87	0.8497	28.27	0.8759
14	BRISK_GA_401				27.51	0.8671	27.04	0.8312	27.49	0.8664
15	KAZE_PSO_41				28.84	0.8932	28.25	0.8644	28.83	0.8930
16	KAZE_GA_623				28.84	0.8932	28.53	0.8731	28.83	0.8924

Red and blue colours indicate corresponding best and second best performance regarding each respective dataset per model

the PSNR and SSIM results of Model 3 (by employing *PSO-FRFS*) are equivalent to those of Model 1. Using KAZE-extracted features, the (PSNR or SSIM) performances of both *PSO-FRFS* and *GA-FRFS* are quite similar on certain datasets (especially, BSD100 and Urban100), whereas *PSO-FRFS* has a slight edge in some of the other datasets (Set5 and Set14). An additional interesting observation is that, while the KAZE features seem to be more descriptive in depicting the information contained within the images than BRISK ones, *PSO-FRFS* working with BRISK features can also produce the second-best results using Model 3 in a number of experiments (see case numbers 5, 9 and 13) on most datasets (Set14, BSD100 and Urban100). The collection of the quantitative evaluation results shows that the proposed approach is effective, especially when utilising KAZE-extracted features with *PSO-FRFS*.

4.4.2 Qualitative evaluation

As examples, the visual representation of the baboon image of Set14, the 'image-002' image of BSD100 and 'image-096' of Urban100 are presented in Figs. 4, 5 and 6, respectively. Note that the (P/S) figures under each of the ISR-returned HR images stand for the PSNR (P) and SSIM (S) values achieved by the corresponding model concerned.

Looking at Fig. 4 more closely, it can be seen that results other than that of running the combination of BRISK-extracted features and *GA-FRFS*, Model 3 produces visual images of the baboon with little distortion from those produced by Model 1, clearly beating those obtainable by Model 2. This shows the potential of applying ANFIS interpolation that is supported by feature selection, conforming to the quantitative findings discussed above.

In general, Fig. 5 illustrates a similar pattern as Fig. 4. Interestingly, the results from the combination of BRISK with *PSO-FRFS* are very similar to those from that of KAZE FE with *PSO-FRFS*. These outcomes once again, reflect the quantitative performance measures and indicate the significance of *PSO-FRFS*, which appears to be less relying upon the choice of any specific feature extraction tool.

The achievements reflected by Figs. 4 and 5 are further reinforced by Fig. 6. In particular, it can be seen from the visual results that the images returned by Model 2 contain certain missing information and noise. Fortunately, the visual results of Model 3 are almost the same as those returned by the reference model (which are quantitatively confirmed by their corresponding PSNR and SSIM measures), demonstrating the great potential for the present approach.

4.4.3 Computational complexity

Computational complexity is herein investigated empirically. For instance, Fig. 7 shows the time consumed by dealing with individual test images from the Set5 dataset. In particular, Fig. 7a presents the average time (in seconds) spent by *PSO-FRFS* via the use of BRISK or KAZE features and Fig. 7b

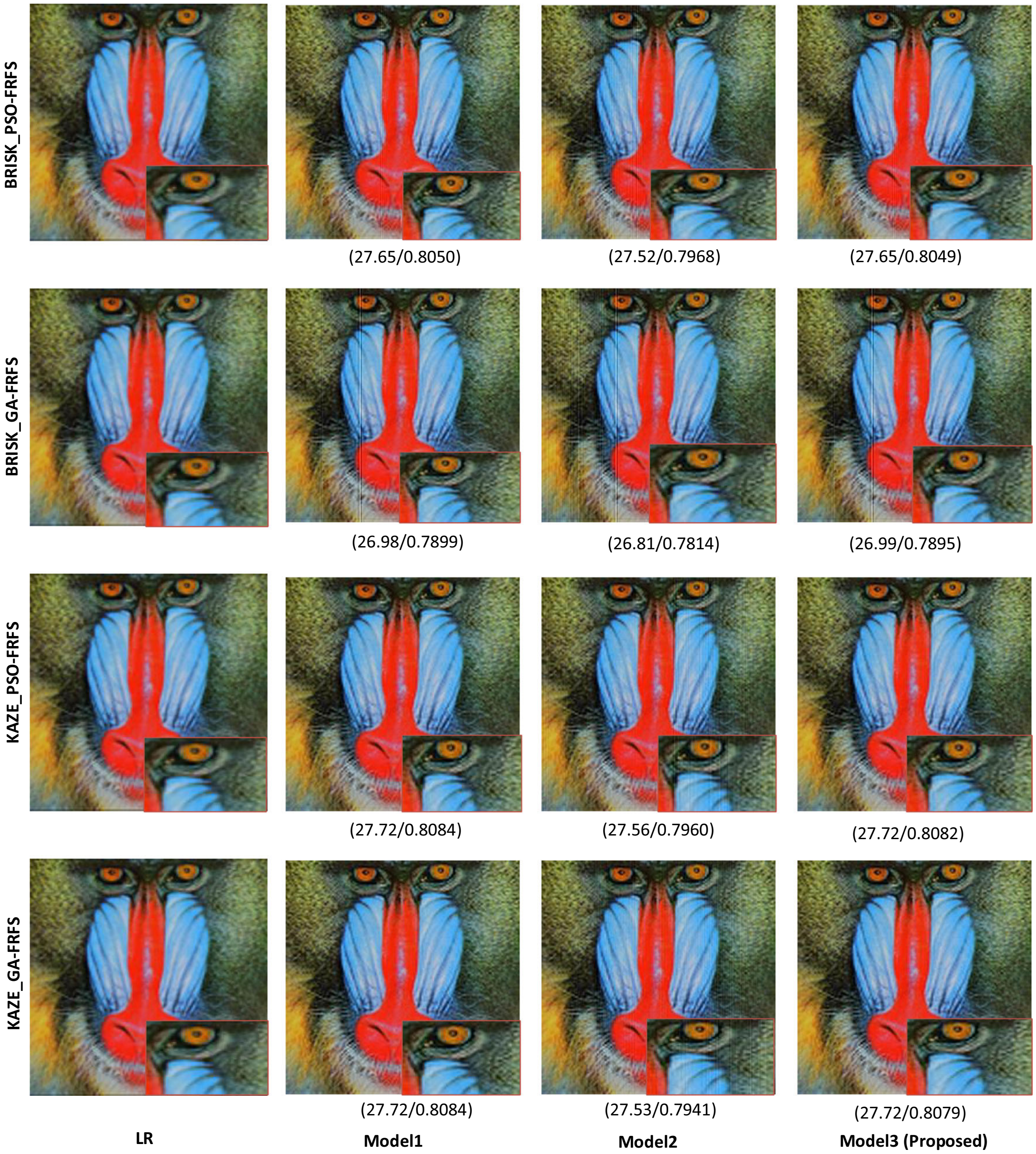


Fig. 4 Super-resolution results for baboon image of Set14 by running *PSO-FRFS* or *GA-FRFS* with BRISK and KAZE features

does that by *GA-FRFS*. It is evident from these figures that using features extracted by KAZE consumes less time than that by BRISK irrespective of how FRFS is implemented. Interestingly, the time consumption is almost the same for both FRFS implementations. Very importantly, either way, the time consumption is practically light-weighted, considering the processing machine used is of rather low capacity, not specifically devised for complicated image analysis tasks.

Examining the results in more detail leads to the following observation that the highest consumed time is under 5 seconds for both child and bird images by the combination of BRISK and *GA-FRFS*. However, a more varying behaviour can be observed in terms of the run time cost for *GA-FRFS*, whilst *PSO-FRFS* behaves more consistently amongst different combinations. This is because *GA-FRFS* has to handle a larger number of features (25% of the total features) than *PSO-FRFS* (which only utilises approximately 5% of the original features). It is not surprising that Model 2 consumes less time than the other two because there is no interpolation employed whilst the data concerned are more restricted and hence, less pattern-matching based inference is carried out. Unfortunately, Model 2 is unable to perform so well as Model

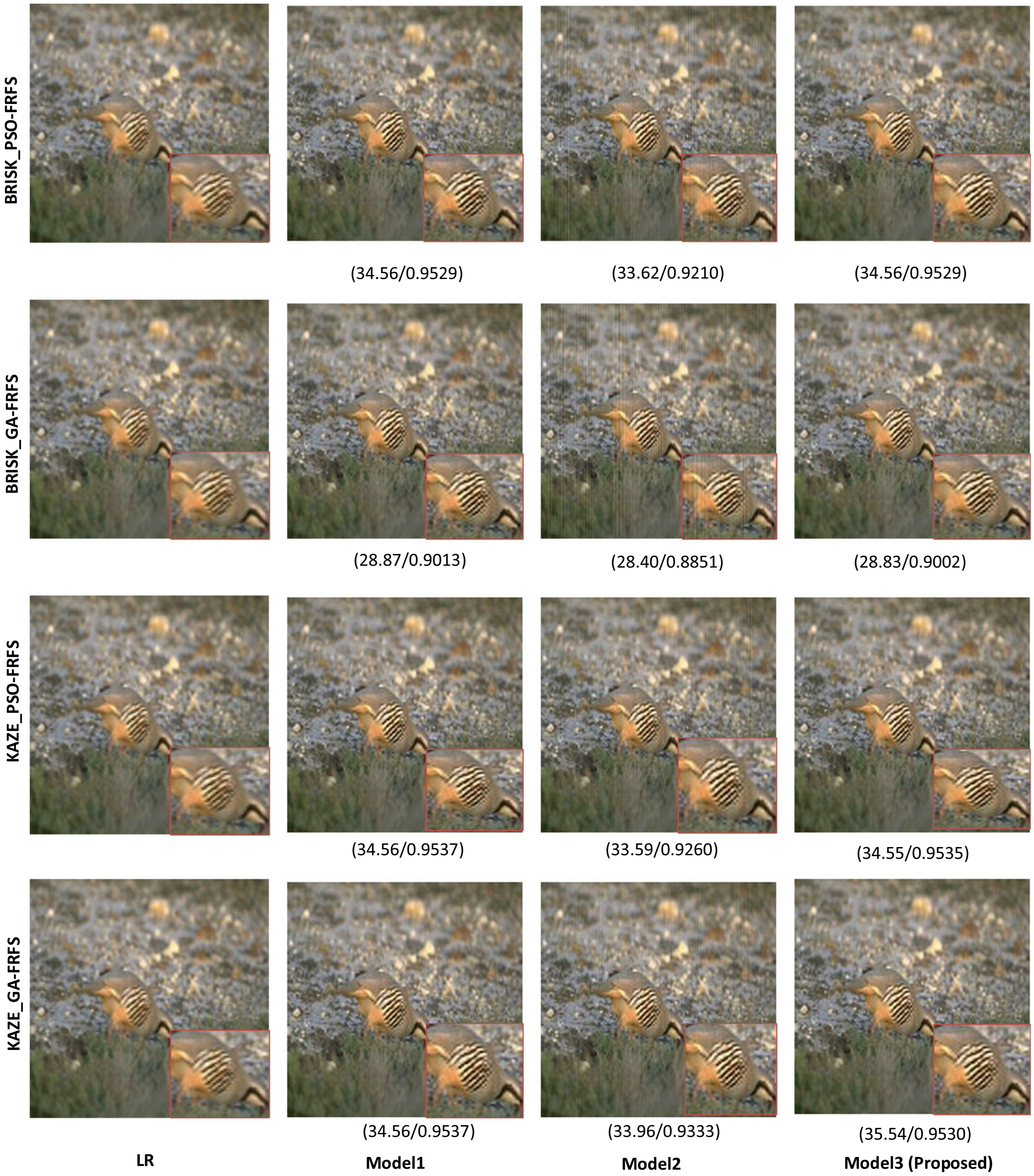


Fig. 5 Super-resolution results for ‘image002’ of BSD100 by running *PSO-FRFS* or *GA-FRFS* with BRISK and KAZE features

3 (which as indicated earlier matches the performance of the reference model), with respect to either of the performance indices employed.

To reveal the time performance more explicitly, Table 2 lists the average computational test time of different models running on the Set5, Set14, BSD100, and Urban100 benchmark datasets. It can be seen that Model 2 outperforms Model 1 and Model 3 (which is again, as expected as stated above), while Model 3 produces the second-best results from this viewpoint. Since Model 3 runs over ANFIS interpolation despite involving sparse data it consumes a bit more time to perform the required task.

Note that the time consumption by (the proposed) Model 3 is generally less than that by (the reference) Model 1. Indeed, Model 3 systematically outperforms the reference model across all experiments except for two specific cases (namely, numbers 8 and 10, where Model 1 implemented with the combination of KAZE and *GA-FRFS* runs a bit faster than Model 3). This may reflect the exceptional situations in which random behaviours of the relevant models have

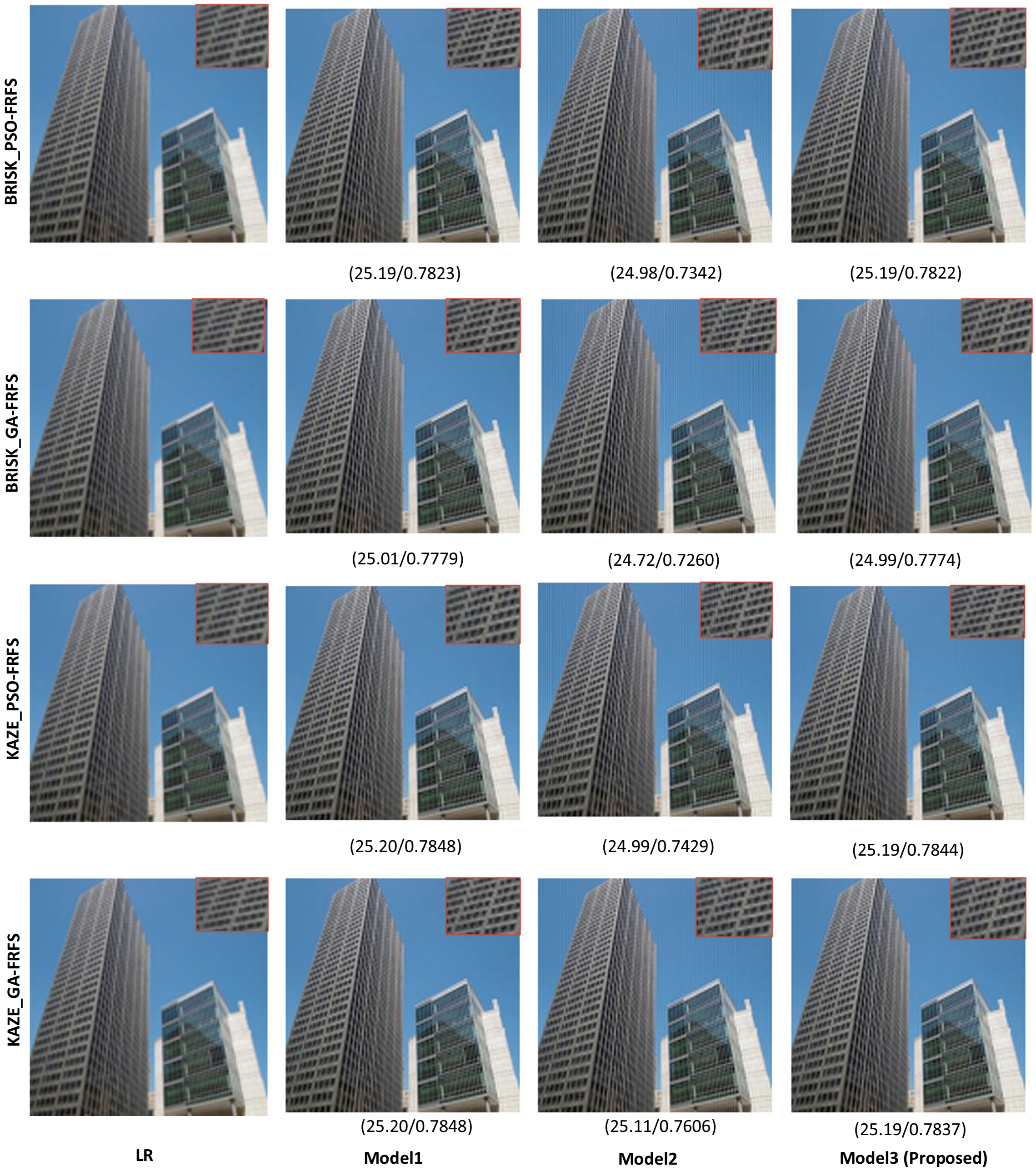
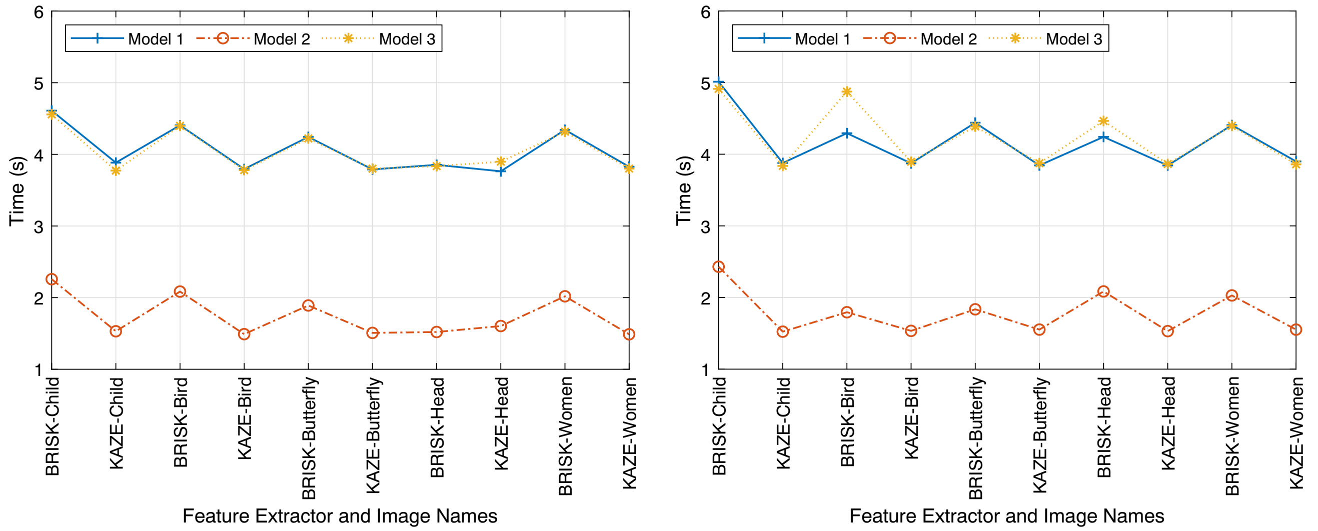


Fig. 6 Super-resolution results for ‘image096’ of Urban100 by running *PSO-FRFS* or *GA-FRFS* with BRISK and KAZE features

gone beyond their statistically expected range. Particularly, the test time for the combination of KAZE and *PSO-FRFS* is the lowest for the Set5 and Set14 datasets, whilst that for *GA-FRFS* using KAZE features performs faster for complex datasets such as BSD100 and Urban100. These results imply that *GA-FRFS* is more suitable for complex datasets whereas *PSO-FRFS* is more computationally efficient for comparatively smaller datasets. While offering flexibility in developing different useful models, these results also provide an empirical methodological guideline for implementational choices of what method to use in practical problem-solving.

5 Conclusion

This paper has presented a novel approach for extending the recent ANFIS interpolation techniques to helping perform image super resolution (ISR) tasks. It is supported by efficient fuzzy rough feature selection (FRFS) algorithms to use just the most important features extracted from a given low resolution image. The paper has introduced two types of technique to reinforce the potential of FRFS, one employing PSO and the other utilising GA to improve the search efficiency of FRFS. Both can work with different feature extraction meth-



(a) Using *PSO-FRFS*.

(b) Using *GA-FRFS*

Fig. 7 Average time cost for ISR on Set5 [21]

ods, as illustrated with the use of two popular mechanisms in the literature (namely, KAZE and BRISK). Together, they form a useful set of techniques to handle challenging ISR problems, with a flexibility of choice on the most appropriate solution when a certain type of image is involved for ISR. The resulting implementations are particularly helpful to address the difficult problems where there is insufficient or missing data for training.

Whilst the results are very promising, the present work also opens up an avenue for further studies. For instance, an initial attempt has been made to provide a methodological guideline for the overall system design, based on the present experimental investigations. However, a theoretical examination of why *PSO-FRFS* is computationally more efficient than *GA-FRFS* for smaller datasets and vice versa for complex datasets requires further study. One particular interest is, in addition to the current computational evaluation, to perform more rigorous analysis of the different combinations of implementation methods, through revealing their underlying mathematical properties. Also, the proposed FRFS-ANFISI

Table 2 Average time results on Set5, Set14, BSD100 and Urban100 by different methods

Exp. #	Experiment Name	Dataset	Model 1 Time (s)	Model 2	Model 3
1	BRISK_PSO_62	Set5	4.291	1.954	4.266
2	BRISK_GA_401		4.475	1.911	4.413
3	KAZE_PSO_41		3.812	1.524	3.810
4	KAZE_GA_623		3.867	1.538	3.866
5	BRISK_PSO_62	Set14	4.317	1.914	4.271
6	BRISK_GA_401		4.237	1.735	4.187
7	KAZE_PSO_41		3.832	1.500	3.798
8	KAZE_GA_623		3.813	1.527	3.823
9	BRISK_PSO_62	BSD100	4.356	1.762	4.190
10	BRISK_GA_401		4.439	1.939	4.486
11	KAZE_PSO_41		3.925	1.532	3.880
12	KAZE_GA_623		3.821	1.528	3.822
13	BRISK_PSO_62	Urban100	4.248	1.851	4.229
14	BRISK_GA_401		4.186	1.636	4.021
15	KAZE_PSO_41		3.956	1.588	3.974
16	KAZE_GA_623		3.830	1.528	3.821

Red and blue colours indicate best and second best performance per dataset approach is herein implemented for images of a small to medium scale. How it may be effectively developed to suit large-scale application problems, such as Martian image super resolution [12] remains active research. Another point to note is that recently, Context-Layered Recurrent Pi-Sigma Neural Network (CLRPSNN) [46] and Memory Recurrent Elman Neural Network (MRENN) [47] have been proposed to perform identification of non-linear systems. These networks are powerful; e.g., facilitated with a context layer, the joint effort of back propagation and Lyapunov-stability method makes CLRSPNN stable and faster for tuning, while MRENN contains a self-feedback feature. These significant characteristics may offer useful means to provide an alternative system modelling approach to the present use of ANFIS. This forms an interesting piece of future work.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Shamsolmoali P, Celebi ME, Wang R (2020) Deep learning approaches for real-time image super-resolution. Neural Comput Appl 32:14519–14520
2. Zhao X, Zhang Y, Qin Y, Wang Q, Zhang T, Li T (2022) Single mr image super-resolution via channel splitting and serial fusion network. Knowl-Based Syst 246:108669
3. Aakerberg A, Nasrollahi K, Moeslund TB (2022) Real-world super-resolution of face-images from surveillance cameras. IET Image Proc 16(2):442–452
4. Chi Y, Li J, Fan H (2022) Pyramid-attention based multi-scale feature fusion network for multispectral pan-sharpening. Appl Intell 52(5):5353–5365
5. Wang P, Sertel E (2023) Multi-frame super-resolution of remote sensing images using attention-based gan models. Knowl-Based Syst 110387
6. Yang J, Wu C, You T, Wang D, Li Y, Shang C, Shen Q (2023) Hierarchical spatio-spectral fusion for hyperspectral image super resolution via sparse representation and pre-trained deep model. Knowl-Based Syst 260:110170
7. Rajput SS, Bohat VK, Arya K (2019) Grey wolf optimization algorithm for facial image super-resolution. Appl Intell 49:1324–1338
8. Shao J, Cheng Q (2021) E-fcnn for tiny facial expression recognition. Appl Intell 51:549–559
9. Zhang Y, Yu L, Sun B, He J (2022) Eng-face: cross-domain heterogeneous face synthesis with enhanced asymmetric cyclegan. Appl Intell 52(13):15295–15307
10. He H, Liang J, Hou Z, Di L, Xia Y (2023) Multi-pose face reconstruction and gabor-based dictionary learning for face recognition. Appl Intell 53(13):16648–16662
11. Tang J, Wan L, Schooling J, Zhao P, Chen J, Wei S (2022) Automatic number plate recognition (anpr) in smart cities: A systematic review on technological advancements and application cases. Cities 129:103833
12. Shang C, Barnes D (2013) Fuzzy-rough feature selection aided support vector machines for mars image classification. Comput Vis Image Underst 117(3):202–213
13. Ismail M, Shang C, Yang J, Shen Q (2021) Sparse data-based image super-resolution with anfis interpolation. Neural Comput Appl 100:1–13
14. Akgun T, Altunbasak Y, Mersereau RM (2005) Super-resolution reconstruction of hyperspectral images. IEEE Trans Image Process 14(11):1860–1875
15. Yang J, Shang C, Li Y, Li F, Shen L, Shen Q (2022) Constructing anfis with sparse data through group-based rule interpolation: An evolutionary approach. IEEE Trans Fuzzy Syst 30(4):893–907
16. Ahmed J, Memon RA, Waqas M, Mangrio MI, Ali S (2018) Selective sparse coding based coupled dictionary learning algorithm for single image super-resolution," in 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). IEEE, pp 1–5
17. Jang J-S (1993) Anfis: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23(3):665–685
18. Jensen R, Shen Q (2008) New approaches to fuzzy-rough feature selection. IEEE Trans Fuzzy Syst 17(4):824–838
19. Wang X, Yang J, Teng X, Xia W, Jensen R (2007) Feature selection based on rough sets and particle swarm optimization. Pattern Recogn Lett 28(4):459–471
20. Jensen R, Mac Parthaláin N (2015) Towards scalable fuzzy-rough feature selection. Infor Sci 323:1–15
21. Bevilacqua M, Roumy A, Guillemot C, Alberi-Morel ML (2012) Low-complexity single-image super-resolution based on nonnegative neighbor embedding
22. Zeyde R, Elad M, Protter M (2010) On single image scale-up using sparse-representations. In: International conference on curves and surfaces. Springer, pp 711–730
23. 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. IEEE, pp 416–423
24. 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. pp 5197–5206
25. Leutenegger S, Chli M, Siegwart RY (2011) Brisk: Binary robust invariant scalable keypoints. In: 2011 International conference on computer vision. Ieee, pp 2548–2555
26. Joshi K, Patel MI (2020) Recent advances in local feature detector and descriptor: a literature survey. Int J Multimed Inf Retr 9(4):231–247
27. Li J, Allinson NM (2008) A comprehensive review of current local features for computer vision. Neurocomputing 71(10–12):1771–1787
28. Alcantarilla PF, Bartoli A, Davison AJ (2012) Kaze features. In: European conference on computer vision. Springer, pp 214–227

29. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vision 60(2):91–110
30. Hwang K, Yoon G, Song J, Yoon SM (2024) Fusing bi-directional global-local features for single image super-resolution. Eng Appl Artif Intell 127:107336
31. Rosten E, Drummond T (2005) Fusing points and lines for high performance tracking. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol. 2. Ieee, pp 1508–1515
32. Rublee E, Rabaud V, Konolige K, Bradski G (2011) Orb: An efficient alternative to sift or surf," in 2011 International conference on computer vision. Ieee, pp 2564–2571
33. Bay H, Tuytelaars T, Gool LV (2006) Surf: Speeded up robust features. In: European conference on computer vision. Springer, pp. 404–417
34. Dhal P, Azad C (2022) A comprehensive survey on feature selection in the various fields of machine learning. Appl Intel 1–39
35. Ji W, Pang Y, Jia X, Wang Z, Hou F, Song B, Liu M, Wang R (2021) Fuzzy rough sets and fuzzy rough neural networks for feature selection: A review. Wiley Interdiscip Rev: Data Min Knowl Discov 11(3):e1402
36. Gad AG (2022) Particle swarm optimization algorithm and its applications: a systematic review. Arch Comput Methods Eng 29(5):2531–2561
37. Katoch S, Chauhan SS, Kumar V (2021) A review on genetic algorithm: past, present, and future. Multimed Tools Appl 80(5):8091–8126
38. Takagi T, Sugeno M (1985) Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cybern SMC–15(1):116–132
39. Su P, Shang C, Chen T, Shen Q (2016) Exploiting data reliability and fuzzy clustering for journal ranking. IEEE Trans Fuzzy Syst 25(5):1306–1319
40. Boongoen T, Shang C, Iam-On N, Shen Q (2011) Extending data reliability measure to a filter approach for soft subspace clustering. IEEE Trans Syst Man Cybern B Cybern 41(6):1705–1714
41. Huang Z, Shen Q (2006) Fuzzy interpolative reasoning via scale and move transformations. IEEE Trans Fuzzy Syst 14(2):340–359
42. Li F, Shang C, Li Y, Yang J, Shen Q (2019) Interpolation with just two nearest neighboring weighted fuzzy rules. IEEE Trans Fuzzy Syst 28(9):2255–2262
43. Baranyi P, Kóczy LT, Gedeon TD (2004) A generalized concept for fuzzy rule interpolation. IEEE Trans Fuzzy Syst 12(6):820–837
44. Timofte R, De Smet V, Van Gool L (2014) A+: Adjusted anchored neighborhood regression for fast super-resolution. In: Asian conference on computer vision. Springer, pp 111–126
45. Zamarashkin N, Osinsky A (2018) On the existence of a nearly optimal skeleton approximation of a matrix in the frobenius norm. In: Doklady Mathematics, vol. 97-2. Springer, pp. 164–166
46. Kumar R (2022) A lyapunov-stability-based context-layered recurrent pi-sigma neural network for the identification of nonlinear systems. Appl Soft Comput 122:108836
47. Kumar R (2022) Memory recurrent elman neural network-based identification of time-delayed nonlinear dynamical system. IEEE Trans Syst Man Cybern: Syst 53(2):753–762

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