



# A Harmony Search-Based Wrapper-Filter Feature Selection Approach for Microstructural Image Classification

Shib Sankar Sarkar<sup>1</sup> · Khalid Hassan Sheikh<sup>4</sup> · Arpan Mahanty<sup>1</sup> · Kalyani Mali<sup>3</sup> · Aniruddha Ghosh<sup>2</sup> · Ram Sarkar<sup>4</sup>

Received: 29 August 2020 / Accepted: 18 December 2020 / Published online: 21 January 2021  
© The Minerals, Metals & Materials Society 2021

## Abstract

Besides chemical composition, microstructure plays a key role to control the properties of engineering materials. A strong correlation exists between microstructure and many mechanical and physical properties of a metal. It has the utmost importance to understand the microstructure and distinguish the microstructure accurately for the appropriate selection of engineering materials in product fabrication. Computer vision and machine learning play a major role to extract the feature and predict the most probable class of a 7-class microstructural image with a high degree of accuracy. Features contain information about the image, and the classification function is defined in terms of features. Feature selection plays an important role in the classification problem to improve the classification accuracy and also to reduce the computational time by eliminating redundant or non-influential features. The current research aims at classifying microstructure image datasets by an improved wrapper-filter based feature selection method using texture-based feature descriptor. Before applying the feature selection method, a feature descriptor, called rotational local tetra pattern (RLTrP), is applied to extract the features from the input images. Then, an ensemble of three filter methods is developed by considering the union of the top-n features selected by Chi-square, Fisher score, and Gini impurity-based filter methods. The objective of this ensemble is to combine all possible important features selected by three filter methods which will be used to create an initial population of the wrapper-based meta-heuristic feature selection algorithm called, harmony search (HS). The novelty of this HS method lies in the objective function, which is defined as a function of Pearson correlation coefficient and mutual information to calculate the fitness value. The proposed method not only optimizes features with reduced dimension but also improves the performance of classification accuracy of the 7-class microstructural images. Moreover, the proposed HS model has also been compared with some standard optimization algorithms like whale optimization algorithm, particle swarm optimization, and Grey wolf optimization on the present dataset, and in every case, the HS method ensures better agreement between feature selection and classification accuracy than the other methods.

**Keywords** Wrapper method · Filter method · Ensemble · Microstructure · Feature selection · Harmony search

## Introduction

Understanding the microstructure of metal is the major concern of material science. The variation of microstructure arises due to irregularly shaped crystals, grain size, the orientation of grain, phase distribution, etc., during thermo-mechanical processing of materials. The variation of microstructure strongly influences many mechanical and physical properties of materials like hardness, strength, ductility, tensile strength, elongation, magnetic properties, etc. [1]. Microstructure characterization with machine learning techniques is an important concept which helps to discover the material properties in material engineering. Many researchers attempt to extract the features from microstructural

✉ Shib Sankar Sarkar  
shib\_sankar\_sarkar@yahoo.co.in

<sup>1</sup> Department of Computer Science and Engineering,  
Government College of Engineering & Textile Technology  
Berhampore, Berhampore 742101, India

<sup>2</sup> Department of Mechanical Engineering, Government  
College of Engineering & Textile Technology Berhampore,  
Berhampore 742101, India

<sup>3</sup> Department of Computer Science and Engineering,  
University of Kalyani, Kalyani 741235, India

<sup>4</sup> Department of Computer Science and Engineering, Jadavpur  
University, Kolkata 700032, India

images using numerous feature descriptors to solve different types of microstructural classification problems.

The microstructural images of metal can be obtained using light optical microscopy (LOM) or scanning electron microscopy (SEM) after proper processing through a method known as etching using a suitable chemical reagent. LOM is a very common available quantification technique for the steel micrograph, and the mixture of color appearances can be used to identify the complex phases. SEM generates images of a specimen by scanning the surface using a focused beam of high-energy electrons. Naturally, SEM does not produce color images, and it is usually represented as a grayscale image. SEM has many advantages over LOM due to its higher resolution, stronger magnification, better depth of field, and ability to the analysis of chemical and structural properties. Due to stronger contrast, SEM is useful to investigate the phase structure and particle. But in LOM many contrasts, color differences and more distinct views are found which make it an aesthetic approach to metallography. Besides higher resolution and magnification in SEM, electron back-scatter diffraction (EBSD) is a powerful method to analyze grain morphology, texture evolution during deformation and phase transformations during heating and cooling. EBSD is a robust technique in materials characterization. However, due to some limitations in the EBSD method [2] microscope images are preferable for microstructure classification. Individually, the EBSD method is not self-sufficient for effective microstructure classification. Britz et al. [3] in their research work have shown that through the correlation of EBSD and LOM, a characterization of the different microstructures is possible.

The microstructure images generated from different sources have variation in contrast, brightness, and color or gray-level intensities. Image texture can be defined as a visual pattern of repeated pixels that has some amount of variability in element appearance and relative position of adjacent elements. The analysis based on image texture has been successfully applied in different fields of engineering and medical applications [4–8]. Various successful attempts have been made by many researchers in microstructural analysis using textural image analysis.

For cast iron, gray-level co-occurrence matrix (GLCM) and local binary pattern (LBP) feature descriptors are used by Gajalakshmi et al. [9] to classify the 3-class microstructural images using the support vector machine (SVM). For steels, Weibel et al. [10] have analyzed the microstructures using Haralick image texture features, calculated from the stepwise rotation of images to make them rotation variant. They have applied this method to distinguish microstructural images as pearlite, martensitic, and bainite using four texture features obtained from GLCM which are contrast, correlation, homogeneity, and energy. Kitahara et al. [11] have applied a transfer learning pipeline and unsupervised

learning method to successfully classify two datasets of microstructural images of surface defects in steel. Gola et al. [12] have used rapid miner as a data mining platform with the support vector machine (SVM) to classify the images into three classes namely martensite, pearlite, and bainite. They have proved that the combination of the data mining process and microstructural parameters could be used in the classification of two-phase steels. Successful usage of texture-based features in the past has motivated us to explore the same by classifying the microstructural images.

## Related Work

The major challenges to implement the feature engineering-based machine learning algorithms are to handle the high dimensionality of the feature set and to enhance the efficiency of the learning models. The feature selection (FS) method proves to be an effective way to overcome this challenge. The main objective of FS is to reduce the dimension of the features by eliminating redundant or irrelevant features and to improve the prediction performance and achieve cost-effective performance. FS can be treated as a critical preprocessing step before the classification task to remove those features that carry no significant role to increase the classification performance. FS is a challenging task with a larger search space. As the number of features increases, the complexity is heightened due to feature interaction.

Depending on the utilized training data, learning methods, evaluation criterion, search strategies, and type of output, FS methods can be classified into various categories [13]. FS methods can be divided into supervised, unsupervised, and semi-supervised models based on utilized training data. Based on learning methods, these are divided into filter [14], wrapper [15], and embedded methods [16]. Two main factors that influence the FS process are the search techniques and evaluation criteria. Search techniques examine the entire search space to find the optimal feature subsets and evaluation criteria evaluate the importance of feature to accompany the search process.

DeCost et al. [17] have applied visual features as a generic microstructural signature to classify 7-class microstructure images (brass/bronze, ductile cast iron, gray cast iron, hypoeutectoid steel, malleable cast iron, superalloy, and annealing twin) using a SVM model as a classifier. The proposed algorithm has shown to be effective with a fivefold cross-validation accuracy of 83%. Chowdhury et al. [18] have applied several feature extraction methods based on texture and shape statistics and also use the pretrained convolutional neural network (CNN) to compute feature vector on the microstructural dataset for 2-class classification problem. These feature vectors are processed with six different techniques that include principle component analysis (PCA),

ANOVA F-statistic (ANOVA), Fisher score, Chi-square, and Gini index to reduce the dimension of the feature vector.

Gola et al. [19] have proposed to use three different types of parameters, namely morphological, textural, and sub-structural to determine the features for reliable classification of microstructural images. In total, they have computed 75 parameters and they have chosen genetic algorithm (GA) as a FS method to reduce the dimension of the parameters. The potential limitation of their research work is that they have attempted to reduce the dimension of parameters maintaining the classification accuracy, so no improvement of accuracy after parameter selection has been observed.

A variety of search techniques have been applied by the researchers to find the best feature subset from a high-dimensional feature set [13]. These searching methods include complete search, greedy search, heuristic search, and random search. However, the problem lies in the fact that they consume too much time to find the best possible solution and many times getting trapped in local optima. A trade-off is required between heuristic and meta-heuristic methods to find an efficient global search technique to improve the quality of FS problems. Most of the meta-heuristic algorithms have taken inspiration from different sources such as GA which is biologically inspired, particle swarm optimization which is swarm-based, and harmony search (HS) which is physics-based. Meta-heuristic works in an iterative manner along with the subordinate heuristics. It is a guided random search technique. It is not starting from a single point rather it starts from multiple points and performs random search and tries to explore the entire search space. It includes the method to avoid getting trapped into local optima and uses search experience intelligently to guide further solutions.

HS is a physics-based meta-heuristic method that provides a problem-independent optimal solution by iteratively examining the entire search space. The HS algorithm is a population-based meta-heuristic algorithm, and it is inspired by the improvisation process of music players. HS algorithm simulates the concept of searching for better harmony in the musical processed. The motivation behind the selection of the HS algorithm in the present work is that it has shown impressive performance in FS in different domains [20].

Das et al. [21] have proposed an algorithm based on HS to reduce the feature set in handwritten Bangla word recognition problem. The HS-based wrapper FS method is capable to reduce the dimension of feature vector from 65 to 48 features, and the proposed algorithm has shown to be effective to enhance the classification accuracy when compared to some standard evolutionary algorithms like GA and PSO.

Gholami et al. [22] have proposed a FS method-based HS algorithm with some modification to the improvisation step. The algorithm is evaluated using the datasets of 18 benchmark problems from the UC Irvine Machine Learning Repository and demonstrates its ability to reduce the

dataset compare with other FS methods. Zainuddin et al. [23] have applied the HS algorithm for FS for epileptic seizure detection and prediction using UCI benchmark datasets, and they have used wavelet neural networks as a classifier. Moayedikia et al. [24] have proposed a technique named SYMON which uses HS algorithm and symmetrical uncertainty to compute the optimal feature subset for high-dimensional imbalanced datasets on especially the microarray datasets. Nekkaal and Boughaci [25] have proposed a hybrid model comprising of HS algorithm and stochastic local search for FS using UCI benchmark datasets

Dash [26] uses a two-stage FS method for high-dimensional microarray dataset. This method includes a technique that combines HS algorithm and Pareto optimization approach. Adaptive HS algorithm-based gene selection technique is applied in the first stage to generate the top 100 features and in the second stage multi-criterion, and Pareto optimal solution is applied to find out the optimal feature subset. The statistical analysis report shows the superiority of this model as compared to other considered approaches.

Ramos et al. [27] have used HS algorithm to select a discriminant subset of features with the optimum-path forest classifier for identifying non-technical losses in power distribution systems. Huang et al. [28] have used a self-adaptive HS (SAHS) algorithm with SVM classifier to locate the optimal feature subsets for music genre classification. HS algorithm as FS has also been applied for accurate classification of speech emotion using the dataset from the Berlin German emotion database (EMODB) and Chinese Elderly emotion database (EESDB) [29]. Saha et al. [30] have used an HS algorithm based on cosine similarity and minimal-redundancy maximal relevance (mRMR) for FS in facial emotion recognition problems with improvements in overall classification accuracy.

## Motivation and Contributions

In data mining and machine learning, the features extracted by some means often come up with a huge number of features and among them some are extraneous and correlated features. In this context, the main objective of FS is to find out a set of uncorrelated features with reduced dimensionality to be used for the classification process under consideration. However, discovering an optimal subset of features that results in maximization of classification accuracy is still a less explored area in the domain of microstructural image classification. This motivates us to propose a FS method applied to texture-based features extracted from microstructural images.

Regarding the feature vector considered here for classification, it is to be noted that there are a few reasons why the proposed method performs better. There are many

texture-based feature descriptors found in the literature; among them local binary pattern (LBP) is one of the most popular ones which has been used in varied domains over the years. However, in the basic LBP, to capture local information around a pixel in an image, it only considers the 8 neighboring pixels. On the other hand, local tetra pattern (LTrP), an extension of the basic LBP, does this by considering two layers of the neighboring pixels. That is why LTrP method and its variants can extract better texture information of the input image which, in turn, helps in classifying the input images belonging to different classes. However, for achieving better classification using this texture-based feature vector through a FS method, we need the most unique and relevant features from the input feature vector. The FS method does that job by finding the most relevant features out of the high-dimensional input feature vector. Our present work attempts to address these issues. In a nutshell, the main contributions of our work include:

- To the best of our knowledge, this is the first study that explores texture-based rotation variant LTrP (RLTrP) in the microstructural image classification.
- An unsupervised FS technique based on the HS algorithm is introduced to find an optimal subset of informative features for improving microstructural image classification.
- The initial population of the HS algorithm is created by forming an ensemble of three filter methods, namely Chi-square, Fisher score, and Gini impurity.
- Instead of using a classifier, PCC and MI values are used to estimate the fitness value which boosts up our proposed FS method.
- Experimental results demonstrate that the proposed wrapper-filter method outperforms other standard meta-heuristic-based FS methods
- Impressive results are obtained in classifying 7-class microstructural images with significantly less number of features.

The remaining section of this paper is presented as follows: “Preliminaries” section presents a brief overview of the concepts of the method used in the work. The proposed model is described in “Proposed Model” section. A brief description of the HS algorithm and the characteristics of the proposed FS method with objective function for improvement are also presented in “Proposed Model” section. The results and discussion are presented in “Results and Discussion” section. The research study of the literature is concluded in “Conclusion” with the outline of future work.

## Preliminaries

We describe five different feature descriptors in “Feature descriptors” section, the three filter methods used in our work in “Dimensionality Reduction” section. The PCC and MI used to define the objective function in the HS algorithm are described in “Objective Functions” section.

## Feature Descriptors

Local variation in intensity is characterized as texture. Image textures are complex visual patterns characterized by shape, size, color, and intensity in an image or selected region of an image. The textures are determined based on the coarseness, fineness, regularity, smoothness, etc. Multiple feature descriptors use the image texture to compute the features of an image.

### Gray-Level Co-occurrence Matrices (GLCM)

Haralick [31] proposed a set of 2D square matrices known as GLCM, a matrix that counts the directional differences between intensities of the neighboring pixels in the image. It is also termed as co-occurrence distribution. From the GLCM, the texture measures are computed and these texture measures imply the variation of intensity at the pixel of interest. Haralick features are computed employing GLCM, which are second-order statistical measures. The co-occurrence matrix with dimension  $X_K \times X_K$  as follows:

$$\begin{bmatrix} s(1, 1) & s(1, 2) & s(1, 3) & s(1, X_K) \\ s(2, 1) & s(2, 2) & s(2, 3) & s(2, X_K) \\ \dots & \dots & \dots & \dots \\ s(X_K, 1) & \dots & \dots & s(X_K, X_K) \end{bmatrix}$$

$X_K$  implies the number of gray levels in the image. GLCM can be computed at any angle and at any offset. Each element [i, j] of this co-occurrence matrix is determined by the sum of the number of times with value i is adjacent to a pixel with value j.

### Scale Invariant Feature Transform (SIFT)

Lowe [32] created an excellent algorithm known as SIFT to extract invariant features from images. This algorithm is attractive for finding interest points in an image. The prime characteristics of this algorithm are that the features are invariant to image scale and rotation, and it contributes a potent matching across a range of affine distortion, the addition of noise, and variation in illumination. The procedure to generate the set of image features is a multi-step pipeline process that involves scale-space extrema detection,

keypoint localization, orientation assignment, and keypoint descriptor.

In the first stage, a scale space is created using an approximation based on difference of Gaussian (DoG) technique. DoG images are formed using the Gaussian blur operator at each octave of the scale space. The interest points that are invariant to scale and orientation are searched over all scales and image locations. In the keypoint localization step, each pixel in the DoG image is compared to its neighboring pixels. Keypoints are selected based on the pixel that is local maximums or minimums. All the low-contrast points are excluded from the final set of keypoints. In the next step, one or more orientations are assigned based upon gradient histogram technique to each keypoint to make it rotation invariant. Peaks in the histogram are considered as the dominant orientations. In the final step, the computation of the feature descriptor for each keypoint is performed. The resulting feature vector consists of 128 orientation histograms.

### Oriented FAST and Rotated BRIEF (ORB)

ORB [33] is built as a combination of FAST and BRIEF techniques as a very fast binary descriptor for feature extraction. ORB produces an optimized mix-and-match result of FAST and rotated BRIEF. Both FAST and BRIEF are well-known feature descriptors because of their good performance and low cost. Multi-scale feature-based scale pyramid of the image is used in the ORB algorithm. At each scale of the image, pyramid FAST and rotated BRIEF are applied. Once ORB has created a pyramid, FAST feature detection algorithm uses the corner detection mechanism to detect the keypoints. For detecting intensity change, the corner orientation is followed by intensity centroids using Rosin's method [34]. The FAST, Harris corner measures, and Rosin method are used at each scale of an image pyramid.

To search the top-n points with the strongest FAST responses, Harris corner measures are applied on the keypoints at each level. The center of gravity of an image is computed with moments to improve the rotation invariance property of this method. For correspondence search, ORB uses multi-probe locally sensitive hashing (MP-LSH), which searches for matches in neighboring buckets when a match fails.

### Local Binary Patterns (LBP)

Ojala et al. in their paper [35] have first introduced a texture-based feature operator called LBP. It is quite effective in encapsulating texture information without any costly computation.

The original description can be found in [35]. Let there are  $N_s$  gray pixels surrounding a center pixel at  $(X_{cen}, Y_{cen})$

having a radius of  $R_{LBP}$  unit. Therefore,  $l$ th pixel position can be easily calculated from Eq. (1).

$$(X_l, Y_l) = \left( X_{cen} + R_{LBP} \cos \left( \frac{2\pi l}{N_s} \right), Y_{cen} - R_{LBP} \sin \left( \frac{2\pi l}{N_s} \right) \right) \quad [l \in \{0, 1, \dots, N_s - 1\}] \quad (1)$$

For center pixel and all  $N_s$  surrounding pixels, the grayscale intensity can be represented as  $I_c, I_0, I_1, \dots, I_{N_s-1}$ , respectively. According to [36], we can define a binary pattern for every possible center pixel with the help of its all  $N_s$  neighboring pixels' grayscale intensity. We can define the binary pattern as follows:

$$Bin_{Feature} = \left\{ F_1(I_0, I_c), F_1(I_1, I_c), \dots, F_1(I_{N_s-1}, I_c) \right\} \quad (2)$$

where

$$F_1(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (3)$$

We can obtain the decimal equivalent of the binary pattern in the following way

$$LBP_{function(N_s, R_{LBP})}(I_c, I_1, \dots, I_{N_s-1}) = \sum_{k=0}^{N_s-1} F_1(I_k - I_c) 2^k \quad (4)$$

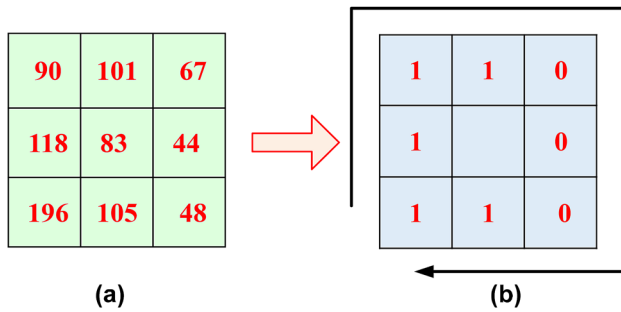
For  $R_{LBP} = 1$ ,  $N_s$  becomes 8. Therefore, for  $R_{LBP} = 1$  and  $N_s = 8$ ,  $Bin_{Feature}$  becomes an 8-bit binary pattern, or its decimal equivalent ranges from 0 to 255. Generally, the histogram containing the frequency of occurrence of all these decimal values is used as a feature.

### Implementation of LBP

For implementation purpose, we take a value of  $R_{LBP} = 1$  and  $N_s = 8$ , so we get an image segment of  $3 \times 3$  to compute a 8-bit binary pattern for each pixel considering all its 8 neighbors. Consider the following example given in Fig. 1, where value of  $I_c = 83$  i.e., the grayscale intensity; all 8 neighbors' grayscale values are also shown in Fig. 1. For each neighboring pixel, a value of 0 or 1 is assigned depending on Eq. 3. Ultimately, for each pixel considering its all 8 neighbors, we end up with an 8-bit binary code, which in turn is converted into its decimal value. This  $3 \times 3$  grayscale image segment gives us the binary pattern 11100011, and this binary pattern gives a decimal value of 227.

### Rotational Variant LBP

The binary pattern obtained from LBP often contains unnecessary details. Therefore, often rotational variant LBP (RLBP) is used to get rid of the unnecessary information.



**Fig. 1** Illustration of LBP calculation process for a 3 x 3 gray image window, where  $N_s = 8$ , and  $R_{LBP} = 1$ . **a** Image segment with actual grayscale intensity values, **b** image segment with grayscale values being replaced with binary value. The binary pattern is generated according to clockwise notation shown in the diagram. The binary pattern = 11,110,001. Here, 1 represents neighboring pixel intensity is greater than center pixel value, 0 vice versa

In [35], RLBP is obtained by rotating the binary pattern to obtain the least decimal value.

$$RotLBP_{(N_s-1)}(I_c, I_1, \dots, I_{N_s-1}) = \min\{Rotate(Bin_{feature}, i | 0 \leq i \leq N_s - 1)\} \tag{5}$$

For  $R_{LBP} = 1$  and  $N_s = 8$ , total possible unique values that can be obtained are 36.

The binary pattern obtained in Fig. 1 can be left-rotated with continuation to form the minimum possible decimal value.

- $(11,100,011)_2 = 227$   $(00,111,110)_2 = 062$
- $(11,000,111)_2 = 199$   $(01,111,100)_2 = 124$
- $(10,001,111)_2 = 143$   $(01,111,100)_2 = 248$
- $(00,011,111)_2 = 031$   $(11,110,001)_2 = 241$
- Min Binary Pattern:  $(00,011,111)_2 = 031$

**Local Tetra Pattern (LTrP)**

There are various reasons for which LBP has been widely used as a texture-based feature. It is computationally less

expensive and it is immune to different lighting effects due to the use of the  $F_1$  for binary pattern generation. But there are few drawbacks, as it uses the first layer of surrounding pixels to generate the binary pattern; hence, it has a huge chance of missing out key features of center pixel locality. It can be overcome by using a rather deeper layer of surrounding pixels to generate features. Murala et al. proposed LTrP as texture-based operators in [36] to overcome the limitations of LBP. They have adopted the idea of different local patterns such as LBP [35], local ternary pattern (LTP) [37], and LDP [38] to define LTrP. The description of the feature is as follows:

Given Im, an image segment, we take the first order derivative along  $0^\circ$  and  $90^\circ$ . We

denote first-order derivative by,  $Im_\alpha^1(I_c) | \alpha = 0^\circ, 90^\circ$ .

$$Im_{(0^\circ)}^1 = I_{(0^\circ)} - I_c \tag{6}$$

$$Im_{90^\circ}^1 = I_{90^\circ} - I_c \tag{7}$$

Here  $I_{0^\circ}$  and  $I_{90^\circ}$  are actually the grayscale intensity

of the pixels aligned with the center pixel at  $0^\circ$  and  $90^\circ$  respectively.

From the value of derivatives, the direction of the center pixel can be considered by the following function:

$$Direction^1(I_c) = \begin{cases} 1, & \text{if } I_{0^\circ}^1 \geq 0, I_{90^\circ}^1 \geq 0 \\ 2, & \text{if } I_{0^\circ}^1 < 0, I_{90^\circ}^1 \geq 0 \\ 3, & \text{if } I_{0^\circ}^1 < 0, I_{90^\circ}^1 < 0 \\ 4, & \text{if } I_{0^\circ}^1 \geq 0, I_{90^\circ}^1 < 0 \end{cases} \tag{8}$$

As there are two possible  $I_{0^\circ}$  pixels for a center pixel, and same for  $I_{90^\circ}$ , for each center pixel, we can have four distinct directions considering all possible combinations. These directions are used for encoding the image segment into a tetra pattern, by defining second-order LTrP, in the following way:

$$LTrP^2(I_c) = \left\{ F_2(Direction^1(I_c), Direction^1(I_0)), F_2(Direction^1(I_c), Direction^1(I_1)), \dots, \dots, F_2(Direction^1(I_c), Direction^1(I_{N_s-1})) \right\} \tag{9}$$

$$F_2(Direction^1(I_c), Direction^1(I_l)) = \begin{cases} 0, & \text{if } Direction^1(I_c) = Direction^1(I_l) \\ Direction^1(I_l), & \text{else} \end{cases} \tag{10}$$

Therefore, we get an 8-bit tetra pattern  $LTrP^2(I_c)$ ; this tetra pattern can in turn generate three 8-bit binary patterns. We can get the three binary patterns in the following way.

$$\text{Let } Direction^1(G_c) = 4,$$

$$BinLTrP_{Direction} = F_3(LTrP^2(I_c)[l|l = 0, 1, \dots, N_s - 1]_{Direction=1,2,3}) \quad (11)$$

$$F_3(x)_{Direction=\mu} = \begin{cases} 1, & \text{if } x = \mu \\ 0, & \text{if } x \neq \mu \end{cases} \quad (12)$$

As each center pixel can have four unique directions,  $4 \times 3 = 12$  binary patterns can be attained from Eq. (11). Murala et al. in [36] also suggested 13 binary patterns based on the magnitude of the local difference operator.

$$M_{I_l} = \sqrt{Im_{0^\circ}^1(I_l)^2 + Im_{90^\circ}^1(Im_l)^2} \quad (13)$$

Therefore, thirteenth binary pattern can be defined as follows:

$$Bin_{13} = \{F_1(M_l - M_{I_c})|l = 0, 1, \dots, 7\} \quad (14)$$

**Rotational Variant Local Tetra Pattern: (RLTrP)**

The method of “Rotational Variant LBP” section can be incorporated into each of the thirteen binary patterns of LTrP to obtain RLTrP.

**Implementation of Proposed Feature Descriptor**

For feature extraction, we have used the grayscale image; therefore, all the input color images are converted into grayscale image before feature extraction algorithm takes place. Main characteristic of LTrP feature extraction procedure is that, unlike LBP, we do not generate the tetra patterns solely based on grayscale intensity; rather, we take the direction of a pixel, with respect to the horizontal and

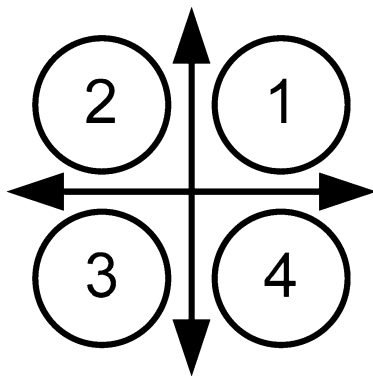


Fig. 2 Visualizing all four directions generated from Eq. 8

vertical neighboring pixels. The direction is given by Eq. 8. Figure 2 describes all four directions possible from Eq. 8. For all other 8 neighboring pixels, we have calculated their corresponding directions with respect to the center pixel. Hence, for generating tetra pattern for each pixel, we take a  $5 \times 5$  window. We have taken one example where center pixel direction is assumed to be 1.

In Fig. 3, red block indicates the center pixel and the blue boxes indicate the neighboring pixel of concern. All the directions of the neighboring pixels can be calculated using Eq. 8. The directions are 4, 3, 1, 3, 1, 2, 3, and 1 from top left to right and bottom left to right. The tetra pattern can be generated using Eqs. 9 and 10 using these directions.

Tetra pattern: 4 3 0 3 0 2 3 0

This tetra pattern is used to generate 3 binary patterns using Eqs. 11 and 12.

Binary pattern 1: 1 0 0 0 0 0 0 0 | Dir=4

Binary pattern 2: 0 1 0 1 0 0 1 0 | Dir=3

Binary pattern 3: 0 0 0 0 0 1 0 0 | Dir=2

This process will repeat for all other three possible directions of the center pixel in total giving 12 binary patterns. Thirteenth pattern is generated using magnitude of difference for any one sequence using Eqs. 13 and 14.

The magnitude difference of center pixel is :  $\sqrt{(8-4)^2 + (9-4)^2} = 6$

The corresponding magnitudes are, respectively,  $\sqrt{(5-7)^2 + (9-7)^2} = 2.8$ ,  $\sqrt{(3-9)^2 + (1-9)^2} = 10$ ,  $\sqrt{(9-1)^2 + (5-1)^2} = 8.9$ ,  $\sqrt{(1-8)^2 + (2-8)^2} = 9.2$ ,  $\sqrt{(8-2)^2 + (9-2)^2} = 9.2$ ,  $\sqrt{(4-3)^2 + (2-3)^2} = 1.4$ ,  $\sqrt{(3-8)^2 + (3-8)^2} = 7$ ,  $\sqrt{(7-3)^2 + (4-3)^2} = 6$ .

Therefore, binary pattern generated by these magnitudes is : 1 1 1 1 0 1 0

Rotational variant is used for all these binary patterns to generate the least decimal values which are used to calculate the histogram for the given image, which are fed to a machine learning model.

In Fig. 4, we have generated 13 LTrP rotational variant images. For each image, we have replaced the actual pixel intensity with the rotational variant decimal value of LTrP code for a particular center pixel.

In Fig. 5, the proposed feature extraction method is explained visually. At the first step, each image is converted into a grayscale image, which is then transformed into the 13 LTrP images. Each image is then used to generate frequency density histogram of the pixel values in the range of 0–255. The rotational variant is used to generate final features.

**Dimensionality Reduction**

The above-mentioned feature descriptors are used to classify the microstructure images. More number of features generally imply better information and better discriminative capability. However, due to the presence of some irrelevant,

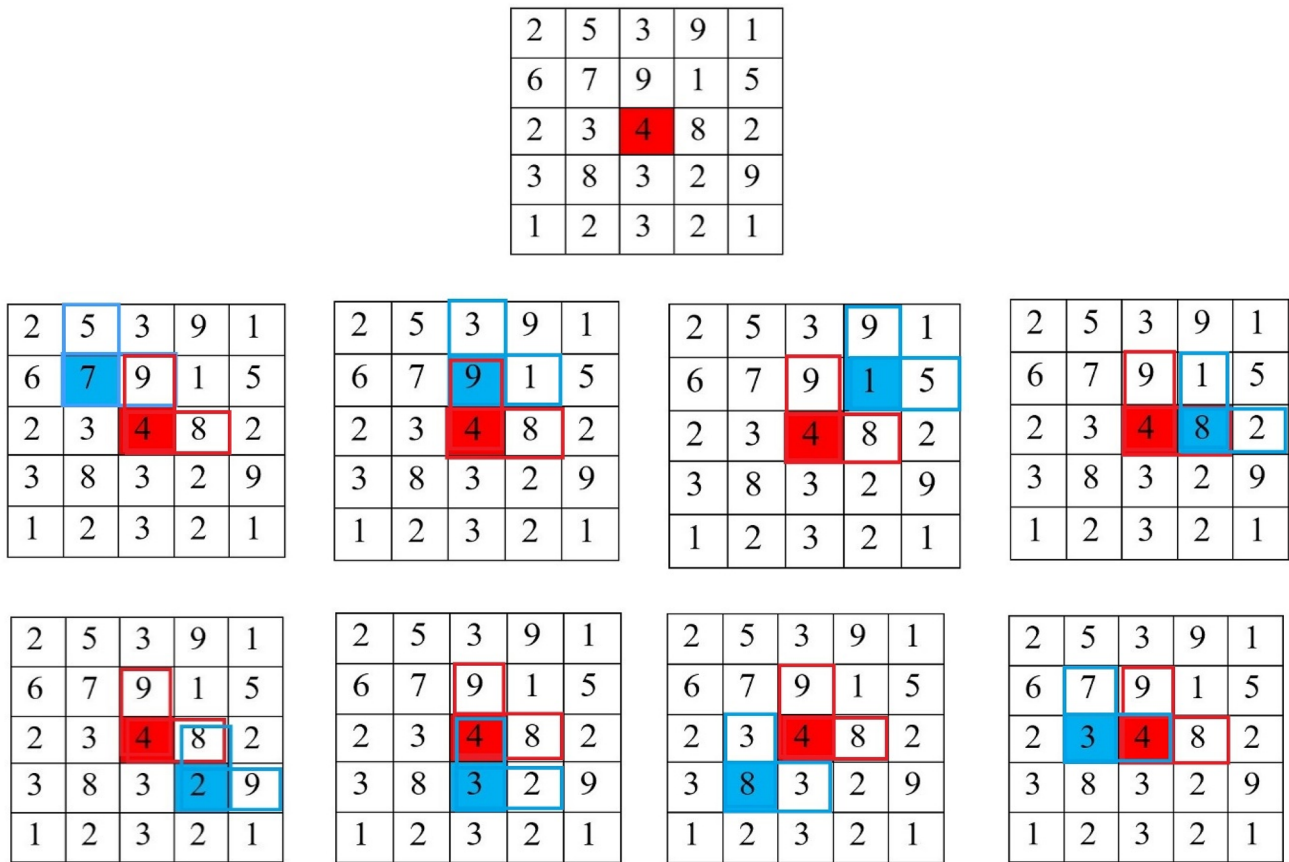


Fig. 3 Illustration of tetra pattern generation. Here, red block is used to indicate the center pixel and blue block is used to indicate neighboring pixel

noisy, and redundant features, the performance of the classifier is usually degraded. Therefore, to maintain or improve the classifier accuracy it becomes necessary to apply FS techniques to avoid the curse of dimensionality. In stage 1, an ensemble of filter methods is developed by considering the union of the top-ranked features of Chi-square, Fisher score, and Gini impurity. A brief description of these filters is given below.

**Chi-Square**

Chi-square ( $\chi^2$ ) [39] is very useful in FS which tests the statistically significant relationship or contingency between the features. The Chi-square statistical test of independence is a method appropriate for testing independence between two categorical variables. The null hypothesis of this statistical test implies that there is no difference between the categorical variables. The computation of  $\chi^2$  value is straightforward and is given by

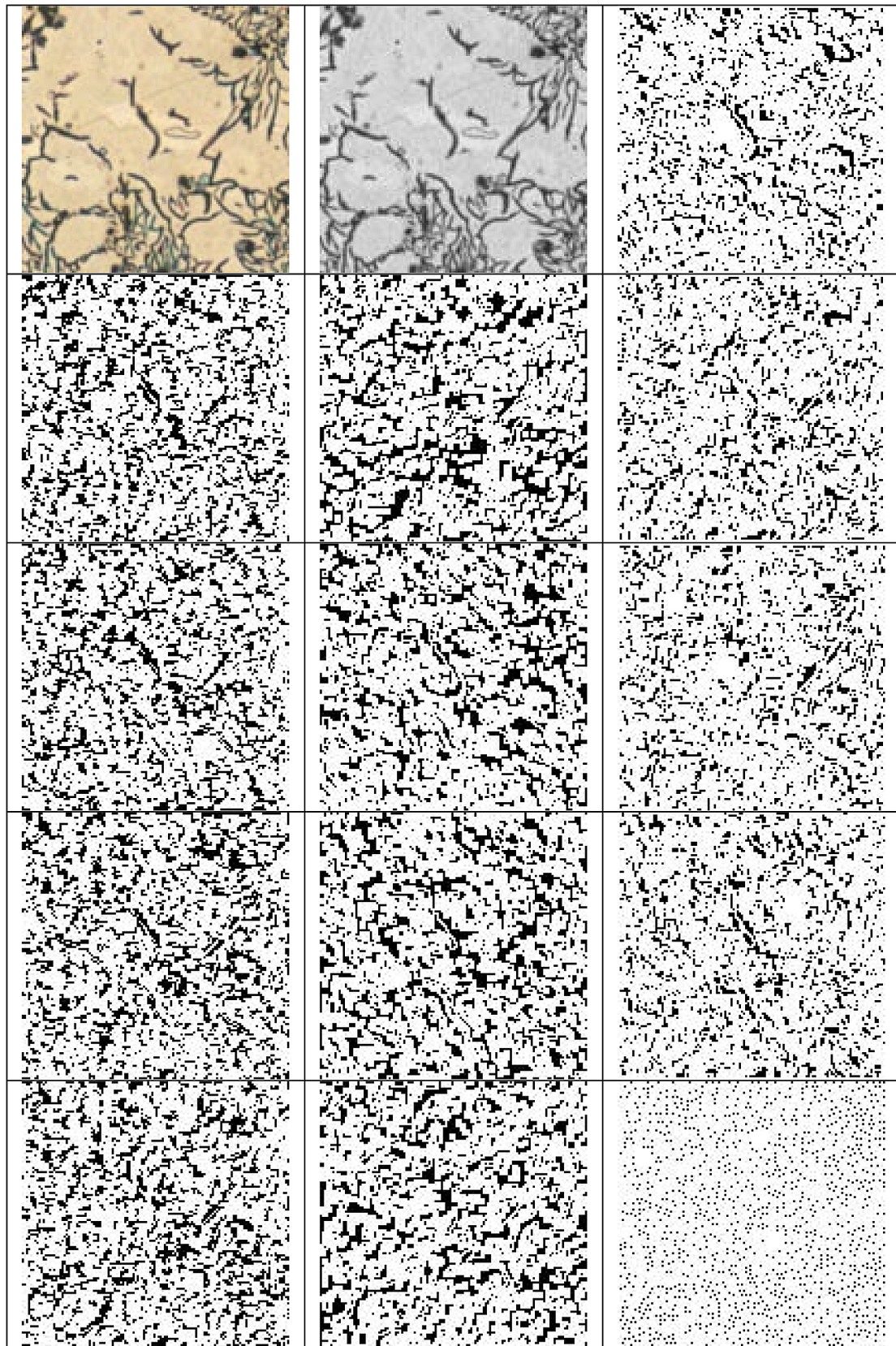
$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(f_{ij} - e_{ij})^2}{e_{ij}} \tag{15}$$

where  $r$  denotes the number of distinct values in the features,  $c$  is the number of distinct values in of the classes.  $f_{ij}$  is the observed value, and  $e_{ij}$  is the expected value. A higher value of  $\chi^2$  indicates that there is no relationship between the features and features with the best Chi-square scores are selected as informative features

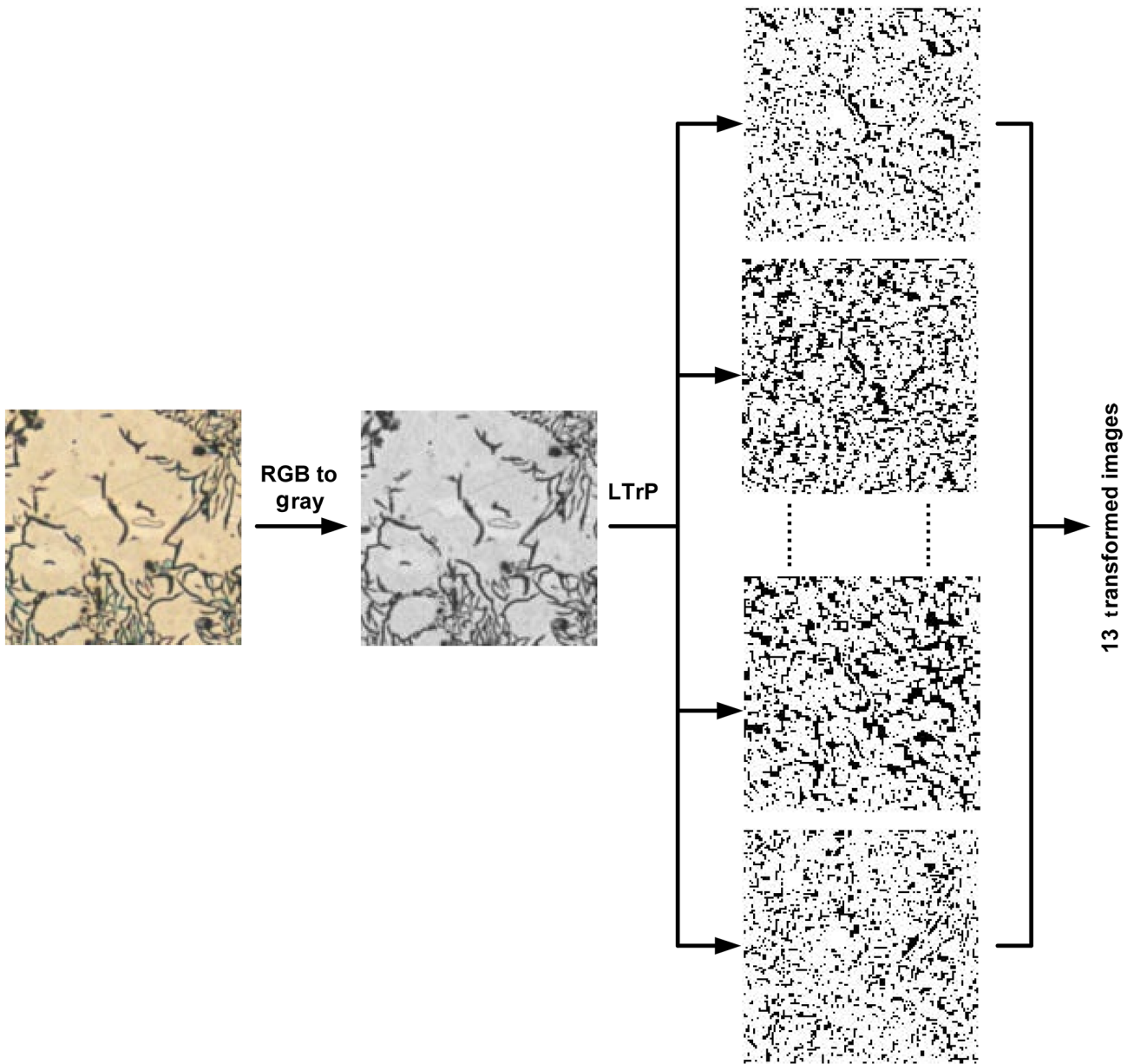
**Fisher Score**

Fisher score [40] is one of the most acceptable similarity-based supervised filter methods. It selects each feature independently based on their score under the Fisher criterion. Fisher score is the ratio of class variance to within-class variance to evaluate feature importance. Fisher score of  $i^{\text{th}}$  feature is computed as

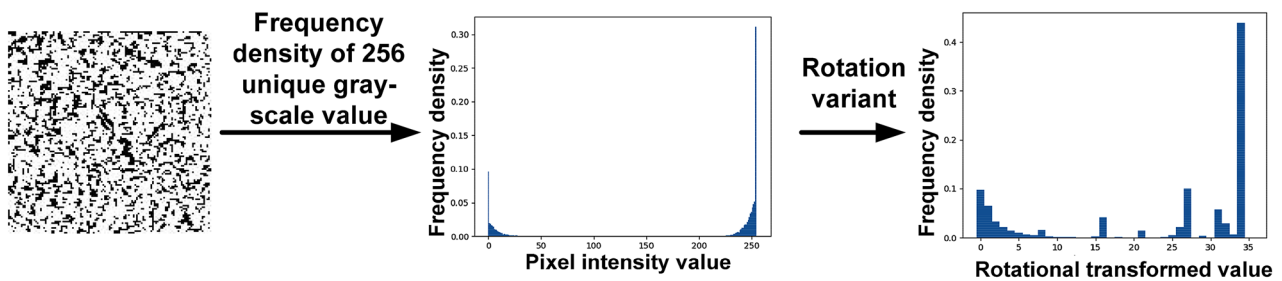




**Fig. 4** Original sample color micrograph of one gray cast iron, its grayscale image and all 13 images generated by our proposed feature extraction method



(a) Image transformation steps



(b) Feature extraction from transformed images

Fig. 5 a–b. Pictorial representation of the proposed feature extraction method

$$F_i = \frac{\sum_{j=1}^c n_j * (u_j - u)^2}{\sum_{j=1}^c n_j * \sigma_j^2} \tag{16}$$

where  $u_j$  and  $\sigma_j$  represent the mean and standard deviation of  $j^{\text{th}}$  class, respectively, corresponding to the  $i^{\text{th}}$  feature, and  $u$  represents the mean of all classes. After evaluation of the Fisher score of each independent feature, the top-n ranked features with the largest score are selected as a suboptimal subset of features.

**Gini Index**

Gini impurity or Gini index is a popular metric that is used in CART (classification and regression tree) and decision tree algorithm for FS [41]. It is an assessment of the possibility of incorrect classification of a randomly chosen element from the dataset conforming to the distribution of labels in the subset. It is computationally efficient and takes a shorter time for execution. Mathematically, Gini impurity for a set of items with n classes can be computed as

$$G(s) = \sum_{i=1}^n p_i * (1 - p_i) \tag{17}$$

where  $p_i$  be the probability of choosing a sample of a certain classification labeled with i. Gini impurity lower bounded by 0 implies that this an ideal split that corresponds to all the members of the set attached to the same class, i.e. the class is perfectly separated. The opposite is true when members of the set are randomly distributed across different classes.

**Objective Function**

For FS purpose, an agent which is kept in the memory is treated as a harmony. The accuracy of the proposed HS method is dependent on the objective function which is determined by the computation using PCC and MI. A brief description of these two methods is given below.

**Pearson Correlation Coefficient (PCC)**

In statistics, PCC measures the linear correlation between variables X and Y. It has a value between -1 and +1. PCC is often represented as  $r_{xy}$ . We can obtain the value of  $r_{xy}$  by the following formula:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \tag{18}$$

where  $r_{xy}$  is the PCC value,  $x_i$  and  $y_i$  represent  $i^{\text{th}}$  sample in X and Y, respectively, and n is the total number of samples.

For the FS problem, the features are defined as:

$$\text{Feature} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_{N_{\text{Feature}}}^1 \\ x_1^2 & x_2^2 & \dots & x_{N_{\text{Feature}}}^2 \\ \vdots & \vdots & & \vdots \\ x_1^{N_{\text{instance}}} & x_2^{N_{\text{instance}}} & \dots & x_{N_{\text{Feature}}}^{N_{\text{instance}}} \end{bmatrix} \tag{19}$$

where  $N_{\text{instance}}$  is total number of instances in the feature set.

$$\text{Class} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^{N_{\text{instance}}} \end{bmatrix} \tag{20}$$

We define,

$$\text{PCC}_{\text{feature}} = [r_{xy}^1 \ r_{xy}^2 \ \dots \ r_{xy}^{N_{\text{Feature}}}] \tag{21}$$

where  $r_{xy}^i$  is PCC value calculated from  $\text{Feature}(:, k)$  and  $\text{Class}$ ,  $k \in \{1, 2, \dots, N_{\text{instance}}\}$ .

Hence,

$$\text{pcc}(h_i) = \frac{\sum_{k=1}^{N_{\text{Feature}}} \text{PCC}_{\text{feature}}(k) * h_i^k}{\sum_{k=1}^{N_{\text{Feature}}} h_i^k} \tag{22}$$

**Mutual Information (MI)**

MI is a measure between two (possibly multi-dimensional) random variables X and Y that quantifies the amount of information obtained about one random variable, through the other random variable. MI is given by

$$I(X; Y) = \sum_{x \in X, y \in Y} P_{xy}(x, y) \log \frac{P_{xy}(x, y)}{P_x(x)P_y(y)} \tag{23}$$

where  $P_{xy}(x, y)$  is the joint probability density function of X and Y and  $P_x(x)$  and  $P_y(y)$  are the marginal density functions. MI determines how similar the joint distribution  $P_{xy}(x, y)$  is to the products of the factored marginal distributions. If X and Y are completely unrelated, then  $P_{xy}(x, y)$  would equal  $P_x(x)P_y(y)$ , and this integral would be zero.

We define,

$$\text{MI}_{\text{feature}} = [I(\text{Feature}(:, 1); \text{Class}) \ I(\text{Feature}(:, 2); \text{Class}) \ \dots \ I(\text{Feature}(:, N_{\text{Feature}}); \text{Class})]$$

Hence,

$$mi(h_i) = \frac{\sum_{k=1}^{N_{Feature}} MI_{feature}(k) * h_i^k}{\sum_{k=1}^{N_{Feature}} h_i^k} \quad (24)$$

## Classifier

Three different types of classifiers, namely SVM, random forest (RF), and K-nearest neighbor (KNN), are used to determine the classification accuracy of the optimal feature subsets obtained by the proposed HA algorithm-based FS process. It is to be noted that the proposed one is an unsupervised FS method, so it has not used any classifier for performance evaluation of the feature subsets. We evaluate the optimal feature subset using SVM classifier where the non-linearity is introduced by the polynomial kernel for transforming data points to a higher-dimensional feature space.

KNN is another classifier (nonparametric algorithm) used in this paper. It is also called a lazy learner algorithm or an instance-based classifier because in the training method it does not learn from the training set immediately; instead, it saves the training examples. At the time of classification, it finds the K training examples  $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$  that are closed to the test example  $x$  and predicts the most frequent class among those  $y_i$ 's that is much similar to the new data.

RF is an ensemble of tree-based supervised learning algorithms used in this work. RF is constructed from decision trees on randomly selected data samples. The intention of the RF is to minimize the variance of the decision trees and produce better results in the final class of prediction by means of voting among decision trees. The number of trees used in the forest for the present work is 500.

## Proposed Model

Multiple feature descriptors, namely basic LBP, RLTrP, GLCM, ORB, and SIFT, are used to extract the features from the microstructural images. The features estimated from these five different feature descriptors are individually trained with SVM, RF, and KNN classifiers to obtain the classification accuracy, and the results are compared to judge best among the five feature descriptors that produce the highest classification accuracy by the majority of the classifiers. Finally, this feature set has opted for FS using the proposed wrapper-filter-based HA algorithm-based FS method.

From the literature survey about FS methods, it can be observed that the general trend of the researchers in different areas is to use the wrapper-filter combination as it ensures a good trade-off between both filter and wrapper methods. Going by this trend, the proposed work introduces

a wrapper-filter FS method for the classification of microstructure images. Initially, it forms an ensemble of three filter methods for finding informative features and setting up as an input to the wrapper method for further improvement in terms of dimension reduction and enhancement of classification accuracy. As a wrapper method, a meta-heuristic method called the HS algorithm is applied to get the minimal subset of features with maximal classification accuracy.

## Feature Selection Using Filter Method

A wide variety of filter methods may be used for ranking of features. However, each of the methods has not shown equally efficient results for a definite feature set. To make stage 1 more efficient and robust, a combination of three filter methods belonging to distinct categories, namely Chi-square (statistics-based), Fisher score (entropy-based), and Gini impurity (similarity-based), is used. The *union* operation is performed for combining the top-ranked features of each of the three filter methods, thereby forming the ensemble of filter methods to obtain important features. The main purpose of this ensemble is to retain the important features that may get missed by any individual filter method.

The approach that followed in this research work is schematically represented in Fig. 6.

## Feature Selection Using Harmony Search

Geem and Kim et al. [42] introduced a meta-heuristic optimization algorithm called HS algorithm. This method took inspiration from musical harmony, a way to determine the merit of musical performance. Musical harmony is a combination of sounds considered pleasing for an aesthetic point of view. Music performance seeks a great state (best possible harmony) determined by aesthetic estimation, while optimizing problem seeks the best state, determined by an objective function.

As described in [42], the HS algorithm has the following steps:

Step 1: initialize the problem and its parameters.

For FS purposes, we determine an agent as a harmony, which is to be stored in HM. A chromosome is a vector consists of 1 s and 0 s, of length  $N_{Feature}$ . It is to be noted that 1 at any index represents the selection of the corresponding feature at that index, and 0 represents reverse. Population is the collection of individual features and here each individual is a harmony. Population size is number of harmony present in the population. The fitness of the agent is determined by the objective function described in “Objective function” section.

Step 2: Initialize HM.

HM is designed as follows,

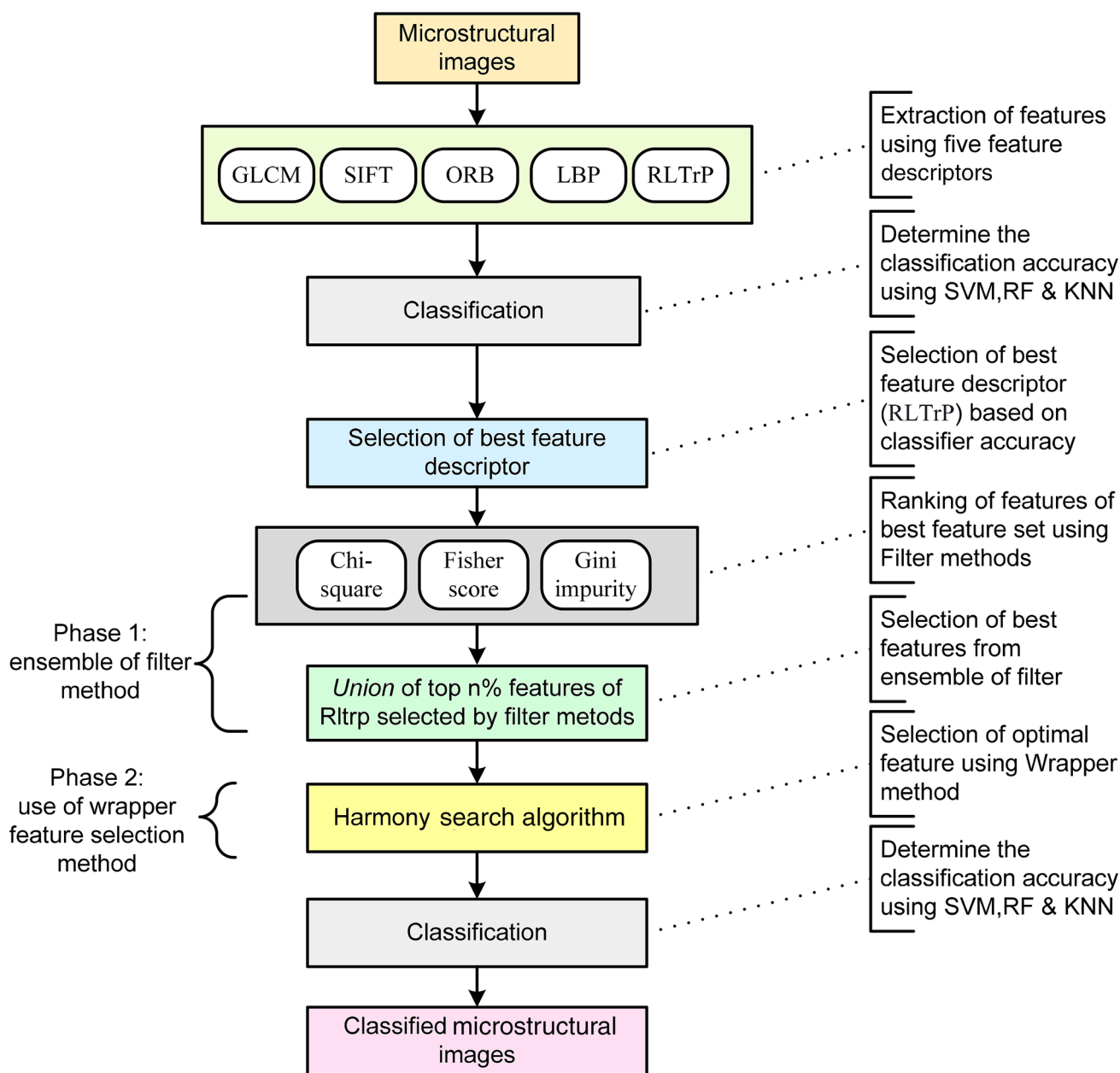


Fig. 6 Block diagram of the proposed wrapper-filter-based feature selection method used for classifying microstructural images

$$HM = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{N_{harmony}} \end{bmatrix} \quad [N_{harmony} \text{ is total number of harmony in } HM] \tag{25}$$

A harmony can be described as follows,

$$h_i = \{ h_i^1 \ h_i^2 \ \dots \ h_i^{N_{Feature}} \} \tag{26}$$

Initially, the HM is filled with random harmonies and the HM is sorted according to the scores of the harmonies generated by the objective function.

Step 3: Improvise a new harmony from HM.

In the improvising procedure, a new harmony is generated by selecting the value of a random harmony from the HM, for each of the  $j \in \{1, 2, \dots, N_{Feature}\}$  and constructing the harmony in the following way:

$$h_{improvise} = [ h_{rand_1}^1 \ h_{rand_2}^2 \ \dots \ h_{rand_{N_{Feature}}}^{N_{Feature}} ] \tag{27}$$

where  $rand_j$  is the index of the randomly chosen harmony from HM for  $j \in \{1, 2, \dots, N_{Feature}\}$ .

#### Step 4: Update HM

The fitness value is calculated using the objective function for  $h_{improvised}$ . If  $h_{improvised}$  overperforms than  $h_{N_{harmony}}$  (which is the least performing harmony), then  $h_{N_{harmony}}$  is replaced with  $h_{improvised}$ . HM is kept sorted according to the fitness value calculated by the objective function.

#### Step 5: Termination

Until the stopping criterion is met for this process, the method continues from **Step 3**.

#### Objective Function:

The objective function is defined as follows,

$$F_{objective}(h_i) = pcc(h_i) * mi(h_i) \quad (28)$$

As described in (28),  $F_{objective}(h_i)$  consists of two parts,  $pcc(h_i)$  &  $mi(h_i)$ .

## Results and Discussion

For the implementation of the proposed method, MATLAB 2018b programming platform has been used. The hardware used for experimentation are Intel® 3rd Generation Core™ i7 3570 processor with 16 GB RAM. Source code of the feature extraction and FS methods can be found at the GitHub links provided in [43] and [44].

### Description of the Dataset

The micrographs are collected from the publicly available Cambridge Dissemination of IT for the Promotion of Materials Science (DoITPoMS) micrograph library [45]. Some sample images used in the classification are shown in Fig. 7. The image dataset consists of micrographs from seven different categories, namely *annealing twin*, *brass/bronze*, *ductile cast iron*, *gray cast iron*, *malleable cast iron*, *nickel-based superalloy*, and *white cast iron*. Each of the original images is cropped into 16 number of square images of size 128x128 pixel. To avoid conflict in the feature extraction process, the cropped images with scale bars are manually excluded from the dataset. Finally, a total of 1201 number of cropped are used in the classification process. The details of the micrograph are listed in Table 1. No filtering process or noise cancelation process is applied to the image dataset for the preprocessing of the image.

This section demonstrates the results of classifying the 7-class microstructural images using the proposed wrapper-filter-based FS method where texture-based features are given as input. Initially, the features of images are extracted using five different feature descriptors (LBP, RLTrP, GLCM, ORB, and SIFT) and the classification task is performed by

three different classifiers such as SVM, RF, and KNN. After comparing the test accuracy, the best feature descriptor is selected for the wrapper-filter FS process. In this classification task, 80% of the image samples are randomly chosen as training data and the rest 20% as testing data. To find out the most informative features in stage 1, an ensemble of the top-ranked features from the three filter methods (Chi-square, Fisher, and Gini impurity) is used which is created by forming a union of top-n % features. The objective of this union operation is to combine the best n-features and to avoid feeding less informative features for FS at stage 2 using the HS algorithm. The success of the HS algorithm in microstructural classification is shown in Table 4.

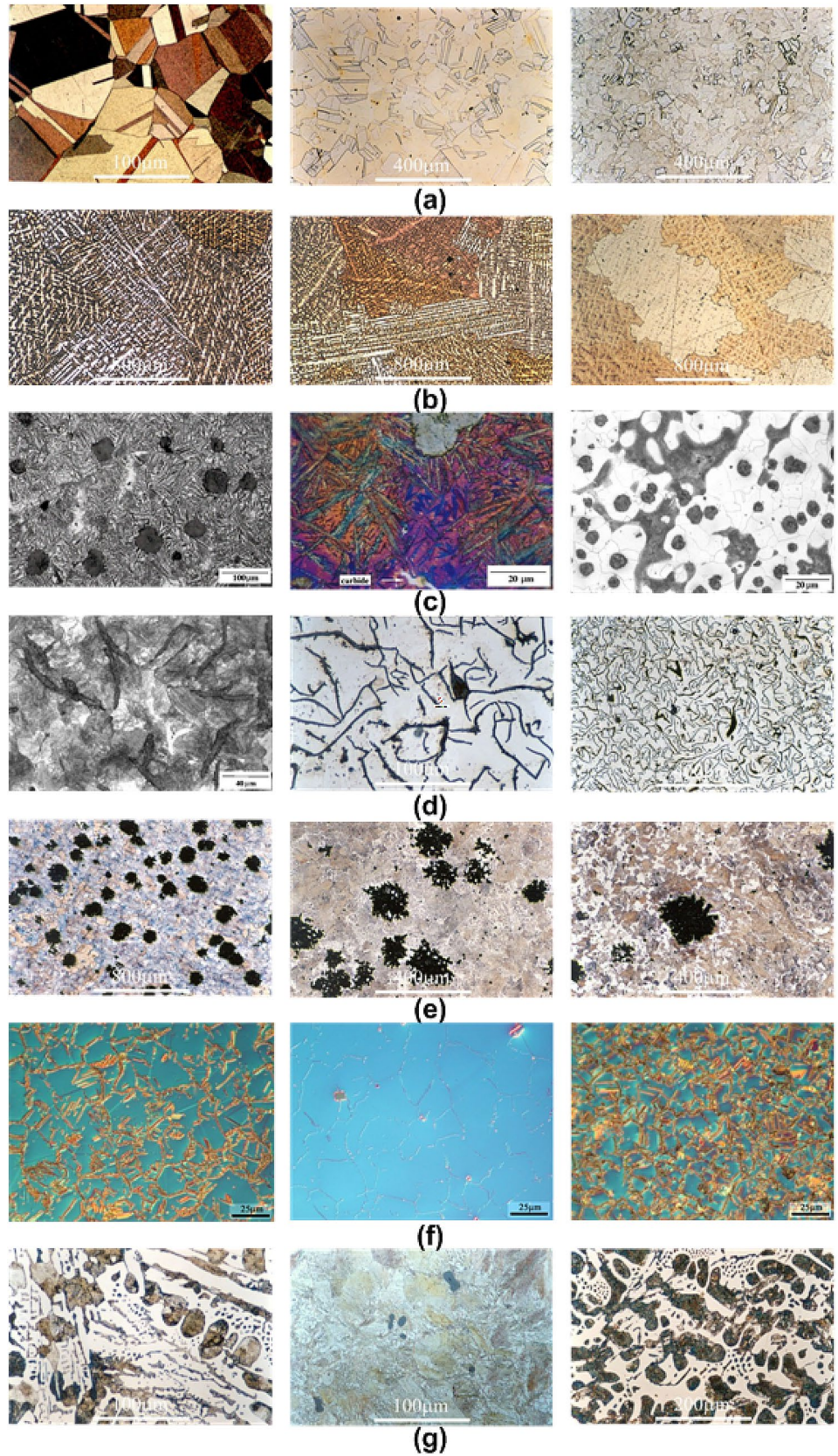
The performance comparison of classification models for five feature descriptors is presented in Table 2. The results obtained by comparing feature descriptor and classifier combinations indicate that RLTrP has achieved the highest classification accuracy of 93.57%, 88.11%, and 87.35% using SVM, KNN, and RF classifiers, respectively, among others with 468 features. The SVM has yielded the maximum classification accuracy among the three classifiers. Figure 8 shows the graphical variation of the accuracies obtained using three classifiers with the features extracted with RLTrP. The other three feature descriptors, namely LBP, SIFT, and GLCM, have also shown promising accuracy with these three classifiers.

To find the optimum number of significant features in the filter method, an approach is taken by determining test accuracies for different percentages of top-ranked features obtained by the ensembles of the three filter ranking methods. The variation of test accuracies with these features obtained from the union of the three ranking methods is listed in Table 3 and depicted in Fig. 9. It is clear from Fig. 9 that the best optimum result is obtained when 50% of the ranked features are considered. This reduced feature set with 399 number of features is selected as an input to the HS algorithm in stage 2 for performing further FS. The objective of the filter method is to select the important features, thereby guiding the search process performed in the wrapper method. The filter method in the first phase is capable of reducing the dimension of features by 15%.

### Results of HS Algorithm

In this section, the results of HS algorithm classifying the microstructural images are reported. Results presented in Table 4 demonstrate that the HS algorithm yields the best results (highest accuracy) with a lower number of features and SVM (polynomial). It achieves the highest test accuracy of 97.19%. In stage 2 the performance of the HS algorithm is compared with the other standard optimization algorithms that include whale optimization algorithm (WOA) [46], particle swarm optimization (PSO) [47], and Grey wolf

**Fig. 7** Examples of micrographs used in the present work: **a** annealing twin, **b** brass/bronze, **c** ductile cast iron, **d** gray cast iron, **e** malleable cast iron, **f** nickel-based superalloy, and **g** white cast iron



**Table 1** Details of the micrograph dataset

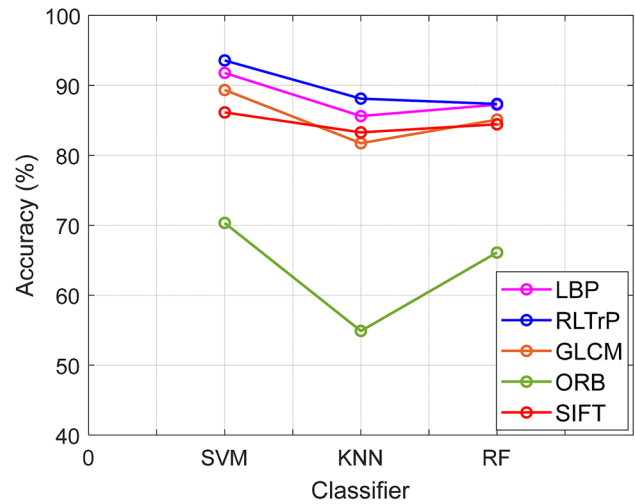
Type of micrograph	# of micrographs	# of cropped images
Annealing twin	8	113
Brass/bronze	20	280
Ductile cast iron	16	254
Gray cast iron	8	118
Malleable cast iron	7	98
Nickel-based superalloy	8	128
White cast iron	15	210
Total	82	1201

optimization (GWO) [48]. The performance comparison is listed in Table 4. It is worth mentioning that the HS algorithm has been consistent to outperform the state-of-the-art FS methods considered here for comparison both in terms of accuracy and percentage of feature reduction, and the HS algorithm is capable to reduce the features by 82%.

Parameter tuning is an important part of any FS method. Two important parameters of our method are population size and number of iterations. We have performed some experiments to set the optimal values of these parameters. Figure 10a shows that the accuracy starts off with 95.41% when the number of iterations is 300 and then increases the stair-case manner and reaches 97.19% at iteration 750 and finally it becomes stagnant and achieves maximum accuracy of 97.96% at iteration 1400. The changes in accuracy with population size as shown in Fig. 10(b) depict that the classification accuracy has reached 95.2% with population size 20 and increased linearly until reached a maximum accuracy of 97.96%. The maximum accuracy achieves when the population size is 30, and with the increase of population from 30 it has been observed that there is a sharp fall in the accuracy and it becomes stagnant around 95% accuracy. Figure 11 depicts the graphical representation of test accuracy using our proposed model and also comparison with the other recent optimization algorithms.

**Table 2** Five different feature descriptors and corresponding classification accuracies (in %) without FS using different classifiers

Feature descriptor	# of features	SVM		KNN		RF	
		Test accuracy (%)	Accuracy on fivefold CV (%)	Test accuracy (%)	Accuracy on fivefold CV (%)	Test accuracy (%)	Accuracy on fivefold CV (%)
LBP	256	94.64	91.73	85.71	85.61	89.54	87.24
RLTrP	468	95.15	93.57	85.46	88.11	91.84	87.35
GLCM	22	91.33	89.34	83.42	81.73	90.56	85.10
ORB	200	81.04	703.3	59.89	54.89	74.73	66.10
SIFT	100	89.95	84.92	84.92	83.28	88.36	84.44



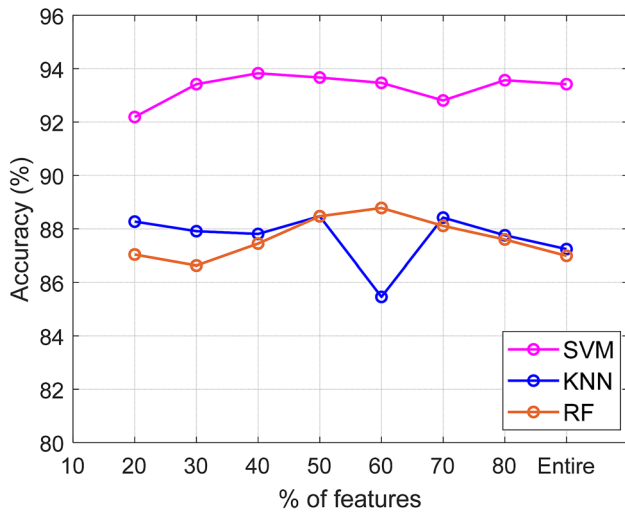
**Fig. 8** Variation of fivefold CV test accuracies for each feature descriptor and classifier without feature selection

**Table 3** Fivefold CV test accuracy (in %) of three classifiers for features (extracted by RLTrP) obtained by the union of the different percentages of top-ranked features of the three filter methods

# of top-ranked features	Test accuracy (%) obtained by		
	SVM	KNN	RF
Entire	93.42	87.24	86.99
20%	92.19	88.27	87.04
30%	93.42	87.91	86.63
40%	93.83	87.81	87.45
50%	93.67	88.47	88.47
60%	93.47	85.46	88.78
70%	92.81	88.42	88.11
80%	93.57	87.76	87.60

The confusion matrix presented in Fig. 12 represents the classification performance of each class predicted by SVM after wrapper-filter feature selection. From this figure, it can be said that the proposed model is competent





**Fig. 9** Variation of fivefold CV classification accuracies (in %) for different percentages of top-ranked features (extracted using RLTrP) obtained from ensemble of the three filter methods in stage 1

to classify every image of the test dataset of four classes (brass, gray cast iron, malleable cast iron, and white cast iron) but it misclassifies only one image of annealing twins, three images of ductile cast iron and seven images of nickel-based superalloy.

### Conclusion

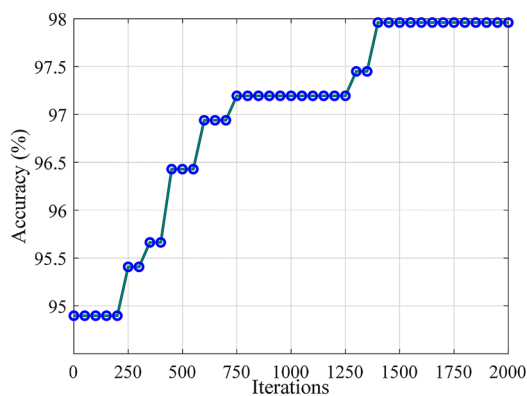
The proposed work demonstrates the usefulness of the wrapper-filter-based FS method using the HS algorithm in classifying the microstructural images aided with some filter methods where texture-based features extracted from the said images are fed as input. By using the proposed wrapper-filter FS method for microstructure classification, it has been observed as follows:

- Classification results for the 7-class microstructural data (annealing twin, brass/bronze, ductile cast iron, gray cast iron, malleable cast iron, nickel-based superalloy, and

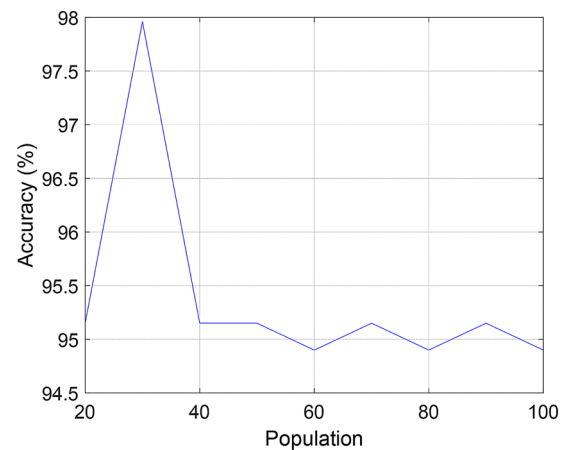
**Table 4** Comparison of results obtained using the proposed FS model in stage 2 along with some standard optimization algorithms for classification of microstructural images

FS method	# of features (original)	# of features (stage 1)	# of features (stage 2)	Test accuracy (%) obtained using		
				SVM	KNN	RF
<b>Proposed</b>	468	399	<b>84</b>	<b>97.19</b>	86.73	91.82
WOA			256	94.90	84.69	90.31
GWO			261	94.39	85.97	89.03
PSO			280	94.13	84.95	89.80

Bold values highlight the best result in all cases

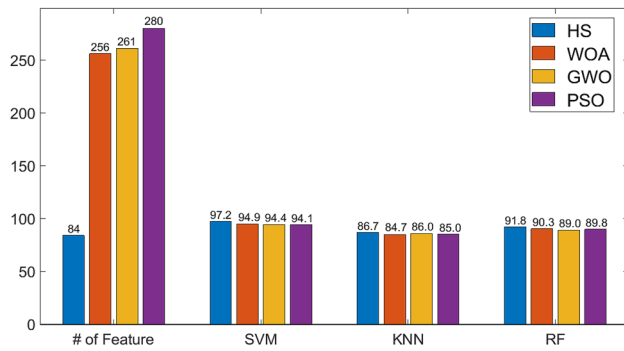


(a)

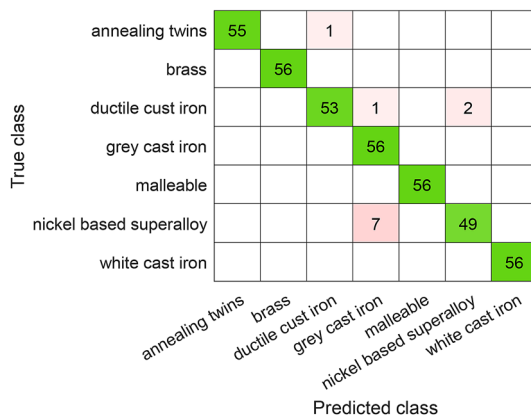


(b)

**Fig. 10** **a** Variation of accuracies with the number of iterations in the HS algorithm using the SVM classifier. **b** Variation of accuracies with population size in HS algorithm using the SVM classifier



**Fig. 11** Comparison of the proposed FS method with some state-of-the-art FS methods for classifying microstructural images



**Fig. 12** Confusion matrix obtained by proposed FS model using SVM classifier showing performance of the microstructural classification after wrapper-filter feature selection

white cast iron) exhibit a high accuracy using texture-based feature descriptor RLTrP with SVM classifier after applying our proposed wrapper-filter feature reduction model.

- The high classification accuracy exemplifies that the texture-based RLTrP feature descriptor is highly favorable for microstructure classification.
- Our proposed meta-heuristic-based HS algorithm applied at stage 2 for FS is also compared with some state-of-the-art algorithms (WOA, GWO, and PSO) which confirms that our proposed method outperforms the others in terms of both classification accuracy and number of selected features.
- Classification results for the 7-class show that after stage 1 (filter method) maximum classification accuracy is achieved by 93.67% and minimizes the number of features by 14.7%. At the final stage, the proposed model is capable to achieve the highest classification accuracy of 97.19% and minimizes the number of features by 82%.

- In the future, we can think of some other competent filter methods at stage 1 of this FS method. Besides, we can add some local search methods to enhance the exploitation capability of the HS algorithm.
- We can explore some other texture-based features also or we can combine some texture-based features which can give complementary information about the input data.
- Another future scope would to apply this method to other microstructural datasets.

## References

1. Clemens H, Mayer S, Scheu C (2017) Microstructure and properties of engineering materials. In: Neutrons and synchrotron radiation in engineering materials science: from fundamentals to applications, 2nd edn. <https://doi.org/10.1002/9783527684489.ch1>
2. Gourgues-Lorenzon AF (2009) Application of electron backscatter diffraction to the study of phase transformations: present and possible future. *J Microscopy* 233(3):460–473. <https://doi.org/10.1111/j.1365-2818.2009.03130.x>
3. Britz D, Webel J, Schneider A, Mücklich F (2017) Identifying and quantifying microstructures in low-alloyed steels: a correlative approach. *Metallurgia Italiana* 3:5–10
4. Carvalho ED, Antônio Filho OC, Silva RR, Araújo FH, Diniz JO, Silva AC, Paiva AC, Gattass M (2020) Breast cancer diagnosis from histopathological images using textural features and CBIR. *Artif Intell Med* 105:101845. <https://doi.org/10.1016/j.artmed.2020.101845>
5. Zhao W, Li S, Li A, Zhang B, Li Y (2019) Hyperspectral images classification with convolutional neural network and textural feature using limited training samples. *Remote Sens Lett* 10(5):449–458. <https://doi.org/10.1080/2150704X.2019.1569274>
6. Mather P, Tso B (2016) Classification methods for remotely sensed data. CRC Press, Boca Raton
7. Weszka JS, Dyer CR, Rosenfeld A (1976) A comparative study of texture measures for terrain classification. *IEEE Trans Syst Man Cybernetics* 4:269–285. <https://doi.org/10.1109/TSMC.1976.5408777>
8. Guan D, Xiang D, Tang X, Wang L, Kuang G (2019) Covariance of textural features: a new feature descriptor for SAR image classification. *IEEE J Selected Topics Appl Earth Observations Remote Sens* 12(10):3932–3942. <https://doi.org/10.1109/JSTAR.S.2019.2944943>
9. Gajalakshmi K, Palanivel S, Nalini NJ, Saravanan S (2018) Automatic classification of cast iron grades using support vector machine. *Optik* 157:724–732. <https://doi.org/10.1016/j.ijleo.2017.11.183>
10. Webel J, Gola J, Britz D, Mücklich F (2018) A new analysis approach based on Haralick texture features for the characterization of microstructure on the example of low-alloy steels. *Mater Charact* 144:584–596. <https://doi.org/10.1016/j.matchar.2018.08.009>
11. Kitahara AR, Holm EA (2018) Microstructure cluster analysis with transfer learning and unsupervised learning. *Integr Mater Manuf Innov* 7(3):148–156. <https://doi.org/10.1007/s40192-018-0116-9>
12. Gola J, Britz D, Staudt T, Winter M, Schneider AS, Ludovici M, Mücklich F (2018) Advanced microstructure classification by

- data mining methods. *Comput Mater Sci* 148:324–335. <https://doi.org/10.1016/j.commatsci.2018.03.004>
13. Xue B, Zhang M, Browne WN, Yao X (2015) A survey on evolutionary computation approaches to feature selection. *IEEE Trans Evolutionary Comput* 20(4):606–626. <https://doi.org/10.1109/TEVC.2015.2504420>
  14. Mitra P, Murthy CA, Pal SK (2002) Unsupervised feature selection using feature similarity. *IEEE Trans Pattern Anal Mach Intell* 24(3):301–312. <https://doi.org/10.1109/34.990133>
  15. Kashef S, Nezamabadi-pour H (2015) An advanced ACO algorithm for feature subset selection. *Neurocomputing* 147:271–279. <https://doi.org/10.1016/j.neucom.2014.06.067>
  16. Duval B, Hao JK, Hernandez Hernandez JC (2009) A memetic algorithm for gene selection and molecular classification of cancer. In: Proceedings of the 11th annual conference on genetic and evolutionary computation, pp. 201–208. <https://doi.org/10.1145/1569901.1569930>
  17. DeCost BL, Holm EA (2015) A computer vision approach for automated analysis and classification of microstructural image data. *Comput Mater Sci* 110:126–133. <https://doi.org/10.1016/j.commatsci.2015.08.011>
  18. Chowdhury A, Kautz E, Yener B, Lewis D (2016) Image driven machine learning methods for microstructure recognition. *Comput Mater Sci* 123:176–187. <https://doi.org/10.1016/j.commat.2016.05.034>
  19. Gola J, Webel J, Britz D, Guitar A, Staudt T, Winter M, Mücklich F (2019) Objective microstructure classification by support vector machine (SVM) using a combination of morphological parameters and textural features for low carbon steels. *Comput Mater Sci* 160:186–196. <https://doi.org/10.1016/j.commatsci.2019.01.006>
  20. Diao R, Shen Q (2012) Feature selection with harmony search. *IEEE Trans Syst Man Cybernetics Part B (Cybernetics)* 42(6):1509–1523. <https://doi.org/10.1109/TSMCB.2012.2193613>
  21. Das S, Singh PK, Bhowmik S, Sarkar R, Nasipuri M (2017) A harmony search based wrapper feature selection method for Holistic Bangla word recognition. *arXiv preprint arXiv:1707.08398*. <https://doi.org/10.1016/j.procs.2016.06.087>
  22. Gholami J, Pourpanah F, Wang X (2020) Feature selection based on improved binary global harmony search for data classification. *Appl Soft Comput*. <https://doi.org/10.1016/j.asoc.2020.106402>
  23. Zainuddin Z, Lai KH, Ong P (2016) An enhanced harmony search based algorithm for feature selection: applications in epileptic seizure detection and prediction. *Comput Electr Eng* 53:143–162. <https://doi.org/10.1016/j.compeleceng.2016.02.009>
  24. Moayedikia A, Ong KL, Boo YL, Yeoh WG, Jensen R (2017) Feature selection for high dimensional imbalanced class data using harmony search. *Eng Appl Artif Intelligence* 57(2017):38–49. <https://doi.org/10.1016/j.engappai.2016.10.008>
  25. Nekkaa M, Boughaci D (2016) Hybrid harmony search combined with stochastic local search for feature selection. *Neural Process Lett* 44(1):199–220
  26. Dash R (2018) An adaptive harmony search approach for gene selection and classification of high dimensional medical data. *J King Saud Univ Comput Inf Sci*. <https://doi.org/10.1016/j.jksuci.2018.02.013>
  27. Ramos CC, Souza AN, Chiachia G, Falcão AX, Papa JP (2011) A novel algorithm for feature selection using harmony search and its application for non-technical losses detection. *Comput Electr Eng* 37(6):886–894. <https://doi.org/10.1016/j.compeleceng.2011.09.013>
  28. Huang Yin-Fu, Lin Sheng-Min, Huan-Yu Wu, Li Yu-Siou (2014) Music genre classification based on local feature selection using a self-adaptive harmony search algorithm. *Data Knowl Eng* 92:60–76. <https://doi.org/10.1016/j.datak.2014.07.005>
  29. Tao Y, Wang K, Yang J, An N, Li L (2015) Harmony search for feature selection in speech emotion recognition. In: 2015 International conference on affective computing and intelligent interaction (ACII), pp. 362–367. IEEE, 2015. <https://doi.org/10.1109/ACII.2015.7344596>
  30. Saha S, Ghosh M, Ghosh S, Sen S, Singh PK, Geem ZW, Sarkar R (2020) Feature selection for facial emotion recognition using cosine similarity-based harmony search algorithm. *Appl Sci* 10(8):2816. <https://doi.org/10.3390/app10082816>
  31. Haralick RM, Shanmugam K, Dinstein IH (1973) Textural features for image classification. *IEEE Trans Syst Man Cybern* 6:610–621. <https://doi.org/10.1109/TSMC.1973.4309314>
  32. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vision* 60(2):91–110
  33. Rublee E, Rabaud V, Konolige K, Bradski G (2011) ORB: an efficient alternative to SIFT or SURF. In: 2011 International conference on computer vision, pp. 2564–2571. IEEE, 2011. <https://doi.org/10.1109/ICCV.2011.6126544>
  34. Rosin PL (1999) Measuring corner properties. *Comput Vis Image Underst* 73(2):291–307. <https://doi.org/10.1006/cviu.1998.0719>
  35. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Mach Intell* 7:971–987. <https://doi.org/10.1109/TPAMI.2002.1017623>
  36. Murala S, Maheshwari RP, Balasubramanian R (2012) Local tetra patterns: a new feature descriptor for content-based image retrieval. *IEEE Trans Image Process* 21(5):2874–2886. <https://doi.org/10.1109/TIP.2012.2188809>
  37. Tan X, Triggs B (2010) Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Trans Image Process* 19(6):1635–1650. <https://doi.org/10.1109/tip.2010.2042645>
  38. Zhang B et al (2009) Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. *IEEE Trans Image Process* 19(2):533–544. <https://doi.org/10.1109/tip.2009.2035882>
  39. Jin X, Xu A, Bie R, Guo P (2006) Machine learning techniques and Chi square feature selection for cancer classification using SAGE gene expression profiles. In: International workshop on data mining for biomedical applications, pp. 106–115. Springer, Berlin, Heidelberg
  40. Gu Q, Li Z, Han J (2012) Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725*
  41. Shang W, Huang H, Zhu H, Lin Y, Youli Q, Wang Z (2007) A novel feature selection algorithm for text categorization. *Expert Syst Appl* 33(1):1–5. <https://doi.org/10.1016/j.eswa.2006.04.001>
  42. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. *Simulation* 76(2):60–68. <https://doi.org/10.1177/003754970107600201>
  43. Khalid H. Sheikh (2019) Local tetra pattern. <https://github.com/khalid0007/Local-Tetra-Pattern>
  44. Sheikh KH (2019) Harmony Search. <https://github.com/khalid0007/Metaheuristic-Algorithms/tree/master/HS>
  45. Barber ZH, Leake JA, Clyne TW (2007) The DoITPoMS project—a web-based initiative for teaching and learning materials science. *J Mater Educ* 29(1/2):7
  46. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
  47. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95 - international conference on neural networks. pp 1942–1948 vol 4 <https://doi.org/10.1109/icnn.1995.488968>
  48. Mirjalili S, Mirjalili S, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>