Highly interpretable hierarchical deep rule-based classifier

Xiaowei Gu[a, b, *, 1]

x.gu3@lancaster.ac.uk

Plamen P. Angelov[a, b, c, 1]

p.angelov@lancaster.ac.uk

[a]School of Computing and Communications, Lancaster University, Lancaster, LA1 4WA, UK

[b]Lancaster Intelligent, Robotic and Autonomous Systems Centre (LIRA), Lancaster University, UK

[c]Honorary Professor at Technical University, Sofia, 1000, Bulgaria

[*]Corresponding author at: School of Computing and Communications, Lancaster University, Lancaster, LA1 4WA, UK.

[1]X. Gu and P. Angelov contributed equally.

Abstract

Pioneering the traditional fuzzy rule-based (FRB) systems, deep rule-based (DRB) classifiers are able to offer both human-level performance and transparent system structure on image classification problems by integrating zero-order fuzzy rule base with a multi-layer image-processing architecture that is typical for ~~the~~ deep learning ~~paradigm~~. Nonetheless, it is frequently observed that the inner structure~~s~~ of DRB ~~classifiers~~ can become over sophisticated and not interpretable for humans when applied to large-scale, complex problems. To tackle the issue, one feasible solution is to construct a tree structural classification model by aggregating the possibly huge number of prototypes identified from data into a much smaller number of more descriptive and highly abstract ones. Therefore, in this paper, we present a novel hierarchical deep rule-based (H-DRB) approach that is capable of summarizing the less descriptive raw prototypes into highly generalized ones and self-arranging them into a hierarchical prototype-based structure according to their descriptive abilities. By doing so, ~~the~~ H-DRB ~~classifier~~ can offer ~~the~~ high-level performance and, most importantly, full transparency and human-interpretability ~~for~~on various problems including large-scale ones. The proposed concept and generical principles are verified through numerical experiments based on a wide variety of popular benchmark image sets. Numerical results demonstrate that the promise of ~~the~~ H-DRB ~~approach~~.

Keywords: Deep rule-based; Hierarchical; Prototype-based; Self-organizing

1 Introduction

Deep learning paradigm is ~~arguably~~ the fastest growing branch of machine learning and artificial intelligence (AI) at the moment [1,2]. Deep neural networks (DNNs) are entirely based on ~~the~~ artificial neural networks and probabilistic type of uncertainties [3]. They have demonstrated eye-catching successes on image classification [4,5], speech processing [6,7] and many other complex problems [8,9] that traditional machine learning approaches are struggling with.

Despite of the impressive advances DNNs have achieved, the research communities and enterprises are increasingly demanding explainable AI [10,11]. Indeed, DNN-based AI algorithms nowadays are more frequently getting involved for making decisions in financial and safety-critical applications [12,13]. However, DNNs are the typical type of "black box" models with a very high level of complexity ~~level~~ that only computers can understand. It is reported that such "black box" models can provide a wrong outcome with high confidence by modifying just one pixel in the input image~~s~~ [14]. The lack of transparency and explainability can pose a significant obstacle, especially for highly regulated, high-risk/high-value industries. Therefore, there is a high demand in developing alternative architectures, learning and model structure paradigms that can ~~provide~~ offer [15]: (1) high levels of precision comparable or surpassing the level achieved by humans or/and by the state-of-the-art methods (including DNNs); (2) be highly transparent, interpretable, easy to explain and use for humans; (3) computationally efficient, fast to train and use; (4) computational resource and training data lean — able to be trained with a single or handful or examples per class, not requiring computer accelerators, such as GPU, HPC, etc.

As one of the pillars of ~~the~~ computational intelligence, fuzzy rule-based (FRB) systems are a mathematical tool to describe ~~the~~ human reasoning and decision-making processes [11]. FRB systems take the form of zero-order,

first-order, or higher-order IF…THEN rules that are highly interpretable by humans, and they have been successfully applied for various classification problems [10,16]. The majority of modern FRB systems are designed for processing nonstationary streaming data “on the fly” by self-updating and self-evolving ~~its~~their system structure and meta-parameters to follow the changing data pattern. The most popular FRB models include, but are not limited to, DENFIS [17], SAFIS [18], PANFIS [19] and IT2FNN [20]. Currently, ~~fuzzy rule-based~~FRB systems have been successfully implemented for many real-world applications [21]. Interested readers are referred to the recent survey [22] for more information. Nonetheless, it is also generally recognized that FRB systems usually could not reach the same level performance as DNNs for very complex, large-scale problems, such as image recognition, due to the simpler system structure and operating mechanism ~~than DNNs~~.

By combining the zero-order self-organizing FRB system with a multi-layer image-processing architecture, a generic approach for image classification named deep rule-based (DRB) classifier is proposed in [23]. Instead of using hundreds of millions of weights which bear no direct and clear link with the problem, ~~the~~ DRB ~~classifier~~ is defined by the extracted/identified meaningful prototypes. It is able to demonstrate highly accurate performance on image classification on par with DNNs, and, at the same time, offers high-level transparency and human-interpretability that are typical for traditional FRB systems only. Due to the nature of ~~the~~ image classification problems, ~~the~~ DRB ~~classifier~~ will demonstrate stronger performance if more images with better quality, higher variation and diversity are provided for training [24]. Given a large-scale training set, depending on the complexity of data structure, ~~the~~ DRB ~~classifier~~ may identify a huge number of prototypes from training samples to achieve highly accurate classification performance surpassing the DNN-based alternatives. However, gaining too many prototypes significantly impairs the transparency and human-interpretability of DRB because ~~they largely increase~~ the overall system complexity ~~of the overall system~~ is largely increased. These prototypes also become a heavy computational burden for both the learning and decision-making processes of the classifier because each training/validation image will be compared with all identified prototypes in terms of their visual similarity.

A commonly-used approach to address large-scale multi-class classification problems is to learn a hierarchical inter-class structure from data ~~to~~for performing classification in a coarse-to-fine manner [25–30]. Training a tree classifier to organize different categories hierarchically based on their visual similarity can effectively improve the computational efficiency during decision-making and also simplify the complexity of the classification task itself [30]. However, the majority of existing hierarchical classifiers suffer from the problem of inter-level error propagation due to the iterative structural optimization process involved during system identification [30,31]. More recently, a novel hierarchical prototype-based (HP) approach described in [32] presented an alternative tree structure for classification bypassing the aforementioned problem by self-organizing prototype-based hierarchies derived from training data per category individually. Nonetheless, the main issue with ~~the~~ HP ~~classifier~~ is that its ~~system~~model ~~structure~~depth and ~~classification performance~~structure are ~~controlled~~influenced by externally controlled parameters.

In this paper, we propose a novel hierarchical deep rule-based (H-DRB) classifier, which is capable of self-organizing a multi-layer premise part for each IF…THEN rule from the identified prototypes per class through an autonomous process free from user- and problem-specific parameters. The bottom layer of the hierarchical premise part is composed of all ~~the~~ prototypes identified directly during the training process, and the top layer consists of much less but more descriptive and representative prototypes. The proposed H-DRB ~~classifier~~ can achieve very high ~~classification~~ performance by using the top-layer prototypes for classification, on par with, or even surpassing its precursor (DRB) as well as many state-of-the-art DNN-based approaches, but with much higher computational efficiency for decision-making. In addition, since the top layer of ~~the~~ H-DRB ~~classifier~~ has orders of magnitude less prototypes, the learned knowledge can be conveniently visualized and explained to end users, ~~which e~~enabl~~es~~ing them to quickly learn the general picture of the problem~~s~~. Numerical examples based on various widely used benchmark image ~~data~~ sets are performed to demonstrate the effectiveness and validity of the proposed concept and general principles.

In summary, ~~the~~ main contributions of this paper include: (1) a new approach that self-organizes multi-layer hierarchical premise parts for zero-order IF…THEN rules; (2) the capability to summarize the extracted knowledge from data into a very small number of highly descriptive and representative prototypes; (3) the ability of performing extremely efficient decision-making with a high level of precision. Moreover, the proposed H-DRB ~~approach~~ is also free from externally controlled parameters, prior assumptions on data generation model as well as the problem of inter-level error propagation.

The remainder of this paper is organized as follows. Section 2 summarizes the architecture, training and validation processes of ~~the~~ DRB ~~classifier~~. The algorithmic process of self-organizing a hierarchical prototype-based structure is presented in Section 3. Numerical examples are given in Section 4, and this paper is concluded by Section 5.

2 Deep rule-based classifier

In this section, the general architecture, algorithmic procedures of ~~the~~ DRB ~~classifier~~ are briefly described to make this paper self-contained. Key notations are summarized in Table 1 for clarity.

Table 1 Definitions of key notations used in this paper.

Notations	Definitions
C	Number of image classes
I	A particular image

$\boldsymbol{x}$	Feature vector of $\mathbf{I}$
D	Data density
D^M	Multimodal data density
$\mathrm{F}(\mathbf{I})$	Discriminative representation extracted from $\mathbf{I}$ by the feature descriptor
$\mathbf{R}^M$	Data space
M	Dimensionality of the data space
N_i	Number of prototypes of the i th class
λ_i	Degree of visual similarity
K_i	Number of processed training images of the i th class
$\boldsymbol{\mu}_i$	Global mean of feature vectors of training images of the i th class.
$\{\mathbf{P}\}_i$	Raw prototypes of the i th class
$\{\boldsymbol{p}\}_i$	Feature vectors of $\{\mathbf{P}\}_i$
$\mathbf{P}_{i,j}$	The j th raw prototype of the i th class
$\boldsymbol{p}_{i,j}$	Feature vector of $\mathbf{P}_{i,j}$
$S_{i,j}$	Cardinality of $\mathbf{P}_{i,j}$
$r_{i,j}$	Radius of the ~~influential area~~ area of influence of $\mathbf{P}_{i,j}$
$\{\mathbf{P}\}_i^t$	Highly descriptive prototypes of the i th class obtained after the t^{th} filtering round
$\{\boldsymbol{p}\}_i^t$	Feature vectors of $\{\mathbf{P}\}_i^t$
$\mathbf{P}_{i,j}^t$	The j th highly descriptive prototype of the i th class obtained after the t^{th} filtering round
$\boldsymbol{p}_{i,j}^t$	Feature vector of $\mathbf{P}_{i,j}^t$
$\mathbb{C}_{i,j}^t$	Cluster formed around $\boldsymbol{p}_{i,j}^t$
$S_{i,j}^t$	Cardinality of $\mathbb{C}_{i,j}^t$
T_i	Number of filtering rounds before the algorithm converges

χ_i^t	Data-driven threshold for identifying neighbouring prototypes

2.1 General architecture

The general architecture of a DRB classifier is given by Fig. 1 [23]. As we can see from this figure, ~~a~~ the typical DRB ~~classifier~~ is composed of the following four components:

(1) Pre-processing module.

This module facilitates the subsequent feature extraction process by involving a number of commonly used pre-processing techniques for data preparation and augmentation. This can largely improve the generalization ability of ~~the~~ DRB ~~classifier~~. Thus, in practice, the pre-processing module usually consists of a number of sub-layers, and these sub-layers have different functionalities, for example, mean subtraction, normalization, rotation, scaling, segmentation [23].

(2) Feature descriptor.

The feature descriptor converts each image to a more informative and meaningful vector form: $\boldsymbol{x}=\mathrm{F}(\mathbf{I})$ by projecting it into a new data space $\mathbf{R}^M$ (M is the dimensionality of the data space) [23]. F $(\cdot)$ stands for the feature extraction process, and $\boldsymbol{x} \in \mathbf{R}^M$ represents the feature vector of a particular image $\mathbf{I}$ extracted by the descriptor, which can be of any types, including the low-level [33] or high-level [34–36] ones. An ensemble of different feature descriptors ~~can~~ may be created to further improve the descriptive ability.

(3) Massively parallel rule base.

The massively parallel rule base is the "learning engine". This component is an ensemble of massively parallel IF...THEN rules identified through a fully autonomous and highly human-interpretable manner. The premise part of each IF...THEN rule is composed of a (possibly huge) number of prototypes identified from the training images of a particular class (thus, one rule per class), and is formulated as [23]:

$$IF\left(\mathbf{I} \sim \mathbf{P}_{i,1}\right) OR\left(\mathbf{I} \sim \mathbf{P}_{i,2}\right) OR \ldots OR\left(\mathbf{I} \sim \mathbf{P}_{i,N_i}\right) \quad THEN\left(Class\, i\right), \tag{1}$$

where " ~ " denotes similarity, which is a form of fuzzy degree of membership; $\mathbf{P}_{i,j}$ is the j th prototype of the i th class, $j = 1,2,\ldots,N_i$; N_i is the number of prototypes of the i th class; $i = 1,2,\ldots,C$ and C is the number of image ~~classes~~categories in the training set.

(4) Decision-maker.

This component determines the semantic label of each validation image.

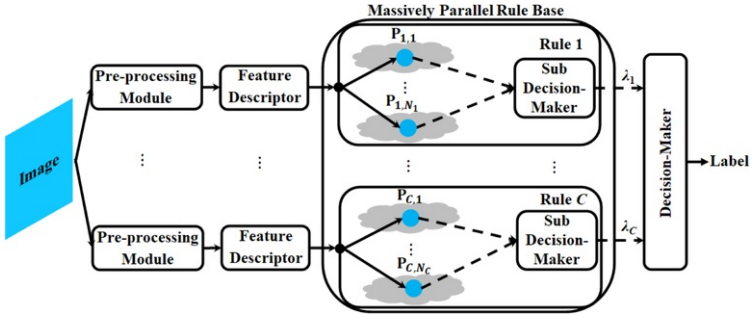


Fig. 1 General architecture of DRB ~~classifiers~~.

2.2 System identification process

In this subsection, the learning process of ~~the~~ DRB ~~classifier~~ is described. In the following algorithmic procedure, the identification process of the i th IF...THEN rule is given. The same process can be applied to all other IF...THEN rules within the same rule base [23].

The identification process of the i^{th} IF...THEN rule:

Step 1. The feature vector of the current image of the i th class, $\mathbf{I}_{i,K_i}$ is extracted and normalized by ~~the~~ its L_2 norm:

$$\boldsymbol{x}_{i,K_i} \leftarrow \frac{\mathrm{F}\left(\mathbf{I}_{i,K_i}\right)}{\left\|\mathrm{F}\left(\mathbf{I}_{i,K_i}\right)\right\|}, \tag{2}$$

where $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$.

If this is the first image, namely, $K_i = 1$, the global meta-parameters of the i th IF...THEN rule are initialized by:

$$N_i \leftarrow 1; \quad \boldsymbol{\mu}_i \leftarrow \boldsymbol{x}_{i,K_i}, \tag{3}$$

where N_i is the number of prototypes; $\boldsymbol{\mu}_i$ denotes the global mean of feature vectors of images of the i th class. ~~The l~~Local meta-parameters of the first prototype of the i th IF...THEN rule are then initialized as:

$$\mathbf{P}_{i,N_i} \leftarrow \mathbf{I}_{i,K_i}; \quad \boldsymbol{p}_{i,N_i} \leftarrow \boldsymbol{x}_{i,K_i}; \quad S_{i,N_i} \leftarrow 1; \quad r_{i,N_i} \leftarrow r_o, \tag{4}$$

where $\boldsymbol{p}_{i,N_i}$ is the feature vector of $\mathbf{P}_{i,N_i}$; S_{i,N_i} is the cardinality of $\mathbf{P}_{i,N_i}$ (number of images associated with $\mathbf{P}_{i,N_i}$); r_{i,N_i} is the radius of the area of influence of $\mathbf{P}_{i,N_i}$; r_o is a small value to stabilize r_{i,N_i} , and $r_o = \sqrt{2\left(1 - cos\left(\frac{\pi}{6}\right)\right)}$. At the end, the i th IF...THEN rule is initialized as:

$$IF\left(\mathbf{I} \sim \mathbf{P}_{i,N_i}\right) \quad THEN\left(Class\, i\right). \tag{5}$$

Otherwise (namely, $K_i \geq 1$), the global mean $\boldsymbol{\mu}_i$ is updated by $\boldsymbol{x}_{i,K_i}$:

$$\boldsymbol{\mu}_i \leftarrow \frac{K_i - 1}{K_i}\boldsymbol{\mu}_i + \frac{1}{K_i}\boldsymbol{x}_{i,K_i}. \tag{6}$$

Step 2. ~~The d~~Data densit~~ies~~y values at $\mathbf{I}_{i,K_i}$ and $\mathbf{P}_{i,1}$, $\mathbf{P}_{i,2}$,..., $\mathbf{P}_{i,N_i}$ are calculated by Eq. (7):

$$D\left(\mathbf{Z}\right) = \frac{1}{1 + \frac{\|z - \boldsymbol{\mu}_i\|^2}{1 - \|\boldsymbol{\mu}_i\|^2}}, \tag{7}$$

where $\mathbf{Z} = \mathbf{I}_{i,K_i}, \mathbf{P}_{i,1}, \mathbf{P}_{i,2}, \ldots, \mathbf{P}_{i,N_i}$; $z = \boldsymbol{x}_{i,K_i}, \boldsymbol{p}_{i,1}, \boldsymbol{p}_{i,2}, \cdots, \boldsymbol{p}_{i,N_i}$.

The nearest prototype, $\mathbf{P}_{i,n^*}$ to $\mathbf{I}_{i,K_i}$ is identified by Eq. (8):

$$n^* = \min_{j=1,2,\ldots,N_i}\left(\left\|\boldsymbol{p}_{i,j} - \boldsymbol{x}_{i,K_i}\right\|\right). \tag{8}$$

Step 3. Condition 1 is examined firstly to see whether $\mathbf{I}_{i,K_i}$ can be a new prototype:

Condition 1:

$$\begin{aligned} &If\left(D\left(\mathbf{I}_{i,K_i}\right) > \max_{j=1,2,\ldots,N_i}\left(D\left(\mathbf{P}_{i,j}\right)\right)\right) \\ &\quad Or\left(D\left(\mathbf{I}_{i,K_i}\right) < \min_{j=1,2,\ldots,N_i}\left(D\left(\mathbf{P}_{i,j}\right)\right)\right) \\ &\quad\quad Or\left(\left\|\boldsymbol{x}_{i,K_i} - \boldsymbol{p}_{i,n^*}\right\| \geq r_{i,n^*}\right) \\ &\quad Then\left(\mathbf{I}_{i,K_i} \quad is \quad a \quad new \quad prototype\right) \end{aligned} \tag{9}$$

If **Condition 1** is met, a new prototype is added by Eq. (10):

$$\begin{aligned} N_i \leftarrow N_i + 1; \quad \mathbf{P}_{i,N_i} \leftarrow \mathbf{I}_{i,K_i}; \quad \boldsymbol{p}_{i,N_i} \leftarrow \boldsymbol{x}_{i,K_i}; \\ S_{i,N_i} \leftarrow 1; \quad r_{i,N_i} \leftarrow r_o; \end{aligned} \tag{10}$$

and the IF...THEN rule is updated accordingly:

$$IF\left(\mathbf{I}\sim\mathbf{P}_{i,1}\right)OR\ldots OR\left(\mathbf{I}\sim\mathbf{P}_{i,N_i}\right)\quad THEN\left(Class\,i\right) \tag{11}$$

Otherwise, ~~the lo~~local meta-parameters of the nearest prototype are updated:

$$\begin{aligned}\boldsymbol{p}_{i,n^*} &\leftarrow \frac{S_{i,n^*}}{S_{i,n^*}+1}\boldsymbol{p}_{i,n^*}+\frac{1}{S_{i,n^*}+1}\boldsymbol{x}_{i,K_i};\\ S_{i,n^*} &\leftarrow S_{i,n^*}+1;\\ r_{i,n^*} &\leftarrow \frac{1}{2}\sqrt{r_{i,n^*}^2+\left(1-\left\|\boldsymbol{p}_{i,n^*}\right\|^2\right)}.\end{aligned} \tag{12}$$

Then, the algorithm goes back to **Step 1** if new images are available.

2.3 Validation process

During ~~this~~the validation process, for a particular validation image $\mathbf{I}$, each IF...THEN rule will produce a score of confidence, $\lambda_i(\mathbf{I})$ based on the visual similarity values between $\mathbf{I}$ and its prototypes following the "nearest prototype" principle:

$$\lambda_i\left(\mathbf{I}\right)=\max_{j=1,2,\ldots,N_i}\left(\lambda_{i,j}\left(\mathbf{I}\right)\right)=\max_{j=1,2,\ldots,N_i}\left(e^{-\left\|x-p_{i,j}\right\|^2}\right), \tag{13}$$

The decision maker, then, determines the label of $\mathbf{I}$ based on the C scores of confidence (namely, $\lambda_1(\mathbf{I}),\lambda_2(\mathbf{I}),\ldots,\lambda_C(\mathbf{I})$) following the "winner takes all" principle:

$$Label\left(\mathbf{I}\right)\leftarrow Class\,i^*;\quad i^*=\underset{i=1,2,\ldots,C}{\operatorname{argmax}}\left(\lambda_i\left(\mathbf{I}\right)\right). \tag{14}$$

3 Self-organizing the premise part into a hierarchical form

In this section, we present an approach to self-organize a multiple-layered premise part for each IF…THEN rule of ~~the~~ DRB ~~classifier~~ based on the identified prototypes, resulting in a hierarchical system structure. The obtained model is renamed as the hierarchical DRB (H-DRB) classifier.

The general architecture of ~~the~~ H-DRB ~~classifier~~ is depicted in Fig. 2, where $\{\mathbf{P}\}_i^0$ at the bottom layer of the massively parallel rule base is the set of all raw prototypes identified during the "one pass" learning process, and $\{\mathbf{P}\}_i^{T_i}$ is a set of highly descriptive prototypes at the top layer; $\{\mathbf{P}\}_i^1$, $\{\mathbf{P}\}_i^2$, ..., $\{\mathbf{P}\}_i^{T_i-1}$ are the sets of prototypes at the first, second, last hidden layers.

In this paper, the hierarchical premise parts of the IF...THEN rules are achieved by clustering. However, despite that many clustering approaches exist [37], in this paper, we are particularly interested in the recently introduced autonomous data partitioning (ADP) algorithm [38]. ADP is non-parametric, fully data-driven and free from prior assumptions and user- and problem-specific parameters. The main benefit for using this algorithm is that ADP can identify ~~the~~ local peaks of multimodal distribution of ~~the~~ existing prototypes, $\{\mathbf{P}\}_i\ (i=1,2,\ldots,C)$ through a non-parametric filtering operation based on ~~the~~ multimodal data density. Each filtering round (assuming the t th round) results in a smaller but more representative group of prototypes, $\{\mathbf{P}\}_i^t$ than the group of prototypes obtained from the previous filtering rounds $\{\mathbf{P}\}_i^{t-1},\{\mathbf{P}\}_i^{t-2},\ldots,\{\mathbf{P}\}_i^0$ ($\{\mathbf{P}\}_i^0=\{\mathbf{P}\}_i$). The filtering operation leads to a small number of highly descriptive prototypes, $\{\mathbf{P}\}_i^{T_i}$ in the end, where T_i denotes the number of filtering round ~~in which~~before ~~the~~ ADP ~~algorithm~~ converges. This naturally ~~gives~~produces a multiple-layered architecture formed by prototypes. With the help of ~~the~~ ADP ~~algorithm~~, ~~the~~ DRB ~~classifier~~ is able to self-organize a hierarchical architecture for the premise part of each IF…THEN rule as Eq. (15), ~~which results~~resulting in the proposed H-DRB ~~classifier~~.

$$\begin{aligned}IF\ &\overbrace{\left(\mathbf{I}\sim\mathbf{P}_{i,1}^{T_i}\right)OR\ldots OR\left(\mathbf{I}\sim\mathbf{P}_{i,H_i}^{T_i}\right)}^{Highly\ \ descriptive\ \ prototypes}\\ THEN\ IF\ &\overbrace{\left(\mathbf{I}\sim\mathbf{P}_{i,1}\right)OR\ldots OR\left(\mathbf{I}\sim\mathbf{P}_{i,m}\right)}^{Prototypes\ \ connected\ \ to\ \ \mathbf{P}_{i,1}^{T_i}}OR\ldots\\ &OR\ \overbrace{\left(\mathbf{I}\sim\mathbf{P}_{i,n}\right)OR\ldots OR\left(\mathbf{I}\sim\mathbf{P}_{i,N_i}\right)}^{Prototypes\ \ connected\ \ to\ \ \mathbf{P}_{i,H_i}^{T_i}}\\ &THEN\left(Class\,i\right),\end{aligned} \tag{15}$$

where $\mathbf{P}_{i,j}^{T_i} \in \{\mathbf{P}\}_i^{T_i}$; H_i is the number of prototypes at the top layer; $\mathbf{P}_{i,1},\ldots,\mathbf{P}_{i,m} \in \{\mathbf{P}\}_i$ are the prototypes at the bottom layer of the hierarchical architecture that are connected to $\mathbf{P}_{i,1}^{T_i}$ through the hidden layers; $\mathbf{P}_{i,n},\ldots,\mathbf{P}_{i,N_i} \in \{\mathbf{P}\}_i$ are the prototypes connected to $\mathbf{P}_{i,H_i}^{T_i}$.

It is worth ~~to~~ notic~~e~~ing that, although the massively parallel rule base of ~~the~~ H-DRB ~~classifier~~ has multiple layers, we are particular interested in the bottom and top layers. The top layer of the massively parallel rule base contains a small number of highly descriptive and generalized prototypes, which are very useful for human experts to interpret the problem, and it can be used for highly efficient classification as well. The bottom layer contains all the identified prototypes with fine details and can be used for showing end-users the full picture of the problems. All ~~the~~ other layers in~~-~~between are the by-products of the multiple-layered architecture identification process, and they are less important compared with the bottom and top ones. Therefore, they can be viewed as hidden layers. By default, ~~the~~ H-DRB ~~classifier~~ will use the top layer for decision-making with Eqs. (13) and (14).

The algorithmic procedure of ~~the~~ ADP ~~algorithm~~ is ~~described~~summarized as follows [38]. Similarly, the algorithm is applied to the prototypes of each class $\{\mathbf{P}\}_i$ independently to find the more descriptive prototypes $\{\mathbf{P}\}_i^{T_i}$.

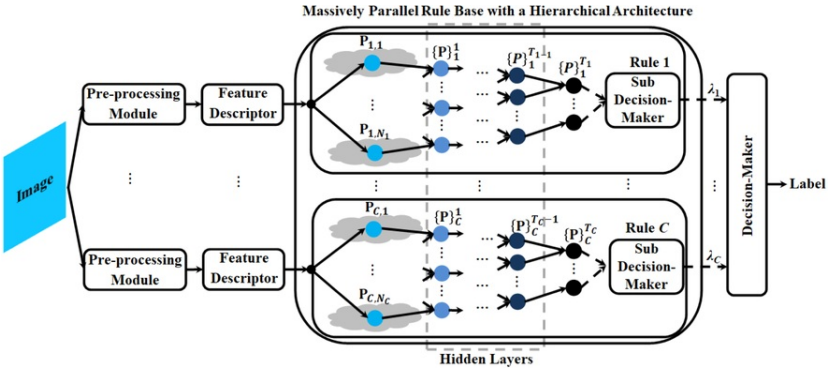


Fig. 2 General architecture of H-DRB ~~classifiers~~.

The identification process of the highly descriptive prototypes of the i th class:

Step 1. Firstly, data densities of the feature vectors, denoted by $\{\boldsymbol{p}\}_i$ of the identified prototypes $\{\mathbf{P}\}_i$ in the feature space are calculated by Eq. (16):

$$D\left(\boldsymbol{p}_{i,j}\right) = \frac{1}{1+\frac{\left\|\boldsymbol{p}_{i,j}-\boldsymbol{\rho}_i\right\|^2}{1-\left\|\boldsymbol{\rho}_i\right\|^2}}, \tag{16}$$

where $\boldsymbol{\rho}_i = \frac{1}{N_i}\sum_{l=1}^{N_i}\boldsymbol{p}_{i,l}$; $j = 1,2,\ldots,N_i$.

The feature vector with the highest data density value (re-denoted by $\boldsymbol{z}_{i,1}$) is identified using Eq. (17):

$$\boldsymbol{z}_{i,1} \leftarrow \boldsymbol{p}_{i,m^*}; \quad m^* = \underset{j=1,2,\ldots,N_i}{\operatorname{argmax}}\left(D_{K_i}\left(\boldsymbol{p}_{i,j}\right)\right), \tag{17}$$

and $\boldsymbol{p}_{i,m^*}$ is removed from $\{\boldsymbol{p}\}_i$, namely, $\{\boldsymbol{p}\}_i \leftarrow \{\boldsymbol{p}\}_i \backslash \boldsymbol{p}_{i,m^*}$.

Then, the feature vector nearest to $\boldsymbol{z}_{i,1}$ is identified and removed from $\{\boldsymbol{p}\}_i$:

$$\boldsymbol{z}_{i,2} \leftarrow \boldsymbol{p}_{i,n^*}; \quad \boldsymbol{p}_{i,n^*} = \underset{\boldsymbol{q}\in\{\boldsymbol{p}\}_i}{\operatorname{argmin}}\left(\left\|\boldsymbol{z}_{i,1}-\boldsymbol{q}\right\|\right), \tag{18}$$

and, $\boldsymbol{z}_{i,3}$ can be identified by finding the feature vector that is closet to $\boldsymbol{z}_{i,2}$. By repeating the same process, one can obtain a rank-ordered sequence, namely, $\{\boldsymbol{z}_{i,1},\boldsymbol{z}_{i,2},\boldsymbol{z}_{i,3},\ldots,\boldsymbol{z}_{i,N_i}\}$.

Step 2. ~~The l~~Local maxima of ~~the~~ data density, denoted by $\{\boldsymbol{p}\}_i^t$ ($t = 1$) are identified from $\{\boldsymbol{z}_{i,1},\boldsymbol{z}_{i,2},\boldsymbol{z}_{i,3},\ldots,\boldsymbol{z}_{i,N_i}\}$ by **Condition 2**:

Condition 2:

$$If\left(D\left(\boldsymbol{z}_{i,j}\right)>D\left(\boldsymbol{z}_{i,j-1}\right)\right)And\left(D\left(\boldsymbol{z}_{i,j}\right)>D\left(\boldsymbol{z}_{i,j+1}\right)\right)\\ Then\left(\boldsymbol{z}_{i,j}\in\{\boldsymbol{p}\}_i^t\right) \tag{19}$$

Step 3. Images with the feature vectors that are the most similar to $\{\boldsymbol{p}\}_i^t$ are selected as the corresponding visual prototypes $\{\mathbf{P}\}_i^t$ at the t th layer of the i th hierarchy ($j=1,2,\ldots,N_i^t$; N_i^t is the cardinality of $\{\boldsymbol{p}\}_i^t$):

$$\mathbf{P}_{i,j}^t\leftarrow\mathbf{I}_{i,n^*};\quad n^*=\underset{k=1,2,\ldots,K_i}{\operatorname{argmin}}\left(\left\|\boldsymbol{p}_{i,j}^t-\boldsymbol{x}_{i,k}\right\|\right). \tag{20}$$

Step 4. Clusters $\mathbb{C}_{i,j}^t\left(j=1,2\ldots,N_i^t\right)$ are formed using $\{\boldsymbol{p}\}_i^t$ to partition the feature space by assigning the feature vector of each training image of the i th class based on the following principle ($k=1,2,\ldots,K_i$):

$$\mathbb{C}_{i,n^*}^t\leftarrow\mathbb{C}_{i,n^*}^t\cup\left\{\boldsymbol{x}_{i,k}\right\};\quad n^*=\underset{j=1,2\ldots,N_i^t}{\operatorname{argmin}}\left(\left\|\boldsymbol{p}_{i,j}^t-\boldsymbol{x}_{i,k}\right\|\right). \tag{21}$$

Then, the multimodal data density at the centre of each cluster is calculated as follows:

$$D^M\left(\boldsymbol{\rho}_{i,j}^t\right)=\frac{S_{i,j}^t}{1+\frac{\left\|\boldsymbol{\rho}_{i,j}^t-\boldsymbol{\rho}_i\right\|^2}{1-\left\|\boldsymbol{\rho}_i\right\|^2}}, \tag{22}$$

where $\boldsymbol{\rho}_{i,j}^t=\frac{1}{S_{i,j}^t}\sum_{\mathbf{P}\in\mathbb{C}_{i,n^*}}\boldsymbol{p}$; $S_{i,j}^t$ is the cardinality of $\mathbb{C}_{i,j}^t$

Step 5. The data-driven threshold, χ_i^t that defines the radius of neighbouring area around each cluster centre is derived from data by Eq. (23):

$$\eta_i^t=\frac{\sum_{p=1}^{N_i^t-1}\sum_{q=p+1}^{N_i^t}\left\|\boldsymbol{\rho}_{i,p}^t-\boldsymbol{\rho}_{i,q}^t\right\|}{N_i^t\left(N_i^t-1\right)};\\ \gamma_i^t=\frac{\sum_{\boldsymbol{x},\boldsymbol{y}\in\{\boldsymbol{\rho}\}_i^t,\boldsymbol{x}\neq\boldsymbol{y},\|\boldsymbol{x}-\boldsymbol{y}\|\leq\eta_i^t}\|\boldsymbol{x}-\boldsymbol{y}\|}{M_{\eta_i^t}};\\ \chi_i^t=\frac{\sum_{\boldsymbol{x},\boldsymbol{y}\in\{\boldsymbol{\rho}\}_i^t,\boldsymbol{x}\neq\boldsymbol{y},\|\boldsymbol{x}-\boldsymbol{y}\|\leq\gamma_i^t}\|\boldsymbol{x}-\boldsymbol{y}\|}{M_{\gamma_i^t}}, \tag{23}$$

where $\{\boldsymbol{\rho}\}_i^t=\left\{\boldsymbol{\rho}_{i,1}^t,\boldsymbol{\rho}_{i,2}^t,\ldots,\boldsymbol{\rho}_{i,N_i^t}^t\right\}$.

Then, for each cluster, $\mathbb{C}_{i,j}^t$, the collection of its neighbouring clusters, denoted by $\{\mathbb{C}\}_{i,j}^{n^*}$ ~~are~~ is identified by **Condition 3**:

Condition 3:

$$If\left(\left\|\boldsymbol{\rho}_{i,j}^t-\boldsymbol{\rho}_{i,k}^t\right\|\leq\chi_i^t\right)\quad Then\left(\mathbb{C}_{i,k}^t\in\{\mathbb{C}\}_{i,j}^{n^*}\right), \tag{24}$$

where $j,k=1,2,\ldots,N_i^t$; $j\neq k$.

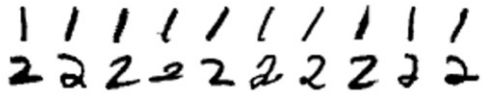
~~The lo~~Local maxima of multimodal data density are identified by **Condition 4**, which are denoted by $\{\boldsymbol{p}\}_i^{t+1}$:

Condition 4:

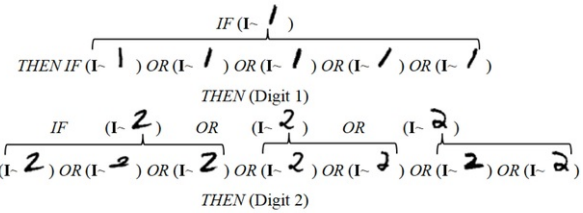
$$If\left(D^M\left(\mathbb{C}_{i,j}^t\right)>\max_{\mathbb{C}\in\{\mathbb{C}\}_{i,j}^{n^*}}\left(D^M(\mathbb{C})\right)\right)\quad Then\left(\boldsymbol{\rho}_{i,j}^t\in\{\boldsymbol{p}\}_i^{t+1}\right), \tag{25}$$

where $j=1,2,\ldots,N_i^t$. After $\{\boldsymbol{p}\}_i^{t+1}$ is identified, the algorithm goes back to **Step 3** and begins a new iteration ($t\leftarrow t+1$) until ~~the prototypes do not change any more~~the algorithm converges.

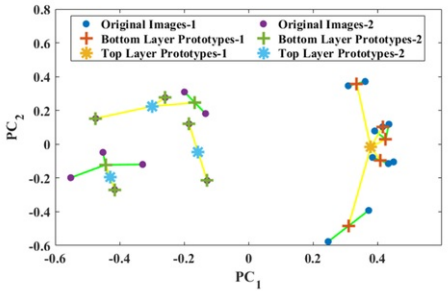
An illustrative example of ~~the~~ H-DRB ~~classifier~~= using MNIST images is given in Fig. 3. In this example, 10 handwritten images of digit 1 and 10 handwritten images of digit 2, respectively, are selected for training ~~the hierarchical deep rule-based classifier~~H-DRB (see Fig. 3(a)); the identified hierarchical IF...THEN rules from these images are given in Fig. 3(b). One may also ~~see~~notice the connections between the top and bottom layer prototypes. The hierarchical architecture is also visualized in a data space formed by the top two principle component analysis (PCA) scores; ~~the l~~Links between ~~the~~ original images and bottom layer prototypes are depicted by the lines in green, and ~~the~~ links between the top and bottom layer prototypes are given by the lines in yellow.



(a) Digits for training



(b) The identified hierarchical IF…THEN rules



(c) Visualization of the hierarchical architecture using PCA scores

Fig. 3 Illustrative example using MNIST images.

4 Numerical experiments and discussions

In this section, we consider a number of popularly studied problems to benchmark our approach. The algorithms were developed on MATLAB R2018a platform, numerical experiments were conducted on a desktop with dual core processor 3.60 GHz × 2 and 16 GB RAM.

4.1 Dataset descriptions and experimental settings

In this paper, eight benchmark datasets are used for numerical examples. ~~The k~~Key information of these datasets is ~~summarized~~given in Table 2. Detailed descriptions of the eight datasets can be found from [39–46].

For MNIST and Fashion MNIST datasets, different numbers of samples (5000, 10000, 20000, 30000, 40000, 50000 and 60000, respectively) ~~form~~from the training sets are randomly chosen for training, and the testing sets are used for classification performance evaluation. In the experiments with these two datasets, the pre-processing module of ~~the~~ H-DRB ~~classifier~~ has only one normalization layer that transforms the pixel value range of ~~the~~ images from $[0,255]$ into $[0,1]$. Here we employ the GIST feature descriptor [47] for feature extraction, which results in a 512 × 1 dimensional feature vector from each image. The architecture of ~~the~~ H-DRB ~~classifier~~ for the two datasets is given in Fig. 4(a).

Table 2 Key information of the eight benchmark datasets.

Dataset		# Images	# Class	# Images Per Class	# Pixels
MNIST [39]	Training set	60000	10	~ 6000	28 × 28
	Testing set	10000		~ 1000	

Fashion MNIST [40]	Training Set	60000		6000	
	Testing set	10000		1000	
RSSCN7 [42]		2800	7	400	400 × 400
Singapore [41]		1086	9	42–179	256 × 256
UCMerced [43]		2100	21	100	
WuHan-RS19 [44]		950	19	50	600 × 600
Caltech101 [45]		8677	101	40–800	$\sim 200 \times 300$
Caltech256 [46]		29780	256	80–800	

For RSSCN7, Singapore, UCMerced, WuHan-RS19, Caltech101 and Caltech256 datasets, we create an ensemble of pre-trained AlexNet [34] and VGG-VD-16 [35] DNN models for feature extraction. The ensemble descriptor will extract a highly discriminative representation from each training/testing image, $\mathbf{I}$ as:

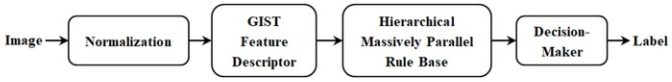
$$\boldsymbol{x} \leftarrow \mathrm{F}(\mathbf{I}) = \left[\frac{\mathrm{AN}(\mathbf{I})}{\|\mathrm{AN}(\mathbf{I})\|}, \frac{\mathrm{VN}(\mathbf{I})}{\|\mathrm{VN}(\mathbf{I})\|}\right]^T \tag{26}$$

where $\mathrm{F}(\mathbf{I})$ stands for the 9192 × 1 dimensional discriminative representation vector of $\mathbf{I}$; $\mathrm{AN}(\mathbf{I})$ and $\mathrm{VN}(\mathbf{I})$ are the 1 × 4096 dimensional activations obtained from the first fully connected layer of the two DNN models [48].

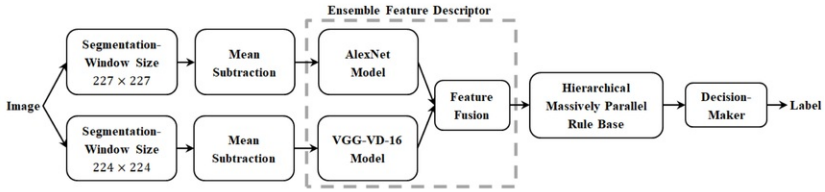
~~F~~Following the common practice, for Singapore dataset, 20% of the images per class are selected out to form the training set and the remaining ones are used for testing. For RSSCN7 dataset, 20% and 50% images per class are used for training, the remaining 80% and 50% are used for testing, respectively. For UCMerced dataset, we follow the 50:50 and 80:20 splits to build the training and testing sets, and the ratios are set as 40:60 and 60:40 for ~~the~~ WuHan-RS19 dataset. For these four benchmark datasets, the pre-processing module of ~~the~~ H-DRB ~~classifier~~ firstly crops five sub-images (centre and four corners [34,49]) from each input image according to the required sizes by the pre-trained DNNs and, then, we further create three new images from each sub-image by flipping it horizontally, vertically and in both directions. Thus, in total, 20 new sub-images are created from each remote sensing image. These sub-images are subtracted by respective their means and passed to the ensemble feature descriptor. Finally, the 9192 × 1 dimensional feature vector of the input image is obtained as the average of the discriminative representations of the 20 sub-images. The architecture for the four remote sensing problems is given in Fig. 4(b).

For Caltech101 dataset, 15 and 30 images per class are randomly selected out and used for training purpose. For Caltech256 dataset, we randomly ~~pick~~select ~~-~~out 15, 30, 45 and 60 images per class for training. The remaining images are used for testing. For these two datasets, the pre-processing module of H-DRB firstly resizes the images into the required sizes by the two DNNs ~~within~~of the ensemble descriptor, and then, performs mean subtraction before feature extraction. The architecture of H-DRB ~~classifier~~ for Caltech datasets is given in Fig. 4(c).

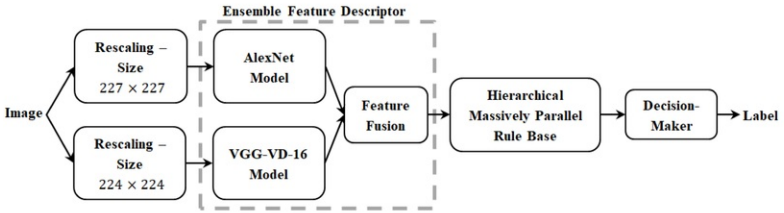
Under the same experimental protocols, we also involve ~~the~~ DRB, support vector machine (SVM) [50], k-nearest neighbour (KNN) [51] and semi-supervised deep rule-based (SSDRB) [52] ~~approaches~~ classifiers for comparison. In the numerical examples, DRB uses the same architecture as H-DRB and serves as the baseline. Both SVM and KNN are the most widely-used generic classifiers by the DNN-based approaches and they have demonstrated very attractive performance on various benchmark problems [53,54]. In this paper, SVM uses linear kernel function, and k is set to be 1 for KNN. ~~For fair comparison, both classifiers use the same inputs as the hierarchical massively parallel rule base of the H-DRB classifier during the training and validation stages. The~~ SSDRB ~~classifier~~ is introduced as a semi-supervised learning extension of DRB. In the numerical experiments, SSDRB also uses the same architecture as H-DRB for image processing. SSDRB follows the offline semi-supervised learning strategy, and the user-controlled parameter φ is set to be 1.2 following [52]. Furthermore, regarding the performance evaluation, we will also report the state-of-the-art results from the ~~existing publications~~ literature for better informed comparison. As mentioned in Section 3, only the top layer of ~~the~~ H-DRB ~~classifier~~ is used for classification unless specifically declared otherwise. The reported numbers of identified prototypes of ~~the~~ H-DRB ~~classifier~~ are calculated from the top layer prototypes by default because prototypes of the bottom layer are no longer involved in decision-making. Note that if H-DRB performs classification with its bottom layer prototypes, the results will be exactly the same as DRB



(a) The architecture for MNIST and Fashion MNIST datasets



(b) The architecture for remote sensing datasets



(c) The architecture for Caltech datasets

Fig. 4 Architectures of ~~the~~ H-DRB ~~classifier~~ used for different benchmark problems.

4.2 Numerical results

The first case we consider ar~~e the~~ MNIST [39] and Fashion MNIST [40] ~~datasets~~. ~~The p~~Performance of ~~the~~ H-DRB ~~classifier~~ in terms of average number of prototypes per class, classification accuracy on testing sets, training and testing time consumption (both in seconds) on the two datasets are tabulated in Table 3, respectively. ~~The p~~Performance of ~~the~~ DRB ~~classifier~~ is also reported as the baseline.

As one can see from Table 3, ~~the~~ DRB ~~classifier~~ has a very large number of prototypes and its classification performance is slightly better than ~~the~~ H-DRB ~~classifier~~. ~~However,~~In contrast, ~~the~~ H-DRB ~~classifier~~ has much less top-layer prototypes in each rule (80% less) but is still able to achieve ~~high~~good classification accuracy rates (in the worst case, 4% lower than DRB). Note that ~~the~~ H-DRB ~~classifier~~ takes, in average, less than 0.02 s to learn from a single image during the training process. During the validation process, as one can see from the same table, DRB takes less than 0.15 s to determine the class label for a testing image, which is already quite efficient. The validation process of H-DRB is even more efficient, which only takes less than 0.02 s for each ~~unlabeled~~ unlabelled image. This only comes with the price of slightly more training time needed for self-organizing its hierarchical structure. Moreover, both, the training and validation processes can be conducted in parallel for each IF...THEN rule within the hierarchical massively parallel rule base of ~~the~~ H-DRB ~~classifier~~ and, thus, the computational efficiency can be further speeded up, in this case, 10 times faster. ~~The p~~Performance (in terms of classification accuracy, %) of ~~the~~ H-DRB ~~classifier~~ is also compared with SVM, KNN, SSDRB and other alternatives, and the comparison results are tabulated in Table 4.

Table 3 Performance of H-DRB and DRB classifiers on MNIST and Fashion MNIST datasets.

Dataset	# Training images	Algorithm	# Prototypes per class	Accuracy,%	Training time, s	Testing time, s
MNIST	5000	H-DRB	45.8	96.3	3.4	16.3
		DRB	237.5	97.1	2.5	132.0
	10000	H-DRB	88.4	97.0	14.5	32.8
		DRB	463.3	97.7	12.4	270.6
	20000	H-DRB	139.2	97.6	80.6	102.2
		DRB	907.4	98.2	73.1	552.3
	30000	H-DRB	242.1	97.9	201.6	147.3

		DRB	1339.5	98.4	182.7	782.1
	40000	H-DRB	310.7	98.1	336.8	180.6
		DRB	1772.0	98.5	302.0	973.5
	50000	H-DRB	377.1	98.2	518.0	216.7
		DRB	2202.9	98.6	460.9	1217.0
	60000	H-DRB	437.4	98.3	749.8	240.0
		DRB	2631.0	98.6	662.5	1407.9
Fashion MNIST	5000	H-DRB	39.9	83.8	3.4	15.3
		DRB	230.9	84.9	2.6	131.0
	10000	H-DRB	76.5	85.1	12.6	24.2
		DRB	457.8	86.0	10.3	224.4
	20000	H-DRB	139.2	85.7	81.1	69.7
		DRB	907.4	87.0	71.0	517.8
	30000	H-DRB	182.7	85.8	202.1	108.8
		DRB	1353.9	87.6	171.6	757.5
	40000	H-DRB	208.4	85.4	384.7	127.2
		DRB	1792.9	88.0	312.4	1009.5
	50000	H-DRB	222.7	84.9	577.5	132.6
		DRB	2229.6	88.4	470.6	1219.9
	60000	H-DRB	232.7	84.9	836.9	135.7
		DRB	2662.5	88.6	673.4	1422.4

~~The~~ H-DRB ~~classifier~~ is further compared with the selected results from literature in terms of classification accuracy (%) and computational efficiency (of the training process), and the comparison is ~~given~~ tabulated in Table 5. Note that many alternative approaches reported in this table, such as [57,58], actually use GPU for computation.

Table 4 Performance comparison on MNIST and Fashion MNIST datasets with partial training set (the best result is in bold).

Dataset	# Training images	H-DRB	DRB	SVM	KNN	SSDRB	eClass1 [55]	TEDA Class [56]
MNIST	5000	96.3	97.1	97.3	96.8	**97.4**	96.9	97.2
	10000	97.0	97.7	**97.9**	97.5	97.8	97.2	97.4
	20000	97.6	**98.2**	**98.2**	98.0	**98.2**	97.3	97.5
	30000	97.9	**98.4**	**98.4**	98.2	**98.4**	97.5	97.7
	40000	98.1	**98.5**	**98.5**	98.3	**98.5**	97.5	97.7
	50000	98.2	**98.6**	98.5	98.3	**98.6**	97.5	97.7
	60000	98.3	**98.6**	**98.6**	98.4	**98.6**	97.5	97.6
Fashion MNIST	5000	83.8	84.9	**86.5**	84.9	85.0	–	–

	10000	85.1	86.0	**87.7**	86.2	86.0	–	–
	20000	85.7	87.0	**88.6**	87.1	87.1	–	–
	30000	85.8	87.6	**88.9**	87.7	87.7	–	–
	40000	85.4	88.0	**89.2**	88.1	88.0	–	–
	50000	84.9	88.4	**89.5**	88.4	88.4	–	–
	60000	84.9	88.6	**89.4**	88.7	88.6	–	–

As one can see from Tables 4 and 5, the proposed H-DRB ~~classifier~~ is able to outperform or, at least, on par with various comparative approaches in terms of classification accuracy. Moreover, its computational efficiency is on the same level of SVM ~~classifiers~~. However, one need to notice that the training process of ~~the~~ SVM ~~classifiers~~ is not parallelizable and limited to offline. Meanwhile, the massively parallel IF...THEN rules of ~~the~~ H-DRB ~~classifier~~ can be trained independently and updated with new data samples recursively.

Table 5 Performance comparison on MNIST and Fashion MNIST datasets with the literature (the best result is in bold).

Algorithm	MNIST		Fashion MNIST	
	Accuracy, %	Training time	Accuracy, %	Training time
H-DRB	98.3	12 min, 29 s	84.9	13 min, 57 s
DRB	98.6	11 min, 2 s	88.6	11 min, 13 s
Logistic Regression [40]	90.9	26 h, 10 min, 12 s	83.9	2 h, 59 min, 26 s
Decision tree classifier [40] Maximum split: 50	88.6	2 min, 14 s	78.9	**36 s**
Passive Aggressive Classifier	88.0	**29 s**	77.3	42 s
Multiple Layer Perception [40] ~~Actitation~~ Activation function: RELU Hidden layer size: 100	97.2	6 min, 55 s	87.1	16 min, 3 s
SVM [40] Polynomial kernel	97.6	1 h, 15 min, 29 s	89.7	1 h, 12 min, 39 s
SVM [40] Gaussian kernel	97.3	48 min, 32 s	**89.7**	1 h, 15 min, 25 s
Committee of 7 DCNNs [57]	99.7	98 h (14 h per DCNN)	–	–
Committee of 35 DCNNs [58]	99.8	490 h (14 h per DCNN)	–	–
Invariant Feature Hierarchies [59]	99.4	No Information	–	–
Two-Stage Predictive Sparse Decomposition [59]	99.5		–	–
DRB Ensemble with GIST Feature [23]	99.3	~ 2 h, 30 min	–	–
DRB Ensemble with HOG Feature [23]	98.9		–	–
DRB Ensemble with Combined GIST and HOC Features [23]	99.3		–	–
Committee of DRB Ensembles with HOG Feature and with GIST Feature [23]	99.3	~ 5 h	–	–
DRB Ensemble Cascade [23]	**99.6**		–	–

In the second case, we consider ~~the~~ RSSCN7 [42], Singapore [41], UCMerced [43] and WuHan-RS19 [44] datasets. ~~The c~~Classification accuracy rates of ~~the~~ H-DRB ~~classifier~~ on the four benchmark datasets are reported in Table 6. ~~The a~~Average numbers of prototypes identified per class of ~~the~~ H-DRB and DRB ~~classifiers~~ are given in Fig. 5. Similarly, we also compare the performance of ~~the~~ H-DRB ~~classifier~~ with the state-of-the-art approaches reported in literature.

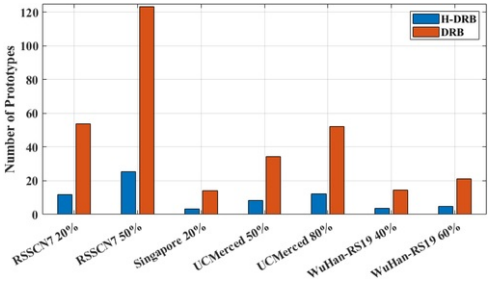


Fig. 5 Average numbers of prototypes identified per class from the remote sensing datasets. The blue bars correspond to ~~the~~ H-DRB ~~classifier~~ and the orange bars correspond to ~~the H~~ DRB ~~classifier~~.

Table 6 Performance comparison on remote sensing datasets.

Algorithm	RSSCN7		Singapore	UCMerced		Wuhan-RS19	
	20% Training	50% Training		50% Training	80% Training	40% Training	60% Training
H-DRB	86.2 ± 0.9	89.8 ± 0.7	97.1 ± 0.6	91.8 ± 0.7	95.5 ± 1.1	93.1 ± 1.1	94.1 ± 0.9
DRB	87.2 ± 0.8	90.8 ± 0.5	97.3 ± 0.4	92.9 ± 0.3	96.7 ± 1.3	93.3 ± 0.8	94.7 ± 0.4
SVM	**88.7 ± 0.9**	91.1 ± 0.5	97.7 ± 0.6	93.3 ± 0.4	96.8 ± 1.3	94.9 ± 1.1	95.9 ± 0.8
KNN	87.6 ± 0.8	**91.4 ± 0.5**	97.7 ± 0.4	94.4 ± 0.6	97.0 ± 0.9	93.6 ± 0.8	95.3 ± 0.7
SSDRB	88.0 ± 0.6	91.1 ± 0.5	**97.8 ± 0.5**	94.1 ± 0.6	97.8 ± 1.2	93.4 ± 1.0	94.9 ± 0.4
Gan et al. [41]	–	–	90.9	–	91.1	–	–
Yang & Newsam [43], [60]	76.3 ± 0.9	81.3 ± 0.6	–	71.9 ± 0.8	74.1 ± 3.3	75.3 ± 1.4	80.1 ± 2.0
Hu et al. [49]	–	–	–	–	98.5	–	98.9
Xia et al. [60]	85.6 ± 1.0	88.9 ± 0.6		94.1 ± 1.0	95.2 ± 1.2	95.1 ± 1.2	96.2 ± 0.6
Wu et al. [61]	–	90.4 ± 0.6	–	–	92.7 ± 0.8	–	–
Wu et al. [62]	–	86.4 ± 0.7	–	–	91.8 ± 1.3	–	–
Zhao et al. [63]	–	89.1	–	–	97.8	–	–
Lazebnik et al. [60,64]	68.9 ± 0.7	72.9 ± 0.9	–	58.3 ± 1.9	62.4 ± 1.9	54.4 ± 2.2	58.5 ± 2.3
Jegou et al. [60,65]	77.3 ± 0.6	82.3 ± 1.2	–	73.2 ± 1.0	78.2 ± 1.7	76.4 ± 2.0	80.8 ± 2.2
Bian et al. [66]	–	–	–	94.2 ± 1.0	95.8 ± 1.0	95.4 ± 0.8	96.4 ± 0.8
Huang et al. [67]	–	–	–	–	93.0 ± 1.2	–	94.3 ± 1.0
Chen et al. [68]	–	–	–	–	90.0 ± 2.1	–	91.0 ± 1.5
Qi et al. [69]	–	–	–	–	91.1 ± 0.7	–	91.7 ± 1.1
Nogueira et al. [70]	–	–	–	–	**99.5 ± 0.5**	–	94.5 ± 1.2
Chaib et al. [71]	–	–	–	–	97.4 ± 1.8	–	98.7 ± 0.2
Wang et al. [72]	–	–	–	**96.8 ± 0.1**	99.1 ± 0.4	**97.5 ± 0.5**	**99.8 ± 0.3**

Shao et al. [73]	–	–	–	–	92.4 ± 0.6	–	94.5 ± 1.0

For better demonstration, we visualize in Fig. 6 the ~~identified~~ prototypes of the top and bottom layers of ~~the~~ H-DRB ~~classifier~~ from ~~the~~ training images of four different classes ~~at~~ identified during one particular experiment using WuHan-RS19 dataset. In this experiment, 40% of ~~the~~ images per class are used for training and the rest are used for testing. The four classes used for visualization are commercial, forest, railway station and residential.

It is well-known that the performance of an image classification algorithm is subject to the image processing and augmentation techniques involved. Therefore, in the following numerical example, we compare the performance of ~~the~~ H-DRB and DRB ~~classifiers~~ with five different image processing architectures on Singapore, UCMerced and WuHan-RS19 datasets. For clarity, the architecture as presented in Fig. 4(a) is re-denoted as Arch. 1. For Arch. 2, the same architecture as Arch. 1 is employed but only the pre-trained VGG-VD-16 model is used for feature extraction. For Arch. 3, ~~the~~ pre-trained CaffeNet [74] is further involved to form the ensemble feature descriptor on the basis of Arch. 1. The architecture used in [75] is involved as Arch. 4, where each input image is segmented to five sub-images (centre and four corners) and the VGG-VD-16 model is used for feature extraction. The architecture used in [76] is involved in the experiment as Arch. 5, where AlexNet ,VGG-VD-16 and CaffeNet are used for creating the ensemble feature descriptor and each input image is segmented to 20 sub-images same as ~~setting~~Arch. 3. However, the operating mechanism of ~~the~~ H-DRB ~~classifiers~~ with Arch. 4 and 5 are different from Arch. 1–3 in the sense that the classifiers are trained and tested with the feature vectors of sub-images instead. Therefore, Arch. 4 and Arch. 5 can better utilize the local semantic information for classification. ~~The p~~Performance comparison between different architectures ~~are~~ is given in Table 7 in terms of classification accuracy (%). ~~The a~~Average training and testing time consumptions (in seconds) of ~~the~~ H-DRB and DRB ~~classifiers~~ with the respective five architectures across the numerical experiments are reported in the same table.

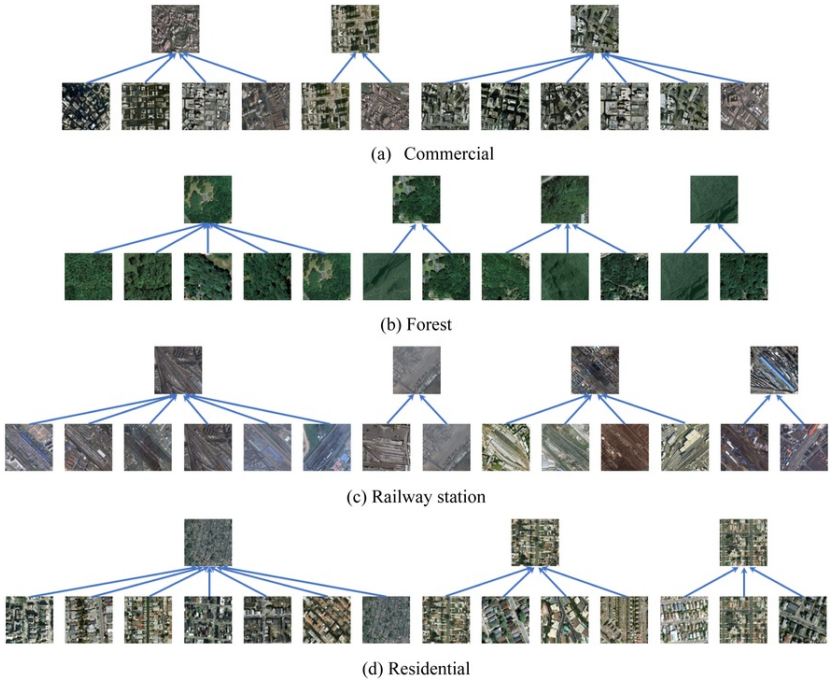


Fig. 6 Example of identified prototypes at the bottom and top layers of ~~the~~ H-DRB ~~classifier~~.

By comparing between Arch. 1–3, one can tell from Table 7 that ~~the~~ both, H-DRB and DRB ~~classifiers~~ perform better if the employed feature descriptor is more powerful. However, the computational complexity will significantly increase if the dimensionality of ~~the~~ feature vectors is too high (see Arch. 3). If the classifier is trained with ~~the~~ segments of images instead (see Arch. 4 and 5), it can achieve even higher accuracy at the price of lower computational efficiency.

In the final case, we consider the well-known Caltech 101 [45] and Caltech256 [46] datasets. The classification accuracy rates of ~~the~~ H-DRB ~~classifier~~ and the comparative alternatives on the two datasets are reported in Table 8. ~~The a~~Average numbers of prototypes that ~~the~~ H-DRB ~~classifier~~ identifies form the training images of each class are given in Fig. 7 and compared with ~~the~~ DRB ~~classifier~~.

Table 7 Performance comparison of H-DRB and DRB with different architectures.

Algorithm		Accuracy, %					Time Consumption, s	
		Singapore	UCMerced		WuHan-RS19		Training	Testing
			50%	80%	40%	60%		
Arch. 1	H-DRB	97.1	91.8	95.5	93.1	94.1	1.8	2.5
	DRB	97.3	92.9	96.7	93.3	94.7	1.5	12.9
Arch. 2	H-DRB	92.4	88.9	93.3	90.0	90.8	0.9	1.6
	DRB	95.6	91.6	96.1	91.4	93.0	0.7	6.6
Arch. 3	H-DRB	96.5	92.8	96.5	93.7	94.7	3.5	4.9
	DRB	96.7	93.7	97.4	94.2	95.1	2.8	22.6
Arch. 4	H-DRB	97.3	90.2	94.5	93.7	94.4	10.0	21.3
	DRB	97.8	90.9	95.8	93.0	93.5	9.8	97.8
Arch. 5	H-DRB	97.8	93.6	97.4	94.9	96.2	653.0	1743.9
	DRB	97.8	94.0	97.3	95.2	96.3	617.0	7424.3

Interestingly, ~~as~~ one can observe from Table 8~~, the~~ that H-DRB performs much better ~~for classification~~ than DRB. This is because images between different classes of the Caltech datasets are more distinctive from each other and, thus, ~~using a small amount of highly descriptive prototypes~~H-DRB can achieve very good classification accuracy using only a smaller amount of highly descriptive prototypes. In ~~this case~~contrast, ~~the~~ DRB ~~classifier will~~ performs worse due to ~~the~~ overfitting.

Table 8 Numerical results on ~~Caltech101 dataset~~Caltech datasets.

Algorithm	Caltech101		Caltech256			
	15	30	15	30	45	60
H-DRB	86.7 ± 0.7	89.5 ± 0.6	**64.7 ± 0.4**	**68.9 ± 0.3**	**71.2 ± 0.3**	73.1 ± 0.3
DRB	84.9 ± 0.6	88.6 ± 0.5	62.4 ± 0.3	67.1 ± 0.3	69.8 ± 0.3	71.9 ± 0.3
SVM	**87.3 ± 1.0**	**90.3 ± 0.9**	Out of System Memory			
KNN	86.7 ± 0.6	90.0 ± 0.5	62.5 ± 0.3	67.2 ± 0.3	69.9 ± 0.3	72.1 ± 0.3
SSDRB	85.5 ± 0.9	89.2 ± 0.7	64.1 ± 0.4	68.0 ± 0.2	70.4 ± 0.3	72.3 ± 0.3
Gao et al. [5]	71.3 ± 0.6	77.6 ± 1.0	35.1 ± 0.4	42.1 ± 0.3	46.0 ± 0.3	48.5 ± 0.3
Xie et al. [77]	76.0	82.5	36.4	45.1	48.0	50.3
Li et al. [78]	–	89.2	–	–	–	**74.9**
Wang et al. [79]	64.0 ± 0.4	71.4 ± 1.2	–	–	–	–
Yang et al. [80]	67.0 ± 0.5	73.2 ± 0.5	27.7 ± 0.5	34.0 ± 0.4	37.5 ± 0.6	40.1 ± 0.9
Saban et al. [81]	68.5	75.0	–	–	–	–
Pan et al. [82]	77.2 ± 0.6	85.8 ± 0.4	36.6 ± 0.6	47.2 ± 0.7	50.8 ± 0.4	52.9 ± 0.5
Zhang et al. [83]	–	–	61.5 ± 0.4	67.7 ± 0.7	69.8 ± 0.5	72.8 ± 0.4

Zhang et al. [84]	–	–	47.6 ± 0.6	55.4 ± 0.6	59.1 ± 0.5	61.7 ± 0.5

4.3 Discussions

Based on the numerical examples presented in this section, the following four remarks are worth noting:

Firstly, it is very interesting to ~~be noticed~~notice that H-DRB can achieve similar or, sometimes, even better classification accuracy rates on various benchmark ~~datasets various~~ problems with a much smaller number of top-layer prototypes (around 80% less) compared with DRB. This raises the question that what is the minimum number of prototypes required by ~~the~~a classifier to achieve the top-level classification performance. This needs to be further studied in order to fully understand the advantages and limitations of prototype-based classifiers.

Secondly, both H-DRB and DRB identify prototypes from training images in a more straightforward manner without any iterative optimization process. Thus, they are outperformed by SVM and KNN in some cases. Since the optimality of prototypes determine the performance of ~~the~~ prototype-based approaches [48], one may need to introduce a prototype optimization mechanism to H-DRB and DRB to maximize their strength.

Thirdly, the performance (accuracy and efficiency) of both H-DRB and DRB is highly subject to their architecture for image processing (see the comparison between five architectures in Table 7). Generally, both approaches perform more accurate classification on testing images if more discriminative representations are extracted from ~~images~~each image and more local information is exploited. ~~On the other hand~~At the same time, their computational efficiency ~~is~~may decrease~~d~~ as a trade-off because of the increase of the dimensionality of feature vectors. Therefore, it is worth exploring alternative ways to fuse the feature vectors extracted by different feature descriptors. In addition, i~~I~~n this paper, only ~~the~~ standard image processing techniques are employed, and the architecture of the classifier has to be designed specifically for each type of problems. It will be a strong novelty to incorporate a more generic, self-adaptive image processing architecture for ~~the proposed~~ H-DRB ~~classifier~~, which is applicable to different types of problems.

Finally, in this paper, both H-DRB and DRB employ standard pre-trained DNNs for feature extraction. As a result, they are outperformed by some of the latest state-of-the-art DNN-based approaches which involve more sophisticated fine tuning and feature selection, such as [70,72]. Therefore, another interesting direction for future work is to involve the fine-tuned DNNs as feature descriptors for the proposed approach, and further employ the state-of-the-art feature selection techniques for dimensionality reduction.

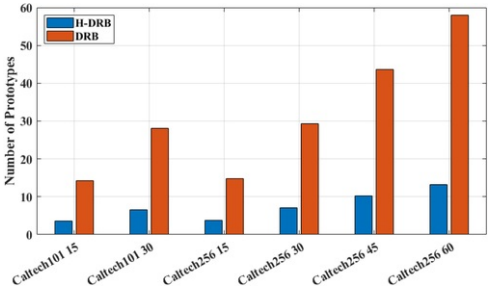


Fig. 7 Average numbers of prototypes identified per class from the Caltech datasets. The blue bars correspond to ~~the~~ H-DRB ~~classifier~~ and the orange bars correspond to ~~the H-~~DRB ~~classifier~~.

5 Conclusion

In this paper, a novel classification approach, called H-DRB, is presented for large-scale multi-class image classification problems. ~~The k~~Key feature that sets H-DRB apart from alternative zero-order FRB systems is its zero-order massively parallel IF…THEN rules with multi-layer premise parts ~~from training images~~ self-organized from ~~data~~training images through an autonomous, non-parametric learning process. Numerical results presented in this paper demonstrated that H-DRB can achieve similar or even better classification accuracy than DRB on various benchmark datasets with a much smaller number of top-layer prototypes (80% less) and much higher computational efficiency (5–10 times faster) for decision-making. The multi-layer premise parts also bring ~~the~~ H-DRB ~~approach~~ the advantage of being highly transparent and human-interpretable for large-scale, complex classification problems.

Nonetheless, we have to admit that the main focus of this paper is to demonstrate the proposed concept and general principles. Only ~~the~~ basic data pre-processing and augmentation techniques were employed by the proposed approach, and the pre-trained DNN models ~~employed~~used for feature extraction were not fine-tuned ~~to improve their descriptive abilities on the particular problems used for experimental demonstration~~. Therefore, there is a large room for performance improvement of ~~the~~ H-DRB ~~classifier~~. As future works, there are a few considerations. Firstly, we will analyse the optimality of the identified prototypes and explore the general principle for determin~~e~~ing the best number of prototypes needed ~~by the proposed approach~~ to achieve the best balance between classification accuracy and computational efficiency. We will also involve fine-tuned DNNs for feature extraction and more advanced image

pre-processing and augmentation techniques to compete for the best performance, and experiment with self-adaptive architectures suitable for different types of problems. Another interesting direction for future work is to apply the proposed approach to video classification by further taking the time and space correlation into consideration.

Declaration of Competing Interest

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

References

[1] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *Nature Methods* **13** (1), 2015, 35.

[2] V. Mnih, et al., Human-level control through deep reinforcement learning, *Nature* **518** (7540), 2015, 529–533.

[3] R. Shwartz-Ziv and N. Tishby, Opening the black box of deep neural networks via information, 2017, arXiv Prepr. arXiv:1703.00810.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[5] S. Gao, L. Duan and I.W. Tsang, DEFEATnet—a deep conventional image representation for image classification, *IEEE Trans. Circuits Syst. Video Technol.* **26** (3), 2016, 494–505.

[6] G. Hinton, et al., Deep neural networks for acoustic modeling in speech recognition, *IEEE Signal Process. Mag.* **29**, 2012, 82–97.

[7] D. Amodei, et al. Deep speech 2: End-to-end speech recognition in english and mandarin, in: International conference on machine learning, 2016, pp. 173–182.

[8] Y. Du, W. Wang, L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, pp. 1110–1118.

[9] Y. Li, H. Zhang, X. Xue, Y. Jiang and Q. Shen, Deep learning for remote sensing image classification: A survey, *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **e1264**, 2018.

[10] J.M. Garibaldi, The need for fuzzy AI, *IEEE/CAA J. Autom. Sin.* **6** (3), 2019, 610–622.

[11] H. Hagras, Toward human-understandable, explainable AI, *Computer (Long. Beach. Calif).* **51** (9), 2018, 28–36.

[12] X. Ding, Y. Zhang, T. Liu, J. Duan, Deep learning for event-driven stock prediction, in: International Joint Conference on Artificial Intelligence, 2015, pp. 2327–2333.

[13] C. Chen, A. Seff, A. Kornhauser, J. Xiao, DeepDriving: Learning affordance for direct perception in autonomous driving, in: IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2722–2730.

[14] J. Su, D.V. Vargas and K. Sakurai, One pixel attack for fooling deep neural networks, *IEEE Trans. Evol. Comput.* 2019, https://doi.org/10.1109/TEVC.2019.2890858.

[15] P.P. Angelov and X. Gu, Toward anthropomorphic machine learning, *IEEE Comput. Mag.* **51** (9), 2018, 18–27.

[16] M. Pratama, J. Lu, E. Lughofer, G. Zhang and M.J. Er, An incremental learning of concept drifts using evolving type-2 recurrent fuzzy neural networks, *IEEE Trans. Fuzzy Syst.* **25** (5), 2017, 1175–1192.

[17] N.K. Kasabov and Q. Song, DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction, *IEEE Trans. Fuzzy Syst.* **10** (2), 2002, 144–154.

[18] H.J. Rong, N. Sundararajan, G. Bin Huang and P. Saratchandran, Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction, *Fuzzy Sets and Systems* **157** (9), 2006, 1260–1275

[19] M. Pratama, S.G. Anavatti, P.P. Angelov and E. Lughofer, PANFIS: A novel incremental learning machine, *IEEE Trans. Neural Netw. Learn. Syst.* **25** (1), 2014, 55–68.

[20] J. Soto, P. Melin and O. Castillo, A new approach for time series prediction using ensembles of IT2FNN models with optimization of fuzzy integrators, *Int. J. Fuzzy Syst.* **20** (3), 2018, 701–728.

[21] I. Ben Ali, M. Turki, J. Belhadj and X. Roboam, Optimized fuzzy rule-based energy management for a battery-less PV/wind-BWRO desalination system, *Energy* **159**, 2018, 216–228.

[22] I. Škrjanc, J. Iglesias, A. Sanchis, D. Leite, E. Lughofer and F. Gomide, Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey, *Inf. Sci. (Ny).* **490**, 2019, 344–368.

[23] P.P. Angelov and X. Gu, Deep rule-based classifier with human-level performance and characteristics, *Inf. Sci. (Ny).* **463–464**, 2018, 196–213.

[24] L. Perez and J. Wang, The effectiveness of data augmentation in image classification using deep learning, 2017, arXiv Prepr. arXiv:1712.

[25] J. Fan, Y. Gao and H. Luo, Integrating concept ontology and multitask learning to achieve more effective classifier training for multilevel image annotation, *IEEE Trans. Image Process.* **17** (3), 2008, 407–426.

[26] M. Marszałek, C. Schmid, Constructing category hierarchies for visual recognition, in: European Conference on Computer Vision, 2008, pp. 479–491.

[27] J. Sivic, B.C. Russell, A. Zisserman, W.T. Freeman, A.A. Efros, Unsupervised discovery of visual object class hierarchies, in: 26th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR, 2008.

[28] J. Fan, et al., HD-MTL: Hierarchical deep multi-task learning for large-scale visual recognition, *IEEE Trans. Image Process.* **26** (4), 2017, 1923–1938.

[29] Y. Qu, et al., Joint hierarchical category structure learning and large-scale image classification, *IEEE Trans. Image Process.* **26** (9), 2017, 4331–4346.

[30] T. Zhao, et al., Embedding visual hierarchy with deep networks for large-scale visual recognition, *IEEE Trans. Image Process.* **27** (10), 2018, 4740–4755.

[31] J. Fan, N. Zhou, J. Peng and L. Gao, Hierarchical learning of tree classifiers for large-scale plant species identification, *IEEE Trans. Image Process.* **24** (11), 2015, 4172–4184.

[32] X. Gu and W. Ding, A hierarchical prototype-based approach for classification, *Inf. Sci. (Ny).* **505**, 2019, 325–351.

[33] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vis.* **60** (2), 2004, 91–110.

[34] A. Krizhevsky, I. Sutskever and G.E. Hinton, Imagenet classification with deep convolutional neural networks, In: *Advances in Neural Information Processing Systems,* 2012, 1097–1105.

[35] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015, pp. 1–14.

[36] C. Szegedy, et al., Going deeper with convolutions, In: *IEEE Conference on Computer Vision and Pattern Recognition,* 2015, 1–9.

[37] C.C. Aggarwal and C.K. Reddy, (Eds.), *Data Clustering: Algorithms and Applications,* 2013, CRC press.

[38] X. Gu, P.P. Angelov and J.C. Principe, A method for autonomous data partitioning, *Inf. Sci. (Ny)* **460–461**, 2018, 65–82.

[39] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE* **86** (11), 1998, 2278–2323.

[40] H. Xiao, K. Rasul and R. Vollgraf, Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms, 2017, arXiv Prepr. arXiv:1708.07747.

[41] J. Gan, Q. Li, Z. Zhang and J. Wang, Two-level feature representation for aerial scene classification, *IEEE Geosci. Remote Sens. Lett.* **13** (11), 2016, 1626–1630.

[42] Q. Zou, L. Ni, T. Zhang and Q. Wang, Deep learning based feature selection for remote sensing scene classification, *IEEE Geosci. Remote Sens. Lett.* **12** (11), 2015, 2321–2325.

[43] Y. Yang, S. Newsam, Bag-of-visual-words and spatial extensions for land-use classification, in: International Conference on Advances in Geographic Information Systems, 2010, pp. 270–279.

[44] G. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, H. Maitre, Structural high-resolution satellite image indexing, in: ISPRS, TC VII Symposium Part A: 100 Years ISPRS—Advancing Remote Sensing Science, 2010, pp. 298–303.

[45] L. Fei-Fei, R. Fergus and P. Perona, Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories, *Comput. Vis. Image Underst.* **106** (1), 2007, 59–70.

[46] G. Griffin, A. Holub and P. Perona, Caltech-256 Object Category Dataset, 2017.

[47] A. Oliva and A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, *Int. J. Comput. Vis.* **42** (3), 2001, 145–175.

[48] X. Gu, P. Angelov and H.J. Rong, Local optimality of self-organising neuro-fuzzy inference systems, *Inf. Sci. (Ny).* **503**, 2019, 351–380.

[49] F. Hu, G.S. Xia, J. Hu and L. Zhang, Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery, *Remote Sens.* **7** (11), 2015, 14680–14707.

[50] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, 2000, Cambridge University Press; Cambridge.

[51] P. Cunningham and S.J. Delany, K-nearest neighbour classifiers, *Mult. Classif. Syst.* **34**, 2007, 1–17.

[52] X. Gu and P.P. Angelov, Semi-supervised deep rule-based approach for image classification, *Appl. Soft Comput.* **68**, 2018, 53–68.

[53] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European Conference on Computer Vsion, 2014, pp. 818–833.

[54] A.B. Sargano, X. Wang, P. Angelov, Z. Habib, Human Action Recognition using Transfer Learning with Deep Representations, in: IEEE International Joint Conference on Neural Networks (IJCNN), 2017, pp. 463–469.

[55] P. Angelov and X. Zhou, Evolving fuzzy-rule based classifiers from data streams, *IEEE Trans. Fuzzy Syst.* **16** (6), 2008, 1462–1474.

[56] D. Kangin, P. Angelov and J.A. Iglesias, Autonomously evolving classifier TEDAClass, *Inf. Sci. (Ny).* **366**, 2016, 1–11.

[57] D.C. Cireşan, U. Meier, L.M. Gambardella, J. Schmidhuber, Convolutional neural network committees for handwritten character classification, in: International Conference on Document Analysis and Recognition, vol.10, 2011, pp. 1135–1139.

[58] D. Ciresan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: International Conference on Computer Vision and Pattern Recognition, 2012, pp. 3642–3649.

[59] K. Jarrett, K. Kavukcuoglu, M. Ranzato, Y. LeCun, What is the best multi-stage architecture for object recognition? in: IEEE International Conference on Computer Vision, 2009, pp. 2146–2153.

[60] G. Xia, et al., AID: A benchmark dataset for performance evaluation of aerial scene classification, *IEEE Trans. Geosci. Remote Sens.* **55** (7), 2017, 3965–3981.

[61] H. Wu, B. Liu, W. Su, W. Zhang and J. Sun, Deep filter banks for land-use, *IEEE Geosci. Remote Sens. Lett.* **13** (12), 2016, 1895–1899.

[62] H. Wu, B. Liu, W. Su, W. Zhang and J. Sun, Hierarchical coding vectors for scene level land-use classification, *Remote Sens.* **8** (5), 2016, 436.

[63] F. Zhao, X. Mu, Z. Yang and Z. Yi, A novel two-stage scene classification model based on feature variable significance in high- resolution remote sensing, *Geocarto Int.* 2019, 1–12.

[64] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 2169–2178.

[65] H. Jégou, et al. Aggregating local descriptors into a compact image representation, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 3304–3311.

[66] X. Bian, C. Chen, L. Tian and Q. Du, Fusing local and global features for high-resolution scene classification, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **10** (6), 2017, 2889–2901.

[67] L. Huang, C. Chen, W. Li and Q. Du, Remote sensing image scene classification using multi-scale completed local binary patterns and fisher vectors, *Remote Sens.* **8** (6), 2016, 1–17.

[68] C. Chen, L. Zhou, J. Guo, W. Li, H. Su, F. Guo, Gabor-filtering-based completed local binary patterns for land-use scene classification, in: IEEE international conference on multimedia big data, 2015, pp. 324–329.

[69] K. Qi, W. Liu, C. Yang, Q. Guan and H. Wu, Multi-task joint sparse and low-rank representation for the scene classification of high-resolution remote sensing image, *Remote Sens.* **9** (1), 2017, 10.

[70] K. Nogueira, O.A.B. Penatti and J.A. dos Santos, Towards better exploiting convolutional neural networks for remote sensing scene classification, *Pattern Recognit.* **61**, 2017, 539–556.

[71] S. Chaib, H. Liu, Y. Gu and H. Yao, Deep feature fusion for VHR remote sensing scene classification, *IEEE Trans. Geosci. Remote Sens.* **55** (8), 2017, 4775–4784.

[72] Q. Wang, S. Member, S. Liu and J. Chanussot, Scene classification with recurrent attention of VHR remote sensing images, *IEEE Trans. Geosci. Remote Sens.* **57** (2), 2019, 1155–1167.

[73] W. Shao, W. Yang, G.-S. Xia, G. Liu, A hierarchical scheme of multiple feature fusion for high-resolution satellite scene categorization, in: International Conference on Computer Vision System, 2013, pp. 324–333.

[74] Y. Jia, et al. Caffe: Convolutional architecture for fast feature embedding $*$, in: ACM International Conference on Multimedia, 2014, pp. 675–678.

[75] X. Gu, P. Angelov, A semi-supervised deep rule-based approach for remote sensing scene classification, in: INNS Big Data and Deep Learning conference, 2019, pp. 257–266.

[76] X. Gu, P. Angelov, Deep rule-based aerial scene classifier using high-level ensemble feature descriptor, in: International Joint Conference on Neural Networks, 2019, p. in press.

[77] L. Xie, Q. Tian, M. Wang and B. Zhang, Spatial pooling of heterogeneous features for image classification, *IEEE Trans. Image Process.* **23** (5), 2014, 1994–2008.

[78] Q. Li, Q. Peng and C. Yan, Multiple VLAD encoding of CNNs for image classification, *Comput. Sci. Eng.* **20** (2), 2018, 52–63.

[79] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3360–3367.

[80] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1794–1801.

[81] A. Shaban, H.R. Rabiee, M. Najibi and S. Yousefi, From local similarities to global coding: A framework for coding applications, *IEEE Trans. Image Process.* **24** (12), 2015, 5074–5085.

[82] Y. Pan, Y. Xia, Y. Song and W. Cai, Locality constrained encoding of frequency and spatial information for image classification, *Multimed. Tools Appl.* **77** (19), 2018, 24891–24907.

[83] C. Zhang, J. Cheng and Q. Tian, Structured weak semantic space construction for visual categorization, *IEEE Trans. Neural Netw. Learn. Syst.* **29** (8), 2018, 3442–3451.

[84] C. Zhang, C. Li, D. Lu, J. Cheng and Q. Tian, Birds of a feather flock together: Visual representation with scale and class consistency, *Inf. Sci. (Ny).* **460–461**, 2018, 115–127.

Highlights

- A generic approach for deep rule-based systems to self-organize a multi-layer structure is proposed.
- The proposed system can offer higher transparency and human-interpretability for large-scale, complex problems.
- The proposed approach can perform highly efficient decision-making with attractive classification precision.
- The effectiveness and validity of the proposed approach are demonstrated on a variety of popular benchmark image ~~data~~sets.

Queries and Answers

Query: Please check whether the designated corresponding author is correct, and amend if necessary.
Answer: Yes, correct.

Query: Please check the corresponding address, and correct if necessary.
Answer: Correct

Query: We find that some of the roles provided for the author(s) Xiaowei Gu, Plamen P. Angelov does not match the list of acceptable roles. Please choose a role from the below list for this author: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing - original draft, Writing - review \& editing.
Answer: Xiaowei Gu: methodology, writing-original draft, writing-review \& editing. Plamen P. Angelov: conceptualization, writing-original draft, writing-review \& editing.

Query: Please check whether the third affiliation given here is okay.
Answer: Okay

Query: Your article is registered as a regular item and is being processed for inclusion in a regular issue of the journal. If this is NOT correct and your article belongs to a Special Issue/Collection please contact immediately prior to returning your corrections.
Answer: Correct.

Query: Please confirm that given names and surnames have been identified correctly and are presented in the desired order and please carefully verify the spelling of all authors' names.
Answer: Yes

Query: Blank spaces have been changed to en dashes in Tables 4, 5, 6, 7 and 8. Please check.
Answer: Correct

Query: Correctly acknowledging the primary funders and grant IDs of your research is important to ensure compliance with funder policies. We could not find any acknowledgement of funding sources in your text. Is this correct?
Answer: Yes, correct.

Query: Please update the status of publication for Ref. [76].
Answer: X. Gu, P. Angelov, Deep rule-based aerial scene classifier using high-level ensemble feature descriptor, in: International Joint Conference on Neural Networks, 2019, pp. 1-7.