Hierarchical Concept-based Explanation in Deep Neural Networks

Yue Liu, Ziyi Yu, Zitu Liu, Zhenyao Yu, Yike Guo, *Fellow, IEEE*, Qun Liu, and Guoyin Wang, *Senior Member, IEEE*

Abstract—Deep neural networks (DNNs) are black-box tools with low explainability due to the huge gap between model representations and human knowledge. Concept-based explanation methods utilize concepts discovered from representations to explain DNN, but they rarely consider the whole-part relationships between concepts. This leads to partial overlap between the discovered concepts and ambiguous semantics in explanations. Here, we propose a Hierarchical Concept-based Explanation method (HCE) that provides clear and trustworthy local explanations. Guided by the defined visual concept tree aligned with human cognition, our Multilevel Concept Extractor based on image segmentation and clustering discovers concepts from representations, and we obtain concept trees of DNN by the designed Concept Tree AutoEncoder. Following this, we define Concept Tree Shapley value satisfying Shapley axioms to quantify the importance of hierarchical concepts on the model decision for each sample. New consistency metrics suitable for local explanation are further proposed to measure the credibility of explanations. Experiments on ImageNet show HCE obtains explanations with clearer concepts for GoogleNet than compared methods, and the average of consistency scores of explanations is improved by 35%. Explanations for 6 other image classification models verify that HCE has a certain universality.

Index Terms—Deep learning, explainability, concept-based explanation, local explanation

1 Introduction

Deep neural networks (DNNs) are widely used in important fields like computer vision with excellent performance [1]. However, the decisions of a DNN are difficult to explain and trust due to the complex mechanisms and enormous parameters within the model. Explainability has become a promising and popular domain for scientific understanding of model decisions, uncovering hidden biases, and beyond [2].

DNN is a black box for humans, and its explainability issue mainly stems from that the knowledge learned by the model is inconsistent with human knowledge. Model knowledge extracted from data by DNN is expressed in the form of vectors or feature maps. It can be called representations, which are not comprehensible to humans. Human knowledge is hierarchical concepts with clear semantics, and the constituent (or whole-part) relationship is one of their basic features. Specifically, concepts at the same level should not intersect, and concepts at different levels have a unidirectional inclusion relationship. For example, when humans observe an image of class cat, they recognize some visual concepts about objects (cat, table, chair, paper, etc., or broadly, background) before recognizing visual concepts about object parts (cat's head, cat's legs, etc.). Concept cat and table do not intersect, and cat is composed of concepts like cat's head, cat's legs, and others. It is obvious that there is a gap between model knowledge and human knowledge. Humans are receptive to concepts but struggle to understand the knowledge learned by a model since the semantics of representations and their internal complex associations are difficult to describe. Establishing a mapping between model knowledge and human knowledge is a common solution for bridging the gap.

Concept-based explanation methods [6], [7], [8], [9] map representations to concepts and quantify the impact of concepts on model decisions as explanations. These methods succeed in discovering concepts from model knowledge and improve the explainability of DNNs. However, they rarely model the constituent relationships between concepts, resulting in the discovered concepts lacking semantic structure and being partially overlapping. Explanations with confusing concept boundaries are easy to contain ambiguous semantics. If concepts have a hierarchy, we can explain a model decision intuitively by identifying the object (e.g., cat itself or background in an image) that has a greater impact. Then the impact of object parts like cat's head and legs can be further explored reasonably due to the constraints of hierarchy. Meanwhile, a concept has different effects in samples due to its diverse expressions, and

- *Yue Liu is with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China, and Shanghai Engineering Research Center of Intelligent Computing System, Shanghai 200444, China. E-mail: yueliu@shu.edu.cn.*
- *Ziyi Yu, Zitu Liu and Zhenyao Yu are with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China. E-mail: {yuziyip, vimotus, 21721601}@shu.edu.cn.*
- *Yike Guo is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong 200444, China, and the Department of Computing, Imperial College, London SW7 2AZ, UK. E-mail: y.guo@imperial.ac.uk.*
- *Qun Liu is with the Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China. E-mail: liuqun@cqupt.edu.cn.*
- *Guoyin Wang is with the Chongqing Key Laboratory of Computational Intelligence, and the Key Laboratory of Cyberspace Big Data Intelligent Security of Ministry of Education, Chongqing University of Posts and Telecommunications, Chongqing 400065, China. E-mail: wanggy@cqupt.edu.cn.*

hierarchical concepts in each sample would support precise local explanations that the above methods lack. Thus, the hierarchy of concepts is essential for further improving explainability.

However, several issues need to be tackled in establishing the mapping between representations and hierarchical concepts and obtaining attribution explanations for a DNN: (i) **How to model and discover hierarchical concepts in a DNN**. The specific structure of hierarchical concepts determines the validity of knowledge expression. Meanwhile, it is hierarchical concepts that emerge from representations that are consistent with model knowledge. Tree structure is a classic and effective structure for expressing knowledge, and employing a concept tree would clarify the semantic relationships among concepts. Establishing a mapping between representations and the concept tree would systematically describe model knowledge. (ii) **How to obtain accurate local explanations.** Local explanation [3], [4], [5] attempts to explain each model decision. It is relatively easy to obtain a local explanation by computing the impact of individual concepts on a model decision. When working with hierarchical concepts, we need to consider the impact of hierarchical structure and interactions between concepts. (iii) **How to evaluate explanations.** A good explanation should be faithful to the DNN, but there are currently no accepted metrics for quantifying the quality of an explanation. We believe that trustworthy explanations are consistent with model decisions and improve the explainability of the model.

Therefore, in order to tackle these challenges to improve the explainability of DNNs, we propose a Hierarchical Concept-based Explanation method (HCE). The main idea of our method is to adopt concept trees to construct a mapping between representations and concepts for describing model knowledge hierarchically, and then quantify the impact of hierarchical concepts from top to down based on the properties of tree structure to explain model decisions credibly. Our method provides new consistency metrics to measure the trustworthiness of explanations and explains several excellent image classification models. Main contributions of this paper can be summarized as follows:

- A visual concept tree including object level and component level is defined to model human concepts and their constituent relationships. Its properties are also discussed.
- We design Multilevel Concept Extractor and Concept Tree AutoEncoder to discover concepts in DNN gradually guided by the visual concept tree, and to generate concept trees, respectively. As a result, a systematic mapping between model knowledge and human knowledge is established.
- We define Concept Tree Shapley value that satisfies Shapley axioms to quantify the importance of concepts. It provides a hierarchical concept-based explanation for each model decision.
- For estimating the trustworthiness of explanations generally, two new instance-level consistency quantification metrics including instance-level smallest sufficient concepts and destroying concepts (ISSC and ISDC) are proposed to confirm the fidelity of explanations to model decision.

Finally, extensive experiments on ImageNet show that explanations from HCE for GoogleNet have clearer concept expressions and the average of consistency scores of explanations is 35% higher than other methods. Explanations for 6 other image classification models verify that HCE has a certain universality.

The rest of this paper is as follows: Section 2 presents related work. Section 3 introduces our explanation method. Section 4 describes experiments and results. Finally, Section 5 summarizes the work.

2 Related Works

2.1 Concept-based Explanation Methods

For improving explainability, feature-based explanations quantify the effect of input features (e.g., pixels in an image) on model decisions and give the feature importance or heatmap as the explanation. These methods typically measure the output variation when a feature is perturbed [16], [17] or they examine the gradient related to the feature [3], [26]. Sample regions that influence model decisions are explicitly labeled, but their semantics are challenging to comprehend further. Therefore, recent studies have focused on concept-based explanations since feature-based explanations are normally not consistent with human understanding.

Concept-based explanations investigate the impact of concepts on model decisions rather than individual features. Local Interpretable Model-agnostic Explanations (LIME) [4] can be regarded as a prototype of concept-based explanation when using image segments for local explanation. Kim *et al.* [6] obtains class-related concept vectors by separating concept instances in the hidden layer of DNN and explains model decisions through directional derivatives. It produces global explanations and explicitly associates concepts with model knowledge, but concept acquisition is limited. Automatic Concept-based Explanations (ACE) [7] uses superpixel segmentation and clustering to obtain concepts in an unsupervised way and begins to concern properties that concepts should satisfy. Meanwhile, some methods [8], [9] focus on extracting concepts from feature maps in the model. Currently, explanation methods care about the properties of concepts for more effective and reasonable explanations. Yeh *et al.* [11] argues that the concept set used for explanation must be sufficient to support model predictions, and it assigns contributions to concepts based on their completeness using Shapley value. Compared with other methods, COncept-based NEighbor Shapley (CONE-SHAP) [12] pays more attention to the physical and semantic neighbors of a concept when evaluating the impact, which achieves better local explanations by leveraging the local structure of concept instances.

Combining generative models to discover concepts in DNNs is a new trend. Ghandeharioun *et al.* [18] generates Concept Traversals by training a generative model from classifier's signals to seek counterfactual explanations. Gat *et al.* [19] finds concepts in the hidden layer of DNN using discrete variational autoencoder and intervention mechanism to determine the concept that can alter the target class.

Georgiev *et al.* [20] generates concept-based explanations for graph neural networks based on the concept bottleneck model and encode-process-decode paradigms [21]. Sarkar *et al.* [22] learns concepts through generative modules and self-supervised tasks to implement self-explaining DNN.

Concept-based explanations still have great potential since how to model concepts systematically is worth discussing. Concepts are closely related to human cognition laws, and modeling of concepts requires the integration of human cognition. For example, granular cognitive computing [30], [41] is a new computing paradigm that extracts different levels of abstract concepts from data. Inspired by human's granularity thinking based problem solving mechanism and the cognition law of 'global precedence', a multiple granularity knowledge expression integrating data and knowledge is proposed.

Different from the above-mentioned methods, we describe model knowledge through concept tree, and then use clear hierarchical concepts to explain model decisions systematically. The organization of our concept structure also benefits from generative model, which effectively constructs an easy-to-operate hidden space.

2.2 Explanation Methods based on Shapley Value

Methods for measuring feature impact are extended to measure the impact of concepts. It is common to compute the impact by gradient, perturbation, or Shapley value. Shapley value is worth considering as it not only satisfies several axioms, but also has wide applications.

Shapley value [10] considers the marginal contribution of a player to distribute the payoff fairly. It can calculate the impact of features for DNNs. Lee *et al.* [5] proposed SHapley Additive exPlanations (SHAP), which introduces Shapley value to explainability research in deep learning. SHAP introduces efficient computational procedures while unifying Shapley value and other popular model-agnostic methods, such as Kernel SHAP [5] based on LIME [4], Deep SHAP [5] based on DeepLIFT [23]. Expected Gradients [24] extends Integrated Gradients [25] by fusing the idea of SHAP and SmoothGrad [26]. Shapley value has also demonstrated success in providing concept-based explanations [11], [12], [13]. However, these methods cannot compute the contributions of concepts with a hierarchy. They cannot answer whether the combination of players will make a difference. Therefore, focusing on the interaction of feature combinations (i.e., concepts) by introducing a hierarchy is interesting.

Some researches based on Shapley value have studied the distribution of payoffs when players have a hierarchical structure. Shapley hierarchical value [15] investigates the situation when players have different permissions (e.g., officers and soldiers). It models the hierarchical relationship of players in the coalition as a directed graph or tree, where each node represents a player, and then calculates the payoffs of players under restricted permutations. Shapley level value [14] continuously groups players, which generates the next player partition based on the previous player partition and more constraints, i.e., the player permutations owned by the next partition are a restricted subset of permutations owned by the previous one. The method calculates players' contributions on the final partitions' permutations. Other methods investigate the transformation of hierarchical structures into level structures [27], or the case when players have different weights [28].

In the above methods, all players in a hierarchy (e.g., all nodes in a tree) constitute a complete set of players. We extend it to a more detailed situation that the node set at any level of the tree is treated as a complete set of players, and the total contribution of nodes in each level is the same.

2.3 Trustworthy Explanation

Evaluating the trustworthiness of an explanation is difficult because there is usually no ground truth available in the dataset to verify the accuracy of the explanation on model decisions. It is an effective way to address this problem by evaluating whether the explanation is faithful to the model decision. We refer to this approach as consistency. For concept-based explanations, consistency means measuring whether important concepts obtained by an explanation method are also important concepts considered by the explained model.

How to measure consistency quantitatively remains an open question. Some methods [7], [12] use the smallest sufficient concepts (SSC) and smallest destroying concepts (SDC) since adding or removing important concepts has a more obvious impact on model predictions. When adding or removing important concepts in images from several classes, SSC or SDC counts the percentage of modified images that maintain the original image predictions. Each class determines the ranking of important concepts based on the average impact of concepts. However, SSC and SDC are commonly employed for evaluating global explanation, which explains an entire model, class, or sets of samples by leveraging concepts of equal importance across diverse samples.

To our knowledge, there is currently no accepted metric to evaluate concept-based local explanations. We evaluate local explanation using two criteria: firstly the importance of a concept varies across samples, and secondly each sample should have its own concept ranking.

3 Method

To enhance human understanding and trust in DNNs, this study explains predictions of DNNs through concept tree that is consistent with human knowledge. This section describes our algorithm in detail. The problem definition is shown in Section 3.1. The explanation process of HCE is shown in Fig. 1. We do not initially know what concepts make up the concept tree of an image class, other than its abstract structure, or logical concept tree (Fig. 1(a1)), which is defined in Section 3.2. Then, Multilevel Concept Extractor segments images gradually to automatically extract multilevel concepts in DNN (Fig. 1(b1)), as discussed in Section 3.3. Concept Tree AutoEncoder reconstructs the whole-part relationships between concepts for each image (Fig. 1(b2)), as discussed in Section 3.4. Thus, we can obtain the class concept tree from image concept trees, where the hierarchical structure of concepts is explicit, and each concept contains different instances (Fig. 1(a2)). We explain

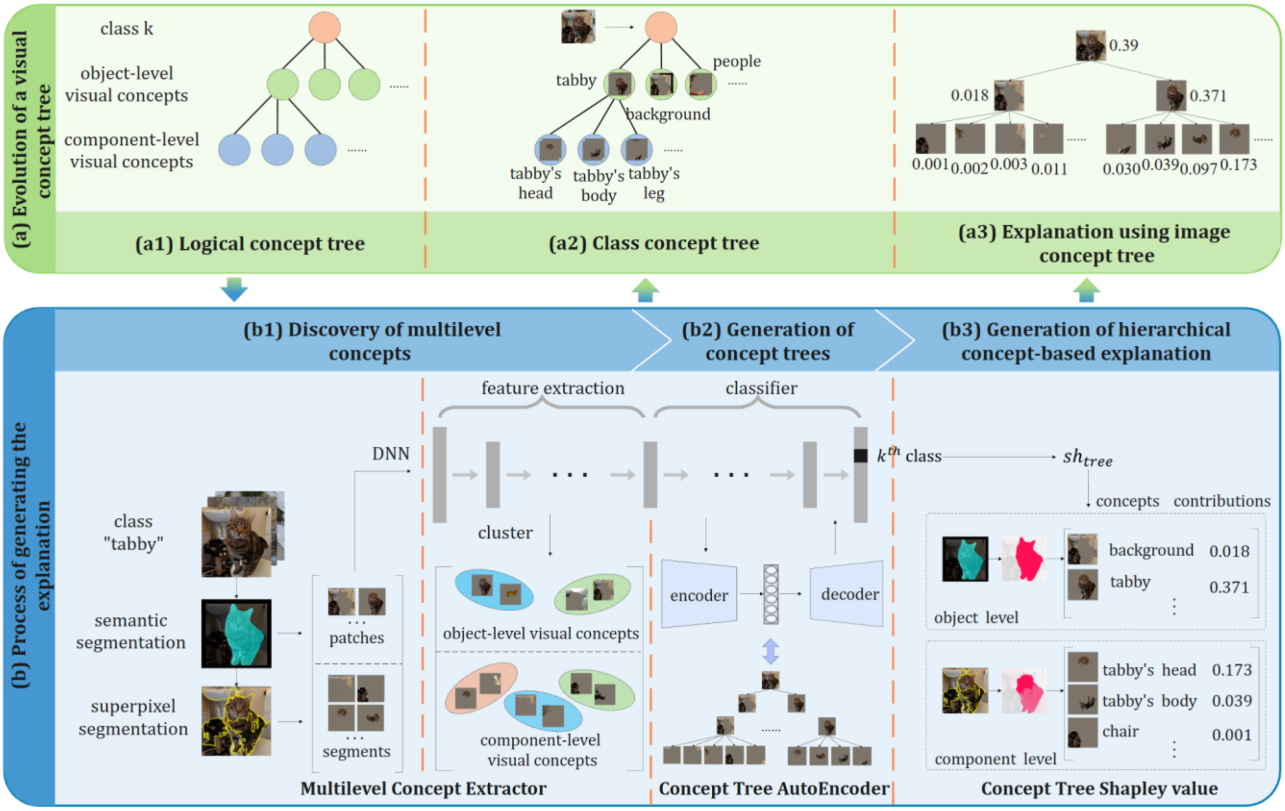
(a) Evolution of a visual concept tree
class k
object-level visual concepts
component-level visual concepts
(a1) Logical concept tree
tabby
people
background
tabby's head
tabby's body
tabby's leg
(a2) Class concept tree
0.39
0.018
0.371
0.001
0.002
0.003
0.011
0.030
0.039
0.097
0.173
(a3) Explanation using image concept tree
(b) Process of generating the explanation
(b1) Discovery of multilevel concepts
(b2) Generation of concept trees
(b3) Generation of hierarchical concept-based explanation
class "tabby"
semantic segmentation
superpixel segmentation
DNN
patches
segments
feature extraction
classifier
cluster
object-level visual concepts
component-level visual concepts
encoder
decoder
k^{th} class
sh_{tree}
concepts
contributions
background 0.018
tabby 0.371
object level
tabby's head 0.173
tabby's body 0.039
chair 0.001
component level
Multilevel Concept Extractor
Concept Tree AutoEncoder
Concept Tree Shapley value

Fig. 1. Overview of HCE. (a) Green part at the top shows the evolution of a visual concept tree, including (a1) logical concept tree, (a2) class concept tree and (a3) explanation using image concept tree. (b) Blue part at the bottom is the process of generating the explanation, including (b1) discovery of multilevel concepts through Multilevel Concept Extractor, (b2) generation of concept trees via Concept Tree AutoEncoder and (b3) generation of hierarchical concept-based explanation using Concept Tree Shapley value.

model classification predictions of images by Concept Tree Shapley value (Fig. 1(b3) and Fig. 1(a3)), as discussed in Section 3.5. Finally, new consistency evaluation indexes are presented in Section 3.6.

3.1 Problem Formulation

In the context of image classification, given an input image x with its class label y, $\hat{y}$ is the prediction of a DNN f. f usually consists of a feature extraction module fe and a classifier module cl. The high-level feature representation z is obtained from x by fe, i.e., $z = fe(x)$. The classifier cl takes the representation z as the input and its output is often normalized by activation function like $softmax$ to obtain a classification result vector p. p can be regarded as a probability distribution, which denotes the probability of x (or z) belonging to each class. $\hat{y}$ actually represents the class label of the largest probability in p. The unexplainable z is the serious obstacle to explain $\hat{y}$.

Our goal is to discover the mapping of representation z to human hierarchical concepts (object-level concepts C_{ob} and component-level concepts C_{co} in concept tree CT), and then quantify the impact of hierarchical concepts in DNN by Shapley value to explain predictions of DNN. Definitions of different levels of concepts and concept trees are given in Section 3.2.

3.2 Definition of a Visual Concept Tree

We expect to find concepts in DNN while concepts are hierarchical. Therefore, in order to describe the general form of hierarchical concepts in the visual domain, we define two levels of concepts and use concept tree to describe the whole-part relationships between concepts.

The cognition process of humans towards an event in nature is commonly from global information of the event to local information, called ‘global precedence’ [29], [30], thus humans are more sensitive to global topological features. In a visual case, humans always focus on the global information of an image when first viewing it, thus different entity objects in the image can be identified rapidly (e.g., cat, person, or more generally, the background).

Definition 1 Object-level visual concepts. *We define the recognizable visual objects as object-level visual concepts C_{ob}, which help humans recognize image structure and make basic judgments (e.g., what is in the image). $C_{ob} = \{c_{ob_1}, c_{ob_2}, ...\}$, where c_{ob} is a specific object-level concept.*

For an object-level visual concept c_{ob} (e.g., cat), the object parts it contains (e.g., cat's head, cat's tail, etc.) possess semantics that describe local information and satisfy the minimum requirements for human cognition. Object parts help humans make more expert judgements (e.g., what is the class of a cat).

Definition 2 Component-level visual concepts. *We define the object parts as component-level visual concepts $C_{co} = \{c_{co_1}, c_{co_2}, ...\}$, where c_{co} is a specific component-level concept.*

A tree (T) is a finite set of $tn(tn \geq 0)$ nodes. It is an empty tree when $tn = 0$. In any non-empty tree, there should be: (i) There is one and only one node called the root node. (ii) When $tn > 1$, the non-root node can be divided into tm ($tm > 0$) disjoint finite sets $T_1, T_2, \dots, T_{tm}$. T_{tm} is called a subtree.

Definition 3 Visual concept tree. *A visual concept tree consists of hierarchical visual concepts and has three levels. The root node of the concept tree is the image class k, which consists of object-level concepts C_{ob}. C_{ob} consists of component-level concepts C_{co}.*

In general, the parent concept in the tree contains child concepts. Since we are now uncertain about concepts in the concept tree, we also call it logical concept tree. Fig. 2 shows an example.

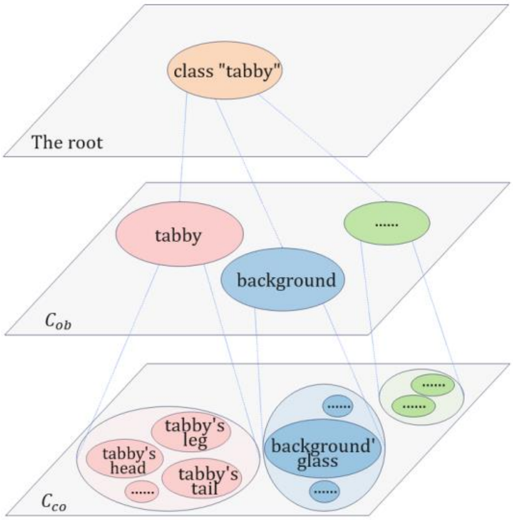

Fig. 2. Visual concept tree of class ‘tabby’.

Concept tree has a natural hierarchical structure, which provides a concrete and clear way to describe the whole-part relationships between concepts. Compared with knowledge graph, concept tree concerns more about the constituent relationship of concepts. There are two implicit properties in the concept tree.

Property 1. *The visual concept tree for different image classes should consist of different hierarchical concepts, i.e., class concept tree.*

Property 2. *Concepts existing in two images of the same class may be different, thus the visual concept tree of an image is part of the class concept tree, i.e., image concept tree. The root node is the image itself.*

Meanwhile, concepts in the concept tree obviously have the following properties.

Property 3 Recursive additivity. *We regard a concept at the position of the parent node as c_{parent} and a concept at the child node as c_{child}. c_{parent} should be the sum of all related c_{child}, as follows:*

$$cin_{parent} = \sum cin_{child}\,, \tag{1}$$

where cin denotes a concept instance of image x, cin_{parent} is an instance of c_{parent} and cin_{child} is an instance of c_{child}.

Specifically, image x consists of all its object-level concept instances, and an object-level concept consists of all its component-level concept instances.

Property 4 Recursive equivalence. *The influence of the parent concept c_{parent} under any task should be equal to that of all its child concepts united together, as follows:*

$$h(cin_{parent}) = h(\sum cin_{child}), \tag{2}$$

where $h(\cdot)$ is a task.

For example, when $h(\cdot)$ is a DNN f, the classification probability of a parent concept instance in image x should be the same as the probability obtained from the union of all related child concept instances. Concept instances are the actual expression of a concept, and they have different forms (superpixels or representations) in different scenes (image or DNN).

3.3 Discovery of Multilevel Concepts

How to use the logical concept tree automatically to discover comprehensible model knowledge is one key step of HCE. We project instances of concepts of different levels in images onto representations, and then cluster the representations to discover multilevel concepts in DNN. In this way, each concept found in DNN can locate its concept instances in images. Image segmentation is a common way of detecting concept instances in images. However, to the best of our knowledge, semantic segmentation algorithms rarely consider hierarchical segmentation due to the annotation cost, meanwhile, superpixel segmentation algorithms are more efficient but similar pixels still have a greater impact on results. Therefore, we design a Multilevel Concept Extractor that conducts superpixel segmentation on the results of semantic segmentation, progressively separating images to discover concepts of different levels in DNN. Fig. 3 shows the process of the extractor. Through the extractor, we know what concepts should exist in the concept tree. Here, a cluster obtained by the clustering algorithm represents a concept.

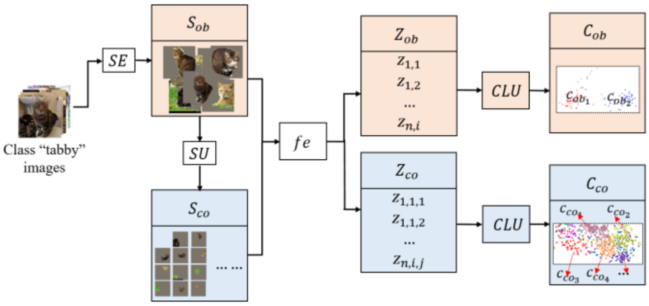

Fig. 3. The process of Multilevel Concept Extractor.

In Fig. 3, we regard $X^k = \{x_1^k, x_2^k, \dots x_n^k\}$ as images belonging to class k (i.e., tabby), and we ignore the superscript k later in the paper for simplicity. First, object-level segmentation uses a semantic segmentation algorithm SE to obtain image patches S_{ob}, as follows:

$$S_{ob} = SE(X) = \{s_{1,1}, s_{1,2}, \dots s_{n,i}\}, i \in N^+, \tag{3}$$

where s is a segment of an image, $s_{n,i}$ is a patch in the image, e.g., $s_{1,1}$ is the first patch of the first image. Patches in

S_{ob} are semantically rich, which can be directly considered as concept instances of object-level concepts.

Then, the component-level segmentation uses a superpixel segmentation algorithm SU further segments S_{ob} to obtain image superpixels S_{co}, as follows:

$$S_{co} = SU(S_{ob}) = \{s_{1,1,1}, s_{1,1,2}, \dots, s_{n,i,j}\}, j \in N^+, \quad (4)$$

where $s_{n,i,j}$ is a superpixel of an image, e.g., $s_{1,1,1}$ is the first superpixel of patch $s_{1,1}$. However, S_{co} cannot be directly regarded as instances of component-level concepts, as superpixels do not always possess complete semantics. The richness of semantics in a superpixel depends on the performance and parameter settings of SU, so a complete component-level concept c_{co} may consist of several superpixels.

We use DNN's feature extraction module fe to obtain representations Z_{ob} and Z_{co},

$$Z_{ob} = fe(S_{ob}) = \{z_{1,1}, z_{1,2}, \dots z_{n,i}\}, \quad (5)$$

$$Z_{co} = fe(S_{co}) = \{z_{1,1,1}, z_{1,1,2}, \dots z_{n,i,j}\}, \quad (6)$$

where $z_{n,i}$ denotes the representation associated with $s_{n,i}$. z represents DNN's understanding of the segment. To match the input of fe, segment s is essentially equivalent to a blank image containing s and the relative position of s is unchanged. Here, we do not use enhanced image segments, e.g., enlarge the superpixel to image size, because it might make segments lose the spatial information intuitively.

Last, we use a clustering algorithm CLU to discover concepts of different levels from representations. Object-level concepts C_{ob} and component-level concepts C_{co}, which describe model knowledge, respectively come from Z_{ob} and Z_{co}. For object-level concepts C_{ob}, CLU determines concepts to which concept instances Z_{ob}(and S_{ob}) belong. For component-level concepts C_{co}, CLU is more interested in determining which z_{co} can constitute a component-level concept instance cin_{co}. For example, cin_{co_1} is the instance of concept c_{co_1}, $c_{co_1} \in C_{co}$ and $cin_{co_1} = \{z_{1,1,1}, z_{1,1,2}\}$. Obviously, the extractor is not limited to specific semantic segmentation, superpixel segmentation, clustering algorithm and DNN. Overall, concepts obtained are traceable and comprehensible.

3.4 Generation of Concept Trees

The discovered multilevel concepts should not be confused with concept tree, as concept instances in DNNs, i.e., representations z, lack the constituent relationship. Obviously, concept instances do not exhibit recursive additivity and recursive equivalence (Property 3.3 and Property 3.4) after the feature extraction of DNN. Mapping representations to another latent space is an effective strategy for imposing constraints on representations, using generative models like AutoEncoder. Therefore, aiming to reconstruct the constituent relationships between multilevel concepts, we propose a Concept Tree AutoEncoder, which uses properties of concepts to constrain the generation of encoding representations. Through the autoencoder, we can obtain hierarchical concepts for each image in DNN. In other word, each image gets its own image concept tree. Meanwhile, we can summarize image concept trees to obtain class concept tree, where the concept hierarchy is relatively explicit, and each concept contains different instances.

The autoencoder consists of Encoder $E(Z; W, b)$ and Decoder $D(EZ; W, b)$, both E and D are feed forward neural networks, W are weights and b are biases. E encodes the representation z as ez and D gets reconstructed representation $\hat{z}$, i.e., $\hat{z} = D(ez) = D(E(z))$. ez is essentially a vector, which should satisfy the additivity and equivalence:

$$ez_{parent} = \sum ez_{child}, \quad (7)$$

$$h(ez_{parent}) = h(\sum ez_{child}), \quad (8)$$

where encoding representation ez_{parent} is a concept instance of c_{parent}, ez_{child} is a concept instance of c_{child} or a part and $h(\cdot)$ is a task. Here, we use the summation of vectors representing ez to achieve additivity. This is because linear superposition is more similar to the additivity of concept instances on the image and is the simplest method, although we know that there are many other methods to achieve the addition of vectors, such as vector stitching or function mapping.

In order to make $\hat{z}$ approximate z, and impose additivity and equivalence on ez, the objective function is defined as

$$\underset{E,D}{\operatorname{argmin}}\, L_{additivity}(E) + L_{recontrust}(E, D) + L_{equal}(D). \quad (9)$$

The first item is additivity loss $L_{additivity}$ which aims to assign the additivity of concepts to ez. The feature extraction module fe has a complex structure and numerous parameters, so a small change in the input may lead to a large difference in the output vector. The blank part of the segment s can be regarded as special information, which disturbs the additivity of the representation z. We use $L_{additivity}$ to constrain the encoder so that ez generated by the encoder is additive. Specifically, $L_{additivity}$ uses Mean Square Error (MSE) to minimize the difference between ez_{parent} and the sum of all its ez_{child}, as follows:

$$L_{additivity}(E) = \mathbb{E}_{ez\sim EZ}\left[MSE(ez_{parent}, \sum ez_{child})\right]. \quad (10)$$

ez cannot be recognized by the classifier of DNN, which makes us unable to estimate the impact of ez on the model prediction. If we reconstruct the ez back to z (actually $\hat{z}$), we can evaluate the effect of ez without modifying the structure of DNN, and the change of ez will also be faithful to DNN. Thus, the second item of the objective function is reconstruction loss $L_{recontrust}$, which aims to minimize the difference between the predictions of DNN generated by z and $\hat{z}$ respectively. We know that the prediction here is the probability distribution p and Kullback-Leibler (KL) divergence is often used to measure the distance between two probability distributions, so $L_{recontrust}$ is defined as

$$L_{recontrust}(E, D) = \mathbb{E}_{z\sim Z}[D_{KL}(p||\hat{p})], \quad (11)$$

where p represents the classification probability distribution of z, e.g., probability values on one thousand image classes, $p = softmax(cl(z))$, $\hat{p}$ is the probability distribution of $\hat{z}$, $\hat{p} = softmax(cl(D(E(z))))$, D_{KL} is KL divergence.

The neural network strives to obtain the global optimal solution, so additivity is limited by the trade-off of network parameters. Meanwhile, as stated before, modules of DNN may produce unanticipated larger differences in results due to small input differences. As a result, the additivity of ez currently obtained fails to satisfy the equivalence further, which affects the construction of concept trees. Therefore, to satisfy the equivalence, the third term of the objective function is the equivalence loss L_{equal}, which uses KL divergence to minimize the difference between predictions obtained from ez_{parent} and $\sum ez_{child}$, as follows:

$$L_{equal}(D) = \mathbb{E}_{ez\sim EZ}\left[D_{KL}(p_{parent}||p_{children})\right], \quad (12)$$

where $p_{parent} = softmax(cl(D(ez_{parent})))$, $p_{children}$ comes from $\sum ez_{child}$ in the same way.

3.5 Generation of Hierarchical Concept-based Explanation

Shapley value [10] considers the marginal contributions of coalition players and equitably calculates players' contributions to distribute payoffs, as follows:

$$\phi_v(w) = \sum_{A\subseteq F\setminus\{w\}} \frac{|A|!(|F|-|A|-1)!}{|F|!}\left[v(A\cup\{w\}) - v(A)\right], \quad (13)$$

where $\phi_v(w)$ denotes the final contribution of player w using the utility function $v(\cdot)$, $|\cdot|$ operator denotes the number of players in a finite set, F is the coalition of all players, $F\setminus\{w\}$ denotes that F does not contain player w, $A\cup\{w\}$ denotes that player w joins a subset A of F and $v(A)$ denotes the payoff of coalition A. When Shapley value applies to image classification, a counterfactual approach is often used to calculate the impact of image segments. In this case, image segment s is equivalent to player w, image x corresponds to union F, x_S corresponds to A and x_S denotes a new image consisting of a subset S of x. Specifically, image x has $|F|$ segments, which can be regarded as a coalition, while segments can be regarded as players. The classification probability value of s or x_S on class k is treated as the contribution of a player or subset. The contribution of s with respect to x_S is represented as:

$$\Delta v_k(s, x_S) = v_k(x_{S\cup\{s\}}) - v_k(x_S). \quad (14)$$

Later we will write v_k as v.

In HCE, we can calculate the impact of a concept directly using ez without repeating the feature extraction, as follows:

$$v(ez) = softmax_k(cl(D(ez))), \quad (15)$$

$$\phi_v(ez) = sh(ez, F), ez \in F, \quad (16)$$

where $softmax_k(\cdot)$ calculates the classification probability value on class k, sh denotes Shapley value and F is the set containing ez here.

Unlike traditional Shapley value algorithms, which calculate payoffs on permutations from the final partition of players, our calculation should satisfy two constraints based on the equivalence of concepts: (i) the contribution of a parent concept should be consistent with the sum of the contributions of all its child, as follows:

$$\phi_v(ez_{parent}) = \sum \phi_v(ez_{child}). \quad (17)$$

(ii) the sum of contributions of nodes in each level of the tree should be consistent, as follows:

$$\phi_v(ez_x) = \sum \phi_v(ez_{ob}) = \sum \phi_v(ez_{co}), \quad (18)$$

where ez_x is the encoding representation of image x, ez_{ob} is the representation of an instance belonging to the object-level concept c_{ob} and ez_{co} corresponds to c_{co}.

In order to satisfy the above constraints and enhance the computational efficiency, we design Concept Tree Shapley value or sh_{tree} to calculate the effect of hierarchical concepts top-down. It combines image concept tree and Shapley value. When computing contributions of concepts in image x, the classification probability $v(ez_x)$ is first calculated as the total contribution $\phi_v(ez_x)$. Then, we let $v(\sum ez_{ob}) = \phi_v(ez_x)$ and the contribution $\phi_v(ez_{ob})$ of each object-level concept instance is calculated by sh. Finally, we let $v(\sum ez_{co}) = \phi_v(ez_{ob})$ and use sh calculate the contribution of each ez_{co}. Therefore, the core of sh_{tree} is a prior calculation constraint:

$$v(\sum ez_{child}) = \phi_v(ez_{parent}). \quad (19)$$

Therefore, the sh_{tree} is defined as

$$\phi_v(ez_i) = \sum_{A\subseteq F\setminus\{ez_i\}} \frac{|A|!(|F|-|A|-1)!}{|F|!}\Big[v(ez_i + \sum_{ez_j\in A} ez_j) - v(\sum_{ez_j\in A} ez_j)\Big]. \quad (20)$$

Unlike the conventional calculation, F is the set consisting of ez_i and its brother concept instances. It is possible that a number of ez_{co} form a real component-level concept instance, so we do not directly use component-level concept instances in the computation. The contribution of component-level concept instance can also be obtained by summing up $\phi_v(ez_{co})$.

We summarize HCE in Algorithm 1, which discovers hierarchical concepts in DNN and obtains explanations for individual samples.

Algorithm 1 Hierarchical Concept-based Explanation.

```
Input: image set $X^k = \{x_1^k, x_2^k, \dots x_n^k\}$ belongs to class $k$.
Output: the explanation for the DNN prediction of $x_i^k$.
Multilevel Concept Extractor:
  for $i \leftarrow 1$ to $n$ do
    Compute patches $S_{ob_i}$ of $x_i^k$ using $SE$.
    for $j \leftarrow 1$ to $len(S_{ob_i})$ do
      Compute superpixels $S_{co_{i,j}}$ using $SU$.
    end for
  end for
  Compute representations $Z_{ob}$ and $Z_{co}$ from $S_{ob}$ and $S_{co}$
  using $fe$ of $f$.
  Compute $C_{ob}$ from $Z_{ob}$ using $CLU$.
  Compute $C_{co}$ from $Z_{co}$ using $CLU$.
Concept Tree AutoEncoder:
  Compute representations $Z_x$ from $X^k$ using $fe$.
  for epoch:
    $Z_{parent} \leftarrow Z_{ob}$, $Z_{child} \leftarrow Z_{co}$.
    Train Encoder $E$ and Decoder $D$ with data $Z_{parent}$ and
    $Z_{child}$ according to Eq. (9).
    $Z_{parent} \leftarrow Z_x$, $Z_{child} \leftarrow Z_{ob}$.
    Train Encoder $E$ and Decoder $D$ with data $Z_{parent}$ and
    $Z_{child}$ according to Eq. (9).
  end for
  Compute $EZ_x$, $EZ_{ob}$ and $EZ_{co}$ from $Z_x$, $Z_{ob}$ and $Z_{co}$ using
  $E$. $EZ$ is the set of encoding representation $ez$.
Concept Tree Shapley value:
  Select an image $x_i^k$ from $X^k$.
  $\phi_v(ez_{x_i}) \leftarrow v(ez_{x_i})$.
  $v(\sum ez_{ob_i}) \leftarrow \phi_v(ez_{x_i})$.
  for $j \leftarrow 1$ to $len(EZ_{ob_i})$ do
    Compute $\phi_v(ez_{ob_{i,j}})$ according to Eq. (20).
    $v(\sum ez_{co_{i,j}}) \leftarrow \phi_v(ez_{ob_{i,j}})$.
    for $k \leftarrow 1$ to $len(EZ_{co_{i,j}})$ do
      Compute $\phi_v(ez_{co_{i,j,k}})$ according to Eq. (20).
    end for
  end for
```

Furthermore, sh_{tree} also satisfies the following four basic axioms of Shapley value, which can also be regarded as explainability axioms.

Efficiency: sh_{tree} automatically satisfies the efficiency, as follows:

$$v(ez_x) = \phi_v(ez_x) = \sum_i \phi_v(ez_{ob_i}) \\ = \sum_i \sum_j \phi_v(ez_{co_{i,j}}), \tag{21}$$

where ez_{ob_i} is the instance of ith object-level concept in image x and $ez_{co_{i,j}}$ can be treated as the instance of jth component-level concept under ez_{ob_i}.

Symmetry: For any equivalent nodes c_i and c_j, they are brother nodes in the same level according to the structure of concept tree. These two nodes lie in the same set F and satisfy $\phi_v(c_i) = \phi_v(c_j)$ under any permutation.

Dummy: For a concept c outside concept tree, theoretically the contribution $\phi_v(c) = 0$. However, v is a DNN and $\phi_v(c)$ will only converge to 0, i.e., $\mathbb{E}[\phi_v(c)] = 0$.

Additivity: Assuming that task h_i computes the contribution of ez on class k_i, task h_j on class k_j and i $\neq j$. $\phi_{h_i+h_j}(ez) = \phi_{h_i+h_j}(ez) + \phi_{h_i+h_j}(ez)$ is true for any $ez \in F$, where F consists of ez and brothers of ez.

3.6 Consistency Evaluation Metrics

This paper concentrates on concept-based local explanation, thus the quality of the explanation is discussed in this section. We design metrics to quantify the credibility of explanations, which evaluate the quality of explanations, since there is no accepted metric. An explanation is considered of high quality and trustworthy when the concept importance ranking obtained by the explanation closely aligns with that of DNN. Therefore, to measure the consistency of explanation on each sample, we design and quantify instance-level smallest sufficient/destroying concepts (ISSC/ISDC) by extending SSC/SDC to the instance-level condition.

In these two metrics, each image should have its own concept ranking and the ranking only includes concepts existing in the image. Performing in the order of decreasing importance of concepts, two metrics can show the trends of classification probability on target class respectively. Therefore, the metric can be quantified as the consistency score, which is scored if the classification probability increases/decreases by less than the last value when a concept is added/deleted in image x, as follows:

$$ISSC_x^d = \frac{\sum_{j=1}^{d-1} \delta(\Delta v_j - \Delta v_{j+1})}{d-1}, \tag{22}$$

$$ISDC_x^d = \frac{\sum_{j=1}^{d-1} \delta(\Delta v_{j+1} - \Delta v_j)}{d-1}, \tag{23}$$

where d is the number of concepts, Δv_j is the variation of classification probability on target class when adding/removing the jth concept and $\delta(\cdot)$ is an indicator function:

$$\delta(u) = \begin{cases} 0, u \leq 0 \\ 1, u > 0 \end{cases}. \tag{24}$$

In short, the higher the score, the more trustworthy the explanation. Hence, class-level ISSC and ISDC can be derived, which use image set belonging to class k, i.e., $X^k = \{x_1^k, x_2^k, \ldots x_n^k\}$. Scores are defined as

$$ISSC_{X^k}^d = \frac{\sum_{i=1}^{n} ISSC_{x_i}^d}{n}, \tag{25}$$

$$ISDC_{X^k}^d = \frac{\sum_{i=1}^{n} ISDC_{x_i}^d}{n}. \tag{26}$$

For simplicity, we ignore the superscript k of x_i.

Here, since the concept contribution calculated by Shapley value considers the marginal contributions of concepts, it is reasonable to add or remove top-d concepts in images and use the variation in probability to measure consistency.

4 Experiments And Results

We focus on image classification task which HCE is suitable for. Firstly, we state experimental settings (Section 4.1). Then, the efficiency of our explanation method is evaluated (Section 4.2). Next, we analyze the consistency of explanations quantitatively (Section 4.3). In the end, the performance of concepts is examined (Section 4.4) and the universality of HCE is discussed (Section 4.5).

4.1 Experimental Settings

4.1.1 Datasets and Explained Models

Our experiments conduct on the ILSVRC2012 dataset (ImageNet) [31] from where we select more than 20 classes. These chosen classes are included in classes supported by the pretrained semantic segmentation model. In a class, about 50 images can provide good explanations.

We use several CNN classification models pretrained on ImageNet as explained models. We mainly use GoogleNet [32] to evaluate the efficiency and consistency of explanations obtained by HCE. Meanwhile, AlexNet [33], Inception-V3 [34], DenseNet-121 [35], ResNet-101 [36], EfficientNet_b0 [37] and VIT_b_16 [38] are employed to evaluate the universality. All pretrained models are provided by PyTorch.

4.1.2 Comparison Methods

To show the advantages of HCE, we compare with explanation methods, among which DEEP SHAP [5] and Expected Gradients [24] are local explanation methods based on input features. LIME [4] is a classical local explanation method. ACE [7] is the global concept-based explanation method, mainly to reflect the lack of explainability of the global explanation method to specific instances here. CONE-SHAP [12] is the most intuitive concept-based explanation method found so far.

4.1.3 Settings of HCE

For images in the same class, Multilevel Concept Extractor acquires concept instances through image segmentation algorithms, and it employs a clustering algorithm to determine concepts to which instances belong. The pretrained semantic segmentation model should be able to segment the subject object and background in each image at least. Here, we use deeplabv3 [39], which supports 21 semantic classes, provided by PyTorch. SLIC [40] is the superpixel segmentation algorithm we use and each object is divided into a maximum of 15 superpixels, since the number of superpixels exponentially affects the time cost of Shapley value. K-Means is used for clustering. Since images in a class have similar structure, the number of clusters is set to 20 for component-level concepts. The number of object-level concepts ranges from 2 to 6.

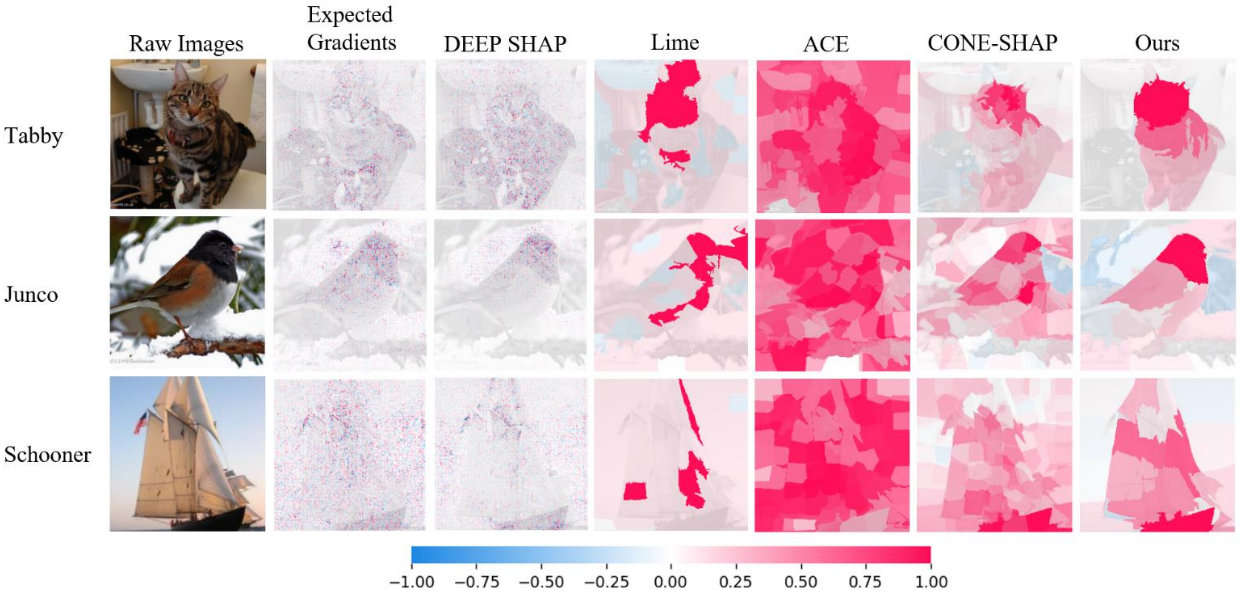

Fig. 4. Explanation heatmaps of different methods.

We use feature vectors obtained from the feature extraction module (e.g., the average pooling layer of GoogleNet) as the input to Concept Tree AutoEncoder. The dimension of each layer of the encoder is half of the previous one and the final compressed dimension is 64, while the decoder is the opposite. Both the encoder and the decoder are 4-layer fully connected networks. It first iterates for 50 epochs, after which the iteration stops if the loss does not decrease within 50 epochs. We use Adam optimizer.

4.2 Evaluating the Efficiency of HCE

We visually compare the explanation effect of HCE with other methods in Section 4.2.1. Then, we show an example of the explanation obtained by HCE in Section 4.2.2.

4.2.1 Evaluating Explanation Effects

To compare the explanation effect of HCE with other methods, we use the tool provided by the SHAP package (https://github.com/slundberg/shap) to generate explanation heatmaps. We highlight the five most important superpixels for LIME. To obtain heatmaps of ACE, we assign TCAV scores to image segments, which follows the setting of CONE-SHAP. We reimplement CONE-SHAP and keep its way of constructing heatmaps. HCE assigns the contribution of the encoding representation *ez* to the superpixel to get heatmaps. Fig. 4 shows heatmaps, in which the importance scores of concepts are normalized between -1 and 1 by dividing the absolute value of the largest score. Deeper red indicates more positive effects and deeper blue indicates more negative.

We present HCE at the component level, which is more commonly seen in other methods. Compared with all methods, HCE considers the hierarchy of concepts, so heatmaps look clearer and explanations are more in line with human cognition. Specifically, we choose images from three classes to show the difference in explanations, while the three classes have different scale size in reality. In Fig. 4, concept-based explanations are more semantically rich and human-friendly than feature-based explanations. LIME's explanations may be unclear if not combined with the original image. ACE focuses on global explanation and sometimes gets relatively close scores for important concepts, so ACE is not precise when explaining a single image. Compared with CONE-SHAP, the heatmap of HCE seems a bit rougher, but it contains clearer semantics and is easier to understand. HCE uses concept tree, which clarifies the whole-part relationships between concepts and makes the boundaries of concepts in the explanation clearer. Concept tree enhances expressions of concepts, such as the head of a cat in the first row of Fig. 4.

4.2.2 Explaining Model Decision

In general, HCE obtains local concept-based explanations, which are hierarchical. For example, given GoogleNet and an image set of class ‘sorrel’, HCE explains the classification result of the image in the set. The explanation consists of images of hierarchical concept instances and the quantitative impacts of concepts on the model decision. Fig. 5 shows the example.

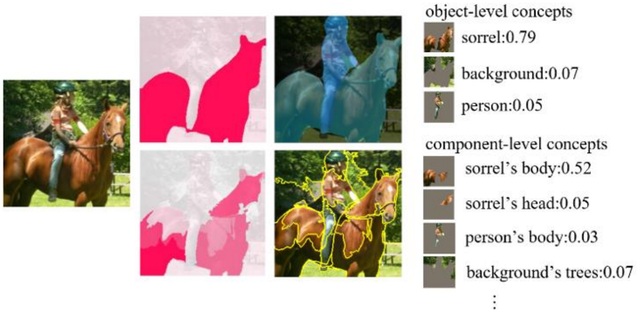

Fig. 5. An explanation of a ‘sorrel’ image.

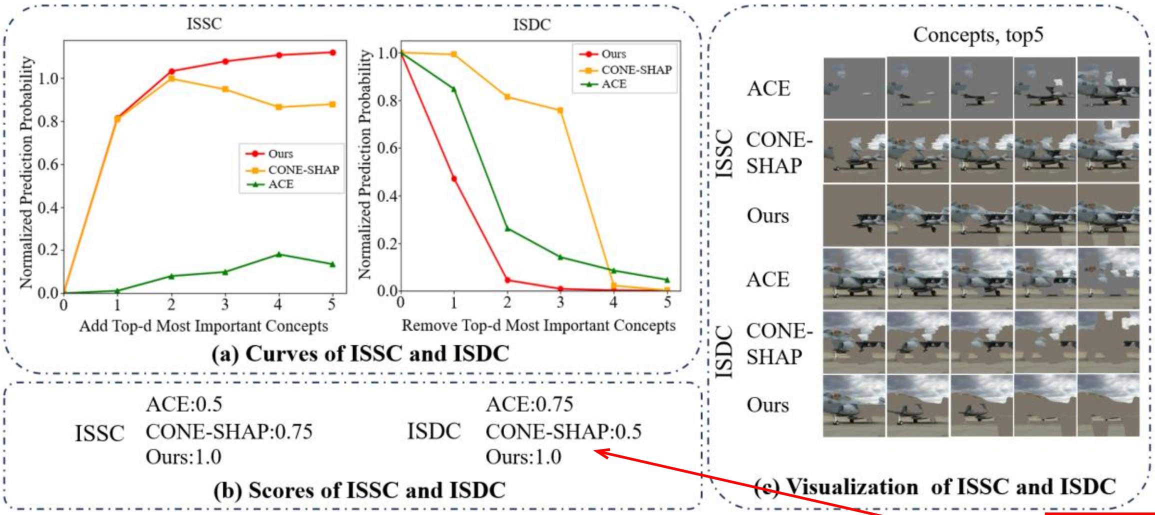

Fig. 6. A 'warplane' image on ISSC and ISDC.

为什么这里ACE和CONE-SHAP不是1

In Fig. 5, for the explained image, HCE first provides normalized hierarchical explanation heatmaps, which indicate the importance of each segment to the model decision. Then, we give segmentation images of heatmaps, so instances of concepts can be found from segmentation images. Finally, importance scores of concept instances are given, which are classification probability values. Scores of component-level concept instances can be obtained from superpixels.

4.3 Evaluating Consistency of Explanations

We use two metrics mentioned in section 3.6, ISSC and ISDC, to measure the consistency of explanations. We validate whether important concepts calculated by HCE are also ones considered important by DNN. We evaluate on two levels, instance level (an image) and class level (image set). The higher the consistency score, the more trustworthy the explanation.

4.3.1 *Instance-Level Evaluation*

Given an image, we observe the variation in the model classification probability on the target class when the top-5 important concepts of the image are added/removed. The ranking of concept importance comes from the sum of concept instances' importance in the image. Fig. 6 shows the results of comparing HCE with ACE and CONE-SHAP, including the visualization of adding and removing concepts. ACE and CONE-SHAP use three different levels of resolution to capture texture, object parts, and objects. In this experiment, HCE compares at the component level while ACE and CONE-SHAP are at the level of object parts.

ISSC adds top-d important concepts to the blank image, while ISDC removes top-d important concepts from the original image. For HCE and CONE-SHAP, scores of important concepts in the image are summed up by scores of segments, and we use scores of globe concepts for ACE. All methods use concepts existing in the image only. For ISSC, the result curve should show an increasing trend, steep first and then slow, while ISDC has a downward trend. In Fig .6(a), the curve of HCE is as expected, while trends of other methods have obvious waviness. The consistency scores also reflect that the explanations obtained by HCE have better consistency (Fig. 6(b)). On the visualization part of the ISSC (Fig. 6(c)), the top-5 important concepts of HCE contain more information about the 'warplane', so our progressive concept discovery reduces the interference of irrelevant concepts effectively and obtains more precise concepts and better consistency.

4.3.2 *Class-Level Evaluation*

$ISSC^d_{X^k}$ and $ISDC^d_{X^k}$ evaluate the consistency of explanations at the class level. We compare with the other methods on more than 20 classes. The number of comparison concepts is set to 5, as the top-5 important concepts have an obvious impact on model decisions, and in some images, there may not be many concepts present. Other settings are the same as the instance level. Table 1 shows the results on 24 classes.

In Table 1, images in the first 18 classes are easy to apply semantic segmentation, and for each class less than 6% of images do not support semantic segmentation. In the next 5 classes, the rate of images for which semantic segmentation did not work ranges from 34% to 84% ('police van' is 34%, 'halftrack' is 72%, 'jeep' is 78%, 'park bench' and 'fire engine' are 84%). Images of the last class cannot undergo semantic segmentation. Overall, HCE achieves the highest scores in most classes. On average, compared with baselines, HCE achieves at least 35% improvement on $ISSC^d_{X^k}$ and 37% on $ISDC^d_{X^k}$. In the class 'Siamese cat' and 'bison', HCE does not achieve a breakthrough probably since objects of such images have many similar pixels. As a result, the effect of our concept discovery on these objects is not very satisfactory and the explanation suffers. For example, the 'Siamese cat' generally has only two colors, black and white. The tail and legs of the cat are both black and long strip while most areas of the cat's body are white. Clustering can easily divide pixels of the same color into a cluster, which affects the recognition of component-level concepts. Overall, the consistency scores from our proposed metrics indicate that the explanations from HCE are more trustworthy.

TABLE 1
SCORES OF ISSC AND ISDC ON CLASS LEVEL

	$ISSC^{d}_{X^k}, d = 5$			$ISDC^{d}_{X^k}, d = 5$		
Class	ACE	CONE-SHAP	HCE (Ours)	ACE	CONE-SHAP	HCE (Ours)
Tabby	0.550	0.455	**0.630**	0.550	0.730	**0.785**
Blenheim Spaniel	0.600	0.425	**0.660**	0.555	0.710	**0.980**
Sorrel	0.565	0.340	**0.695**	0.565	0.690	**0.925**
Junco	0.528	0.475	**0.755**	0.583	0.655	**0.910**
Bighorn	0.588	0.445	**0.840**	0.523	0.665	**1.000**
Warplane	0.495	0.550	**0.895**	0.497	0.570	**0.895**
Pop Bottle	0.548	0.445	**0.830**	0.502	0.665	**0.990**
Motor Scooter	0.515	0.485	**0.760**	0.550	0.620	**0.980**
Ballplayer	0.505	0.397	**0.825**	0.465	0.670	**0.985**
Siamese Cat	**0.630**	0.445	0.628	0.530	0.625	**0.702**
Siberian Husky	0.585	0.455	**0.625**	0.585	0.650	**0.880**
Timber Wolf	0.540	0.455	**0.645**	0.510	0.675	**0.960**
Schooner	0.525	0.410	**0.735**	0.490	0.615	**0.820**
Canoe	0.568	0.440	**0.730**	0.523	0.650	**0.995**
Bulbul	0.508	0.505	**0.800**	0.572	0.645	**0.940**
Black Swan	0.505	0.505	**0.930**	0.585	0.595	**0.790**
Mountain Bike	0.525	0.500	**0.730**	0.525	0.750	**0.890**
Bison	**0.565**	0.435	**0.565**	0.580	0.710	**0.865**
Police Van	0.485	0.395	**0.733**	0.495	0.615	**0.938**
Half Track	0.475	0.355	**0.535**	0.510	0.640	**0.885**
Jeep	0.550	0.420	**0.795**	0.465	0.655	**0.960**
Park Bench	0.620	0.445	**0.768**	0.527	0.725	**0.985**
Fire Engine	0.465	0.400	**0.560**	0.480	0.715	**0.795**
Zebra	0.533	0.675	**0.943**	0.553	0.625	**0.960**
Average	0.540	0.450	**0.730**	0.530	0.660	**0.910**

4.4 Evaluating the Impact of Concepts on Model Predictions

We use two metrics, SSC and SDC, to analyze the impact of concepts on model predictions. These two metrics are similar to measuring the performance of concepts at the model level. Using 1000 images from 20 classes, we observe the top-1 classification probability of the DNN when adding or removing concepts in images, and calculate the rate of images whose predictions keep constant. In other words, the rate is equal to the prediction accuracy of DNN. HCE is evaluated at the component level, while ACE and CONE-SHAP are at the level of object parts. Global concepts of HCE are obtained in the same way as CONE-SHAP. To avoid differences in the existence of concepts due to different segmentation methods and the number of clusters, i.e., the top-d important global concepts may not always appear in every image, all methods use 5 concepts existing in the image. Tables 2 and 3 show the results. Following ACE, we set the original accuracy to 0.8.

SSC increases top-5 importance concepts and the accuracy of the model gradually increases, while SDC opposite. Similar trends for HCE and other methods, while HCE outperforms other methods on most items of SDC and on all items of SSC. Through HCE, the performance of top-5 concepts is improved by 20% at least. However, our top-1 concept decreases less on SDC than other methods, probably because our concept is a bit coarser than other methods (due to the difference in segmenting images). In addition, the difference in clustering may be the reason why HCE is better. Although HCE and CONE-SHAP set the same number of clusters, the former uses only component-level superpixels while the latter uses superpixels of three levels. There are some overlaps between superpixels at multiple levels, which may affect the performance of CONE-SHAP on metrics at the level of the object part.

TABLE 2
SCORES OF SSC (THE HIGHER THE BETTER)

	SSC (add top-d most important concept)				
	d=1	d=2	d=3	d=4	d=5
ACE	0.0176	0.0360	0.0464	0.0656	0.0768
CONE-SHAP	0.0888	0.1584	0.2384	0.3504	0.4480
HCE (Ours)	**0.5792**	**0.7384**	**0.7632**	**0.7680**	**0.7664**

TABLE 3
SCORES OF SDC (THE LOWER THE BETTER)

	SSC (remove top-d most important concept)				
	d=1	d=2	d=3	d=4	d=5
ACE	**0.6144**	0.5072	0.4368	0.3672	0.3088
CONE-SHAP	0.6216	0.4920	0.3376	0.2056	0.1160
HCE (Ours)	0.6840	**0.4616**	**0.2432**	**0.1312**	**0.0912**

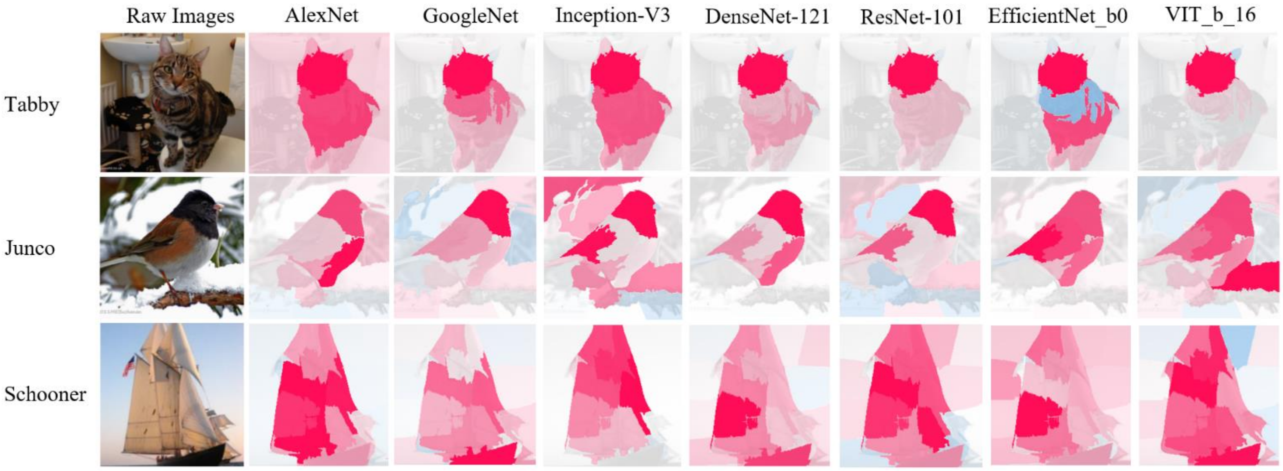

Fig. 7. Explanation heatmaps of same-task models.

4.5 Evaluating the Universality of HCE

In order to verify the applicability of HCE beyond Googlenet, we further evaluate its universality on 6 other excellent image classification models. Fig. 7 shows that HCE can identify important concepts and explain model decisions for different DNNs. All results are displayed at the component level.

In Fig. 7, HCE discovers the commonality and specificity of DNNs in terms of judgment conditions. For example, in the first row, all models consider the cat's head to be the most important for class 'tabby', while the different models in the third row consider different parts of the sail to have a strong influence on class 'schooner'. Although they are different designs of the same series, GoogleNet focuses more on the hull of 'schooner' than Inception-V3. VIT_b_16 focuses more on the upper part of the sail, while other models are more interested in the lower part. In the second row, we can distinguish which models have attention to the background of an image. DenseNet-121 and EfficientNet_b0 pay little attention to the background of the image, while other models make decisions based on partial background. In a single model, EfficientNet_b0 tends to use more concepts for decisions. Explanations for different DNNs demonstrate that HCE has a certain universality.

5 Conclusions And Future Work

Explainability helps humans understand model decisions and verify whether the model reasonably obtains decisions. One of the main reasons for the lack of explainability is that the internal knowledge of DNN is hard to understand. In this paper, we investigate how representations can be associated with human knowledge to explain model internal knowledge systematically, and aim to make model decisions comprehensible to humans. We propose HCE, a local explanation method that uses concept tree to describe model internal knowledge and explain decisions. HCE defines concept trees to model concepts and their constituent (or whole-part) relationship, and establishes a systematic mapping between model representations and human concepts by Multilevel Concept Extractor and Concept Tree AutoEncoder we designed. Compared to other concept-based explanation methods, HCE provides clearer hierarchical local explanations for samples by the defined Concept Tree Shapley value. It satisfies four axioms of Shapley value, considering the interaction of concepts. Our new consistency metrics (ISSC and ISDC) quantify the credibility of explanations, and the average of consistency scores of explanations is 35% higher than other methods. Thus, HCE provides trustworthy explanations for GoogleNet and improves human understanding of GoogleNet effectively. In addition, explanations for 6 other image classification models verify that HCE has a certain universality.

To improve explainability, the mutual correlation between human knowledge and model knowledge has become a new research trend. Interaction between humans and models in deep learning research where data-driven and knowledge-driven converge may help further ensure the credibility and reliability of the model. It could be an interesting direction to focus on the bi-directional comprehensibility of humans and models in the future.

Acknowledgments

This work is supported by the State Key Program of National Nature Science Foundation of China (Grant No. 61936001), the Training Program of the Major Research Plan of the National Natural Science Foundation of China (No. 92270124), the Program of National Nature Science Foundation of China (Grant No. 52073169), the National Nature Science Foundation of China (Grant No. 62221005), the Natural Science Foundation of Chongqing Nos. cstc2021ycjh-bgzxm0013 and cstc2019jcyj-cxttX0002, the key cooperation project of Chongqing Municipal Education Commission (Grant No. HZ2021008). We appreciated the High Performance Computing Center of Shanghai University and Shanghai Engineering Research Center of Intelligent Computing System for providing the computing resources and technical support.

REFERENCES

[1] Z. W. Li, F. Liu, W. J. Yang, S. H. Peng and J. Zhou, “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,” *IEEE Trans. Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999-7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.

[2] Y. Zhang, P. Tino, A. Leonardis and K. Tang, “A Survey on Neural Network Interpretability,” *IEEE Trans. Emerging Topics In Computational Intelligence*, vol. 5, no. 5, pp. 726-742, Oct. 2021, doi: 10.1109/TETCI.2021.3100641.

[3] R. R. Selvaraju, M. Cogswell, A. Das, R Vedantam1, D Parikh and D Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” *Int. J. Computer Vision*, vol. 128, no. 2, pp. 336-359, Feb. 2020, doi: 10.1007/s11263-019-01228-7.

[4] M. T. Ribeiro, S. Singh, and C. Guestrin, “"Why Should I Trust You?" Explaining the Predictions of Any Classifier,” *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 1135-1144, 2016.

[5] S. M. Lundberg and S. I. Lee, “A Unified Approach to Interpreting Model Predictions,” *Proc. Adv. Neural Information Processing Systems*, pp. 4765-4774, 2017.

[6] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas and R. Sayres, “Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV),” *Proc. Int. Conf. Machine Learning*, pp. 2668–2677, 2018.

[7] A. Ghorbani, J. Wexler, J. Zou and B. Kim, “Towards Automatic Concept-based Explanations,” *Proc. Adv. Neural Information Processing Systems*, pp. 9273-9282, 2019.

[8] W. b. Wu, Y. X. Su, X. X. Chen, S. L. Zhao, I. King, M. R. Lyu and Y. W. Tai, “Towards Global Explanations of Convolutional Neural Networks with Concept Attribution,” *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 8649-8658, 2020.

[9] R. H. Zhang, P. Madumal, T. Miller, K. A. Ehinger and B. I. P. Rubinstein, “Invertible Concept-based Explanations for CNN Models with Non-negative Concept Activation Vectors,” *Proc. 35th AAAI Conf. Artificial Intelligence*, pp. 11682-11690, 2021.

[10] L. S. Shapley, “A value for n-person games,” *Contributions to the Theory of Games*, pp. 307–317, 1953.

[11] C. Yeh, B. Kim, S. Arık, C. L. Li, T. Pfister and P. Ravikumar, “On Completeness-aware Concept-Based Explanations in Deep Neural Networks,” *Proc. Adv. Neural Information Processing Systems*, pp. 20554-20565, 2020.

[12] J. Li, K. Kuang, L. Li, L. Chen, S. Y. Zhang, J. Shao and J. Xiao, “Instance-wise or Class-wise? A Tale of Neighbor Shapley for Concept-based Explanation,” *Proc. 29th ACM Int. Conf. Multimedia*, pp. 3664-3672, 2021.

[13] J. B. Chen, L. Song, M. J. Wainwright and M. I. Jordan, “L-Shapley and C-Shapley: Efficient Model Interpretation for Structured Data,” *Proc. Int. Conf. Learning Representations*, 2019.

[14] E. Winter, “A value for cooperative games with levels structure of cooperation,” *Internat. J. Game Theory*, vol. 18, no. 2, pp. 227-240, 1989, doi: 10.1007/BF01268161.

[15] R. van den Brink, P. J. J. Herings, G. van der Laan and A. J. J. Talman, “The Average Tree permission value for games with a permission tree,” *Economic Theory*, vol. 58, no. 1, pp. 99-123, 2015, doi: 10.1007/s00199-013-0796-5.

[16] V. Petsiuk, A. Das, and K. Saenko, “Rise: Randomized input sampling for explanation of black-box models,” 2018, *arXiv:1806.07421*.

[17] R. Fong, M. Patrick, and A. Vedaldi, "Understanding deep networks via extremal perturbations and smooth masks," *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 2950-2958, 2019.

[18] A. Ghandeharioun, B. Kim, C. L. Li, B. Jou, B. Eoff and R. W. Picard, “DISSECT: Disentangled Simultaneous Explanations via Concept Traversals,” *Proc. Int. Conf. Learning Representations*, 2022.

[19] I. Gat, G. Lorberbom, I. Schwartz and T. Hazan, “Latent Space Explanation by Intervention,” *Proc. 36th AAAI Conf. Artificial Intelligence*, pp. 679-687 2022.

[20] D. Georgiev, P. Barbiero, D. Kazhdan, *et al.*, “Algorithmic Concept-Based Explainable Reasoning,” *Proc. 36th AAAI Conf. Artificial Intelligence*, pp. 6685-6693, 2022.

[21] J. B. Hamrick, K. R. Allen, V. Bapst, T. Zhu, K. R. McKee, J. B. Tenenbaum and P. W. Battaglia, “Relational inductive bias for physical construction in humans and machines,” *Proc. 40th Annu. Conf. Cognitive Science Society (CogSci)*, 2018.

[22] A. Sarkar, D. Vijaykeerthy, A. Sarkar and V. N. Balasubramanian, “A Framework for Learning Ante-hoc Explainable Models via Concepts,” *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 10276-10285, 2022.

[23] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning Important Features Through Propagating Activation Differences,” *Proc. Int. Conf. Machine Learning*, pp. 3145-3153, 2017.

[24] G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg and S. I. Lee, “Improving performance of deep learning models with axiomatic attribution priors and expected gradients,” *J. Nature Machine Intelligence*, vol. 3, no. 7, pp. 620-631, Jul. 2021, doi: 10.1038/s42256-021-00343-w.

[25] M. Sundararajan, A. Taly and Q. Yan, “Gradients of counterfactuals,” 2016, *arXiv:1611.02639*.

[26] D. Smilkov, N. Thorat, B. Kim, F. Viegas and M. Wattenberg, “Smoothgrad: removing noise by adding noise,” 2017, *arXiv:1706.03825*.

[27] M. Alvarez-Mozos, R. van den Brink, G. van der Laan and O. Tejada, “From hierarchies to levels: new solutions for games with hierarchical structure,” *Int. J. Game Theory*, vol. 46, no. 4, pp. 1089-1113, Nov. 2017, doi: 10.1007/s00182-017-0572-z.

[28] M. Besner, “Weighted Shapley hierarchy levels values,” *J. Operations Research Letters*, vol. 47, no. 2, pp. 122-126, March. 2019, doi: 10.1016/j.orl.2019.01.007.

[29] L. Chen, “Topological structure in visual perception,” *J.Science*, vol. 218, no. 4573, pp. 699-700, Nov. 1982, doi: 10.1126/science.7134969.

[30] G. Y. Wang, “MGCC: Multi-Granularity Cognitive Computing,” *Proc. Rough Sets: International Joint Conference (IJCRS)*, pp. 30-38, 2022.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh1, S. Ma, Z. H. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and F. F. Li, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Computer Vision*, vol. 115, no. 3, pp. 211-252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.

[32] C. Szegedy, W. Liu, Y. Q. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, “Going Deeper with Convolutions,” *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 1-9, 2015.

[33] A. Krizhevsky, “One weird trick for parallelizing convolutional neural networks,” 2014, *arXiv: 1404.5997*.

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 2818-2826, 2016.

[35] G. Huang, Z. Liu, L. van der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 2261-2269, 2017.

[36] K. M. He, X. Y. Zhang, S. Q. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 770-778, 2016.

[37] M. X. Tan, and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *Proc. Int. Conf. Machine Learning*, pp. 6105-6114, 2019.

[38] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. H. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *Proc. Int. Conf. Learning Representations*, 2021.

[39] L. C. Chen, G. Papandreou, F. Schroff and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," 2017, *arXiv: 1706.05587.*

[40] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Susstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," *IEEE Trans. Pattern Analysis And Machine Intelligence*, vol. 34, no. 11, pp. 2274-2281, Nov. 2012, doi: 10.1109/TPAMI.2012.120.

[41] G. Y. Wang, "Data-Driven Granular Cognitive Computing," *Proc. Rough Sets: International Joint Conference (IJCRS)*, pp. 13-24, 2017.

Yue Liu obtained her B.S. and M.S. degrees in Computer Science from Jiangxi Normal University in 1997 and 2000, respectively. She got her Ph.D. degree in Control Theory and Control Engineering from Shanghai University (SHU) in 2005. She was a curriculum R&D manager at the Sybase-SHU IT Institute of Sybase Inc. from July 2003 to July 2004 and a visiting scholar at the University of Melbourne from September 2012 to September 2013. At present, she is a professor at SHU. Her current research interest focuses on research of machine learning, data mining, and AI for materials science.

Ziyi Yu received the B.S. degree from the School of Computer and Engineering, Shanghai University, Shanghai, China, in 2018. He is currently pursuing the M.S. degree in Shanghai University, Shanghai, China. His research interests include Data Mining and Deep Learning Interpretability.

Zitu Liu received the M.S. degree from the Compute Science, Heilongjiang University, Heilongjiang, China, in 2020. He is currently pursuing the Ph.D. degree in Shanghai University, Shanghai, China. His main research interests include Data Mining and Deep Learning Interpretability.

Zhenyao Yu received the B.S. degree from the College of Communication and Information Technology, Xi'an University of Science and Technology, Shaanxi, China, in 2021. He is currently pursuing the M.S. degree in Shanghai University, Shanghai, China. His research interests include data mining and deep learning interpretability.

Yike Guo is the vice president of Hong Kong University of Science and Technology and professor at Imperial College London. He is an IEEE fellow, fellow of the Royal Academy of Engineering (FREng), member of the Academia Europaea (MAE), fellow of the British Computer Society and a trustee of the Royal Institution of Great Britain. Professor Guo has published over 200 articles, papers, and reports. His current research interests focus on data mining, machine learning, and dig data of science.

Qun Liu received her B.S. degree from Xi'An Jiaotong University in China in 1991, and the M.S. degree from Wuhan University, in China in 2002, and the Ph.D from Chongqing University in China in 2008. She is currently a Professor with Chongqing University of Posts and Telecommunications. Her current research interests include complex and intelligent systems, neural networks, and intelligent information processing.

Guoyin Wang received the B.S., M.S., and Ph.D. degrees from Xi'an Jiaotong University, Xian, China, in 1992, 1994, and 1996, respectively. He was at the University of North Texas, and the University of Regina, Canada, as a visiting scholar during 1998-1999. Since 1996, he has been at the Chongqing University of Posts and Telecommunications, where he is currently a professor, the Vice-President of the University, the director of the Chongqing Key Laboratory of Computational Intelligence, the director of the Key Laboratory of Cyberspace Big Data Intelligent Security, Ministry of Education. He was the director of the Institute of Electronic Information Technology, Chongqing Institute of Green and Intelligent Technology, CAS, China, 2011-2017. He is the author of over 10 books, the editor of dozens of proceedings of international and national conferences and has more than 300 reviewed research publications. His research interests include rough sets, granular computing, knowledge technology, data mining, neural network, and cognitive computing, etc. Dr. Wang was the President of International Rough Set Society (IRSS) 2014-2017. He is a Vice-President of the Chinese Association for Artificial Intelligence (CAAI), and a council member of the China Computer Federation (CCF). He is a Senior Member of IEEE, a Fellow of IRSS, CAAI and CCF.