Deep Fuzzy Tree for Large-Scale Hierarchical Visual Classification

Yu Wang, Qinghua Hu, Senior Member, IEEE, Pengfei Zhu, Linhao Li, Bingxu Lu, Jonathan M. Garibaldi, Senior Member, IEEE, and Xianling Li

Abstract—Deep learning models often use a flat softmax layer to classify samples after feature extraction in visual classification tasks. However, it is hard to make a single decision of finding the true label from massive classes. In this scenario, hierarchical classification is proved to be an effective solution and can be utilized to replace the softmax layer. A key issue of hierarchical classification is to construct a good label structure, which is very significant for classification performance. Several works have been proposed to address the issue, but they have some limitations and are almost designed heuristically. In this article, inspired by fuzzy rough set theory, we propose a deep fuzzy tree model which learns a better tree structure and classifiers for hierarchical classification with theory guarantee. Experimental results show the effectiveness and efficiency of the proposed model in various visual classification datasets.

Index Terms—Deep learning, fuzzy rough set, hierarchical classification, label structure.

I. INTRODUCTION

In recent years, deep learning has received widespread attention and achieved remarkable performance in visual classification tasks [1], [2]. With the tremendous growth of data, the number of labels in these tasks has been growing fast, which makes it more difficult to recognize the true class from thousands of candidate classes directly. Fortunately, a hierarchical structure often exists in these massive labels, which helps organize data with huge amounts of classes efficiently [3]. With a hierarchical structure, a difficult classification task is divided into several easier subtasks and hence gets solved more effectively. Babbar et al. [4] and Partalas et al. [5] compare hierarchical classification and flat classification, and conclude that hierarchical classification performs better than flat in such difficult task theoretically.

Currently, most network structures of deep learning use the softmax layer to classify samples [6]. Although this obtains good performance in many types of visual classification tasks, it still challenging to deal with large-scale tasks with massive labels. Inspired by [4] and [5], we believe that replacing softmax with a deep hierarchical tree system for classification can solve the problem effectively. In this scenario, a key problem is to construct a good label structure, which has been shown to significantly affect the performance of hierarchical classification [7], [8].

Many efforts have been dedicated to building a good structure for visual classification tasks. First, expert-designed ontologies are widely used for large-scale visual classification. They are often in the form of a semantic structure such as WordNet for ImageNet [9], and reflect the relations independent of data. However, such ontologies depend heavily on expert knowledge and are not always available. More importantly, the feature information of data is often ignored [7], and classification accuracy decreases if the semantic hierarchy is inconsistent with the feature space of data [10].

To solve this problem, data-driven hierarchical structure learning methods have been designed in many works. Some researchers try to build a tree structure by recursively assigning the classes of greatest confusion into one group [11]–[13]. Specifically, a 1-vs-all SVM is trained first, and then used to obtain the confusion matrix, to which the hierarchical clustering method is applied. The method has proven to be effective, but the performance heavily depends on the reliability of SVM, and it is particularly challenging when facing some difficult tasks, such as those with massive labels or unbalanced data.

Alternatively, several works have proposed to construct class affinity matrix by measuring interclass similarity or distance without a classifier. Fan et al. use the Euclidean distance between the central point of each class to describe the interclass distance [14], Dong et al. apply the averaged kernel distance [15], Qu et al. utilize the mean and variance to measure [7], and Zheng et al. introduce the Hausdorff distance [8]. Although some improvements are obtained in some visual classification tasks, these algorithms depend on some assumptions. For example, [14] assumes the data distribution is ball-shaped, [7] requires data respect Gaussian distributions, while [8] need data without much noise. More significantly, they are designed...
Fig. 1. Framework of deep fuzzy tree. In the training phase, the deep features of training samples are first extracted, then interclass similarities are measured and similarity matrix is created in step \textbf{A} for tree construction, and a tree structure with base fuzzy rough classifiers is built by hierarchical community detection in step \textbf{B}. In the test phase, after deep feature extraction step, test samples are assigned by fuzzy rough classifiers at each node of the tree from the root to the leaves (step \textbf{C}).

heuristically instead of theoretically for hierarchical classification, hence cannot measure the interclass similarity more properly.

In this article, we propose a Deep Fuzzy Tree (DFT) model to solve the large-scale classification tasks with massive labels by replacing the softmax layer in the deep learning network with a newly designed hierarchical classification method (see Fig. 1). We aim to learn a better tree structure and classifiers for hierarchical classification by measuring interclass similarity more appropriately. As fuzzy rough sets have shown their importance in some classification tasks [16]–[18], we leverage the theory of lower approximation and design a new dual fuzzy interclass similarity to both measure the interclass similarity and set the base classifiers of the tree with theoretical guarantee. Then, we detect the communities with higher visual similarity by community detection methods recursively and obtain a hierarchical tree structure. To deal with large-scale tasks, fast adaptation is designed with the help of vector quantization. Experimental results show that the proposed DFT model learns a more reasonable tree structure, which further improves the performance of deep learning and the hierarchical classification tasks.

The main contributions of this article are summarized as follows.

1) A new DFT framework is proposed to construct the tree structure and learn the base classifiers for large-scale hierarchical visual classification tasks. We verify the effectiveness and efficiency on various visual classification datasets in comparison with some state-of-the-art algorithms.

2) A new interclass similarity measurement dual fuzzy interclass similarity is proposed and used for both tree learning and base classifier setting. We theoretically prove that the new measurement can lead to a tighter generalization error bound of hierarchical classification.

The rest of this article is organized as follows. Section II reviews the preliminary knowledge hierarchy structure and hierarchical classification for this article. Section III describes the details of the proposed deep fuzzy tree algorithm. Section IV shows the theoretical results of the proposed model with respect to the error bound of hierarchical classification. Experimental results on various datasets are presented in Section V. Finally, Section VI concludes this article.

II. PRELIMINARIES

A. Class Hierarchy

Class hierarchy organizes classes into a hierarchical structure where granularities are from the coarse-grained to the fine-grained. There are two kinds of structures in class hierarchy, trees, and directed acyclic graphs (DAG). We focus on tree structures in this article since trees are the most common and widely used.

A tree hierarchy organizes the class labels into a tree-like structure to represent a kind of “IS-A” relationship between labels [19]. Specifically, Kosmopoulos et al. point out that the properties of the “IS-A” relationship can be described as asymmetry, antireflexivity, and transitivity [20]. We define a tree $T$ which has a pair of properties $(\mathcal{V}, \mathcal{E}, \prec)$ with a set of nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, where $\mathcal{E}$ represents a set of edges between nodes in different levels, and $\prec$ represents the parent–children relationship between nodes connected by edges (“IS-A” relationship), formulated as follows:

1) Asymmetry: if $v_i \prec v_j$ then $v_j \not\prec v_i$ for $\forall v_i, v_j \in \mathcal{V}$;

2) Antireflexivity: $v_i \not\prec v_i$ for $\forall v_i \in \mathcal{V}$;

3) Transitivity: if $v_i \prec v_j$ and $v_j \prec v_k$, then $v_i \prec v_k$ for $\forall v_i, v_j, v_k \in \mathcal{V}$;

Generally, there are several types of nodes in a tree hierarchy. For node $v_i$:

1) its parent node is denoted by $P_i$;

2) its child nodes are denoted by $C_i$, and $|C_i|$ is the number of child nodes of $v_i$;
As shown in previous works, it is reasonable that classes with high visual similarity are grouped into one super-class [7], [14]. The key point in this issue lies in the proper measurement of similarity between different classes. Given the interclass similarity matrix, clustering or community detection methods can be applied to build a tree. However, the designed measurements in previous works almost have some assumptions as discussed in Section I. Inspired by the theory of fuzzy rough sets, we propose a new measurement for better describing the similarity relations between different classes.

Given universe $U$ representing a group of objects, let $R$ be a fuzzy with similar relation to $U$, with generated features $B$. Assume that $R(x, y)$ monotonically decreases with the distance between $x$ and $y$, and $\forall x, y, z \in U$, $R$ has the following properties:

\begin{align}
(1) & \quad R(x, x) = 1 \\
(2) & \quad R(x, y) = R(y, x) \\
(3) & \quad R(x, z) \geq \min_y (R(x, y), R(y, z)).
\end{align}

For any subset $X \subset \langle U, R \rangle$, the lower approximation operator is defined as

$$R_{\downarrow}(X) \equiv \inf_{y \in U} S(N(R(x, y)), X(y))$$

(1)

where $S$ represents fuzzy triangular conorm ($T$-conorm), $N$ is a negator, and the standard negator is defined as $N(x) = 1 - x$. Assume there a set of classes $Y = \{d_1, d_2, \ldots, d_i, \ldots, d_N\}$. For

$$\forall x \in U, d_i(x) = \begin{cases} 0, & x \notin d_i \\ 1, & x \in d_i \end{cases}$$

As proved by Moser [22], [23], given $T^{cos} = \max(ab - \sqrt{1 - a^2} \sqrt{1 - b^2}, 0)$, kernel functions which have the property of reflexivity also have $T^{cos}$-transitive property. Hence, given $U$ and kernel function $k$ which have reflexivity and $T^{cos}$-transitive property, the fuzzy lower approximation operator of fuzzy subset $d_i \subset U$ can be formulated as

$$k_{\downarrow}d_i(x) = \inf_{y \in U} S(N(k(x, y)), d_i(y)) = \inf_{y \notin d_i} (1 - k(x, y)).$$

(2)

If we use Gaussian kernel to extract the relations in fuzzy rough calculations, (2) is transformed into

$$k_{\downarrow}d_i(x) = \inf_{y \notin d_i} \left(1 - \exp \left(-\frac{||x - y||^2}{\sigma} \right) \right).$$

(3)

Equation (3) shows that the lower approximation membership degree of $x$ to $d_i$ depends on the nearest sample with different label, since it needs to search the nearest sample in the space of $U - d_i$. Intuitively, the fuzzy lower approximation describes the certain degree of a sample $x$ belonging to a specific class $d_i$. By considering the fuzzy lower approximation of all samples in $d_i$, the interclass similarity can be measured appropriately.
Definition 3.1: Given a set of samples \((x_i, y_i)\), where \(x_i \in \{x_1, \ldots, x_m\}, y_i \in C = \{C_1, \ldots, C_n\}\), the fuzzy interclass similarity (FIS) is defined as

\[
K(C_{ab}) = \sum_{i=1}^{m} \frac{1}{m} \left( \inf_{x_a \in C_a} (1 - k(x_a, x_b)) \right)
\]

where

\[
\forall x_{a_i}, \ y_{a_i} = C_a \\
\forall x_{b_i}, \ y_{a_i} = C_b
\]

\(k(x, y)\) is the kernel function. Commonly, we use Gaussian kernel in the computation, so (4) turns into

\[
K(C_{ab}) = \sum_{i=1}^{m} \frac{1}{m} \left( \inf_{x_a \in C_a} \left(1 - \exp \left(-\frac{\|x_a - x_b\|^2}{\sigma}\right)\right) \right).
\]

Moreover, since the distance or similarity measurement should be symmetric intuitively, we define the dual fuzzy interclass similarity (DFIS) as follows:

Definition 3.2: DFIS is the average of the bidirectional fuzzy interclass similarity:

\[
K(C_a, C_b) = \frac{1}{2} (K(C_{ab}) + K(C_{ba})).
\]

The similarity of each class pair is calculated according to (6), and then the similarity matrix of all labels can be obtained. Moreover, we prove theoretically that the proposed interclass similarity obtains upper bounds to the generalization error bound of hierarchical classification (shown in Section IV).

B. Tree Construction With Hierarchical Community Detection

With the proposed dual fuzzy interclass similarity, the interclass similarity matrix is appropriately computed. Then, the tree structure is obtained by grouping the more similar nodes as a granule recursively. Inspired by the community detection structure in the social network which reveals the organization of people [and its application to confusion graph (as proposed by Jin et al. [24]), we apply the adaptive modularity community detection algorithm hierarchically to explore the communities through the large number of classes. In contrast to the conventional use of spectral clustering, the adaptive modularity community detection methods automatically find the local optimal solution to the community modularities, without processing the similarity matrix to meet the needs of spectral clustering properties.

We utilize the fast community detection algorithm proposed by Blondel et al. [25], and extend it to hierarchical community detection. For the flat version of community detection, the key point is to compute the modularity \(Q_k\) of the \(k\)th community. In our hierarchical scenario, we need to compute \(Q_k^v\) at node \(v\) in the tree structure

\[
Q_k^v = \frac{1}{2m} \sum_{i,j} \left( S_{i,j} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)
\]

where \(S_{i,j}\) is the similarity degree of the edge between class \(i\) and class \(j\), \(k_i = \sum_j (S_{i,j})\) is the sum of similarity degrees of the edges attached to the vertex class \(i\), \(c_i\) is the community to which vertex class \(i\) is assigned, and the \(\delta(p, q)\) function is 1 if \(p = q\) and 0 otherwise, and \(m = \frac{1}{2} \sum i_j S_{i,j}\).

Then, we follow the method of [25] which optimizes local modularity changes and the aggregated community at each node \(v\), and apply the algorithm recursively to build the tree structure. In this way, we automatically compute the optimal number of communities at each node of the tree, which removes the need for the parameter setting of node numbers and tree depth in [7], [14].

However, using the aforementioned hierarchical community detection algorithm may sometimes generate a tree structure with few singleton nodes which have only one child node. We delete the redundant nodes in the constructed tree by using

\[
TN_{pat}(v) = \Phi_{v\epsilon v}(\pi(v), \mu(\pi(v)))
\]

where \(\pi(v)\) is the parent node of the current node \(v\), \(\mu(\pi(v))\) = 1 if the number of child nodes of node \(\pi(v)\) is greater than 1, and 0 otherwise. For \(\Phi(\alpha, \beta)\), if \(\beta = 1\), then it means there is no singleton nodes to be removed. In this case, let \(\Phi(\alpha, \beta) = \alpha\). Otherwise, \(\Phi(\alpha, \beta) = \beta\) if node \(\pi(v)\) is a singleton node \((\mu(\pi(v)) = 0)\). In this case, we need to search the next parent node in ancestor node set \(\nu\) for node \(v\).

C. Classification With Node-Wise Tree Classifier

Given a learned tree structure \(T\), a test sample needs to be assigned to one of the leaf nodes in the classification task. Similarly, we adopt the framework of classical Pachinko Machine to classify samples starting from the root node until a leaf node is reached. Consistent with the theory of fuzzy lower approximation utilized in measuring the interclass similarity, we use the fuzzy rough classifier (FRC) as proposed by An et al. [26] as the base classifier, which assigns samples to the candidate class with the largest membership of the samples to the fuzzy lower approximation of the class. In other words, we utilize the lower approximation of samples to first measure the distance between classes appropriately, and then to assign the test samples to the right candidate class, which can help improve the overall performance.

Moreover, using FRC as the base classifier of the tree can reduce the training time of the overall algorithm significantly as it does not need to be trained in advance before using the testing phase.

D. Fast Adaptation to Large-Scale Tasks

Constructing the affinity matrix with all samples is time-consuming, especially for large-scale tasks. Therefore, in this article, fast adaptation is proposed to solve this problem. With the aim of reducing sample numbers, vector quantization is used to generate a few representative points for each class. Although this method reduces the complexity remarkably, it does not lose much performance, as shown in our experiments. We introduce vector quantization briefly in the following.
Recall that $x \in \mathcal{X}$ is a sample of $d$-dimensional vector, we aim to obtain a reconstruction vector $\mathbf{q}_i \in \mathcal{X}$ ($1 \leq C \leq d$) through $\mathbf{q}_i = \mathbf{Q}(x_i)$. When $x_i$ is quantized as $\mathbf{q}_i$, a distortion measure $d(x_i, \mathbf{q}_i)$ can be defined between $x_i$ and $\mathbf{q}_i$. The overall average distortion with $m$ samples is written as $D_T = \frac{1}{m} \sum_{i=1}^{C} \sum_{j=1}^{C} d(x_i, q_{i})$. To split the original space into $C$ cells, each cell $C_i$ is associated with a reconstruction vector $\mathbf{q}_i$.

We use mean-square-error (mse) $d(x, q) = \frac{1}{N} \sum_{k=1}^{N} ||x_k - q_k||^2$ as the distortion measurement, and this process turns to be the well-known K-means algorithm. Representative points $r_{C_i}$ are generated for each class $C_i$ by setting a proportion $\eta$ of samples in the class. Note that at least one representative point is ensured for each class.

With the aforementioned parts, we summarize the proposed deep fuzzy tree model. In the training phase, deep features are first extracted and the interclass similarity is measured based on the features of training samples, and then hierarchical community detection is applied to build a tree structure (see Algorithm 1). In the test phase, after feature extraction, test samples are assigned by the fuzzy rough set classifier at each node of the tree structure (see Algorithm 2).

IV. THEORETICAL RESULTS

To find the rationality and features of the proposed model, we analyze the generalization error bound of hierarchical classification, and prove theoretically that the designed fuzzy interclass similarity can help improve the performance of hierarchical classification in the following proposition.

Proposition 1: Let $\mathbf{S} = \{(x_i, y_i)\}_{i=1}^{n}$ be a set of samples drawn independent and identically distributed (i.i.d.) based on a probability distribution $D$. Let $\mathbf{X}$ be the membership of $x_i$ to class $y_i$, and $\Phi(\mathbf{X})$ be the kernel function for $\mathbf{X}$. Given a tree structure $T$, the empirical data-dependent error of hierarchical classification with $H_{err}D[f]$ is upper bounded by the fuzzy interclass similarity $K(\mathbf{S})$.

Proof: The Rademacher complexity is widely used to analyze the generalization error bound of a classification problem. We utilize the theorem presented by Shawe-Taylor and Cristianini for the Rademacher data-dependent generalization bound in this issue $H_{err}D[f] \leq H_{err}S[f] + 2R_Dp_{cur}(G(F)) + 3\sqrt{\ln(2/\delta)}m$ (9) where $F$ is a set of hypotheses, $H_{err}D(f) = E_D[f]$ represents the generalization error of a hierarchical classifier, and $H_{err}S[f] = E_S[f]$ represents the empirical error of a hierarchical classifier, with the minimum probability of $1 - \delta$.

In the scenario of hierarchical classification, we define $f \in F$ as $\mathcal{F} = \{ f : (x, y) \in \mathcal{X} \times \mathcal{Y} \rightarrow \langle \mathbf{X}(x, y), \Phi(\mathbf{X}) \rangle \}$ (10) where $\mathbf{X}(x, y)$ is the membership of $x$ to class $y$, and $\Phi(\mathbf{X})$ is the kernel function for $\mathbf{X}$.
to explore the influence of interclass measurement as

$$
\hat{R}_m(G(F)) = \mathbb{E}_\sigma \left[ \sup_{g_f \in G} \left\{ \frac{1}{m} \sum_{i=1}^{m} \sigma_i g_f(x_i, y_i) \right\} \right] = \mathbb{E}_\sigma \left[ \sup_{g_f \in G} \left\{ \frac{1}{m} \sum_{i=1}^{m} \sigma_i \min_{v \in \Psi(y_i)} \left( f(x_i, v) - \max_{v' \in \Psi(v)} f(x_i, v') \right) \right\} \right]
$$

(11)

where $\sigma_i$ s are Rademacher variables which are independent uniform random variables whose values are chosen from the set $\{-1, +1\}$. Here we utilize the definition of Babbar et al. [4] for hierarchical classification, which develops the multiclass margin in [29] as

$$
G(F) = \{ g_f : (x, y) \in \mathcal{X} \times \mathcal{Y} \mapsto \min_{v \in \Psi(y_i)} \left( f(x_i, v) - \max_{v' \in \Psi(v)} f(x_i, v') \right) \}.
$$

(12)

Then, (11) can be transformed with the construction of $c$ as

$$
\hat{R}_m(G(F)) \leq \mathbb{E}_\sigma \left[ \sup_{g_f \in G} \sum_{v, v' \in \mathcal{Y}^2} \frac{1}{m} \sum_{i \in \mathcal{C}} \sigma_i \left( f(x_i, v) - f(x_i, v') \right) \right]
$$

(13)

where

$$
i = [1, 2, \ldots, m]
$$

$$\mathcal{C} : c(f, x_i, y_i) = (v, v').
$$

As $\forall i, j \in \{ l | c(f, x_i, y_i) = (v, v') \}^2, i \neq j, \mathbb{E}_\sigma [\sigma_i, \sigma_j] = 0$, inequality (13) can be relaxed according to Jensen’s inequality as follows:

$$
\hat{R}_m(G(F)) \leq \frac{1}{m} \sup_{g_f \in G} \sum_{v, v' \in \mathcal{Y}^2} \sum_{v' \in \Psi(v)} \left( \mathbb{E}_\sigma \left[ \sum_{i \in \mathcal{C}} \sigma_i \left( f(x_i, v) - f(x_i, v') \right) \right] \right)^2
$$

$$
= \frac{1}{m} \sup_{g_f \in G} \sum_{v, v' \in \mathcal{Y}^2} \sum_{v' \in \Psi(v)} \sum_{i \in \mathcal{C}} \left( \langle \Delta x_{i, v}, \Phi(\Lambda) \rangle - \langle \Delta x_{i, v'}, \Phi(\Lambda) \rangle \right).
$$

(14)

(14) is in the form of

$$
\frac{1}{m} \sup_{g_f \in G} \sum_{v, v' \in \mathcal{Y}^2} \sum_{v' \in \Psi(v)} \left( \inf_{x_{i, v}} \left( 1 - \exp \left( -\frac{||x_{i, v} - \hat{x}_{i, v}||^2}{\sigma} \right) \right) \right)
$$

$$
- \inf_{x_{i, v'}} \left( 1 - \exp \left( -\frac{||x_{i, v} - \hat{x}_{i, v'}||^2}{\sigma} \right) \right)
$$

$$
= \frac{1}{m} \sup_{g_f \in G} \sum_{v, v' \in \mathcal{Y}^2} \sum_{v' \in \Psi(v)} \left( \mathcal{K}_{x_{i, v}}(C_v) - \mathcal{K}_{x_{i, v'}}(C_v') \right)
$$

$$
\leq \sum_{v, v' \in \mathcal{Y}^2} \sum_{v' \in \Psi(v)} \frac{\mathcal{K}_{x_{i, v}}(C_v, C_v')}{m}
$$

(15)

where the node $v$ is the ground truth node, and the node $v'$ is the other node which is the most similar node with respect to a certain sample $x_i$.

Inequality (15) shows that for the sample $x_i \in \mathcal{S}$, the generalization upper bound is influenced by the fuzzy interclass similarity between nodes $v$ and $v'$. For each node in the tree, the error upper bound will be smaller if the fuzzy interclass similarity between the ground truth node and other sibling nodes is larger. In the proposed deep fuzzy tree model, we realize inequality (15) by clustering the classes based on the interclass matrix with respect to the fuzzy interclass similarity to maintain large interclass similarity at each non-leaf node. The proposition also demonstrates that setting FRC as the base classifier can further improve the classification performance along with the tree structure built upon fuzzy interclass similarity.

V. EXPERIMENTS

A. Dataset and Implementations

We perform experiments on various datasets (see Table I). Typically, these datasets are all organized by using a semantic tree structure, which shows the hierarchical relations between classes.

1) PASCAL VOC [30]: a visual object classes dataset which is a benchmark in visual object category recognition and detection. It has 34 828 images with 20 classes.

2) ILSVRC65 [31]: a visual object image dataset which is the subset of ImageNet. It has 17 1000 samples in 65 different classes.
3) Stanford Cars [32]: a car image dataset which aims to address the fine-grained classification problem. It has 16,185 samples in 196 different classes.

4) Cifar-100 [33]: an image dataset containing 60,000 samples in 100 classes, with 600 images in each class.

5) Caltech256 [34]: an image dataset with various types of classes. It has 30,607 image samples and 256 class labels.

6) SUN [35]: a scene understanding dataset with 397 kinds of scenes. We modify it by leaving out the categories that have more than one parent labels and samples with multiple labels. Finally, SUN dataset turns into 324 classes with at least 100 images per category.

7) ImageNet 1 K [36]: a large-scale image classification dataset which contains 1000 categories. Each dataset is split into a training subset and a test subset by 80% and 20%, respectively. The training subset is used to construct the hierarchical structure, while the test subset is used to obtain the classification results. All the results shown are the average of five-time running results, and all the experiments are executed on an Intel Core i7-600 running the Windows 8 operating system at 3.40 GHz with 32 GB memory.

B. Comparison Methods

We compare various algorithms including tree construction, semantic ontology, and deep learning baseline model.

1) Label Tree (LT) [12]: builds the label tree based on the confusion degree of classification results. It first learns a multiclass SVM and gets the confusion matrix through classification as the affinity matrix. Then, the tree is built through hierarchical spectral clustering method.

2) Mean-Vector-Based Tree Learning (MeanVT) [14]: considers the distance between the center of different classes as the distance between different classes and hence builds the similarity matrix. Then, it uses hierarchical spectral clustering to build the tree according to the similarity matrix.

3) Mean-Variance-Based Visual Tree Construction (MeanVarVT) [7]: constructs a similarity matrix by using the mean vector and the variance vector of each class to measure the distance between different classes. Then, it uses hierarchical spectral clustering to build a tree according to the similarity matrix.

4) Enhanced Visual Tree Construction (EnhancedVT) [8]: first proceeds through active sampling to choose a small part of samples reflecting the features of the dataset, then applies the Hausdorff distance to construct the affinity matrix. Finally, it utilizes hierarchical spectral clustering method to build the label tree structure.

5) Ontology (OTG): is the expert-designed semantic tree structure. It reflects the thinking manner of human beings and helps organize the datasets.

6) Standard VGG Net [6]: is the conventional VGG deep learning model, which uses softmax layer to classify the sample to all the candidate classes in a flat manner. We use pretrained VGG-19 Net fine-tuned with various datasets in this article for deep feature extraction, and replace its softmax layer with the proposed deep fuzzy tree algorithm.

For a fair comparison with the quality of the tree structure, we obtain the different interclass relation matrix and apply the same community detection method to build the tree structure for [7], [8], [12], [14].

C. Results on Classification Performance

In order to investigate the performance of the proposed model, we compare DFT with other state-of-the-art models on six visual datasets, and use classification accuracy of the original test labels to assess the performance. In the proposed DFT, we set parameter $\eta = 0.05$, and choose the best parameter $\sigma$ from the candidate set $\{10^{-2}, 10^{-1}, \ldots, 10^{0}\}$. The results are shown in Table II.

There are three aspects of these results that merit discussion. First, flat classification versus hierarchical classification. Table II shows that Standard VGG-19 Net performs well in datasets without too many labels in comparison with all hierarchical methods, such as PASCAL VOC, ILSVRC 65, and Cifar 100. Just as [4] and [5] pointed out, flat classifier performs better in the easy tasks while a hierarchical classifier is good at handling some difficult tasks by dividing the hard task into many easier subtasks. DFT outperforms the flat VGG-19 Net in most of the datasets with a large number of labels, which verifies the conclusion of [4] and [5].

Second, ontology versus data-driven hierarchy. Table II demonstrates that ontology-based method performs well in ILSVRC 65, Cifar 100, and SUN in comparison with data-dependent hierarchical methods except DFT, which suggests that human knowledge is very helpful for determining structure and classification. However, there are large gaps between ontology-based and data-dependent hierarchical methods in PASCAL VOC and Stanford Cars. This suggests that building the tree structure without data does not work well in all the classification tasks, as a semantic gap exists in these tasks which needs to be improved by considering data information. Third, different hierarchical structure learning algorithms. It can be seen from Table II that our proposed DFT performs better than other hierarchical methods in all the datasets. MeanVT generally fails to get a good performance since utilizing one central point to represent all the samples can hardly express all information in data, and will only be effective if the data distributions of all the classes are ball-shaped. MeanVarVT improves this problem by using the mean and variance of data, thereby improving performance noticeably. However, it also has a prior assumption that the data distribution is Gaussian, which is often the case, but is hard to verify for data distributions of massive classes. In contrast, the performance of EnhancedVT in datasets with fewer labels is generally better than in datasets with massive labels. The limit for this algorithm is that the Hausdorff distance is sensitive to abnormal points, and this is inevitably in datasets with massive labels. Finally, although the LabelTree algorithm performs relatively well in various datasets, it cannot ensure the reliability of SVM for all the datasets, especially for those with huge amounts of labels.

Moreover, to further explore whether the observed differences are statistically significant, the Friedman test [37] for multiple
comparisons, together with the Bonferroni–Dunn post hoc test [38] to identify pairwise differences, is applied on the all seven datasets. In the Friedman test, given \( k \) compared algorithms and \( N \) datasets, let \( r_i^j \) be the rank of the \( j \)th algorithm on the \( i \)th dataset, and \( R_i = \frac{1}{N} \sum_{i=1}^{N} r_i^j \) be the average rank of algorithm \( i \) among all datasets. The null-hypothesis of the Friedman test is that all the algorithms are equivalent in terms of classification accuracy. Under null-hypothesis, the Friedman statistic is distributed according to \( \chi^2_F \) with \( k-1 \) degrees of freedom

\[
\chi^2_F = \frac{12N}{k(k+1)} \left( \sum_{i=1}^{k} R_i^2 - k(k+1)^2 \right) \tag{16}
\]

where \( F_F \) follows a Fisher distribution with \( k-1 \) and \((k-1)(N-1)\) degrees of freedom. The average rank of all the algorithms in terms of classification accuracy is listed in Table II, and the value \( F_F = 26.649 \) is computed according to (16). With seven algorithms and seven datasets, the critical value for \( \alpha = 0.05 \) of \( F((7-1), (7-1) \times (7-1)) \) is 2.3638, so the null-hypothesis can be rejected. Thus, all the algorithms are not equivalent in terms of classification accuracy, and there exist significant differences between them.

Then the Bonferroni–Dunn post hoc test is leveraged to detect if the DFT algorithm is better than the existing ones on the all seven datasets. Specifically, the performance of the two compared algorithms is significantly different if the distance between the averaged ranks exceeds the critical distance (CD): \( CD_{\alpha} = q_{\alpha} \sqrt{\frac{k(k+1)}{6N}} \), where \( q_{\alpha} \) is given in [39, Table 5]. Note that \( q_{0.1} = 2.394 \) with \( k = 7 \), so \( CD_{0.1} = q_{0.1} \sqrt{\frac{7 \times 7}{6 \times 6}} \). Fig. 3 visually shows the CD diagrams in terms of classification accuracy, in which the lowest (best) ranks appear on the right of the x-axis. The bars show the estimated range of ranks, such that algorithms for which the bars do not overlap horizontally are statistically different. From Fig. 3, therefore, in terms of classification accuracy, DFT performs statistically better than MeanVT, MeanVarVT, Ontology, and EnhancedVT, but there is no statistical difference between DFT, Standard VGG, and LabelTree. According to conclusion of [4], [5], and Table II, DFT is more appropriate to deal with data with a large number of labels than Standard VGG, so we do another Bonferroni–Dunn test of DFT and Standard VGG across four datasets with many labels, i.e., Stanford Cars, Caltech 256, SUN, and ImageNet 1 K. In this test, \( q_{0.1} = 1.645 \) with \( k = 2 \) leads to \( CD_{0.1} = 0.8225 \). The result is shown in Fig. 4, which demonstrates that there exists statistical difference between DFT and Standard VGG, which verifies that DFT performs better than Standard VGG in datasets with a large number of labels.

D. Case Study on Learned Tree Structures

To better understand the features of structures, we analyze and visualize the PASCAL VOC dataset in Fig. 5. Fig. 5(a) is the expert-designed semantic structure, and Fig. 5(b) is the structure learned by DFT. There are some slight differences between these two structures, which are annotated in red. In the semantic structure, Aeroplane and Bird belong to superclass Vehicles and Animal, respectively. However, they are grouped into the same superclass in the structure learned by DFT. Actually, the images of Aeroplane and Bird are very similar in visualization, and it is more reasonable to group them into one superclass. Similarly, Bus is the fine-grained class of 4-Wheeled Vehicles, whereas Train is not. In contrast, they are assigned to the same superclass in the structure learned by DFT. It is shown that the images of Train are almost locomotive images, which have a lot.

| Datasets       | Standard VGG | Ontology | LabelTree | MeanVT | MeanVarVT | EnhancedVT | DFT    |
|----------------|--------------|----------|-----------|--------|-----------|------------|--------|
| PASCAL VOC     | 75.55 \((1)\) | 73.40 \((7)\) | 74.66 \((4)\) | 74.50 \((5)\) | 73.43 \((6)\) | 75.19 \((3)\) | 75.51 \((2)\) |
| ILSVRC 65      | 85.70 \((3)\) | 85.64 \((4)\) | 86.68 \((2)\) | 85.30 \((7)\) | 85.40 \((6)\) | 85.42 \((5)\) | 87.27 \((1)\) |
| Stanford Cars  | 58.13 \((2)\) | 51.74 \((6)\) | 55.15 \((3)\) | 51.00 \((7)\) | 52.30 \((5)\) | 52.51 \((4)\) | 59.90 \((1)\) |
| Cifar 100      | 72.72 \((1)\) | 70.09 \((4)\) | 70.89 \((3)\) | 68.91 \((5)\) | 67.72 \((7)\) | 68.68 \((6)\) | 72.53 \((2)\) |
| Caltech 256    | 82.11 \((3)\) | 80.54 \((7)\) | 81.31 \((5)\) | 81.17 \((6)\) | 82.19 \((2)\) | 81.61 \((4)\) | 82.67 \((1)\) |
| SUN            | 82.91 \((2)\) | 81.25 \((6)\) | 82.05 \((3)\) | 81.28 \((5)\) | 80.93 \((7)\) | 81.67 \((4)\) | 84.41 \((1)\) |
| ImageNet 1K    | 61.98 \((2)\) | 60.55 \((5)\) | 60.56 \((4)\) | 60.45 \((7)\) | 60.50 \((6)\) | 61.90 \((3)\) | 62.11 \((1)\) |

Average Rank: 2.000, 5.571, 3.000, 6.000, 6.000, 4.143, 1.286

The best result obtained for each dataset is highlighted in bold.
Fig. 5. Tree structures comparison on PASCAL VOC dataset. The learned tree structure groups class with similar shape (Bird and Aeroplane, Train and Bus) into one super-class, which reflects the visual feature similarity of various classes in the dataset better. (a) Expert-designed semantic tree structure. (b) Tree structure learned by the proposed DFT algorithm.

Fig. 6. Visualization of the learned tree structures. The learned tree structure removes the class Mushroom from the group Fruit and Vegetables, which corrects the unreasonable local parts of the semantic tree structure addressed by [33].

of resemblance with the images of Bus, and hence it is more reasonable to group them together.

Moreover, we visualize the learned tree structures of Cifar 100 datasets in Fig. 6, and it can be explicitly shown that the relations between different classes are deeply connected. It is worth noting that the learned structure corrects mistakes in the original human-designed structure. As stated in [33], Mushroom is grouped into Fruit and Vegetables for convenience, but they really do not belong to that group. Our learned tree structure correctly places Mushroom into a separate new group, as shown in Fig. 6.

E. Results on Efficiency of Tree Construction

To explore the efficiency of our algorithm, we run all the hierarchical models on all seven datasets and compare the
enhancedVT perform well in the cost of decrease in efficiency. Ontology generally comes just behind MeanVT and MeanVarVT, and enhancedVT obtains the best level. The efficiency of the proposed DFT generally comes just behind MeanVT and is comparable to MeanVarVT. To further explore the statistical differences, the Friedman test and Bonferroni–Dunn post hoc test are again applied. In the Friedman test, the null-hypothesis is that all the tree learning algorithms are considered equivalent in terms of run time. According to (16), \( \chi^2_F = 22.456 \) and \( F_F = 24.303 \) with five tree learning algorithms and seven datasets. Therefore, we can reject the null-hypothesis and conclude that there are significant differences between the algorithms. With the rejection of the null-hypothesis, the Bonferroni–Dunn post hoc test can be proceeded to explore if the algorithms are compared to each other. In this case, it is used to explore if the proposed algorithm is statistically better than others.

The Bonferroni–Dunn post hoc test result is shown in Fig. 7 with the CD diagrams in terms of run time. From this, we can conclude that the proposed DFT is comparable with MeanVT and MeanVarVT, with no statistical differences between them. However, LabelTree and enhancedVT are not so efficient in comparison with these three algorithms. LabelTree first trains a multiclass SVM and then uses it to get the confusion matrix, so the efficiency is heavily influenced by the training of SVM. enhancedVT aims to find the most important samples for data, requiring large amounts of time to get the selected sample set. The reason for enhancedVT being less efficient than LabelTree in some datasets is mainly due to the process of selecting sample set in terms of computing multiple features of data, which takes up most of the running time. It is worth noting that LabelTree and enhancedVT perform well in the cost of decrease in efficiency. On the other hand, MeanVT and MeanVarVT algorithm are very efficient, but their performance is not as good as other hierarchical methods since they cannot measure the similarity between classes accurately by assuming data distribution as ball-shaped or Gaussian. The proposed DFT obtains the best level (with LabelTree) of classification accuracy while its efficiency is comparable with MeanVT, MeanVarVT, and StandardVGG, which indicates that DFT is both effective and efficient.

### F. Parameter Analysis

There are two parameters in the proposed model, \( \sigma \) and \( \eta \). To explore the influence of each parameter, various combinations are explored in this experiment. We choose \( \sigma \) ranging from \{10^{-2}, 10^{-1}, \ldots, 10^{2}\} and \( \eta \) ranging from \{0.05, 0.15, \ldots, 0.65\}. Generally, we investigate how much influence on classification accuracy made by different combinations of \( \sigma \) and \( \eta \) in Fig. 8 and Figs. 9–14 (see in the supplementary material), where Figs. 8 and 9 show some recommended choices of the parameters and Figs. 9–14 describe the influence of all the alternatives for the parameters.

First, \( \eta \) is the ratio of representative samples in each class, and the larger the \( \eta \) is, the more original information the data contains. Fig. 8 shows that the classification accuracy first increases from \( \eta = 0.05 \) and then decreases until \( \eta = 0.45 \). This indicates that even though using more samples is helpful for training the model, inappropriate ratios of representative points will lead to a decrease in performance. The reason may be that the general features of data can be partly reflected by few representative points but gradually being better until there are sufficient points to describe the overall features. Moreover, a larger value of \( \eta \) can help improve performance since more information of data is leveraged. Interestingly, \( \eta = 0.05 \) appears to be a good choice for balancing the effectiveness and efficiency experimentally, which indicates that small data can also produce good results if they can reflect the general information of the dataset properly.

Second, Fig. 9 shows that the optimal (or near-optimal) value of \( \sigma \) is different for various datasets. With the fixed value of \( \eta \), values of \( \sigma \) in the range \{10^2, 10^3\} achieve good performance in most datasets. Furthermore, we also plot all the combination of \( \eta \) and \( \sigma \) to explore more details of the parameter influence in Figs. 9–14 of the supplementary material. Generally, we find that the classification accuracy is broadly similar when each

| Datasets         | LabelTree | MeanVT | MeanVarVT | enhancedVT | DFT  |
|------------------|-----------|--------|-----------|------------|------|
| PASCAL VOC       | 140(4)    | 49(1)  | 57(3)     | 142(5)     | 56(2)|
| ILSVRC 65        | 125(5)    | 38(1)  | 50(3)     | 110(4)     | 42(2)|
| Stanford Cars    | 189(4)    | 33(1)  | 58(2)     | 247(5)     | 64(3)|
| Cifar 100        | 1218(4)   | 339(1)| 614(2)    | 1861(5)    | 887(3)|
| Caltech 256      | 769(5)    | 250(1)| 371(3)    | 563(4)     | 352(2)|
| SUN              | 2501(4)   | 930(1)| 1604(3)   | 5693(5)    | 1143(2)|
| ImageNet 1K      | 400932(5) | 93941(1)| 131941(2)| 282236(4) | 133716(3)|
| Avg. Rank        | 4.429     | 1.000  | 2.571     | 4.571      | 2.429|

Fig. 7. Comparison on running time of tree construction phase between Deep Fuzzy Tree and other algorithms with the Bonferroni–Dunn test.
parameter has a value in the recommended interval, but there are significant differences if inappropriate values of the parameters are selected.

VI. CONCLUSION

In this article, we propose a new DFT framework, aiming at gaining the benefit from hierarchical classification to help deep learning solve classification tasks with a large numbers of labels. With the help of the theory of fuzzy rough sets, dual fuzzy interclass similarity is designed to learn a better tree structure along with setting fuzzy rough classifier as the base classifier. It is further proved theoretically to be effective for hierarchical classification. By using community detection methods, a tree structure can be constructed by hierarchically detecting the most similar communities. To deal with large-scale tasks, fast adaptation is designed by using vector quantization. The performance of the proposed DFT algorithm shows the effectiveness and efficiency in comparison with the standard deep learning model and state-of-the-art hierarchical classification models.

REFERENCES

[1] C. Spampinato, S. Palazzo, I. Kavasis, D. Giordano, N. Souly, and M. Shah, “Deep learning human mind for automated visual classification,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2017, pp. 4503–4511.

[2] N. Lu, T. Li, X. Ren, and H. Miao, “A deep learning scheme for motor imagery classification based on restricted Boltzmann machines,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 25, no. 6, pp. 566–576, Jun. 2017.

[3] H. Zhao, P. Zhu, P. Wang, and Q. Hu, “Hierarchical feature selection with recursive regularization,” in Proc. Int. Joint Conf. Artif. Intell., 2017, pp. 3483–3489.

[4] R. Babbar, I. Partalas, E. Gaussier, and M. R. Amini, “On flat versus hierarchical classification in large-scale taxonomies,” in Proc. Int. Conf. Neural Inform. Process. Syst., 2012, pp. 1824–1832.

[5] I. Partalas, I. Partalas, E. Gaussier, and M. R. Amini, “Learning taxonomy adaptation in large-scale classification,” J. Mach. Learn. Res., vol. 17, no. 1, pp. 3350–3386, 2016.

[6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

[7] Y. Qi et al., “Joint hierarchical category structure learning and large-scale image classification,” IEEE Trans. Image Process., vol. 26, no. 9, pp. 4331–4346, Sep. 2017.

[8] Y. Zheng, J. Fan, J. Zhang, and X. Gao, “Hierarchical learning of multi-task sparse metrics for large-scale image classification,” Pattern Recognit., vol. 67, pp. 97–109, 2017.

[9] O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vision, vol. 115, no. 3, pp. 211–252, 2015.

[10] M. Sun, W. Huang, and S. Savarese, “Find the best path: An efficient and accurate classifier for image hierarchies,” in Proc. IEEE Int. Conf. Comput. Vision, 2014, pp. 265–272.

[11] G. Griffin and P. Perona, “Learning and using taxonomies for fast visual categorization,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2008, pp. 1–8.

[12] S. Bengio, J. Weston, and D. Grangier, “Label embedding trees for large multi-class tasks,” Adv. Neural Inform. Process. Syst., pp. 163–171, 2013.

[13] B. Liu, F. Sadeghi, M. Tappen, O. Shamir, and C. Liu, “Probabilistic label trees for efficient large scale image classification,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2013, pp. 843–850.

[14] J. Fan, J. Peng, L. Gao, and N. Zhou, “Hierarchical learning of tree classifiers for large-scale plant species identification,” IEEE Trans. Image Process., vol. 24, no. 11, pp. 4172–4184, Nov. 2015.

[15] P. Dong, K. Mei, N. Zheng, H. Lei, and J. Fan, “Training inter-related classifiers for automatic image classification and annotation,” Pattern Recognit., vol. 46, no. 5, pp. 1382–1395, 2013.

[16] Q. Hu, L. Zhang, Y. Zhou, and W. Pedrycz, “Large-scale multimodality attribute reduction with multi-kernel fuzzy rough sets,” IEEE Trans. Fuzzy Syst., vol. 26, no. 1, pp. 226–238, Feb. 2018.

[17] A. Tan, W. Z. Wu, Y. Qian, J. Liang, J. Chen, and J. Li, “Intuitionistic fuzzy rough set-based granular structures and attribute subset selection,” IEEE Trans. Fuzzy Syst., vol. 27, no. 3, pp. 527–539, Mar. 2019.

[18] F. Feng, H. Fujita, M. I. Ali, R. R. Yager, and X. Liu, “Another view on generalized intuitionistic fuzzy soft sets and related multiattribute decision making methods,” IEEE Trans. Fuzzy Syst., vol. 27, no. 3, pp. 474–488, Mar. 2019.

[19] C. N. Silla Jr. and A. A. Freitas, “A survey of hierarchical classification across different application domains,” Data Mining Knowl. Discovery, vol. 22, pp. 31–72, 2011.

[20] K. Aris, P. Ioannis, G. Eric, G. Palouras, and I. Androutsopoulos, “Evaluation measures for hierarchical classification: A unified view and novel approaches,” Data Mining Knowl. Discovery, vol. 29, no. 3, pp. 820–865, 2015.
D. Koller and M. Sahami, “Hierarchically classifying documents using very few words,” in Proc. Int. Conf. Mach. Learn., 1997, vol. 97, pp. 170–178.

B. Moser, “On the t-transitivity of kernels,” Fuzzy Sets Syst., vol. 157, no. 13, pp. 1787–1796, 2006.

B. Moser, “On representing and generating kernels by fuzzy equivalence relations,” J. Mach. Learn. Res., vol. 7, pp. 2603–2620, 2006.

R. Jin, Y. Dou, Y. Wang, and X. Niu, “Confusion graph: Detecting confusion communities in large scale image classification,” in Proc. 26th Int. Joint Conf. Artif. Intell., 2017, pp. 1980–1986.

V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” J. Statistical Mech., vol. 2008, no. 10, pp. 155–168, 2008.

S. An, Q. Hu, and D. Yu, Case-Based Classifiers With Fuzzy Rough Sets. Berlin, Germany: Springer, 2011.

P. L. Bartlett and S. Mendelson, “Rademacher and gaussian complexities: Risk bounds and structural results,” J. Mach. Learn. Res., vol. 3, pp. 463–482, 2003.

J. Shawe-Taylor and N. Cristianini, Kernel Methods for Pattern Analysis. Cambridge, UK: Cambridge University Press, 2004.

Y. Guermeur, “Sample complexity of classifiers taking values in RQ, application to multi-class SVMs,” Commun. Statist. Theory Methods, vol. 39, no. 3, pp. 543–557, 2010.

M. Everingham et al., “The Pascal visual object classes (VOC) challenge,” Int. J. Comput. Vision, vol. 88, no. 2, pp. 303–338, 2010.

D. Jia, “Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2012, pp. 3450–3457.

J. Krause, M. Stark, D. Jia, and F.-F. Li, “3D object representations for fine-grained categorization,” in Proc. IEEE Conf. Comput. Vision Workshops, 2014, pp. 554–561.

A. Krizhevsky, “Learning multiple layers of features from tiny images,” Tech. Rep., Dept. Comput. Sci., Univ. Toronto, 2009.

G. Griffin, A. Holub, and P. Perona, Caltech-256 Object Category Dataset. California, USA: California Institute of Technology, 2007.

J. Xiao, J. Hays, and K. A. Ehinger, “Sun database: Large-scale scene recognition from abbey to zoo,” Comput. Vision Pattern Recognit., pp. 3485–3492, 2010.

O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vision, vol. 115, no. 3, pp. 211–252, 2015.

M. Friedman, “A comparison of alternative tests of significance for the problem of m rankings,” Ann. Math. Statist., vol. 11, no. 1, pp. 86–92, 1940.

O. J. Dunn, “Multiple comparisons among means,” J. Am. Statist. Assoc., vol. 56, no. 293, pp. 52–64, 1961.

J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” J. Mach. Learn. Res., vol. 7, pp. 1–30, 2006.

Yu Wang received the B.S. degree in communication engineering and the M.S. degree in software engineering from Tianjin University, Tianjin, China, in 2013 and 2016, respectively. He is currently working toward the Ph.D. degree in computer application techniques with the School of Intelligence and Computing, Tianjin University, China.

His research interests include hierarchical learning and image classification in data mining and machine learning.

Qinghua Hu received the B.S., M.S., and Ph.D. degrees in energy science engineering from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively. He was a Postdoctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong, from 2009 to 2011. He is currently the Dean of the School of Artificial Intelligence, the Vice Chairman of the Tianjin Branch of China Computer Federation, and the Vice Director of the SIG Granular Computing and Knowledge Discovery, and the Chinese Association of Artificial Intelligence. He has published over 200 peer-reviewed articles. His current research interests include uncertainty modeling in big data, machine learning with multi-modality data, and intelligent unmanned systems.

Dr. Hu is an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS, Acta Automatica Sinica, and Energies.

Pengfei Zhu received the B.S. degree in power engineering and the M.S. degree in power machinery and engineering from the Harbin Institute of Technology, Harbin, China, in 2009 and 2011, respectively, and the Ph.D. degree in computer vision from Hong Kong Polytechnic University, Hong Kong, China, in 2015.

Currently, he is an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. He has authored/co-authored more than 30 articles in conferences and journals. His research interests include machine learning and computer vision.

Dr. Zhu is the local arrangement Chair of International Joint Conference On Rough Sets (IJCRS) 2015, China Conference on Machine Learning (CCML) 2017, and Chinese Conference on Computer Vision (CCCV) 2017.

Linlao Li received the B.S. degree in applied mathematics, the M.S. degree in computational mathematics, and the Ph.D. degree in computer science from Tianjin University, Tianjin, China, in 2012, 2014, and 2019, respectively.

He is currently working as an Assistant Professor with the School of Artificial Intelligent, Hebei University of Technology, Tianjin. His research interests include quantization and hashing learning, sparse signal recovery, background modeling, and foreground detection.

Bingxu Lu received the B.S. degree in computer science and technology from the Nanjing Agriculture University, Nanjing, China, in 2016. Currently, he is working toward the Ph.D. degree in computer application techniques with the College of Intelligence and Computing, Tianjin University, China.

His research interests include deep learning and computer vision.

Jonathan M. Garibaldi received the B.Sc. (Hons.) degree in physics from Bristol University, Bristol, U.K., in 1984, and the M.Sc. degree in intelligent systems and the Ph.D. degree in uncertainty handling in immediate neonatal assessment from the University of Plymouth, Plymouth, U.K., in 1990 and 1997, respectively.

He is Head of the School of Computer Science, University of Nottingham, Nottingham, U.K., and leads the Intelligent Modelling and Analysis (IMA) research group. The IMA research group undertakes research into intelligent modeling, utilizing data analysis and transformation techniques to enable deeper and clearer understanding of complex problems. He has made many theoretical and practical contributions in fuzzy sets and systems, and in a wide range of generic machine learning techniques in real-world applications. He has published over 200 articles on fuzzy systems and intelligent data analysis. His main research interests include modeling uncertainty and variation in human reasoning, and in modeling and interpreting complex data to enable better decision-making, particularly in medical domains.

Dr. Garibaldi is the Editor-in-Chief of the IEEE TRANSACTIONS ON FUZZY SYSTEMS (2017). He has served regularly in the organizing committees and program committees of a range of leading international conferences and workshops, such as FUZZ-IEEE.

Xianling Li received the B.S., M.S., and Ph.D. degrees in power mechanical engineering from the Harbin Institute of Technology, Harbin, China, in 2006, 2007, and 2011, respectively.

He is a Senior Engineer with the Science and Technology on Thermal Energy and Power Laboratory, Wuhan, China. His interests include big data, fault diagnosis of complex power system, and intelligent control.