A Multiobjective Genetic Algorithm to Evolving Local Interpretable Model-Agnostic Explanations for Deep Neural Networks in Image Classification

Bin Wang, *Student Member, IEEE*, Wenbin Pei, *Member, IEEE*, Bing Xue, *Senior Member, IEEE*, and Mengjie Zhang, *Fellow, IEEE*

***Abstract*—Deep convolutional neural networks have become a dominant solution for numerous image classification tasks. However, a main criticism is the poor explainability due to the black-box characteristic, which hurdles the extensive usage of deep convolutional neural networks. To address this issue, this article proposes a new evolutionary multiobjective-based method, which aims to explain the behaviors of deep convolutional neural networks by evolving local explanations on specific images. To the best of our knowledge, this is the first evolutionary multiobjective method to evolve local explanations. The proposed method is model agnostic, i.e., it is applicable to explain any deep convolutional neural networks. ImageNet is used to examine the effectiveness of the proposed method. Three well-known deep convolutional neural networks—VGGNet, ResNet, and MobileNet, are chosen to demonstrate the model-agnostic characteristic. Based on the experimental results, it can be observed that the local explanations are understandable to end users, who need to check the sensibility of the evolved explanations to decide whether to trust the predictions made by the deep convolutional neural networks. Furthermore, the local explanations evolved by the proposed method improves the confidence of deep convolutional neural networks making the predictions. Finally, the pareto front and convergence analyses indicate that the proposed method can form a good set of nondominated solutions.**

***Index Terms*—Evolutionary deep learning, explainable machine learning, image classification, local explanations.**

Manuscript received 31 March 2022; revised 5 July 2022 and 2 October 2022; accepted 25 November 2022. Date of publication 30 November 2022; date of current version 1 August 2024. This work was supported in part by the Marsden Fund of New Zealand Government under Contract VUW1913 and Contract VUW1914; in part by the Science for Technological Innovation Challenge (SfTI) Fund under Grant E3603/2903; in part by the University Research Fund at Victoria University of Wellington under Grant 223805/3986; in part by the Ministry of Business, Innovation and Employment (MBIE) Data Science Strategic Science Investment Fund (SSIF) under Contract RTVU1914; in part by the National Natural Science Foundation of China (NSFC) under Grant 61876169; in part by the Huayin Medical under Grant E3791/4165; and in part by MBIE Endeavor Research Programme under Contract C11X2001. *(Corresponding author: Bin Wang.)*

Bin Wang, Bing Xue, and Mengjie Zhang are with the School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6140, New Zealand (e-mail: bin.wang@ecs.vuw.ac.nz; bing.xue@ecs.vuw.ac.nz; mengjie.zhang@ecs.vuw.ac.nz).

Wenbin Pei is with the School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China (e-mail: peiwenbin@dlut.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TEVC.2022.3225591.

Digital Object Identifier 10.1109/TEVC.2022.3225591

I. Introduction

DEEP convolutional neural networks (DCNNs) have achieved remarkable performance for image classification tasks. Researchers have improved the classification accuracy by increasing the depths of DCNNs, such as AlexNet [1] and VGGNet [2]. They have also explored various network topologies to obtain better performance, e.g., ResNet [3] and DenseNet [4]. In the most recent literature, novel techniques have been proposed to reduce the high computational cost of training DCNNs. For example, in Mobilenetv2 [5], the traditional convolutional layer was replaced by a depth separable convolutional layer, and in Shufflenet [6], a reshuffle strategy was proposed. Overall, DCNNs have dominated the field of image classification.

However, the explainability of DCNNs is poor due to the complex black-box nature, but the explainability is important, especially when it comes to critical decision-making scenarios, such as medical diagnosis and law enforcement. Therefore, the research area of explainable deep learning (XDL) [7], [8], [9], [10] has drawn broad attention to interested researchers. Various branches of XDL have developed, e.g., the model translation direction [11], [12], and the local approximation approach [13], [14], [15]. This article is mainly inspired by an outstanding local approximation method called local interpretable model-agnostic explanations (LIMEs) [14] due to a couple of advantages. First, the image that needs to be explained is segmented into a group of interpretable features, which are readable to humans. The *local explanation* obtained by LIME is comprised of a subset of interpretable features, which are straightforward for end users to understand. Furthermore, compared with the model translation methods [11], [12], LIME usually takes less time to obtain the explanations.

This article endeavors to further advance LIME-based methods from the following two points. First, the proposed method further reduces the computational cost by avoiding the expensive sampling process of generating a large set of perturbed images. Second, the number of interpretable features in an explanation needs to be defined before applying LIME on a specific image. Since the number of interpretable features may affect the local explanations for different images, it may require expert interference to tweak this number for each image, which could be tedious. Therefore, the proposed method aims to eliminate the human effort of fine-tuning

the method by automatically searching for the number of interpretable features.

Evolutionary computation (EC) has been excessively investigated and demonstrated its effectiveness in automating the process of designing DCNNs. Some researchers [16], [17], [18], [19], [20], [21] focus on the classification accuracy, some [22], [23], [24], [25], [26], [27] target to reduce the model size while keeping a competitive classification accuracy, and others [28], [29], [30], [31], [32], [33], [34] endeavorf to reduce the computational cost of the searching process of DCNNs while retaining good performance. However, EC techniques have rarely been used in XDL, especially for the local approximation approaches.

This article aims to develop an evolutionary multiobjective method for evolving local explanations for DCNNs in image classification to address the limitations of LIME. Two objectives: 1) minimizing the number of interpretable features in an explanation and 2) maximizing the confidence of the DCNN making a specific prediction based on the corresponding explanation, are designed. The confidence of the DCNN making a specific prediction often increases when the number of interpretable features grows from 1 because more meaningful features can increase the DCNN's confidence of make a specific prediction until redundant features or noises are introduced, so minimizing the number of interpretable features potentially conflicts the objective of maximizing the confidence. Since the two objectives are potentially conflicting with each other, there is no single best solution, but multiple tradeoff solutions. Therefore, this article proposes a multiobjective EC-based model-agnostic method (called MO-LIME) to evolve local explanations (i.e., a subset of meaningful superpixels that a DCNN replies on to make a prediction) by simultaneously optimizing the aforementioned objectives to find a set of tradeoff solutions. MO-LIME is based on the nondominated sorting genetic algorithm II (NSGA-II) [35], which is a very popular evolutionary multiobjective approach and demonstrated encouraging performance in various problems [36], [37], and [38].

A. Contributions

The proposed MO-LIME conquers the limitations of LIME—the expensive sampling process and the prefixed number of interpretable features. To the best of our knowledge, this is the first multiobjective method to evolve local explanations. The main contributions of this article are summarized as follows.

1) This article proposes an NSGA-II-based model-agnostic method, which maximizes the probability of the DCNN model predicting a local explanation as a specific class label (i.e., the class label predicted by the DCNN model for the corresponding image), and minimizes the number of the selected local interpretable features. The probability could imply the confidence of a DCNN in making a specific prediction, so it could be used to select interpretable features that impact the decision made by the DCNN. Minimizing the number of interpretable features is also beneficial because it is difficult/time consuming to examine a large number of interpretable features by the end users.
2) A new encoding strategy is designed to encode the local interpretable features into real-value vectors, so NSGA-II can be straightforwardly used to optimize the vectors. A vector represents the local interpretable features in a local explanation, where the value of each dimension controls whether the corresponding interpretable feature is selected or not. The proposed method takes the advantage of the flexible encoding and the powerful search ability of EC methods to avoid the sampling process of LIME that is computationally intense. With the powerful search ability of EC algorithms, it obtains local explanations ten times faster than LIME [14].
3) After a set of nondominated solutions are evolved by NSGA-II, a new method is designed to select only two nondominated solutions. NSGA-II could obtain many nondominated solutions in the pareto front, so it would require considerable effort to examine all of them. To mitigate the demand of human effort, this article proposes to only select two that are expected to be pivotal for the end users to decide whether to trust the prediction or not.
4) A thorough analysis is performed to assess the meaningfulness of the evolved local explanations and the performance of the proposed method. First, the evolved local explanations are analyzed, which shows the effectiveness of the proposed method. Second, it can be observed the confidence of the DCNN making a specific prediction on the local explanation (that focuses on the confidence) is increased from the confidence of the DCNN on the original image. Finally, the pareto front and the convergence of the evolutionary process are exhibited to further support the effectiveness of the proposed method. Overall, this article demonstrates the powerfulness of using EC to solve an emerging issue, which could potentially encourage the EC community to explore the EC approaches to solve a much wider range of issues in the future.

B. Organization

The remainder of the article is outlined as follows. In Section II, the background knowledge is introduced. Section III elaborates the proposed method and Section IV presents the experimental design. The results are reported and analyzed in Section V. Section VI concludes the article and points out future work.

II. Background

In this section, first of all, we explain why DCNN models are regarded as a black-box model by taking VGGNet as an example. This is the main motivation of XDL research. After that, we review the related works in the research domain, and then introduce simple linear iterative clustering (SLIC) superpexiels [39], which the proposed method relies on.

A. VGGNet—Black-Box Model

A typical DCNN called VGGNet [2] is drawn in Fig. 1. VGGNet is composed of very small (3×3) convolutional filters, pooling layers, and fully connected layers at the end.

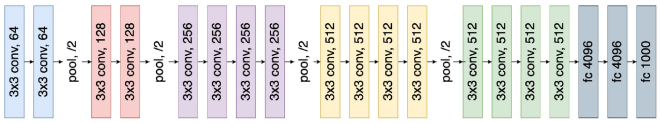

Fig. 1. VGGNet-19 architecture [2].

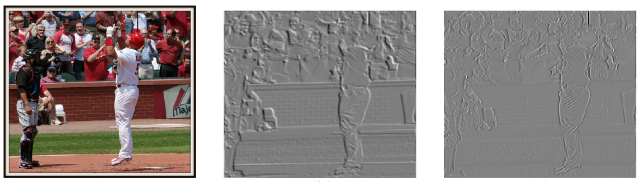

Fig. 2. Sample feature maps of VGGNet. (a) Shows the original image of a baseball player. (b) and (c) are feature maps extracted from VGGNet.

Fig. 1 shows one example of VGGNet of 19 layers, and the configuration of each layer is manually designed. The convolutional filters are used to extract feature maps. Fig. 2(b) and (c) are two example feature maps of the original image [i.e., Fig. 2(a)], which are extracted by the first convolutional layer. Clearly, it is hard for humans to recognize the baseball players from the two extracted feature maps. The recognizability of the extracted feature maps could get worse in the deeper layers. Therefore, the behaviors of the layers in DCNNs are not understandable to humans without expert knowledge. This results in the black-box characteristic of DCNNs.

B. Model-Agnostic Techniques for Explainability

Model-agnostic techniques depend mainly on model simplification, feature relevance explanation techniques, and visualization techniques [40].

LIME [14] is one of the most well-known approaches in the category of generating model-agnostic explanations through model simplification. The main idea of LIME is straightforward, i.e., a prediction of a black box or an over-complicated model is explained by an interpretable model, such as a linear model or a decision tree, etc. LIME has the following steps. First, to explain the prediction of a black-box model on instance x, the instances around x are sampled at random, and then each of the sampled instances is weighed according to the distance between the sampled instance and x. Second, the black-box model is used to predict labels (or probabilities) of the weighted instances. Third, the weighted instances with the predicted labels are the data to train an interpretable model, e.g., DT.

LIME has been extended in [15], where a new model-agnostic explanations method (called *anchors*) was designed and proposed by considering to maximize the coverage region of an explanation represented by an if-then rule. In [41], LIME has been applied to generate three kinds of explanations for music content analysis. For feature relevance explanation techniques, the behaviors of a black-box model are explained by measuring the influence or relevance of each feature on a prediction [40], [42].

C. SLIC Superpixels

Superpixel techniques provide a crucial way to represent an image conveniently and compactly. Without the loss of generality, an image is segmented into multiple superpixels, each of which is a group of similar pixels. For image classification, the use of superpixels with deep learning could reduce computational costs and make the subsequent image processing tasks easier [39]. To date, many algorithms have been designed and proposed to generate superpixels, mainly, including graph-based and gradient-ascent-based algorithms [43], [44], [45]. Among the existing superpixel generation methods, SLIC is one of the most commonly used methods [14], [15]. In LIME, an image is segmented into a number of superpixels based on SLIC. We will introduce SLIC in more detail because it is also the main technique used by the proposed method in this article.

To segment an image, SLIC employs k-means clustering to group pixels into superpixels according to the color proximity and spatial proximity. A matrix is used to represent a cluster center $C_i = [l_i\ a_i\ b_i\ x_i\ y_i]^{\mathrm{T}}$, where l_i, a_i, and b_i are from the pixel color vector in the CIELAB color space, and x_i and y_i are the pixel positions.

Note that, for large superpixels, the spatial proximity outweighs the color proximity because the image size influences the spatial distance. As a consequence, when calculating the Euclidean distance between each pixel and a cluster center, the priority of the spatial proximity is higher than that of the color proximity. To avoid this, a normalized distance measure is defined as follows [39]:

$$D = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)m^2} \tag{1}$$

where $d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}$, $d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$, S is a grid interval ($S = \sqrt{(N_p/k)}$, where N_p is the number of pixels), and m is used to weigh the relative importance between the color similarity and spatial proximity (m is in the range of [1, 40]).

The steps of SLIC are as follows.

1) *Initialization:* k cluster centers $C_i = [l_i\ a_i\ b_i\ x_i\ y_i]^T$ are initialized. It is achieved by sampling pixels at the regular grid interval S. The cluster centers move to seed locations of the lowest gradient position in a 3×3 neighborhood.
2) Repeat the following steps.
   a) For each cluster center C_i, the distance D between each pixel i and C_i is calculated according to (1) to assign pixel i to the nearest cluster.
   b) Updates cluster centers.
   c) Calculate the residual error E.
3) *Termination:* The process comes to the end when $E \leq$ *threshold* (*threshold* is a predefined parameter).

D. Related Works

The methods for XDL could be categorized into three classes: 1) *Visualization methods*; 2) *Intrinsic methods*; and 3) *Model distillation* [46]. Visualization methods rely on a

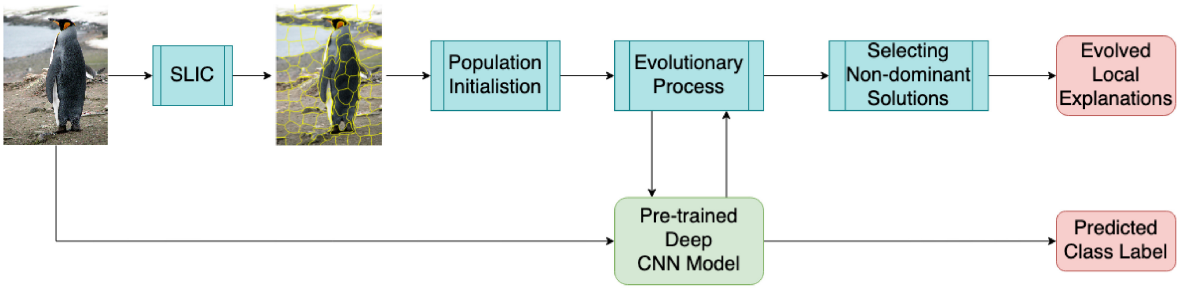

Fig. 3. Overall framework of the proposed method.

common explanatory form—*saliency maps*, which is usually not easy to obtain. For example, Deconvolution [47] adds *Deconvnet* layers to the DCNN to form a Deconvnet, and requires to retrain the Deconvnet to obtain saliency maps. CAM [48] redesigns the DCNN by using global average pooling to generate the class activation maps, and also retains the modified DCNN to acquire saliency maps. DeepLIFT [49] does not require a redesign of network architecture or retraining, but the *reference input* needs to be carefully chosen, and the choice of the target layer, where the DeepLIFT rules are applied to calculate the contribution scores, needs to be decided by domain experts. Overall, the process of obtaining saliency maps is not easy, and may require relevant machine learning expertise to redesign the DCNN and retrain the modified DCNN. For intrinsic methods, the decisions and their corresponding explanations are simultaneously provided by the model as the output, e.g., Single-Modal Weighting [50], Explanation Association [51], and Model Prototype [52]. Intrinsic methods need to redesign the model architecture based on DCNNs that do not include explanations. In this article, the proposed method aims to explain the existing DCNNs to end users without any machine learning expertise, so neither intrinsic methods nor visualization methods suit the purpose. Meanwhile, model distillation methods distil the knowledge encoded in DCNNs into a representation that is explainable. One branch is model translation, which is to learn a simpler model to replicate the behaviors of DCNNs [11], [12]. However, model translation methods may sacrifice the performance. Besides, it may also require domain expertise to understand the behaviors of the simple model. The other branch is a local approximation with LIME [14] as the most popular method. LIME finds a subset of interpretable features, which are intuitive to end users without requiring machine learning expertise. The proposed method is targeted at improving LIME by mitigating two limitations of LIME: 1) high computational cost and 2) predefining the number of interpretable features.

III. Proposed Method

In this section, we introduce the overall framework of the proposed MO-LIME method, and then elaborate every component in more detail.

A. Overall Framework

The overall framework of the proposed method is drawn in Fig. 3. There are two branches in the overall framework—the explanation branch on the top row and the prediction branch on the bottom row. Regarding the explanation branch, the input image is passed to the segmentation algorithm, i.e., SLIC. Each cell divided by the yellow lines/curves in the output image of SLIC is a superpixel, i.e., an interpretable feature. The population is then initialized by creating individuals representing a subset of the segments in an image. The evolutionary learning process starts with the initial population and the pretrained DCNN model that needs to be explained, and then produces a set of nondominated solutions as the evolved local explanations. At the end, some of the nondominated solutions are selected as the final evolved local explanations to explain the DCNN model. For the prediction branch, the pretrained DCNN model predicts the class label of the input image. Both the predicted class label and the evolved local explanations are presented as the final results to the end users, so they can decide whether to trust the predicted class label by checking the reasonableness of the evolved local explanations.

B. Uninterpretable Features Versus Interpretable Features

As this article targets to evolve local explanations, the interpretable features need to be introduced first. In standard DCNN models, each pixel of the input image is an input feature. The pixels are numeric values, which are not interpretable to humans. Inside the DCNN models, the convolution filters are utilized to extract feature maps from the input image. Fig. 4(b) shows an example of the extracted feature map from the input image of Fig. 4(a). The shape of the king penguin can be roughly observed in Fig. 4(b), but humans may not be able to distinguish between the king penguin and other penguins, such as the little blue penguins and the yellow-eyed penguins, by just referring to the contour of the penguin in the feature map. It is also arguable that if the original image was not given, most people might not even be able to tell that Fig. 4(b) is an extracted feature map of a penguin. Overall, both the input features, i.e., the image pixels and the extracted feature maps are not interpretable to humans.

In LIME [14], the segments, a.k.a. the superpixels, obtained by applying SLIC to the original image, are adopted as the interpretable features. Fig. 4(c) exhibits 100 superpixels achieved by segmenting the original image by SLIC. Each superpixel contains a group of similar neighborhood pixels. When pulling a superpixel from Fig. 4(c), it is feasible for humans to recognize it. For example, when humans see a superpixel that contains the black feet, the jaw with an orange stripe, or the golden-orange spoon-shaped head markings on

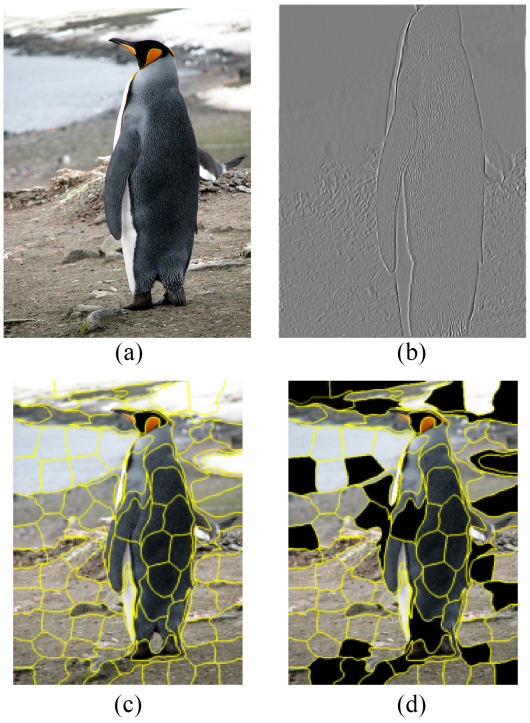

Fig. 4. (a) Original input image of a king penguin. (b) Sample of the uninterpretable feature maps extracted by VGGNet. (c) Segmented input image by SLIC, where each of the segments is an interpretable feature. (d) Sample of the evolved local explanations.

the side of the neck, it is straightforward for them to recognize the superpixel from the original image. Therefore, it is reasonable to deem a superpixel as an interpretable feature [14].

The final output of the proposed method is a subset of the superpixels, based on which the DCNN model makes the prediction. Fig. 4(d) shows an example of a set of selected superpixels. The superpixels that are blacked out are dumped superpixels, which may be irrelevant features or noises for the DCNN model to make the right decision. However, the superpixels that have not been blacked out are considered as the selected interpretable features. The group of the selected interpretable features is called the *local explanation* [14].

In Fig. 4(d), the local explanation contains two most important interpretable features, which are the two segments showing the bright golden-orange spoon-shaped head markings on the side of the neck and the orange stripe along the lower jaw. Since the two features are the key to identify king penguins, the DCNN model could use them to make the prediction and the local explanation could improve the confidence of end users. Conversely, if the two key features are not included, the end users should doubt the prediction of the DCNN model.

C. Encoding Strategy

In a local explanation, each of the interpretable features is either selected or not selected. The proposed encoding strategy encodes the probability of each interpretable feature being selected as a real value from 0 to 1 in each dimension of the encoded vector. Then, NSGA-II [35] can be easily employed to evolve the local explanations.

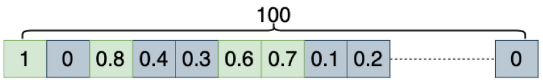

Fig. 5. Example of the encoded vector.

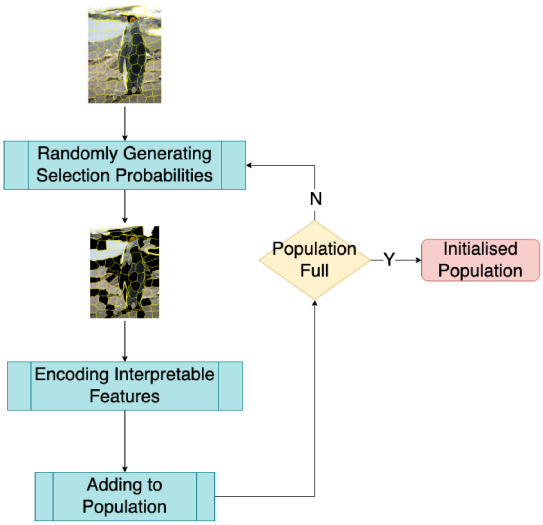

Fig. 6. Population initialization.

Fig. 5 indicates a sample vector to encode the interpretable features of Fig. 4(d). As shown in Fig. 4(d), there are 100 superpixels, and the dimensionality of the encoded vector is set to 100. Each of the dimensions is referred to as a sueperpixel in Fig. 4(d) numbered from the top-left to the bottom-right. In the local explanation, the interpretable features are either selected or not selected (i.e., backed out), so it is a binary operation. To accommodate the binary operation in the real-value vector, the real value in each dimension could be considered as the probability of being selected, and a threshold of 0.5 is set as the selection boundary. In other words, if the value of the specific dimension in the vector is greater than 0.5, the corresponding interpretable feature will be selected. Otherwise, the interpretable feature will not be selected. In Fig. 5, the dimensions with the light green background represent the selected features, and the dimensions with the greyish color represent the interpretable features that are blacked out.

During the evolutionary learning process, each individual is a real-value vector as shown in Fig. 5. However, the output of the proposed method cannot just be the individual because a real-value vector cannot explain the behaviors of DCNN models. Therefore, the encoded vector needs to be decoded back to the corresponding local explanation by blacking out the superpixels whose corresponding values in the individual are not greater than 0.5. The decoding process converts an encoded vector similar to Fig. 5 to a local explanation shown in Fig. 4(d).

D. Population Initialization

After the encoding strategy is designed, the population can be initialized as illustrated in Fig. 6.

First, the probability of each superpixel being selected is randomly generated. In the example of 100 superpixels, a

real-value vector of 100 dimensions, with each dimension being randomly sampled from [0, 1], is generated. As per the probability threshold defined in the encoding strategy, some superpixels are selected while others are blacked out as shown in the image positioned below the probability generation step in Fig. 6. Next, the superpixels with the selection probabilities are encoded into a real-value vector as described in Section III-C. Then, the encoded vector, a.k.a. the individual is added to the population until the number of individuals reaches the predefined population size.

E. Objectives

In this article, the major targets of the local explanations are to extract the meaningful superpixels and to remove the noisy superpixels that could confuse the DCNN model. To achieve the above targets, the proposed method needs to simultaneously optimize the following two objectives:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}. \tag{2}$$

1) *Objective 1:* Maximizing the probability of the DCNN model predicting a local explanation as a specific class label on an image. Specifically, the local explanation represented by the individual, i.e., an image that only displays the selected superpixels, is passed to the DCNN model. The output of the DCNN is z_j for $j = 1, 2, \ldots, K$, where K is the total number of class labels. Suppose the predicted class for the original image is i, the probability of the local explanation represented by the individual can be calculated by (2), which is the first objective; the reason for choosing this objective is that the DCNN model should be more confident to make the prediction when the noisy superpixels are removed from the input image. Therefore, the probability of the prediction should be improved as well. To obtain the probability as the objective value, a pretrained DCNN model is passed to the objective evaluation, which is also the DCNN model that needs to be explained by the proposed method. The probability is obtained in the last layer of the pretrained DCNN model by passing the input image that needs to be predicted.
2) *Objective 2:* Minimize the number of interpretable features displayed to the end users. The end users could well benefit from this because it takes less time for them to examine fewer superpixels, and the examination process could also be simplified due to a smaller number of superpixels. To fulfil the above objective, the number of interpretable features selected by the proposed method is set as the second objective. Note that the first objective is a probability value, which ranges from 0 to 1. However, the range of the second objective is based on the number of superpixels. In the example of 100 superpixels in Fig. 5, the value of the second objective could be from 1 to 100. In the proposed method, the values of the second objective are normalized to reach the same range as the first objective, i.e., the number of selected superpixels is divided by the total number of superpixels. The generalized equation of calculating the value of the second objective is defined as follows:

$$f_2 = \frac{n_s}{n_t} \tag{3}$$

where f_2 is the normalized value of the second objective, n_s is the number of selected superpixels, and n_t is the number of total superpixels.

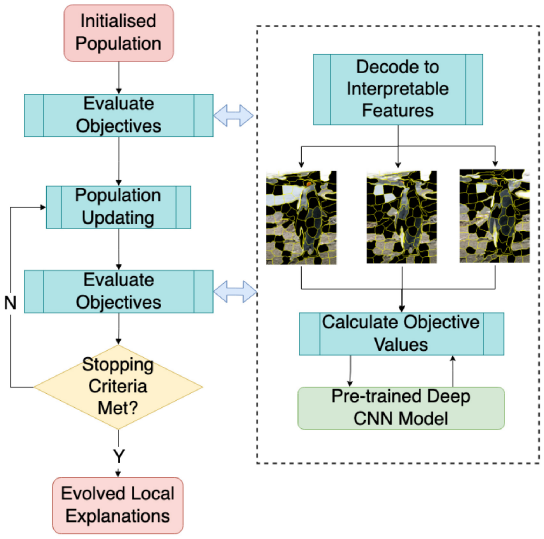

Fig. 7. Evolutionary learning process.

F. Evolving Local Explanations

Now, it is straightforward to apply NSGA-II to evolve the local explanations after all the above steps are designed. The evolutionary learning process is illustrated in Fig. 7. First, the population is initialized according to Section III-D. Second, the individuals in the population are evaluated, and their corresponding objective values are assigned. The population is passed to the objective calculation process on the right of Fig. 7. Each individual in the population is decoded into the corresponding local explanation described in Section III-C. The values of the two objectives are calculated according to Section III-E, which are then used to update the objective values of the corresponding individuals. Third, the population is updated by applying NSGA-II operators. Fourth, the objectives of the individuals in the population are updated by following the same process as that of updating the objectives of the initialized population. Fifth, the stopping criterion is checked. When the stopping criterion is met, the evolutionary learning process stops, and the evolved nondominant solutions are produced. Otherwise, the process goes back to the population updating step until the stopping criterion is met.

G. Selecting Nondominated Solutions From Pareto Front as Final Local Explanations

The outcome of NSGA-II [35] is a pareto front, which contains a set of nondominated solutions. It is not practical to present all the tradeoff solutions to the end users because it is exhaustive for them to examine all. Based on the two objectives, this article proposes to present two nondominated solutions from the pareto front—one with the best value of the

first objective and the other with the best possible value of the second objective given a higher confidence than the original image.

The first nondominated solution is selected by simply searching for the solution that achieves the best value on the first objective. It is worth checking the local explanation with the highest probability because it may contain all of the superpixels that are meaningful to the DCNN. The local explanation is expected to include the important characteristics of the object and the relevant background to identify the object. This could also be a good reference for the end users to decide whether to trust the model.

The second nondominated solution is selected by following two steps. First, the proposed method selects the solutions that achieve a higher probability than the original probability achieved by the pretrained DCNN on the input image. Then, from the selected nondominated solutions, one solution is identified as the second target solution if it achieves the best value on the second objective (i.e., with the fewer superpixels than other selected nondominated solutions). The intuitions behind this are as follows. First, it makes more sense to examine the evolved local explanations that could improve the confidence of the DCNN model in making the prediction. Furthermore, it is easier for the end users to check the local explanations with a smaller number of interpretable features.

Note that the first selected nondominated solution could be compared with the second. By performing this comparison, the common interpretable features and the disparate interpretable features in the two solutions can be observed by the end users. The common interpretable features can represent the key features that are decisive for the DCNN model to make the prediction. The disparate interpretable features are most likely to be the relevant background features. The end users can check how the common and disparate interpretable features reflect the object and the background, respectively.

IV. Experiment Design

A. Benchmark Dataset and Selected DCNNs

ImageNet [1] is the most widely used benchmark dataset to evaluate DCNNs for image classification. There are 1000 class labels, 1.2 million training images, and 150 000 validation images in the ImageNet dataset [1]. As the proposed method targets to explain DCNNs, ImageNet was selected as the benchmark dataset to evaluate the effectiveness of the proposed method. Another advantage of the ImageNet dataset is that it consists of real images from the Internet, which could indicate the performance of the proposed method on real-world images. Besides, most of the state-of-the-art DCNNs have pretrained models available on the ImageNet dataset.

The proposed method could be applied to evolve local explanations for any DCNNs, e.g., VGGNet [2], ResNet [3], Inception [53], CiCNet [54], Xception [55], DenseNet [4], and MobileNet [56]. However, it is not feasible to list the experimental results on all of the above DCNNs. In the experiments, three DCNNs are selected—VGGNet, ResNet, and MobileNet with two primary reasons. The first is that each of the three DCNNs represents different DCNN architectures. To be specific, VGGNet is a DCNN with feed-forward topology, ResNet introduces short-cut connections in the topology, and MobileNet proposes a new depth separable convolution filter. The second is that the pretrained models of these three DCNNs are available in most of the popular deep-learning frameworks, which makes the reproduction of the experiments easier. In addition, a further experiment is performed to explain the predictions of the newly proposed Vision Transformer [57]. Transformer has become the state-of-the-art for natural language processing tasks, which was only introduced to solve computer vision tasks dubbed about a year ago. The proposed method is also evaluated on explaining Vision Transformer, which is quite different from other popular DCNNs. Due to the page limit, the comparison results with Vision Transformer are displayed in supplementary materials to this article.

B. Parameter Settings

In Section III-C, the dimension of the encoded vector in the example is 100, i.e., 100 superpixels. However, different values could be used for the number of superpixels. Based on the images of the benchmark dataset—ImageNet, different parts of the images could be separated by segmenting the images to 100 superpixels. This is the major consideration of this parameter. Another point taken into account is the computational cost of evolving the local explanations. As the number of superpixels decides the dimension of the encoded vector, a larger number of superpixels exponentially enlarges the search space, which affects the efficiency of the proposed method. To sum up, the guideline for setting the number of superpixels is to find a relatively small number that could clearly separate different parts of the objects in an image. In other words, end users can examine several values, e.g., 20, 50, 100, 150, and 200 in our case by visualizing whether key features can be separated clearly. Since the proposed method is designed to explain specific predictions, the number of superpixels can be easily adjusted by end users for a given prediction. The parameters for NSGA-II are set according to the community convention [35], [58], which are shown in Table I. In NSGA-II, the mutation rate p_m is calculated by $p_m = (1/n)$, where n is the number of decision variables for real-coded GAs [35].

All the experiments were ran on a single GPU card with the specific model of GeForce GTX 1080.

V. Results and Analysis

The section is devoted to analyzing and discussing the results received from the experiments to examine the effectiveness and efficiency of the proposed MO-LIME method. In the first place, we report the local explanations evolved by MO-LIME based on three CNN models (i.e., **VGGNet**, **ResNet**, and **MobileNet**) on the four images that are randomly selected from the ImageNet dataset. For each CNN model, we visualize two nondominated solutions selected by the proposed method introduced in Section III-G.

A. Evolved Local Explanations

1) Explanations for the Image of Lion: Fig. 8 shows the original image of a lion, and the six solutions of the evolved

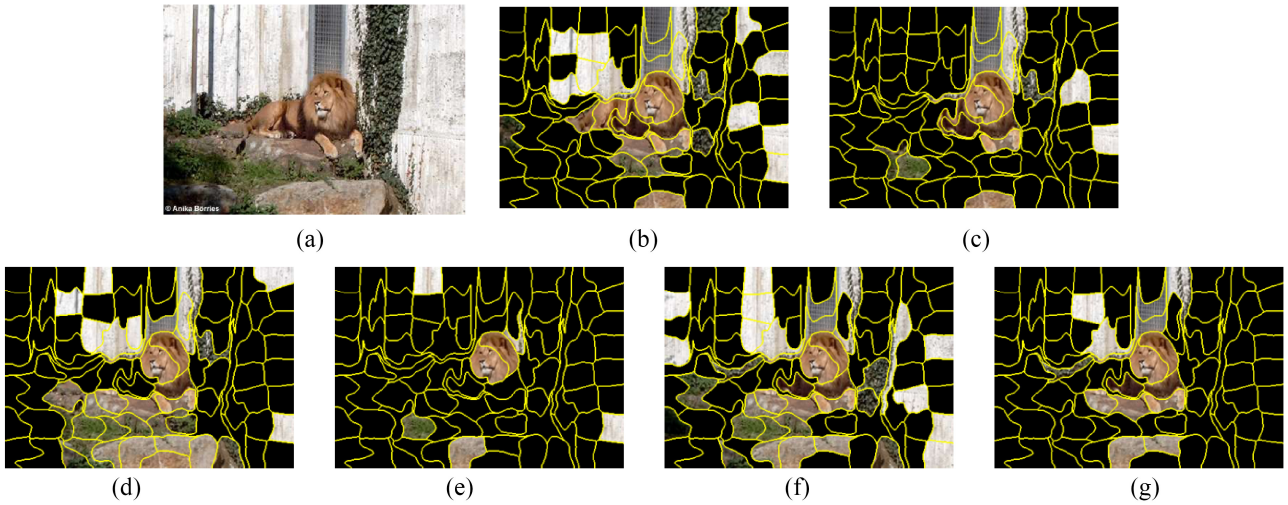

Fig. 8. Evolved local explanations for the image of a lion. From left to right, it is the original image, two evolved local explanations based on **VGGNet**, **ResNet**, and **MobileNet**, respectively. (a) Original image. (b) 1st evolved local explanation for VGGNet. (c) 2nd evolved local explanation for VGGNet. (d) 1st evolved local explanation for ResNet. (e) 2nd evolved local explanation for ResNet. (f) 1st evolved local explanation for MobileNet. (g) 2nd evolved local explanation for MobileNet.

TABLE I
PARAMETER SETTINGS

Parameter	Value
crossover rate	0.9
population size	100
number of generations	50

local explanations for explaining the predictions of the three CNN models on this image. As shown in Fig. 8(a), a lion raises its head, lying on the stone in a corner.

In Fig. 8(b) and (c), two of the evolved local explanations—the two nondominated solutions described in Section III-G, are visualized for explaining the prediction of VGGNet on this image. According to Fig. 8(b), MO-LIME is able to extract interpretable features about the face, head, body, front, and back legs, as the explanation for end users to understand why VGGNet predicts this image into the lion category. Note that this solution from the pareto front achieves the best value on the first objective. As can be seen from Fig. 8(b), in order to increase the chance of predicting this image into the lion category, all of the important interpretable features about the lion are successfully selected. Moreover, most of the irrelevant features in the background are effectively filtered. This could further improve the confidence of VGGNet for making the prediction. For the second solution shown in Fig. 8(c), it performs better than the first solution shown in Fig. 8(b) on the second objective. Obviously, the number of the selected interpretable features is significantly smaller than that of the solution shown in Fig. 8(b). It also shows that the most important features are successfully selected, i.e., facial and head features. The selected interpretable features are informative enough for VGGNet to make the prediction without any other interpretable features, such as body features. Moreover, in Fig. 8(c), a larger number of noisy interpretable features in the background are removed than the first solution shown in Fig. 8(b). By considering the prediction along with the evolved local explanations in Fig. 8(b) and (c), end users may find it easier and more confident to decide whether the prediction is trustworthy or not.

Fig. 8(d) and (e) visualized the evolved local explanations to explain the prediction of ResNet on this image. In Fig. 8(d), MO-LIME successfully extracts facial, head, front legs as interpretable features to explain the prediction of ResNet. Although the body features are not extracted, ResNet is still able to predict correctly based on the extracted interpretable features that are vital to identify the lion. As to the second solution shown in Fig. 8(e), only a few interpretable features are selected to explain the prediction of ResNet. Nearly all of the irrelevant features in the background are effectively filtered by MO-LIME. More importantly, the most discriminative features, i.e., facial features and head contour, are extracted.

Fig. 8(f) and (g) show the evolved explanations for the prediction of MobileNet on this image. Similar to that in Fig. 8(d), the facial features, head, and front legs are also extracted by MO-LIME as interpretable features for explaining the prediction of MobileNet, as shown in Fig. 8(f). It is indicated that the interpretable features about the lion, which are extracted by the first solution in Fig. 8(f), are also extracted by the solution in Fig. 8(g). However, compared to that in Fig. 8(f), most of the irrelevant features in the background are removed, so it is easier for end users to identify and understand important features.

As can be seen from the six solutions of explanations based on the three different CNN models, the proposed MO-LIME method can extract the most discriminative features, i.e., facial features and head contour, as the explanation to identify the lion.

2) Explanations for the Image of Schooner: Fig. 9(a) shows a schooner with four fore-and-aft sails on its two masts, sailing on the sea. The image is predicted into the schooner category by VGGNet, ResNet, and MobileNet.

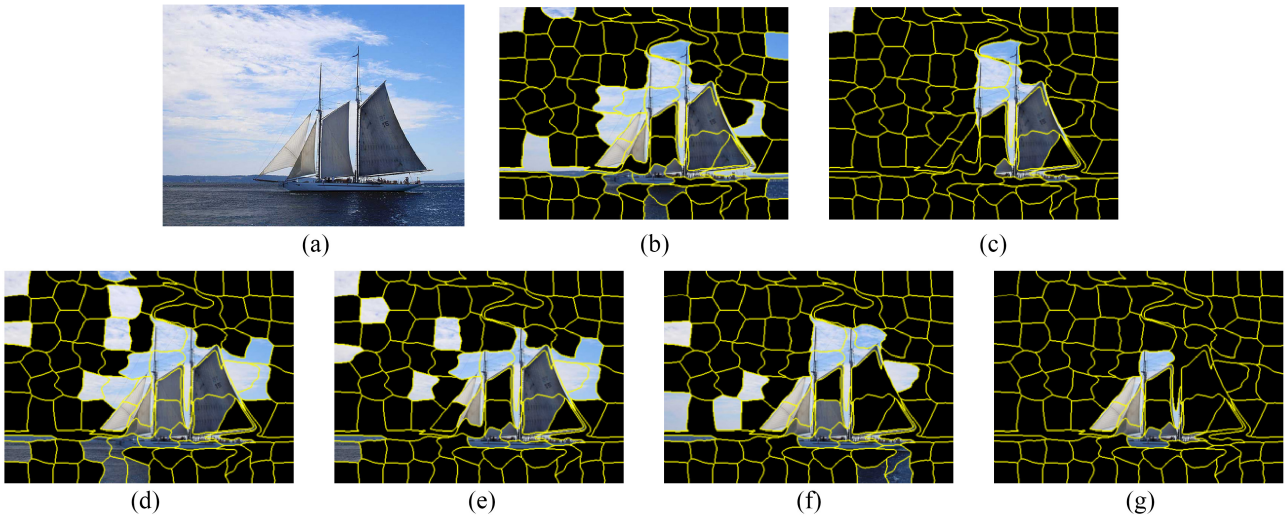

Fig. 9. Evolved local explanations for the image of a schooner. From left to right, it is the original image, two evolved local explanations based on **VGGNet**, **ResNet**, and **MobileNet**, respectively. (a) Original image. (b) 1st evolved local explanation for VGGNet. (c) 2nd evolved local explanation for VGGNet. (d) 1st evolved local explanation for ResNet. (e) 2nd evolved local explanation for ResNet. (f) 1st evolved local explanation for MobileNet. (g) 2nd evolved local explanation for MobileNet.

The explanations evolved for explaining the prediction of VGGNet are visualized in Fig. 9(b) and (c). In Fig. 9(b), it shows that interpretable features about three sails and two masts are selected as the explanation. In Fig. 9(c), the interpretable features about one sail and two masts are selected, which are informative enough for VGGNet to make the prediction. Moreover, nearly all of the irrelevant features in the background are removed to increase the confidence of VGGNet on the prediction.

To explain the prediction of ResNet on this image, as can be seen from Fig. 9(d), all of the key interpretable features about the four sails and two masts are extracted as the explanation. Based on Fig. 9(e), MO-LIME selects interpretable features about three sails and two masts as the explanation, and effectively removes most of the irrelevant features in the background.

Fig. 9(f) and (g) show the evolved explanations by MO-LIME for explaining the prediction of MobileNet on this image. In Fig. 9(f), interpretable features about three sails and two masts are selected to increase the chance of predicting the image into the schooner category by MobileNet. Very interestingly, as shown in Fig. 9(g), all of the irrelevant features in the background are filtered.

Similar to the evolved explanations for the image of a lion in Fig. 8, MO-LIME has the capability of selecting the most informative features as the explanation for the image of a schooner, i.e., sails and masts, which is not limited to a specific CNN model.

3) Explanations for the Image of Baseball Players: In Fig. 10(a), a pitcher wearing a red helmet is jumping, and another baseline player (wearing a black helmet) stands on his left. There are many spectators in the auditorium. The image is predicted into the baseball player category by VGGNet, ResNet, and MobileNet.

The prediction of VGGNet on this image is explained by MO-LIME and the evolved explanations are shown in Fig. 10(b) and (c). As can be seen from the two figures, the important features explaining the baseball pitcher, i.e., the helmet, arms, and legs, are effectively extracted to explain the prediction on this image. In Fig. 10(b), MO-LIME also extracts some interpretable features about the green lawn, baseball field, and spectators. This could somehow help VGGNet distinguish this image from other classes about people in ImageNet. In Fig. 10(c), only a few key interpretable features are selected and the features related to the green lawn are removed.

Fig. 10(d) and (e) visualize the evolved explanations for ResNet, and Fig. 10(f) and (g) visualize the evolved explanations for MobileNet. In Fig. 10(d), MO-LIME extracts important features, such as the back and legs of the pitcher in this image. In both Fig. 10(d) and (e), MO-LIME extracts some interpretable features about the lawn, baseball field, and spectators to increase the confidence of ResNet. Similar to the evolved explanations for ResNet, Fig. 10(f) and (g) also demonstrate the effectiveness of MO-LIME in selecting informative features as model-agnostic explanations (such as the back and legs of the pitcher, the lawn, the baseball field, and spectators) to end users to assist them to determine whether the prediction is trustworthy or not.

4) Explanations for the Image of Computer Monitors: Fig. 11 shows the original image of computer monitors and the local explanations evolved by MO-LIME to explain the prediction of VGGNet, ResNet, and MobileNet. In Fig. 11(a), there are three computer monitors of different sizes (one laptop and two desktop personal computers) on the desk. In addition, a router and a handful of notebooks/magazines are in the front of the two desktop personal computers. This image is predicted into the computer monitor category.

For the explanations with the best values for the first objective, it can be seen from Fig. 11(b), (d), and (f) that the common ground is that the features used to explain the two larger monitors, router, and the keyboard of the laptop are

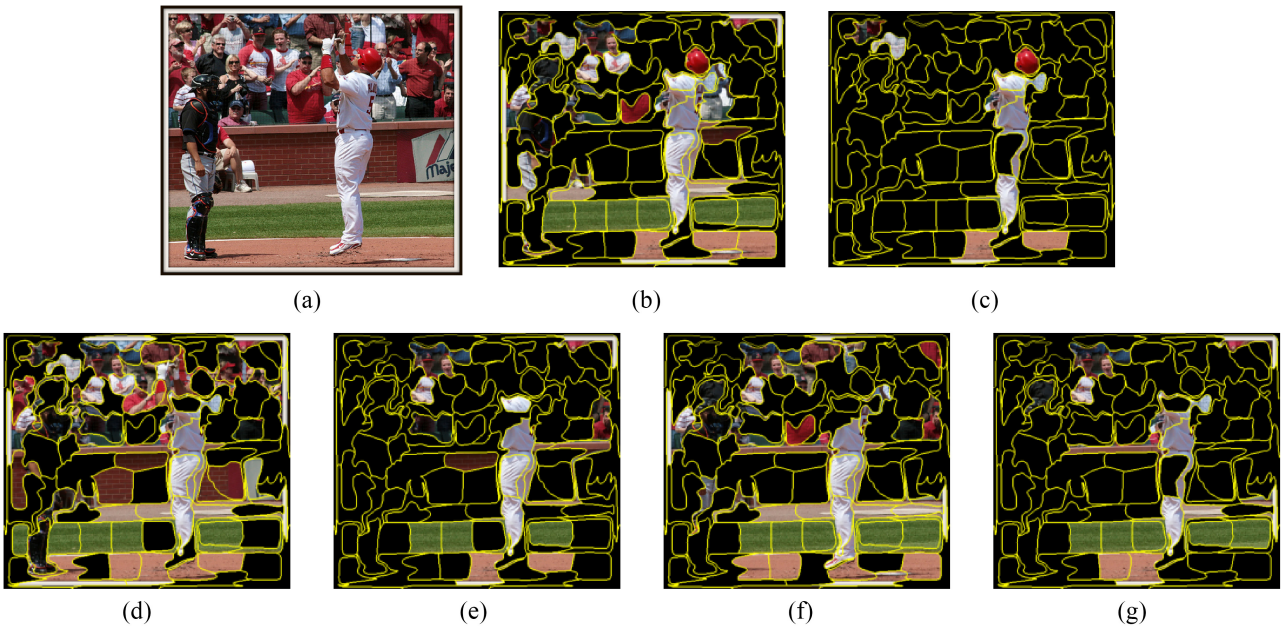

Fig. 10. Evolved local explanations for the image of a baseball player. From left to right, it is the original image, two evolved local explanations based on **VGGNet**, **ResNet**, and **MobileNet**, respectively. (a) Original image. (b) 1st evolved local explanation for VGGNet. (c) 2nd evolved local explanation for VGGNet. (d) 1st evolved local explanation for ResNet. (e) 2nd evolved local explanation for ResNet. (f) 1st evolved local explanation for MobileNet. (g) 2nd evolved local explanation for MobileNet.

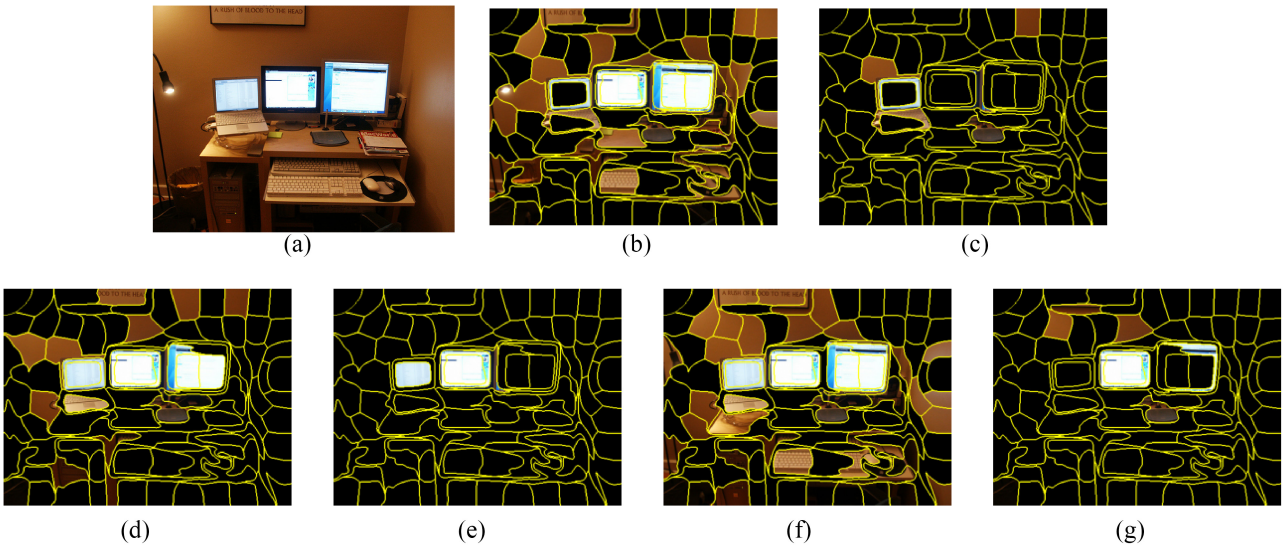

Fig. 11. Evolved local explanations for the image of a computer monitor. From left to right, it is the original image, two evolved local explanations based on **VGGNet**, **ResNet**, and **MobileNet**, respectively. (a) Original image (b) 1st evolved local explanation for VGGNet. (c) 2st evolved local explanation for VGGNet. (d) 1st evolved local explanation for ResNet. (e) 2nd evolved local explanation for ResNet. (f) 1st evolved local explanation for MobileNet. (g) 2nd evolved local explanation for MobileNet.

effectively selected. However, Fig. 11(b) shows that the feature explaining the laptop monitor is not selected (but a feature of the screen frame is selected). This may not cause serious performance loss since the selected interpretable features in Fig. 11(b) are informative enough for VGGNet to predict the image into the computer monitor category.

By looking into Fig. 11(c), (e), and (g), it seems that VGGNet, ResNet, and MobileNet can use a few interpretable features to make the prediction. For example, in Fig. 11(e), MO-LIME only selects three interpretable features about the monitor of a desktop personal computer and the screen frame as the local explanation to explain the prediction of ResNet on this image.

Note that for explaining a prediction made by a CNN model, MO-LIME provides end users with two solutions to help them decide whether the prediction is trustworthy or not. The two solutions could somehow provide supplementary explanations to each other. For example, in Fig. 11(g), MO-LIME selects features explaining a computer monitor, but in Fig. 11(f), all of the interpretable features about the three monitors are effectively selected to increase the confidence of MobileNet to predict this image into the computer monitor

TABLE II
FIRST OBJECTIVE—PROBABILITIES OF **MO-LIME WITH VGGNET, RESNET AND MOBILENET**

	Original (Probability)	MO-LIME (Probability)	
Images	**Original**	**Best**	**Mean ± Std**
VGGNet			
Lion	0.9997	0.999995	0.999982 ± 0.000012
Schooner	0.7938	0.9919	0.9864 ± 0.0033
Baseball player	0.9411	0.9995	0.9992 ± 0.0002
Monitor	0.3825	0.8066	0.7634 ± 0.0226
ResNet			
Lion	0.8167	0.9973	0.9920± 0.0047
Schooner	0.9528	0.9978	0.9936 ± 0.0026
Baseball player	0.9770	0.9990	0.9978 ± 0.0008
Monitor	0.2688	0.9190	0.8723 ± 0.0197
MobileNet			
Lion	0.9111	0.9945	0.9895 ± 0.0040
Schooner	0.8783	0.9947	0.9904 ± 0.0029
Baseball player	0.9714	0.9991	0.9981± 0.0006
Monitor	0.3926	0.9029	0.87789 ± 0.0158

1: Std stands for standard deviation.

TABLE III
TRAINING TIME OF **MO-LIME WITH VGGNET, RESNET, AND MOBILENET (IN SECONDS)**

Images	Training Time (Shortest)	Training Time (Mean ± Std)
VGGNet		
Lion	59.7020	70.9354 ± 7.5369
Schooner	64.9522	65.9657 ± 0.6102
Baseball player	69.8477	71.7756 ± 0.9481
Monitor	65.2579	66.8131 ± 0.7857
ResNet		
Lion	42.9649	54.2428 ± 7.1115
Schooner	49.0864	50.2651± 0.6870
Baseball player	53.7376	55.1131 ± 0.7820
Monitor	48.6974	50.3731 ± 0.8768
MobileNet		
Lion	42.9649	54.2428 ± 7.1115
Schooner	49.0864	50.2651± 0.6870
Baseball player	53.7376	55.1131 ± 0.7820
Monitor	48.6974	50.3731 ± 0.8768

1: Std stands for standard deviation.

category. Oppositely, in Fig. 11(f), some less important features in the background are selected, but in Fig. 11(g), nearly all of the irrelevant features are effectively removed. After presenting the two solutions to the end users, it is relatively intuitive and informative for them to trust the prediction.

B. Pareto Front Analysis

Fig. 12 shows pareto fronts evolved by MO-LIME with VGGNet, ResNet and MobileNet on the four images, respectively. Generally speaking, the nondominated solutions are regularly distributed across an entire pareto front. In addition, as can be shown from Fig. 12, the convergence to a pareto front is usually different when MO-LIME is used to explain a prediction of different CNN models on an image.

In Fig. 12(a)–(c), it can be found that a majority of the nondominated solutions in a pareto front are able to achieve a very good performance on the first objective (i.e., very close to 1.0). This demonstrates that MO-LIME could make a CNN model become more confident in making a prediction on an image. Note that in Section V-A, we visualized two solutions selected from a pareto front to show the evolved local explanations for explaining the prediction of a CNN model on an image, and found that the two solutions sometimes provide complementary explanations as additional evidence or further references. Basically, MO-LIME is capable of providing end users with a set of tradeoff solutions from a pareto front to meet requirements of different users.

C. Performance Comparisons

The VGGNet section of Table II reports the original probabilities achieved by VGGNet in the second column and the results of MO-LIME on the four images. Overall, on all of the four images, the proposed MO-LIME method enhances the original probabilities achieved by VGGNet, based on the comparisons between the mean probability and original probability on each image. In more detail, on the three images (i.e., lion, schooner, and baseball player), the mean probabilities achieved by MO-LIME are very close to 1, and the gap between the mean and best probabilities is very narrow. The VGGNet section of Table II also shows that the standard deviation of results from the 30 runs is very small on every image, which could demonstrate the good stability of the proposed method. On the image of computer monitors, MO-LIME significantly improves the original probability from 0.3825 to 0.7634. The improvements in probabilities could make users feel more confident to trust the predictions or easier to determine whether these predictions are trustworthy or not. The VGGNet section of Table III reports the training time of MO-LIME. MO-LIME consumes around 1 m on each image.

The ResNet section of Table II reports the results of MO-LIME with ResNet on the four images. First of all, MO-LIME is able to increase the original probabilities achieved by ResNet on the four images, based on the comparisons between the original probabilities and the mean probabilities of MO-LIME. Particularly on the image of Monitor, MO-LIME enhances the original probability of ResNet from 0.2688 to 0.8723. By comparing the original probabilities achieved by VGGNet and ResNet, ResNet performs better than VGGNet on images of Schooner (15.90% higher) and Baseball player (3.59% higher), while it performs worse than VGGNet on images of Lion (18.3% lower) and Monitor (11.37% lower). However, it is noteworthy that the mean probabilities achieved by MO-LIME with VGGNet and ResNet are very close on three images, i.e., lion, schooner, and baseball player. This shows that the evolved local explanations could improves the confidence of all three DCNN models, which supports the model-agnostic characteristic of the proposed method.

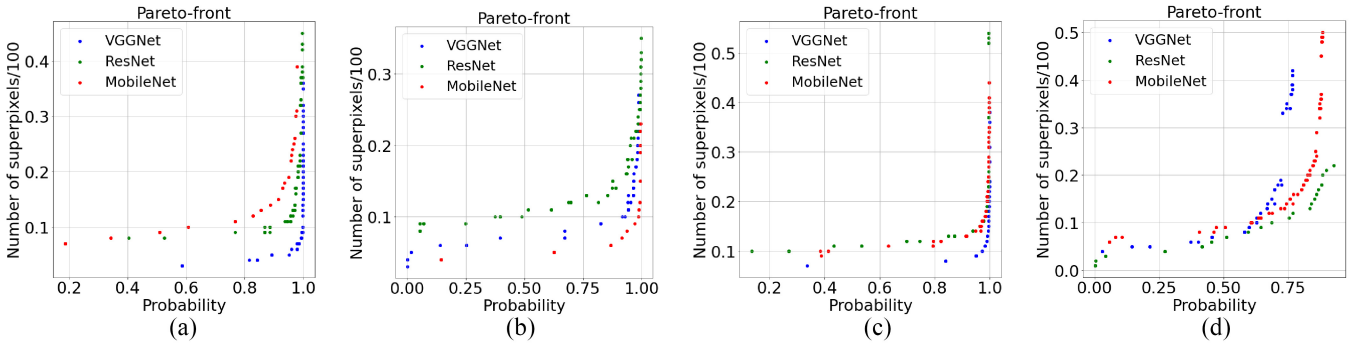

Fig. 12. Pareto fronts for the images of lion, schooner, baseball player, and monitor. (a) Lion. (b) Schooner. (c) Baseball player. (d) Monitor.

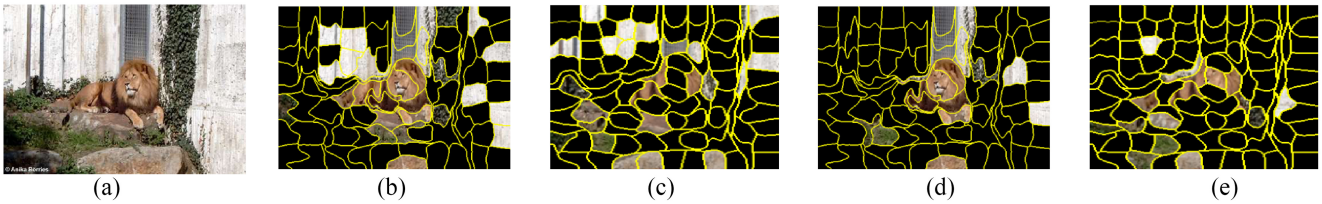

Fig. 13. Comparison of local explanations for the image of a lion between MO-LIME and LIME based on **VGGNet**. (a) Original image. (b) MO-LIME first solution. (c) LIME first solution. (d) MO-LIME second solution. (e) LIME second solution.

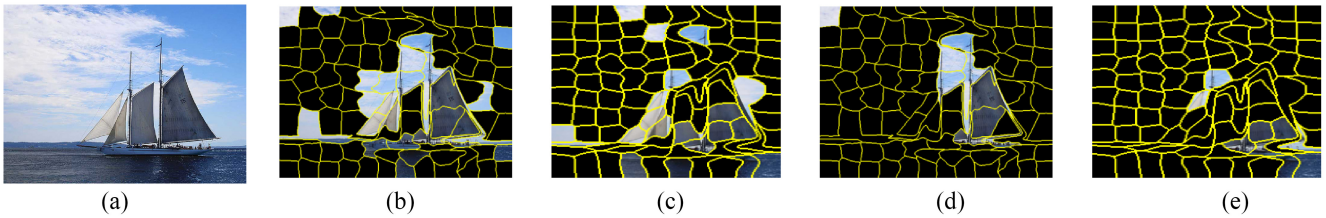

Fig. 14. Comparison of local explanations for the image of a schooner between MO-LIME and LIME based on **VGGNet**. (a) Original image. (b) MO-LIME first solution. (c) LIME first solution. (d) MO-LIME second solution. (e) LIME second solution.

The MobileNet section of Table II shows the results of MO-LIME based on MobileNet on the four images. It is not difficult to find that the original probabilities of MobileNet are also improved by MO-LIME on all of the four images, which are similar to that for VGGNet and MobileNet. By comparing the results on the four images, according to Table II, it is found that the mean probabilities of MO-LIME based on the three CNN models are very close. Therefore, it further proves the proposed MO-LIME is applicable to evolve model-agnostic explanations for any deep models.

By analyzing the training time reported in the ResNet section and the MobileNet section of Table III, it is noteworthy that MO-LIME consumes very similar training time when working with VGGNet or MobileNet, due mainly to roughly the same computational costs in the objective evaluation process of MO-LIME.

D. Comparison With LIME

LIME [14] needs to set the number of interpretable features. However, the proposed method automatically evolves the number of interpretable features. To perform fair comparisons, the number of interpretable features for LIME is set to the same as the nondominated solution found by MO-LIME. Two LIME explanations for the first nondominated solution and the second nondominated solution are obtained by using the corresponding number of interpretable features. Section V-A shows that MO-LIME achieves meaningful explanations for all three of the DCNN models. The comparison between LIME and MO-LIME is performed based on VGGNet due to the page limit.

In Fig. 13, the facial characteristics of the lion are clearly presented in the two nondominated solutions of MO-LIME. However, LIME only includes parts of the facial characteristics. For example, the long hair of the lion is the main feature in the first solution of LIME, while the eyes and nose are added in the second solution of LIME. Therefore, MO-LIME provides more meaningful features in this particular case.

By comparing the first and second solutions obtained by LIME and MO-LIME in Fig. 14, both LIME and MO-LIME can spot parts of the two masts and several sails as the interpretable features. Although the interpretable features in the explanations of LIME and MO-LIME are not the same, the difference is marginal, so LIME and MO-LIME perform similar to each other in the case of explaining VGGNet on the schooner image.

By further examining the explanations of LIME and MO-LIME on the images of a baseball player and a monitor in Figs. 15 and 16. Both of the first and second solutions

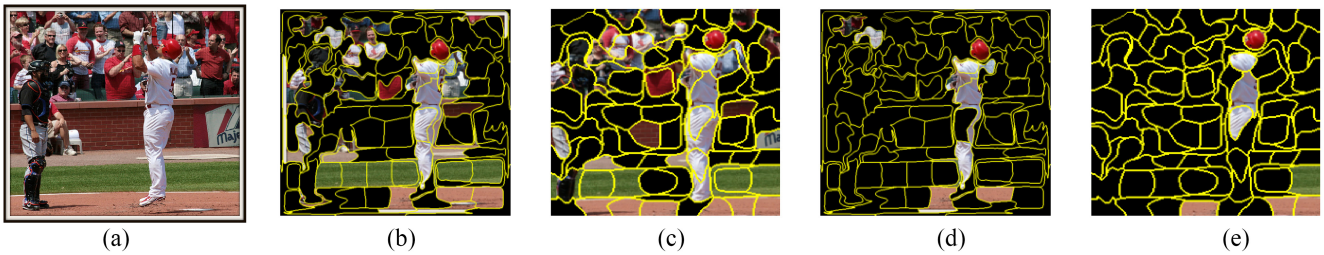

Fig. 15. Comparison of local explanations for the image of a baseball player between MO-LIME and LIME based on **VGGNet**. (a) Original image. (b) MO-LIME first solution. (c) LIME first solution. (d) MO-LIME second solution. (e) LIME second solution.

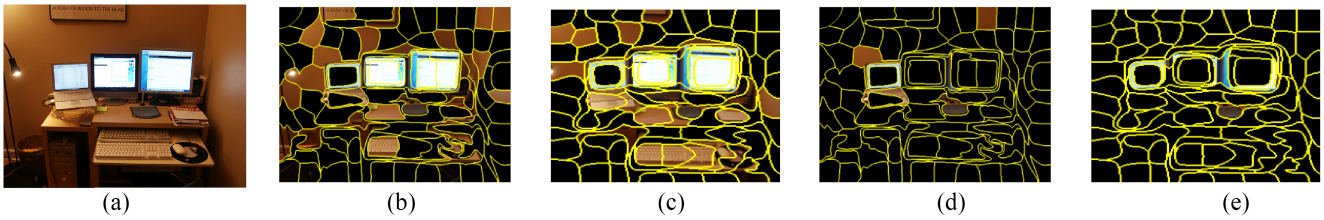

Fig. 16. Comparison of local explanations for the image of a monitor between MO-LIME and LIME based on **VGGNet**. (a) Original image. (b) MO-LIME first solution. (c) LIME first solution. (d) MO-LIME second solution. (e) LIME second solution.

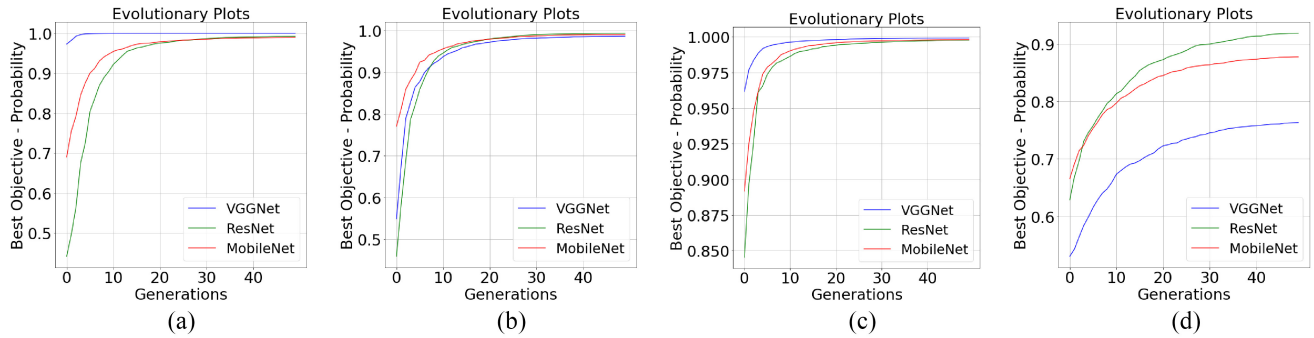

Fig. 17. Convergence curves. From left to right, there are convergence curves for the images of lion, schooner, baseball player, and monitor, respectively. (a) Lion. (b) Schooner. (c) Baseball player. (d) Monitor.

found by LIME and MO-LIME for both of the images include very similar interpretable features. This means that LIME and MO-LIME explain VGGNet on the two local images—a baseball player and a monitor in a very similar way. To sum up, both LIME and MO-LIME can achieve meaningful local explanations.

E. Convergence Analysis

Fig. 17 shows the convergence curves of MO-LIME on the four images, where the blue, green, and red curves indicate the convergence curves of MO-LIME based on VGGNet, ResNet, and MobileNet, respectively.

As can be seen from the three curves in Fig. 17(a), all of the three evolutionary learning processes have converged for the image of lion. Clearly, the three curves start from different original probabilities based on the three CNN models, and then keep an upward trend in the following generations, ending with roughly the same probabilities for the image of lion. As to VGGNet, MO-LIME starts from a high probability and then converges at earlier generations (the blue curve). Fig. 17(b) shows the convergence curves on the image of schooner. Similar to the curves in Fig. 17(a), the starting points of the three curves in Fig. 17(b) are obviously different, while the three curves end with almost the same probabilities after reaching a steady state. According to Fig. 17(c), for the image of a baseball player, the overall trend of the three convergence curves is similar to that for the images of lion and schooner.

Fig. 17(d) shows the convergence curves of MO-LIME on the image of monitor. It can be seen that the three curves keep an increasing trend and become relatively stable after around 40 generations. The blue curve for VGGNet is under the other two curves for ResNet and MobileNet across all the generations. By further looking into the results about this image in Table II, the mean probability of MO-LIME with VGGNet is also lower than that of MO-LIME with ResNet and MobileNet. In Fig. 11(d) and (f), the features about all the three monitors are selected as the explanations based on ResNet and MobileNet, while according to Fig. 11(b), the interpretable feature about the laptop monitor is not selected based on VGGNet. However, the selected interpretable features in Fig. 11(b) are still informative enough for humans to understand the prediction of VGGNet.

Note that the major goal of this article is not only to enhance the probability of a prediction but more importantly to evolve

explanations that are expected to assist end users to understand behaviors behind the prediction of an over-complicated model.

VI. Conclusion and Future Work

In conclusion, the overall goal of proposing a multiobjective EC-based model-agnostic method to effectively and efficiently evolve local explanations for DCNNs in image classification has been successfully achieved by attaining the following contributions. First, two objectives, which contribute to obtaining good-quality local explanations, have been designed. Second, an encoding strategy has been designed to encode the interpretable features, after which NSGA-II has been applied to optimize the two objectives simultaneously. Third, a new method of selecting two nondominated solutions, which could benefit the end users most, have been developed. Third, the in-depth analyses on the evolved local explanations and the convergence curves have been performed to gain profound insights. From the experimental results, two main claims are supported—the evolved local explanations are understandable to humans, and the proposed method is model agnostic, which can be used to explain any DCNNs. Finally, the local explanation method is important in real-world applications, especially in the areas that need essential trust and understanding of the learned models, such as the medical domain and the legal field. This is another contribution to the EC community to demonstrate the powerfulness of EC algorithms in real-world applications.

This article has presented a method to explain the behaviors of DCNNs on specific predictions to help end users decide whether to trust the predictions or not. Another potential usage of local explanations is to assist end users to choose DCNN models. In LIME, a method has been designed to select a small number of examples from the whole training dataset that could well represent the whole dataset. The local explanations are obtained based on the selected examples, and then the local explanations are evaluated. If the local explanations for all the selected examples on a DCNN makes more sense than another, the former DCNN could be selected due to better explanations. However, LIME was only targeted at selecting a subset of examples for text classification tasks. It would be promising to propose an effective and efficient method to select a subset of images that can be used to select DCNNs for image classification. Moreover, as there are other methods in XDL to explain DCNNs, it would be interesting to investigate new methods to interpret the behaviors of DCNNs in a broad sense. Furthermore, investigating other more advanced methods for generating superpixels could potentially achieve better results. A typical example could be image classification for binary images, where SLIC cannot be used to generate superpixels, so it is important to explore other segmentation methods in addition to SLIC.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[6] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6848–6856.

[7] H. Lee et al., "An explainable deep-learning algorithm for the detection of acute intracranial haemorrhage from small datasets," *Nature Biomed. Eng.*, vol. 3, no. 3, pp. 173–182, 2019.

[8] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (XAI): Toward medical XAI," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 4793–4813, Nov. 2021.

[9] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 5, pp. 726–742, Oct. 2021.

[10] L. Zou et al., "Ensemble image explainable AI (XAI) algorithm for severe community-acquired pneumonia and COVID-19 respiratory infections," *IEEE Trans. Artif. Intell.*, early access, Feb. 25, 2022, doi: 10.1109/TAI.2022.3153754.

[11] N. Frosst and G. Hinton, "Distilling a neural network into a soft decision tree," 2017, *arXiv:1711.09784*.

[12] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu, "Interpreting CNNs via decision trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6261–6270.

[13] Q. Zhang, R. Cao, F. Shi, Y. N. Wu, and S.-C. Zhu, "Interpreting CNN knowledge via an explanatory graph," 2017, *arXiv:1708.01785*.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you? Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Min.*, 2016, pp. 1135–1144.

[15] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations." in *Proc. AAAI*, vol. 18, 2018, pp. 1527–1535.

[16] M. Suganuma, S. Shirakawa, and T. Nagao, "A genetic programming approach to designing convolutional neural network architectures," in *Proc. Genet. Evol. Comput. Conf.*, 2017, pp. 497–504.

[17] B. Wang, Y. Sun, B. Xue, and M. Zhang, "Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2018, pp. 1–8.

[18] B. Wang, Y. Sun, B. Xue, and M. Zhang, "A hybrid differential evolution approach to designing deep convolutional neural networks for image classification," in *Proc. Aust. Joint Conf. Artif. Intell.*, 2018, pp. 237–250.

[19] R. Miikkulainen et al., "Evolving deep neural networks," in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*. San Diego, CA, USA: Elsevier, 2019, pp. 293–312.

[20] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 394–407, Apr. 2020.

[21] B. Wang, B. Xue, and M. Zhang, "A hybrid GA-PSO method for evolving architecture and short connections of deep convolutional neural networks," in *Proc. Trends Artif. Intell.*, 2019, pp. 650–663.

[22] T. Elsken, J. H. Metzen, and F. Hutter, "Efficient multi-objective neural architecture search via Lamarckian evolution," 2018, *arXiv:1804.09081*.

[23] B. Wang, Y. Sun, B. Xue, and M. Zhang, "Evolving deep neural networks by multi-objective particle swarm optimization for image classification," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, 2019, pp. 490–498.

[24] Z. Lu et al., "NSGA-Net: Neural architecture search using multi-objective genetic algorithm," in *Proc. Genet. Evol. Comput. Conf.*, 2019, pp. 419–427.

[25] Y. Chen et al., "RENAS: Reinforced evolutionary neural architecture search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4787–4796.

[26] B. Wang, B. Xue, and M. Zhang, "Particle swarm optimization for evolving deep convolutional neural networks for image classification: Single-and multi-objective approaches," in *Deep Neural Evolution*. Singapore: Springer, 2020, pp. 155–184.

[27] J. Z. Liang, E. Meyerson, and R. Miikkulainen, "Multiobjective Coevolution of deep neural network architectures," U.S Patent 16 671 274, Jul. 2020.
[28] H. Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu, "Hierarchical representations for efficient architecture search," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–13.
[29] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. 33rd AAAI Conf. Artif. Intell. (AAAI)*, 2019, pp. 4780–4789.
[30] B. Wang, B. Xue, and M. Zhang, "Particle swarm optimisation for evolving deep neural networks for image classification by evolving and stacking transferable blocks," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2020, pp. 1–8.
[31] J. Ren, Z. Li, J. Yang, N. Xu, T. Yang, and D. J. Foran, "EIGEN: Ecologically-inspired GENetic approach for neural network structure searching from scratch," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9059–9068.
[32] H. Zhang, Y. Jin, R. Cheng, and K. Hao, "Efficient evolutionary search of attention convolutional networks via sampled training and node inheritance," *IEEE Trans. Evol. Comput.*, vol. 25, no. 2, pp. 371–385, Apr. 2021.
[33] B. Wang, B. Xue, and M. Zhang, "Surrogate-assisted particle swarm optimization for evolving variable-length transferable blocks for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3727–3740, Aug. 2022.
[34] J. Dong, B. Hou, L. Feng, H. Tang, K. C. Tan, and Y.-S. Ong, "A cell-based fast memetic algorithm for automated convolutional neural architecture design," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 17, 2022, doi: 10.1109/TNNLS.2022.3155230.
[35] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[36] M. G. C. Tapia and C. A. C. Coello, "Applications of multi-objective evolutionary algorithms in economics and finance: A survey," in *Proc. IEEE Congr. Evol. Comput.*, 2007, pp. 532–539.
[37] T. T. Teo et al., "Optimization of fuzzy energy-management system for grid-connected microgrid using NSGA-II," *IEEE Trans. Cybern.*, vol. 51, no. 11, pp. 5375–5386, Nov. 2021.
[38] X. Ma, Z. Huang, X. Li, Y. Qi, L. Wang, and Z. Zhu, "Multiobjectivization of single-objective optimization in evolutionary computation: A survey," *IEEE Trans. Cybern.*, early access, Dec. 22, 2021, doi: 10.1109/TCYB.2021.3120788.
[39] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[40] A. B. Arrieta et al., "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020.
[41] S. Mishra, B. L. Sturm, and S. Dixon, "Local interpretable model-agnostic explanations for music content analysis." in *Proc. ISMIR*, 2017, pp. 537–543.
[42] A. Henelius, K. Puolamäki, H. Boström, L. Asker, and P. Papapetrou, "A peek into the black box: Exploring classifiers by randomization," *Data Min. Knowl. Disc.*, vol. 28, no. 5, pp. 1503–1529, 2014.
[43] A. P. Moore, S. J. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
[44] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 211–224.
[45] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 6, pp. 583–598, Jun. 1991.
[46] N. Xie, G. Ras, M. van Gerven, and D. Doran, "Explainable deep learning: A field guide for the uninitiated," 2020, *arXiv:2004.14545*.
[47] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.
[48] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.
[49] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3145–3153.
[50] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[51] R. Iyer, Y. Li, H. Li, M. Lewis, R. Sundar, and K. Sycara, "Transparency and explanation in deep reinforcement learning neural networks," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, 2018, pp. 144–150.
[52] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, "This looks like that: Deep learning for interpretable image recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.
[53] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.
[54] Y. Pang, M. Sun, X. Jiang, and X. Li, "Convolution in convolution for network in network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1587–1597, May 2018.
[55] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.
[56] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
[57] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[58] T. Back, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford, U.K.: Oxford Univ. Press, 1996.

Bin Wang (Student Member, IEEE) is currently pursuing the Ph.D. degree in computer science with the School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand.

He has been severing as a Reviewer for top international journals, such as IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, and IEEE TRANSACTIONS ON CYBERNETICS.

Wenbin Pei (Member, IEEE) received the Ph.D. degree from the Victoria University of Wellington, Wellington, New Zealand, in 2021.

She is currently an Assistant Professor with the Dalian University of Technology, Dalian, China.

She has been serving as a Reviewer for international journals, such as IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and IEEE TRANSACTIONS ON CYBERNETICS.

Bing Xue (Senior Member, IEEE) received the B.Sc. degree from the Henan University of Economics and Law, Zhengzhou, China, in 2007, the M.Sc. degree in management from Shenzhen University, Shenzhen, China, in 2010, and the Ph.D. degree in computer science from the Victoria University of Wellington (VUW), Wellington, New Zealand, in 2014.

She is currently a Professor of Computer Science and the Program Director of Science with the School of Engineering and Computer Science, VUW. She has over 200 papers published in fully refereed international journals and conferences and her research focuses mainly on evolutionary computation, machine learning, evolving deep neural networks, image analysis, and multiobjective machine learning.

Mengjie Zhang (Fellow, IEEE) received the B.E. and M.E. degrees from Artificial Intelligence Research Center, Agricultural University of Hebei, Baoding, Hebei, China, in 1989 and 1992, respectively, and the Ph.D. degree in computer science from RMIT University, Melbourne, VIC, Australia, in 2000.

He is currently a Professor of Computer Science, the Head of the Evolutionary Computation Research Group, and the Associate Dean (Research and Innovation) with the Faculty of Engineering. He has published over 800 research papers in refereed international journals and conferences. His current research interests include evolutionary computation, particularly with application areas of image analysis, multiobjective optimization, and transfer learning.