Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Ensemble-based genetic algorithm explainer with automized image segmentation: A case study on melanoma detection dataset

Hossein Nematzadeh [a,c,*], José García-Nieto [a,b,c], Ismael Navas-Delgado [a,b,c], José F. Aldana-Montes [a,b,c]

[a] ITIS Software, Universidad de Málaga, Arquitecto Francisco Peñalosa 18, Malaga, 29071, Spain
[b] Biomedical Research Institute of Málaga (IBIMA), Universidad de Málaga, Malaga, Spain
[c] Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Malaga, Spain

ARTICLE INFO

Keywords:
Explainable Artificial Intelligence
Local Interpretable Model-agnostic Explanations
Deep learning
Genetic algorithm
Melanoma dataset

ABSTRACT

Explainable Artificial Intelligence (XAI) makes AI understandable to the human user particularly when the model is complex and opaque. Local Interpretable Model-agnostic Explanations (LIME) has an image explainer package that is used to explain deep learning models. The image explainer of LIME needs some parameters to be manually tuned by the expert in advance, including the number of top features to be seen and the number of superpixels in the segmented input image. This parameter tuning is a time-consuming task. Hence, with the aim of developing an image explainer that automizes image segmentation, this paper proposes Ensemble-based Genetic Algorithm Explainer (EGAE) for melanoma cancer detection that automatically detects and presents the informative sections of the image to the user. EGAE has three phases. First, the sparsity of chromosomes in GAs is determined heuristically. Then, multiple GAs are executed consecutively. However, the difference between these GAs are in different number of superpixels in the input image that result in different chromosome lengths. Finally, the results of GAs are ensembled using consensus and majority votings. This paper also introduces how Euclidean distance can be used to calculate the distance between the actual explanation (delineated by experts) and the calculated explanation (computed by the explainer) for accuracy measurement. Experimental results on a melanoma dataset show that EGAE automatically detects informative lesions, and it also improves the accuracy of explanation in comparison with LIME efficiently. The python codes for EGAE, the ground truths delineated by clinicians, and the melanoma detection dataset are available at https://github.com/KhaosResearch/EGAE.

1. Introduction

The paramount role of Artificial Intelligence (AI) in reasoning, learning complex computational tasks for Decision Support Systems (DSSs), and data analytics is not hidden from anyone. The DSSs utilize AI aspects (such as medical DSSs) that may deal with human lives. As such, it is a vital need for clinicians (as one of the primary target audiences identified by Arrieta et al. [1]) to understand the pieces of evidence behind the decisions of DSS [2]. In fact, the explanation of the predictions has reached the same level of importance as the predictions' accuracy in recent years. Thus, clinicians will not adopt accurate predictive models with weak explanations. In addition, trustworthy AI in the ethical AI context also makes clinicians solely adopt simultaneously high performance and interpretable prediction models [3]. Generally, the need for explanation of machine learning algorithms increases as the sophistication and complexity of learning grows. Ensemble models and deep learning have the highest levels of complexity. In contrast, regressions, decision trees, and rule-based models are explicit and clear so that the effect of each predictor variable on the response variable is simply traceable [4]. Explainable AI (XAI) emerged to address this problem through series of methods to unveil the obscurity of complex ML models somehow. Literature also reveals the rising evolution of the scientific trends in XAI during the last years [5,6]. Explainers can be divided from three different perspectives. The first well-known classification is model-agnostic against model-specific. Unlike model-specific explainers that are learning model-dependent, model-agnostic explainers are more flexible in model selection and can be used to evaluate and compare the performance of different models. In other major classification, explainers are categorized into intrinsic against

[*] Corresponding author at: ITIS Software, Universidad de Málaga, Arquitecto Francisco Peñalosa 18, Malaga, 29071, Spain.
E-mail addresses: hnematzadeh@uma.es, hn_61@yahoo.com (H. Nematzadeh), jnieto@uma.es (J. García-Nieto), ismael@uma.es (I. Navas-Delgado), jfaldana@uma.es (J.F. Aldana-Montes).

https://doi.org/10.1016/j.compbiomed.2023.106613
Received 28 October 2022; Received in revised form 5 January 2023; Accepted 28 January 2023
Available online 5 February 2023

post-hoc. Intrinsic explainers are Machine Learning (ML) models that are interpretable such as decision trees and linear models. Post-hoc explainers are applied to the model for interpretability after training the learning model. Finally, explainers could be either local or global so that local explainers explain the individual predictions while global explainers explain the entire model. A good explainer also needs to be interpretable so that the explanation is readily understandable to the human user [1].

One of the main applications of XAI is the explanation of prediction achieved by the deep learning models in Melanoma cancer [7,8]. Melanoma cancer is a skin pigmentation disorder that develops from melanocytes which can be diagnosed by biopsy. The early diagnosis in the golden time is essential. Otherwise, it is difficult to treat using chemotherapy or radiation therapy because abnormal melanocyte cells spread rapidly to other tissues of the body and become highly metastasis and increase the number of deaths [9]. American cancer society reported that the 5-year relative survival rates for melanoma skin cancer decreases to 30% when the cancer spread to distant parts of the body [10]. In one classification, skin pigmentation disorders are categorized into three groups: melanoma, nevus, and seborrheic-keratosis. Nevi and seborrheic keratoses are benign non-cancerous disorders. Nevi are moles, and seborrheic-keratoses are more common in elderlies. Deep learning is being tested for the classification and prediction of melanoma against non-cancerous classes [11]. However, many layers and huge parametric space make Deep Neural Networks (DNNs) a complex black box learning model. Therefore, explaining such models to clinicians in the context of trustworthy AI is inevitable. There are well-known explainers which have been used for explaining the prediction of deep learning on melanoma datasets including Gradient-weighted Class Activation Mapping (Grad-CAM) [12], SHapley Additive exPlanations (SHAP) [7], and Local Interpretable Model-agnostic Explanations (LIME) [13]. Inspired from LIME, the concentration of this paper is on developing an image explainer with an automized image segmentation so that the existing gaps, contributions, and novelty of this paper are described in the following.

LIME is a post-hoc explainer that can show meaningful areas in a given sample image (with its image explainer library) on a deep learning model trained for melanoma prediction [13]. The user intervention in specifying the number of superpixels of the intended input image for explanation and the number of top features to be seen within the image (which is not always straightforward for experts to determine them manually) in the process of explanation of LIME was the inspiration and motivation of the contributions of this research. As such, to address this gap the current research is done with the following contributions:

1. To develop a heuristic approach to intelligently discard trivial solutions in the initial population that leads to better convergence of GA
2. To develop a model-agnostic explainer for images by ensembling consecutive genetic algorithms (GAs) namely, Ensemble-based Genetic Algorithm Explainer (EGAE)
3. To include EGAE with consensus and majority voting strategies for automatic specification of the important pixels positively contributed to classification

Thereby, the novelty of this paper is to improve LIME formulation by substituting the surrogate model in LIME image explainer (usually linear regression) with genetic algorithms. To the best of our knowledge, this is the first attempt in the field to eliminate expert intervention in manually determining the number of superpixels of the input image as well as the final top features contributed positively in classification.

The rest of this paper is organized as follows. Section 2 divides the existing works into two categories and clarifies in what category the paper belongs. Section 3 explains GA and LIME briefly. Section 4 presents the proposed method (EGAE) including three phases. Section 5 presents the experiments and illustrates the results. This section also argues about the findings. Section 6 states the concluding remarks and outlines the advantages and limitations of EGAE.

2. Related works

As stated earlier in Section 1, this paper tries to resolve the problem of user intervention in the process of LIME image explainer. Even though investigation of the literature does not reveal any clear attempt to address this gap by existing studies, but there are still some other papers that tried to add value to the existing LIME in other directions. Thus, Section 2.1 identifies the recent papers that resolved the recognized limitations of LIME in different directions. In contrast, the goal of some other research in Section 2.2 are to enhance the image segmentation algorithm to help doctors in diagnosing diseases. The results of this paper are directly compared with the existing LIME to show how the proposed methodology is able to resolve the existing gaps.

2.1. Research on LIME improvements

There are bulks of research that tried to either directly improve LIME in different directions under recognized gaps or compare LIME with existing explainers based on well-known metrics such as execution time, reproducibility of the results, etc. Perhaps one of the most recent papers that concentrated on solely comparison of LIME with existing explainers on melanoma detection dataset is the work by Hurtado et al. [7] in which it was technically argued how LIME outperforms SHapley Additive exPlanations (SHAP) for the differential diagnosis of pigmented skin lesions in a melanoma dataset. Generally, SHAP has a core kernel explainer which is slow but appropriate for any model. Likewise, SHAP has optimized variations (deep explainer and gradient explainer) which can be used for deep learning. Hurtado et al. [7] experimentally investigated the difference between LIME (with three segmentation algorithms to show the top 5 superpixels) and SHAP gradient explainer for a pre-trained ResNet model on a pigmentation skin dataset containing three classes (melanoma, nevus, and seborrheic-keratosis). The results showed that LIME had better reproducibility and execution time than SHAP gradient explainer. Following the success of LIME and its extensive application on image data, literature reveals many attempts to improve LIME as a leading image explainer including Anchor LIME [14], KL-LIME [15], NormLIME [16], LIME-Aleph [17], MPS-LIME [18], and the most recent one LIMEcraft [8] so that each one tried to add a value to the existing LIME. Anchor LIME [14] used anchors to support high precision and precise coverage of interpretable explanations. Anchors are some parts of the image that are sufficient for prediction. Moreover, Anchor LIME superimposed another image over the rest of the superpixels, instead of hiding the rest of the superpixels as LIME does. Anchor LIME enhanced model prediction on unseen data. KL-LIME [15] was an extension of LIME to Bayesian models appropriate for different types of predictions. It worked by minimizing the Kullback–Leibler divergence computed between the predictive distributions of the original model and the explanation models' predictive distributions. The proposed method was evaluated on the benchmark MNIST digit dataset [19] with a concentration on the classification of 3s and 8s. NormLIME [16] was a new metric for feature importance. It was indeed an attempt to aggregate the local explanation of LIME to form a global explanation by adding proper normalization to the computation of the global weights for features. NormLIME also used the MNIST dataset for experimental evaluations. LIME-Aleph [17] explained a classifier decision based on logic rules calculated from Inductive Logic Programming Aleph which yielded a richer explanation. Generally, LIME forgoes the relationship between specific parts in an image as an explanatory factor. This method was tested on blocksworld domain images. Modified Perturbed Sampling operation for LIME (MPS-LIME) [18] improved the existing sampling operation in LIME. It was done by paying attention to complicated correlation between features by converting the superpixel image into an undirected graph. Various experiments on Google's pre-trained Inception neural network revealed that MPS-LIME had better fidelity, understandability, and efficiency than LIME. LIMEcraft [8] claimed that LIME might have static and meaningless explanations because explanations of LIME do not take into account the semantic meaning of the explained objects. Thus, LIMEcraft allowed user intervention to select semantically consistent regions and examine the prediction for the image instance. The main algorithm of LIMEcraft lets the users decide among several options: to upload a prepared mask of superpixels, draw an irregular shape path, choose the number of superpixels inside and outside the mask, or change image features by editing them (editing images increases the robustness of the model). LIME used the quick shift segmentation algorithm, but LIMEcraft used manual or predefined superpixel selections followed by the K-means clustering algorithm. The experimental results revealed that LIMEcraft improved model safety.

2.2. Research on image segmentation

While many works are focusing on improving and adding value to the existing LIME, there exist other works which concentrate on proposing totally other image segmentation techniques. Binjun et al. [20] used ANN to recognize lung cancer and then the lesion area was automatically selected using the proposed lung cancer segmentation algorithm. Binjun et al. [20] used Dice similarity coefficient measure and Average Surface Distance (ASD) as the main evaluation indicators as well as Positive Predictive Values (PPV) as the auxiliary evaluation to show how the segmentation results of the algorithm proposed in this study were close to the doctor's annotations. Literature also shows many attempts to develop novel segmentation techniques including Qi et al. [21], Wang et al. [22] and Su et al. [23]. Qi et al. [21] proposed a multi-level image segmentation model (MIS-XMACO) to improve the effectiveness and efficiency of image segmentation in Covid-19 X-rays. MIS-XMACO used ant colony optimization with both directional crossover (DX) and mutation (DM) strategy initially. MIS-XMACO also incorporated two-dimensional (2D) histograms, 2D Kapur's entropy, and a nonlocal mean strategy. MIS-XMACO resulted in superior segmentation than other models at different threshold levels. Wang et al. [22] proposed an image segmentation method that performs well for the situations where it is difficult to distinguish the target lesion boundary against the background. The proposed method utilized graph theory and guided feathering so that guided feathering algorithm was initially used for roughly separation of the foreground from background image. Then, graph-based algorithm was used to accurately segment the images. Finally, the segmented images were merged to create the result. Su et al. [23] proposed a multilevel thresholding image segmentation (MTIS) method based on an enhanced multi-verse optimizer (CCMVO). Inspired from original multi-verse optimizer CCMVO also used horizontal and vertical search mechanisms. The combination of MTIS and CCMVO had good segmentation results on COVID-19 chest radiography datasets based on Feature Similarity Index (FSIM), the Peak Signal to Noise Ratio (PSNR), and the Structural Similarity Index (SSIM) evaluation metrics.

One of the significant limitations of LIME (and the approaches based on it) is the need for the user's intervention. Thus, the user must manually determine the number of top features and the number of superpixels in the input image. The top features are those for which the best superpixels convey helpful information regarding the prediction class identified by the $num\_features$ parameter in LIME library. Likewise, the user (expert) must also specify the number of superpixels along with a segmentation algorithm manually. As such, this paper proposes an automatic explanation of the prediction model using Ensemble-based Genetic Algorithm Explainer (EGAE). EGAE also implicitly increases the accuracy of explanations compared with LIME by detecting and discarding less important features LIME keeps. In addition, EGAE is an interpretable explainer that is also model-agnostic, post hoc, and local. This proposal tries to keep an acceptable rate of reproducibility for explanations. Thereby, EGAE belongs to the category of research intended to add value to LIME, similar to those in Section 2.1.

3. Preliminaries

This section briefly presents general concepts of both Genetic Algorithm (GA) for solving discrete problems and LIME. Additionally, the limitations of LIME on automatic explanation are discussed.

3.1. Genetic algorithm

Genetic Algorithms [24] belong to the family of guided random search evolutionary algorithms and thus, suffer a lack of reproducibility of the results. However, GAs guarantee optimization by discovering and recombining good building blocks of solutions. GAs have three well-known operators (namely selection, cross-over, and mutation) in which mutation is the prominent operator that guarantees optimization. The selection operator assigns probabilities to solutions based on their fitness. Then, these probabilities are used to select the best parents for recombination in the exploitation phase by cross-over. Mutation provides good coverage of the search space by exploring new regions. Different methods and variations are available for each operator and GAs, respectively. The different variations of GAs refer to how they combine operators. GAs are particularly practical for problems with discrete search spaces like N-Queens [25], image encryption [26] or mathematical functions and optimizations [27], among others. Algorithm 1[1] shows one of the famous variations of GA used in this research:

1. The problem is defined initially in line 1 by specifying the fitness function and number of genes in chromosomes.
2. Line 2 sets the parameters of GA (population size, maximum number of iterations, cross-over percentage, number of offsprings generated from cross-over, mutation percentage, number of mutants) and those needed for the parent_selection function in line 6 (for example, selection pressure if the parent_selection function is a roulette wheel).
3. The initialization of GA is done so that the initial population (*Pop*) is generated and evaluated in lines 3–4, and the *best_solution* is stored accordingly.
4. The cross-over population (Pop_{C_i}) and mutants (Pop_{M_i}) are both generated and evaluated so that the population in the next generation ($Pop_{(i+1)}$) is created together with Pop_i in the main loop of GA starting from line 5 to line 14. There are some options for termination criterion in line 5, including reaching the time limit (maximum number of iterations), passing time (consecutive iterations) without improvement in the fitness function, and reaching satisfactory fitness.

3.2. LIME

Local Interpretable Model-agnostic Explanations (LIME) is an explanation technique approximating any black box learning model [28,29]. The general idea of LIME with its image explainer is to generate a specified number of images (π_x) in the vicinity of the image that needs explanation (x) using a segmentation algorithm initially. Then, the prediction model (f) is used to predict $\hat{y}$ (the set of predicted labels for (π_x)). Next, LIME weighs (π_x) and calculates (w_x) using a distance metric (the default distance metric is cosine) to calculate the distance of each member of (π_x) to x. The general formulation of LIME is stated in Eq. (1), where g is a surrogate model with low complexity and high interpretability, which belongs to the class of G. Eq. (1) minimizes $\epsilon(x)$ so that $L(f, g, (\pi_x))$ and $\Omega(g)$ are the locality-aware loss and the complexity of explanation. Assuming g is a linear regression surrogate model. As such, a linear model (lm) can be fitted using (π_x), $\hat{y}$, and (w_x)

[1] Variables are italicized to enhance the readability of algorithms in this research

Algorithm 1 One of the variations of Genetic Algorithm

```
Input: $problem$: a problem
Output: $best\_solution$
Define the $problem$
Set GA parameters
Generate random initial population of $Pop_{(i=0)}$
Evaluate $Pop_{(i=0)}$ and find $best\_solution_{(i=0)}$
while termination criterion is not satisfied do
    $Pop_{S_i} \leftarrow$ parent_selection $(Pop_i)$
    $Pop_{C_i} \leftarrow$ cross_over $(Pop_{S_i})$
    Evaluate $Pop_{C_i}$
    $Pop_{M_i} \leftarrow$ mutate $(Pop_i)$
    Evaluate $Pop_{M_i}$
    $Pop_{(i+1)} \leftarrow$ Generate next generation from $Pop_i$, $Pop_{C_i}$, and $Pop_{M_i}$
    Find $best\_solution_{(i+1)}$ from $Pop_{(i+1)}$
    $i \leftarrow i + 1$
end while
Return $best\_solution$
```

to calculate coefficients ($\hat{\beta}$) as in Eq. (2). Finally, the list of $\hat{\beta}$ is sorted and the superpixels contribute to the prediction label are recognized. The user can manually select the best coefficients to visualize the best superpixels accordingly.

$$\epsilon(x) = \arg\min_{g\in G} L(f, g, (\pi_x)) + \Omega(g) \tag{1}$$

$$lm \sim ((\pi_x), \hat{y}, (w_x)) \tag{2}$$

The limitation of the LIME image explainer is the intervention of users in determining the number of superpixels in (π_x) that is a time-consuming task since it is not always clear whether the number of superpixels should be high or low. Furthermore, the user also needs to determine the number of top features (best coefficients in ($\hat{\beta}$) in case of using linear model regression for g) to be seen. Thus, this paper decreases user intervention by defining an automatic approach for the explanation.

4. Proposed method

Fig. 1 reflects three contributions of the paper discussed in the introduction section by illustrating three phases of EGAE in an abstract technical view. First, the sparsity of chromosomes in GAs is determined based on a heuristic search in phase 1. This helps a better generation of the initial population in GA and leads to better convergence by selecting non-trivial solutions in the initial population of GA. Second, multiple GAs are executed consecutively so that each GA is devised for a specified number of superpixels of the input image. The result of each GA, which is a unique image with some active superpixels, is finally recorded at the end of phase 2. It is tried to evaluate the input image using different sizes of superpixels with respective GAs (the size of superpixels is determined at the beginning of execution of each GA). Thus, the unique images in phase 2 contain important areas of the image, including small to large superpixels. Third, the recorded images are finally ensembled in phase 3. EGAE generates two images based on consensus and majority voting strategies from existing images computed in phase 2. Consensus voting and majority voting images contain pixels of the input image that automatically explain the classification result. Consensus voting has a rigorous approach to explanation which could negatively affect the interpretability of the explanations in some minority cases. However, it provides good accuracy of explanations. In contrast, majority voting usually has lower accuracy of explanation than consensus voting but has more stability and provides better interpretability. As such, using both explanations (through two voting strategies) simultaneously can provide a good perception to the expert on the classification logic. The input of EGAE is an image divided into superpixels, and the outputs are two images generated using pixels that positively contribute to classification. The remaining subsections explain and elaborate on each phase of EGAE.

4.1. Determining the sparsity of chromosomes

Each chromosome in EGAE equals a unique perturbation. Each perturbation equals (π_x) in Eq. (1) or Eq. (2). As such, chromosomes are encoded in binary with genes of 0 and 1 so that 0 and 1 show the inactiveness and activeness of the respective superpixel within the input image. The sparsity of chromosomes in the initial population of GAs refers to the number of genes with 0 value (inactive superpixels) in each chromosome with respect to the number of genes with 1 value (active superpixels). EGAE controls this sparsity using a heuristic algorithm. The general idea is to identify the portion of the image that mainly contributes to prediction. If this portion is a vast part of the image, EGAE decreases the sparsity of chromosomes. Otherwise, the solutions in the initial population of the GA may have very low $\hat{y}$ and could not improve the fitness considerably during consecutive iterations using cross-over and mutation operators. In other words, controlling the sparsity of chromosomes results in more informative chromosomes in the GA's initial population, which affects better and more efficient convergence. Thus, initially, the input image is divided into small superpixels (h) using a segmentation algorithm. Then, the fitness of all possible perturbations of the input image ($2^h - 1$) are calculated heuristically (excluding the perturbation in which all superpixels are inactive). As such, the weighted sum scalar fitness function shown in Eq. (3) is calculated based on the prediction accuracy and the number of active superpixels in each perturbation. The best perturbation has the greatest accuracy (accuracy of the prediction model f in Eq. (1)) with the least number of active superpixels. The weights of α and β are experimentally fixed to 0.7 and 0.3, respectively. Finally, the perturbation with the best fitness is selected for sparsity calculation. The calculation of chromosomes' sparsity is shown using a piecewise function in Eq. (4). v is the number of active superpixels in the best perturbation. Eq. (4) shows that as v/h increases, the sparsity decreases by increasing the value of φ in [0.5, 0.9] accordingly. The value of 0.9 for φ reveals that with the probability of 0.9 a gene in a binary chromosome is 1 and otherwise is 0. Algorithm 2 shows how, heuristically the sparsity of the chromosomes in the initial population of GA can be calculated.

$$Fitness = \alpha\ (accuracy\ (perturbation)) + \beta\ (\frac{(h - v + 1)}{h}) \tag{3}$$

$$\begin{cases} \varphi = 0.5, & v/h \le 0.5 \\ \varphi = v/h, & 0.5 < v/h < 1 \\ \varphi = 0.9, & v/h = 1 \end{cases} \tag{4}$$

Generally, Algorithm 2 starts by defining $temp$ as a variable to record the fitness of the best perturbation of the input image and follows by creating an empty list of z with size 2^h in line 2 (recalling that h shows the entire number of superpixels within an image and equals 5 in this research). Next, each element of z is initialized with a binary array within $[1, 2^h]$ in line 4. The binary array will be used to generate the respective perturbed image. For example, the binary array of $[1, 1, 0, 0, 0]$ shows that the perturbed image only has two active superpixels. As such, the perturbation of the input image that has identical active/inactive superpixels to the binary array is generated using perturb function in line 5 and recorded in $perturbed\_image$. Line 6 calculates the $yhat\_max$ and line 7 calculates the respective label and records it in $predicted\_label$. Line 8–10 records both fitness (based on Eq. (3)) and the number of active superpixels of the best $perturbed\_image$ in $temp$ and v respectively. At the end, the sparsity is calculated in lines 13–19 based on Eq. (4). Generally, the greater φs denote less sparse chromosomes in EGAE.

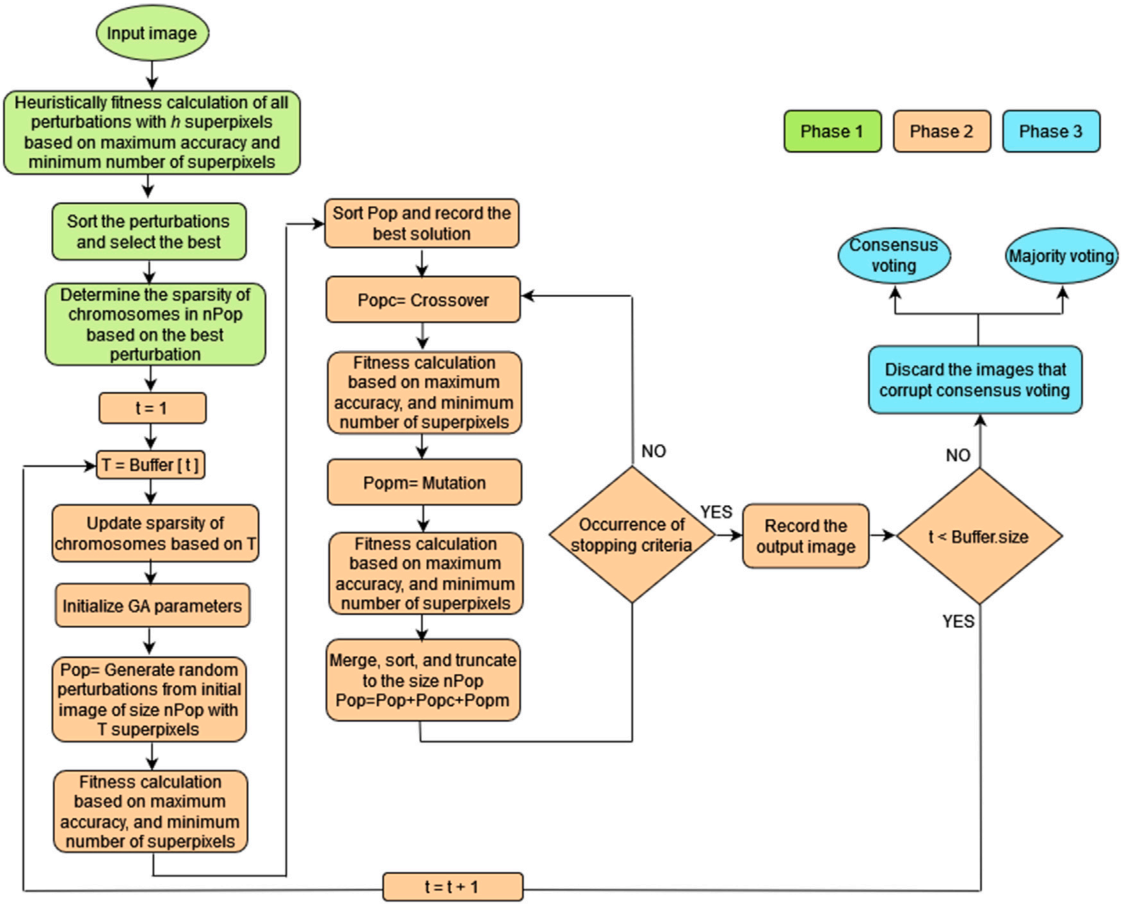
Input image
Heuristically fitness calculation of all perturbations with h superpixels based on maximum accuracy and minimum number of superpixels
Sort the perturbations and select the best
Determine the sparsity of chromosomes in nPop based on the best perturbation
t = 1
T = Buffer [t]
Update sparsity of chromosomes based on T
Initialize GA parameters
Pop= Generate random perturbations from initial image of size nPop with T superpixels
Fitness calculation based on maximum accuracy, and minimum number of superpixels
Sort Pop and record the best solution
Popc= Crossover
Fitness calculation based on maximum accuracy, and minimum number of superpixels
Popm= Mutation
Fitness calculation based on maximum accuracy, and minimum number of superpixels
Merge, sort, and truncate to the size nPop Pop=Pop+Popc+Popm
Occurrence of stopping criteria
NO
YES
Record the output image
t < Buffer.size
NO
YES
Discard the images that corrupt consensus voting
Consensus voting
Majority voting
Phase 1
Phase 2
Phase 3
t = t + 1

Fig. 1. Phases of EGAE.

Algorithm 2 Sparsity calculation of chromosomes

```
Input: $h$, $image$, $model$, $real\_label$
Output: $\varphi$
$temp \leftarrow 0$
$z = [None] * 2^h$
for $i \leftarrow 1$ to $2^h$ do
    $z[i] \leftarrow$ decimal_to_binary($i$)
    $perturbed\_image \leftarrow$ perturb($image, z[i]$)
    $yhat\_max \leftarrow$ max($model$.predict($perturbed\_image$))
    $predicted\_label \leftarrow model$.predict($perturbed\_image$).index($yhat\_max$)
    if (fitness($perturbed\_image$) $> temp$) and ($predicted\_label = real\_label$) then
        $temp \leftarrow$ fitness($perturbed\_image$)
        $v \leftarrow$ sum($z[i]$)
    end if
end for
if $v/h \leq 0.5$ then
    $\varphi \leftarrow 0.5$
else if $0.5 < v/h < 1$ then
    $\varphi \leftarrow v/h$
else
    $\varphi \leftarrow 0.9$
end if
Return $\varphi$
```

4.2. Genetic algorithms

This section shows how multiple GAs are executed consecutively to generate respective multiple images with informative sections from the input image. Each execution is based on a different number of superpixels. As stated in Section 4.1, the chromosomes are binary, and the sparsity of chromosomes is calculated based on φ. Our experiments show that executing 5 GAs with 10, 15, 20, 25, and 100 superpixels is an appropriate approach, which leads to satisfactory results. This aspect is shown by the Buffer array in Fig. 1. The Buffer.size is set to 5, and the buffer elements are set to 10, 15, 20, 25, and 100 accordingly. The sparsity of chromosomes specified in Section 4.1 is appropriate when the number of superpixels is not very large. As such, this section starts by updating the sparsity calculated from Section 4.1 so that even if v/h is less than one, but the number of superpixels equals 100, φ is updated to 0.9. The general parameters of EGAE are shown in Table 1 as well. Phase 2 then continues by generating an initial population (Pop) of size nPop as well as fitness calculation of the members of Pop based on Eq. (3). Then, Pop members are sorted, and the best solution is identified. EGAE repeats selection, cross-over, and mutation until stopping criteria occur. First, EGAE selects the best parents for applying cross-over and then applies bit string mutation to generate Pop_c and Pop_m populations respectively as stated in Fig. 1. For this purpose, EGAE uses Roulette Wheel Selection (RWS) in Algorithm 3 so that the probability measure of each solution (P_i) is calculated using Boltzmann distribution with a selection pressure of θ and fitness of each solution (f_i) as shown in Eq. (5) before applying RWS. WorstFit in Eq. (5) makes θ independent of the fitness function. Selection pressure ranges in $[0, \infty)$ so that zero gives the same chance to all solutions and

Table 1
General parameters of GAs.

GA parameters	# of superpixels				
	10	15	20	25	100
nPop	5	10	15	15	35
Pc	0.9				
Pm	0.4				
Size of Popc	2 * round (Pc * nPop/2)				
Size of Popm	round (Pm * nPop)				
Cross-over	Single-point				
Mutation	Bit string				
Selection	Roulette wheel				
Maximum number of iterations	150				
Stopping criteria	10 consecutive iterations without fitness improvement or reaching the maximum number of iterations				

∞ assigns the entire chance to the best solution as stated in Eq. (6). EGAE selects the selection pressure θ so that Eq. (7) holds in which H is the set of half-best solutions. Based on the experiments, GAs with greater number of superpixels in EGAE have smaller selection pressures.

$$P_i = \frac{e^{-}\theta(\frac{\frac{1}{f_i}}{\frac{1}{WorstFit}})}{\sum_j e^{-}\theta(\frac{1}{f_j})} \tag{5}$$

where

$$P_i \propto -\theta(\frac{1}{f_i}), \quad \sum_i P_i = 1$$

$$\begin{cases} \theta = 0, & P_i = \frac{1}{nPop} \forall i \\ \theta \to \infty, & \begin{cases} P_i = 1, & \textit{best solution} \\ P_i = 0, & \textit{others} \end{cases} \end{cases} \tag{6}$$

$$\sum_{i \in H} P_i = 0.8 \tag{7}$$

Algorithm 3 RWS

```
Input: $P$
Output: $i$
$r \sim U(0,1)$
$C_i \leftarrow \sum_{j \in 1}^{i} P_j$
$i \leftarrow \min \{j \mid r \leq C_j\}$
Return $i$
```

Algorithms 4 and 5 show single-point cross-over and mutation, respectively. Even though in Algorithm 5, one gene of the chromosome is flipped, if the number of superpixels is greater than 25, 3 genes of the chromosome are flipped in implementation to improve exploration. In contrast to single-point cross-over that uses RWS, selection in bit string mutation is random. After mutation, Pop, Pop_c, and Pop_m, are merged, sorted based on the calculated fitness from Eq. (3), and truncated to the size of $nPop$. As such, the best solution in Pop is the answer of the first generation in GA. Finally, if stopping criteria occur, EGAE will record the best solution as the explainable image with the most informative superpixels of the input image for a given GA.

4.3. Ensemble of GAs

The final phase of EGAE is ensembling the results (images) of the GAs from phase 2. EGAE shows two images to explain the model: consensus voting and majority voting. Consensus voting refers to the intersection of images, a rigorous approach to explaining the prediction model. Majority voting has a lenient approach so that the resulting image contains the pixels that appeared in most images.

Algorithm 4 Single-point cross-over

```
Input: $x_1$, $x_2$
Output: $y_1$, $y_2$
$nVar \leftarrow$ length($x_1$)
$c \leftarrow$ randint($1, nVar - 1$)
$y_1 \leftarrow x_1\ [0, c]\ +\ x_2\ [c\ :]$
$y_2 \leftarrow x_2\ [0, c]\ +\ x_1\ [c\ :]$
Return $y_1$, $y_2$
```

Algorithm 5 Bit string mutation

```
Input: $x$
Output: $y$
$nVar \leftarrow$ length($x$)
$j \leftarrow$ randint($1, nVar$)
$y \leftarrow x$
$y[j] \leftarrow 1 - x[j]$
Return $y$
```

In some minority cases, consensus voting may lose interpretability to achieve high accuracy. This is where majority voting could be used to assure the interpretability of the explanation. Consensus voting and majority voting ensure the accuracy and interpretability of the results simultaneously. It should be noted that the intersection of all images could be an empty image indicating one or some of the images spot distinct sections of the input image that corrupt the intersection. This has many reasons, such as an inappropriately great number of segmentations for an image with a wide area of explanability. As such, these segmentations could generate many global optima for EGAE so that GA may detect a distinct superpixel as an explainable lesion in each run. This also implicitly reveals that the GA with inappropriate segmentation should be discarded. Therefore, it is initially investigated whether to discard one or some images for voting. Algorithm 6 generally detects and discards redundant images within a loop and sends valid images to Algorithm 7 for applying voting strategies to generate respective explanations. In this research, EGAE incorporates 5 GAs and, thus, generates 5 images as the input of Algorithm 6 (t=5). Algorithm 7 checks whether an identical pixel (i, j) is active in all ($consensus\_voting\_image$) or majority ($majority\_voting\_image$) of valid images in $list\_of\_valid\_images$ in lines 7–13.

Algorithm 6 Detection of redundant images

```
Input: $image_1, image_2, ..., image_t$
Output: $CV$, $MV$
$temp\_list\_of\_images[1, ..., t]\ \leftarrow [\ image_1, image_2, ..., image_t]$
while intersection of $temp\_list\_of\_images = \emptyset$ do
    $redundant\_images \leftarrow$ Images which cause intersection to fail
end while
$list\_of\_valid\_images \leftarrow$ Remove $redundant\_images$ from $temp\_list\_of\_images$
$(CV,\ MV) \leftarrow$ voting($list\_of\_valid\_images$)
Return $CV$, $MV$
```

Algorithm 7 The voting function to calculate CV and MV

```
Input: $list\_of\_valid\_images$
Output: $consensus\_voting\_image$, $majority\_voting\_image$
no_img ← length($list\_of\_valid\_images$)
$[n, m]$ ← dimension(images in $list\_of\_valid\_images$)
$consensus\_voting\_image(n, m)$ ← zeros$(n, m)$
$majority\_voting\_image(n, m)$ ← zeros$(n, m)$
for $i \leftarrow 1$ to $n$ do
    for $j \leftarrow 1$ to $m$ do
        $temp\_pixels$ ← Select pixels in position$(i, j)$ from $list\_of\_valid\_images(1, ..., no\_img)$
        $no\_active\_pixels$ ← Calculate number of active pixels in position$(i, j)$ in $temp\_pixels$
        if all pixels in $temp\_pixels$ are active then
            $consensus\_voting\_image(i, j)$ ← Highlight the pixel in position$(i, j)$
        else if $no\_active\_pixels > \lfloor \frac{no\_img}{2} \rfloor$ then
            $majority\_voting\_image(i, j)$ ← Highlight the pixel in position$(i, j)$
        end if
    end for
end for
Return $consensus\_voting\_image$, $majority\_voting\_image$
```

5. Experimental results

This section initially provides relevant information about the dataset under study, the predictive deep learning model, its architecture, and the experimental setups, including hardware/software used in Section 5.1 followed by the metrics to evaluate the explanation in Section 5.2. Then, the results and respective discussions are illustrated and explained in Sections 5.3 and 5.4, respectively.

5.1. Data description

The melanoma detection dataset is available in Kaggle repository[2] as a classification dataset. The dataset consists of one cancerous label (melanoma) with 374 samples. Likewise, it has two non-cancerous labels including 254 seborrheic-keratosis , and 1372 nevus.

So, we get a total of 2000 images showing an imbalance. Thus, oversampling the minority classes is used to balance the training data. It is worth mentioning that neither undersampling the majority class nor a simultaneous mixture of oversampling the minority classes and undersampling the majority class was not as effective as oversampling the minority class in resulting in better outcomes. Furthermore, data augmentation techniques, including rescaling, rotating, width-shift, height-shift, and horizontal-flip are applied to training and validation data (recalling that oversampling is solely done for training data and test data do not need to be neither balanced nor augmented as unseen data). After balancing, the distribution of each class in the training data equals 1372 and the entire training data contains 4116 observations. Afterwards, the pre-trained ResNet50 convolutional Deep Learning model is reused in a customized model with a test accuracy of 0.73 (the best achieved). The best weights are saved as a separate file for reusability [7]. Table 2 shows the characteristics of the melanoma dataset before and after balancing the training data. Fig. 2, illustrates the images that are used for the evaluation of EGAE. The selected images in Fig. 2 represent EGAE well and understandably (both for clinicians and computer scientists). Hence, this paper focuses on eight images for experimental analysis from test data that could better represent the outputs calculated by EGAE. The prediction model correctly classified the label for the selected test images with accuracy of almost 100%. The images in Fig. 2 also have scale, hair, and blue sign as possible noises. All the experiments have been conducted in a virtualization environment on a private, high-performance cluster computing platform. This infrastructure is located at the Ada Byron Research Center at the University of Málaga (Spain). It comprises several IBM hosting racks for storage, virtualisation units, server compounds, and backup services. Our virtualization platform is hosted in this computational environment. Concretely, this platform is made up of a CPU with Intel(R) Xeon(R) Gold 6130 @ 2.10 GHz, maximum 2 TB of HDD, maximum 64 GB of RAM, and Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-1049-kvm x86 64). All simulations of EGAE are coded and executed in Python 3.9[3] software environment. EGAE uses the Simple Linear Iterative Clustering (SLIC) segmentation algorithm. Likewise, LIME uses the default parameters in the Python LIME library except for the segmentation function, which is set to SLIC for having a fair comparison with EGAE.

Table 2
Specification of melanoma detection dataset.

Data	Total Observations	Melanoma	Seborrheic-keratosis	Nevus
Train before balancing	2000	374	254	1372
Train after balancing	4116	1372	1372	1372
Validation	150	30	42	78
Test	600	117	90	393

5.2. Measurement criteria

The Number of Function Evaluations (NFE) and error of explanation are the criteria used for performance analysis. Most of the research in XAI evaluates the accuracy of the explanation empirically. However, this paper proposes a numerical approach to explain the error of calculation. As such, the measurement criteria are introduced as follows:

Number of Function Evaluations (NFE): The NFE refers to the number of times the fitness function in an evolutionary algorithm (GA in this paper) is called, which explicitly shows the number of images EGAE uses for explanation. The NFE is a fair metric more reliable than the CPU time, particularly when different algorithms under different implementations are compared. In other words, the number of images the explainer uses for explanation is fairer as a measurement criterion instead of the execution time spent. The general formula of NFE for evolutionary algorithms is shown in Eq. (8) in which I is the number of solutions in the initial population and O is the total number of offsprings generated in each iteration. Eq. (9) is a particular case of Eq. (8) used in this paper. The NFE in EGAE is calculated while execution, however, the number of images required by LIME for evaluation is manually allocated prior to execution.

$$NFE = I + [O \times number\ of\ iterations] \tag{8}$$

$$NFE = Pop + [(Pop_c + Pop_m) \times number\ of\ iterations] \tag{9}$$

Explanation error: The good explanation should necessarily emulate and unveil the classifiers' decision-making procedure. Moreover, it should sufficiently meet the clinicians' diagnosis. Thus, the clinicians from Hospital Regional Universitario de Malaga were asked to specify the informative sections from lesions that mainly contribute to the diagnosis ($actual\_explanation$). This information is used to calculate the distance from the result of EGAE ($calculated\_explanation$) in Eq. (10) using Euclidean distance. The best case is obviously when the $calculated\_explanation$ conforms $actual\_explanation$ so that the error

[2] In URL https://www.kaggle.com/datasets/wanderdust/skin-lesion-analysis-toward-melanoma-detection

[3] In URL https://www.python.org/downloads/release/python-390/

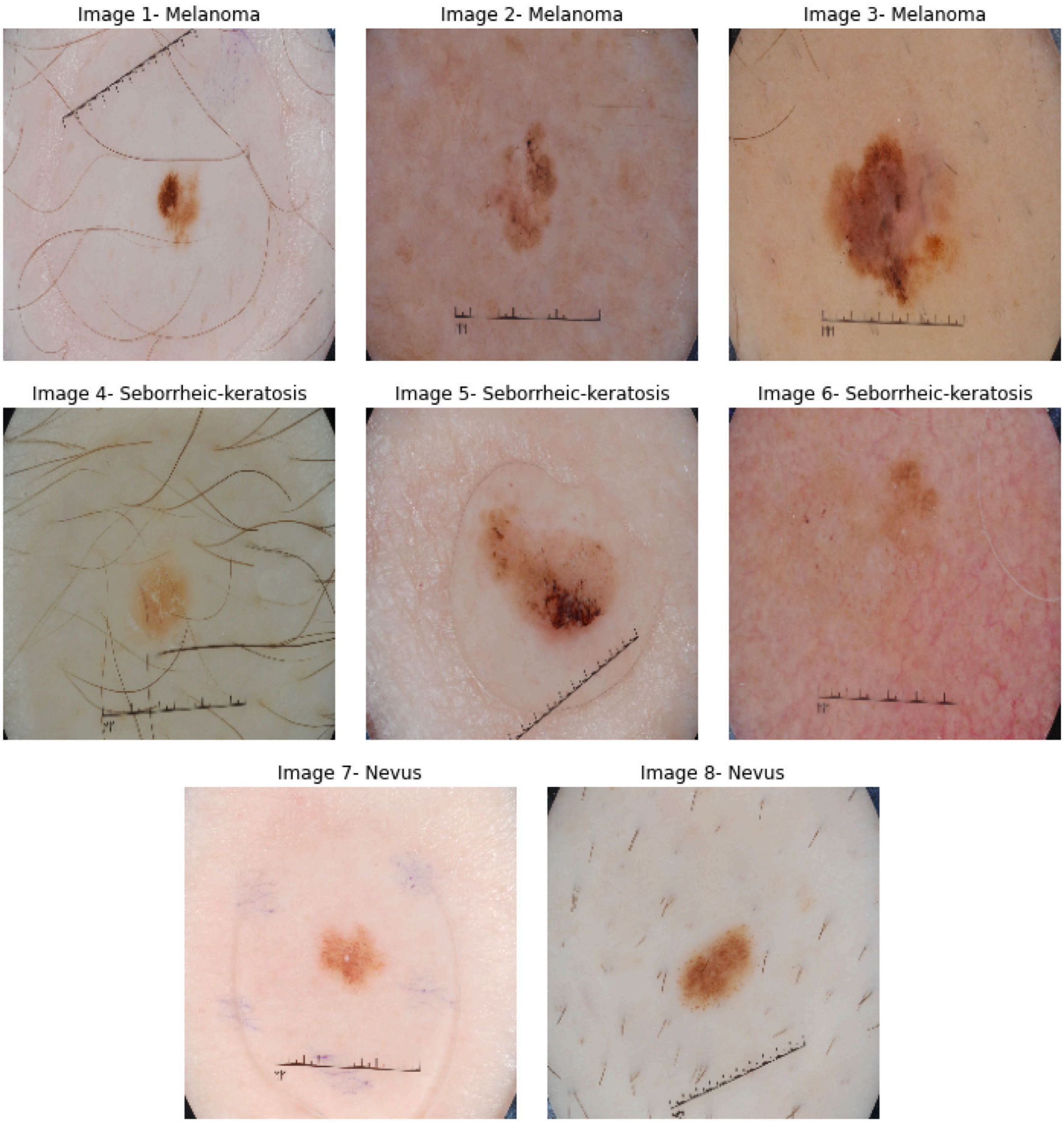
Image 1- Melanoma
Image 2- Melanoma
Image 3- Melanoma
Image 4- Seborrheic-keratosis
Image 5- Seborrheic-keratosis
Image 6- Seborrheic-keratosis
Image 7- Nevus
Image 8- Nevus

Fig. 2. Illustrations of selected test data for evaluation of EGAE.

in Eq. (10) equals zero. Nonetheless, the worst case differs image by image. Thus, the normalized error was calculated to transform the error values into [0,1] for all images using Eq. (11). The less the $normalized\_error$ in Eq. (11) is, the more accurate the explanation is. The delineations of clinicians are also illustrated using SLIC segmentation algorithm in Fig. 3.

$$error = \text{Euclidean}\quad \text{dist}(actual\_explanation - calculated\_explanation) \tag{10}$$

$$normalized\_error = \frac{error - min\_error}{max\_error - min\_error} = \frac{error}{max\_error} \tag{11}$$

5.3. Performance analysis

Fig. 4 shows the results of Consensus Voting (CV) and Majority Voting (MV) for images of Fig. 2 in 3 consecutive runs. The illustrations in Fig. 4 show that EGAE generally converges into the informative lesions of the figures in Fig. 2. The intuitive investigation in Fig. 4 also reveals that MV does not typically discard the informative lesion in all images. However, it contains more non-informative sections than CV, which affects the accuracy of the explanation accordingly. In contrast, CV tends to discard more features of the images and may discard the informative sections of the image as a side effect in the minority of cases. Fig. 4 also illustrates the reproducibility of the results in 3 consecutive runs. EGAE does not guarantee the reproducibility of the results (like LIME and many other existing model-agnostic explainers). This issue has three main reasons. First, the essence of EGAE is optimization with genetic algorithms, and evolutionary algorithms do not guarantee complete reproducibility, especially in case of EGAE, which is based on multiple GAs. Second, the inputs of EGAE are multiple segmented images, but the results (CV and MV) are based on pixels, making it harder for the results to be completely reproducible. Third, due to using multiple GAs with different levels of segmentation, unexpected situations could happen, such as having multiple global optima in one of the GAs. This will directly affect the reproducibility of CV and, subsequently, MV. Even though EGAE is not completely reproducible, illustrations in Fig. 4 show that it has a high degree of reproducibility (with either of CV or MV, depending on the image). For example, the reproducibility of MV is better than CV in images 2 and 3. As another example, it is also intuitively evident that CV is more reproducible than MV in image 1. Moreover, based on the methodology illustrated in

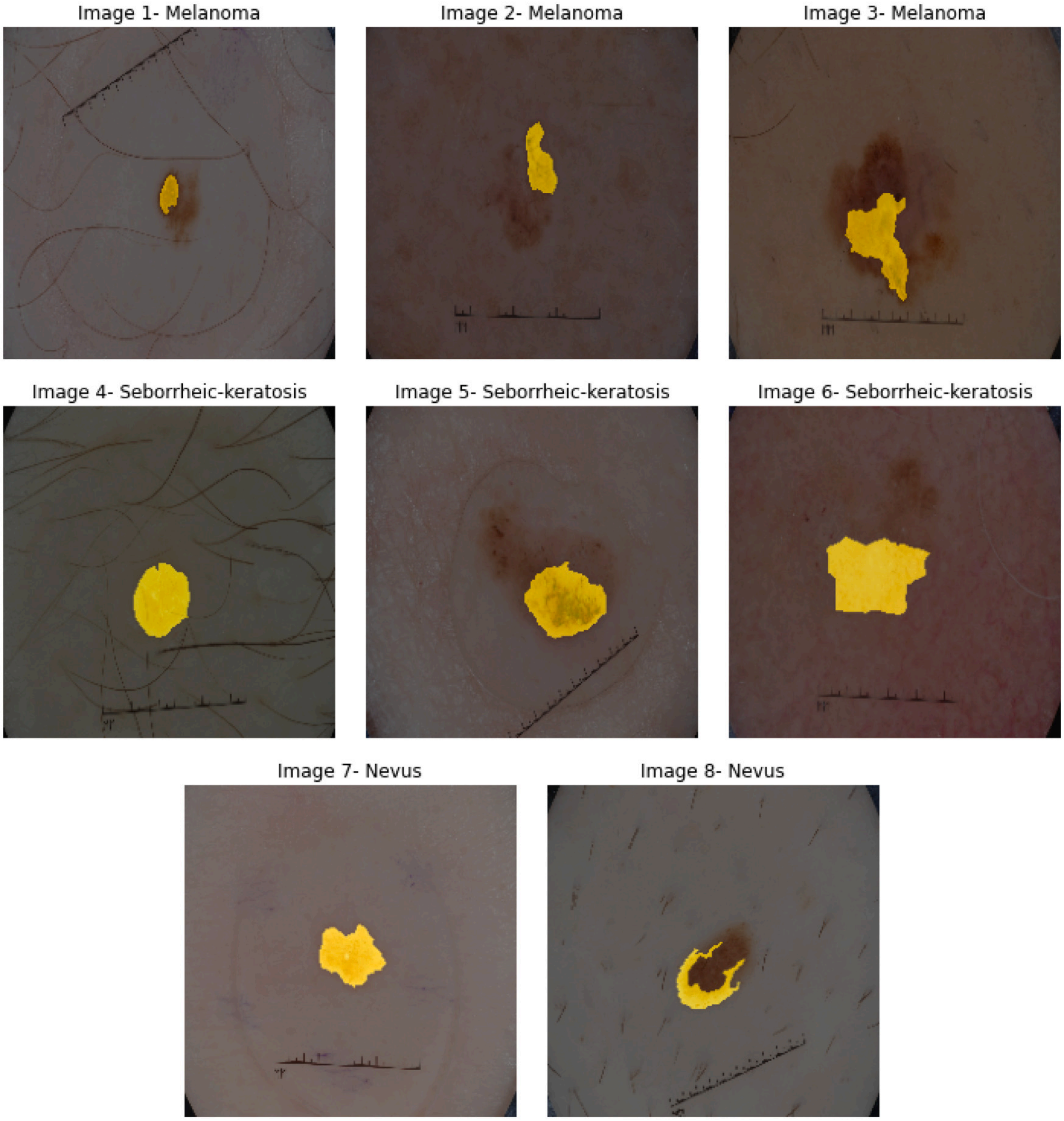

Image 1- Melanoma

Image 2- Melanoma

Image 3- Melanoma

Image 4- Seborrheic-keratosis

Image 5- Seborrheic-keratosis

Image 6- Seborrheic-keratosis

Image 7- Nevus

Image 8- Nevus

Fig. 3. Delineations of the clinician generated by SLIC segmentation algorithm.

Fig. 1, the images that corrupt consensus voting (images that cause consensus voting to be an empty image) are identified for discarding at the beginning of phase 3.

Fig. 5 shows the average number of images (in 5 consecutive runs) contributing to voting for each image in Fig. 2. This figure also shows that the least number of images contributing to voting is for image 3, with an average value of 3.8. Likewise, the average images contributing to voting for all images in Fig. 2 are 4.78, which reveals that EGAE does not aggressively discard images. Fig. 6 shows the NFE of EGAE in 5 consecutive runs and the average NFE for each image. Generally, the NFE in GA increases as the number of superpixels in the input image increases. As such, the GAs with 100 and 10 superpixels in the input image have the greatest and smallest NFE. Fig. 6 illustrates that EGAE has an average NFE of less than almost 6000 in the majority of the images. The minimum average NFE is for image 3, and the maximum average NFE is recorded for image 6. The average NFE in Fig. 6 shows that EGAE uses considerably few images for explanation. Recalling that EGAE includes 5 GAs with a search space of $2^{10} + 2^{15} + 2^{20} + 2^{25} + 2^{100}$ number of images, the average NFE recorded in Fig. 6 is indeed insignificant compared with the search space. This confirms the usage of an evolutionary algorithm in explanation.

This paper's main inspiration is to automate the explanation by eliminating the intervention of the user to determine the number of superpixels in the input image (It is now decided by experts manually with a segmentation algorithm in LIME) and the number of top features (It is now specified by experts intuitively with the $num\_features$ parameter in LIME). Although Fig. 4 shows that EGAE does not discard the informative lesion of the images, the information in Table 3 and Fig. 7 specify to what extent the explanation of Fig. 4 is close to the clinician delineation compared with LIME. Table 3 investigates the accuracy of explanation using Eq. (11). For this reason, LIME is executed five times with 10, 15, 20, 25, and 100 superpixels in the input image. The $num\_samples$ parameter in LIME library, which shows the size of the neighborhood to learn the linear model is set to 50, 1000, 2000, 2000 when the input image is segmentized into 10, 15, 20, and 25 superpixels respectively. However, the parameter $num\_samples$ is set to the average NFE calculated in Fig. 6 for LIME, when the input image is segmentized to 100 superpixels (to have a fair comparison between LIME and EGAE). Then, for each case, the group of superpixels that LIME identifies positively to contribute to the prediction are kept, and the rest of the superpixels are discarded. Next, the normalized

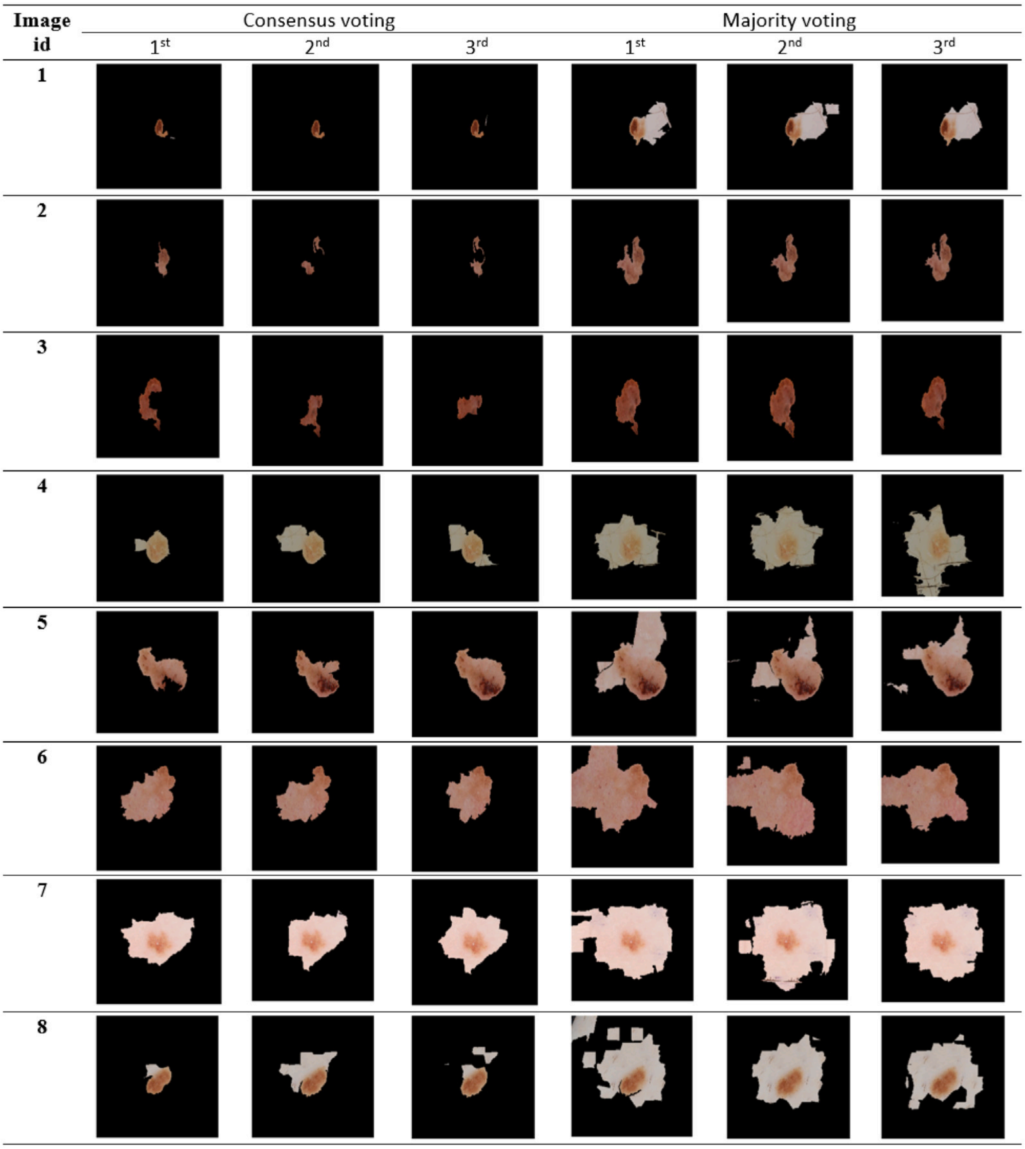

Fig. 4. The results of EGAE in three runs.

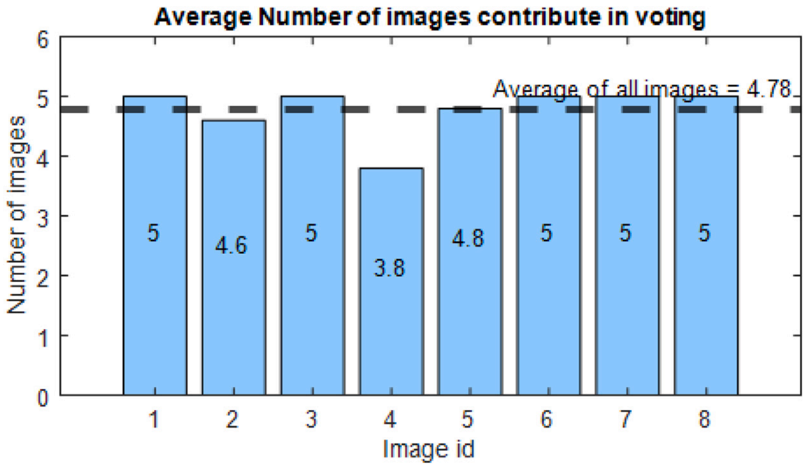

Fig. 5. Average images contribute to voting.

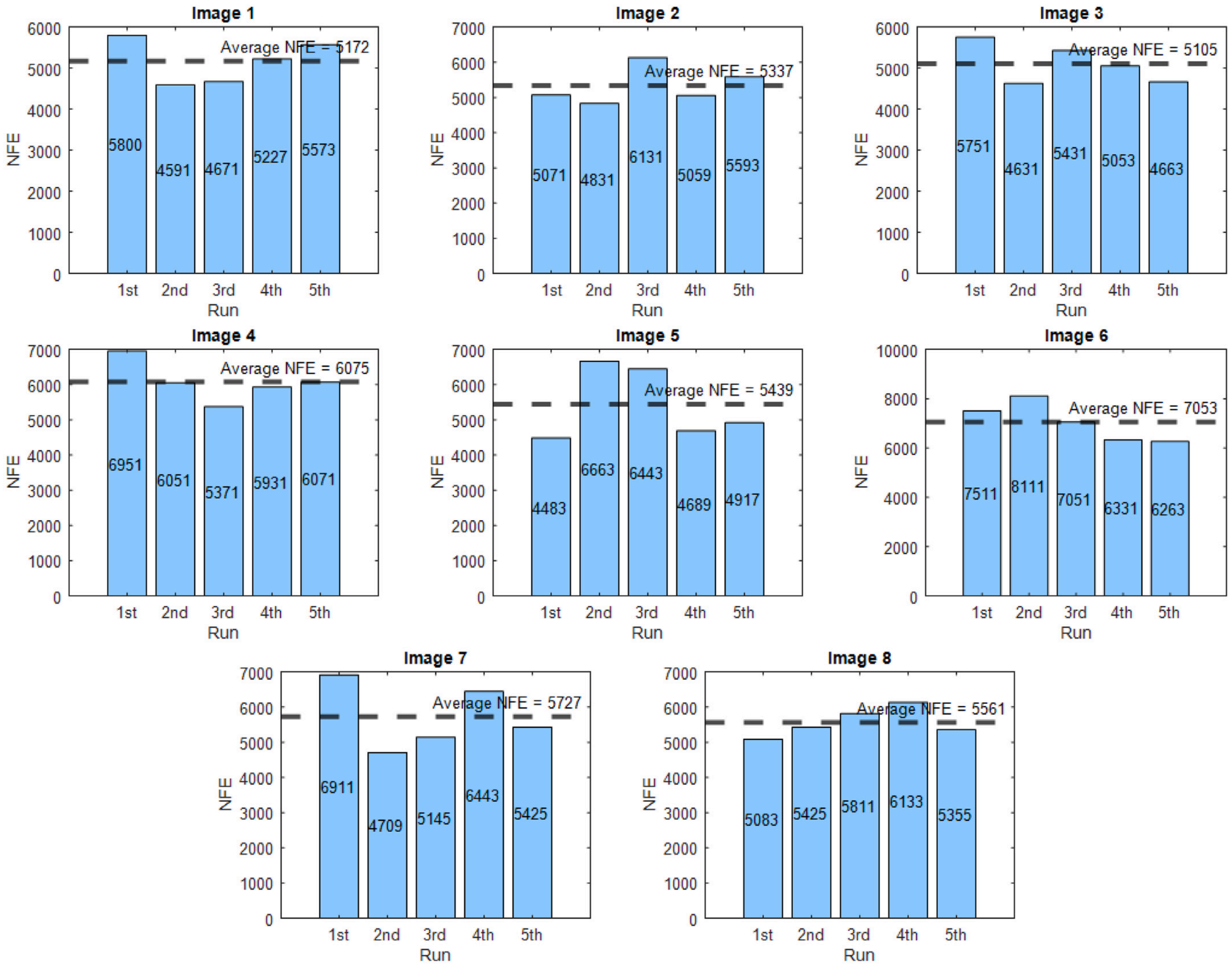

Fig. 6. The NFEs of EGAE on test data.

euclidean distance of the *actual_explanation* delineated by clinicians against *calculated_explanation* by LIME is calculated. Likewise, the same has been done for EGAE results.

Table 3 shows that both Consensus Voting (CV) and Majority Voting (MV) are very close to *actual_explanation* in comparison with LIME considering that CV is slightly better than MV. This confirms that LIME keeps too many insignificant superpixels, which EGAE discards. Fig. 7 investigates the accuracy of explanations in another way. In this case, the clinician intervenes to determine the number of segmentations in the input image. This solution gradually increases the top features (starting from 1) until all sections of the *actual_explanation* are seen. In other words, the clinician repeatedly adds to the top features (the *num_features* parameter in LIME library) until the set of top features can completely cover the delineated area specified by the clinician. Finding the best segmentations of the input image and the number of top features within that segmentation is time-consuming. However, EGAE with CV strategy slightly outperforms LIME even by manual determination of the parameters (number of superpixels in the input image and *num_features*) in images 1, 2, 3, and 6. Additionally, CV has also acceptable results for the rest of the images. Moreover, it is evident that CV has better accuracy of explanation in comparison with MV in all cases in Fig. 7, which is also clearly discussed in Table 3 and intuitively proved in Fig. 4. The results of MV are not good for images 4 and 7 but still acceptable in other images. The information in Fig. 7 shows that automatic EGAE not only has a good accuracy of explanation but also sometimes outperforms LIME even with the manual determination of parameters, particularly with CV based on Eq. (11).

Fig. 8 shows the performance graph of EGAE, which starts from the initial population at iteration 0 (the first generation that is created randomly) and continues by applying selection, cross-over, and mutation starting from generation 1 until convergence. Fig. 8 confirms the maximum number of 150 iterations is enough for convergence, as the number of iterations rarely increases 100 for an input image with 100 segmentations. Likewise, as the number of superpixels in the input image increases from 10 to 100, fitness improvements increase accordingly. Fig. 8 also shows how fitness continuously and smoothly increases through iterations, particularly when the number of superpixels equals 100. This, in turn, implicitly confirms the use of optimization and GA in the explanation.

Fig. 9 clearly shows the effect of determining the sparsity of chromosomes (φ) in the first phase of EGAE. As a particular case, Fig. 9 illustrates the comparison of $\varphi = 0.9$ against $\varphi = 0.5$ (random assignment of 0 and 1 to genes) while the number of superpixels in the input image equals 100. The idea is that when the number of segmentations in the input image is large (100, for example) and $\varphi = 0.5$, half of the superpixels in the input image would be active. The problem is that these active superpixels may not cover the informative area of the image. Thus, the initial population may contain trivial solutions

Table 3
Normalized explanation error of LIME in 5 cases compared with automatic EGAE.

Image id	LIME 100	LIME 25	LIME 20	LIME 15	LIME 10	EGAE with Consensus Voting	EGAE with Majority Voting
1	0.77	0.74	0.74	0.68	0.60	**0.06**	0.25
2	0.75	0.80	0.84	0.76	0.82	**0.10**	0.16
3	0.67	0.64	0.51	0.66	0.48	**0.12**	0.15
4	0.75	0.80	0.75	0.77	0.79	**0.21**	0.42
5	0.71	0.67	0.76	0.74	0.68	**0.20**	0.41
6	0.79	0.74	0.70	0.68	0.78	**0.33**	0.46
7	0.71	0.71	0.71	0.79	0.87	**0.32**	0.57
8	0.70	0.68	0.74	0.74	0.80	**0.18**	0.48

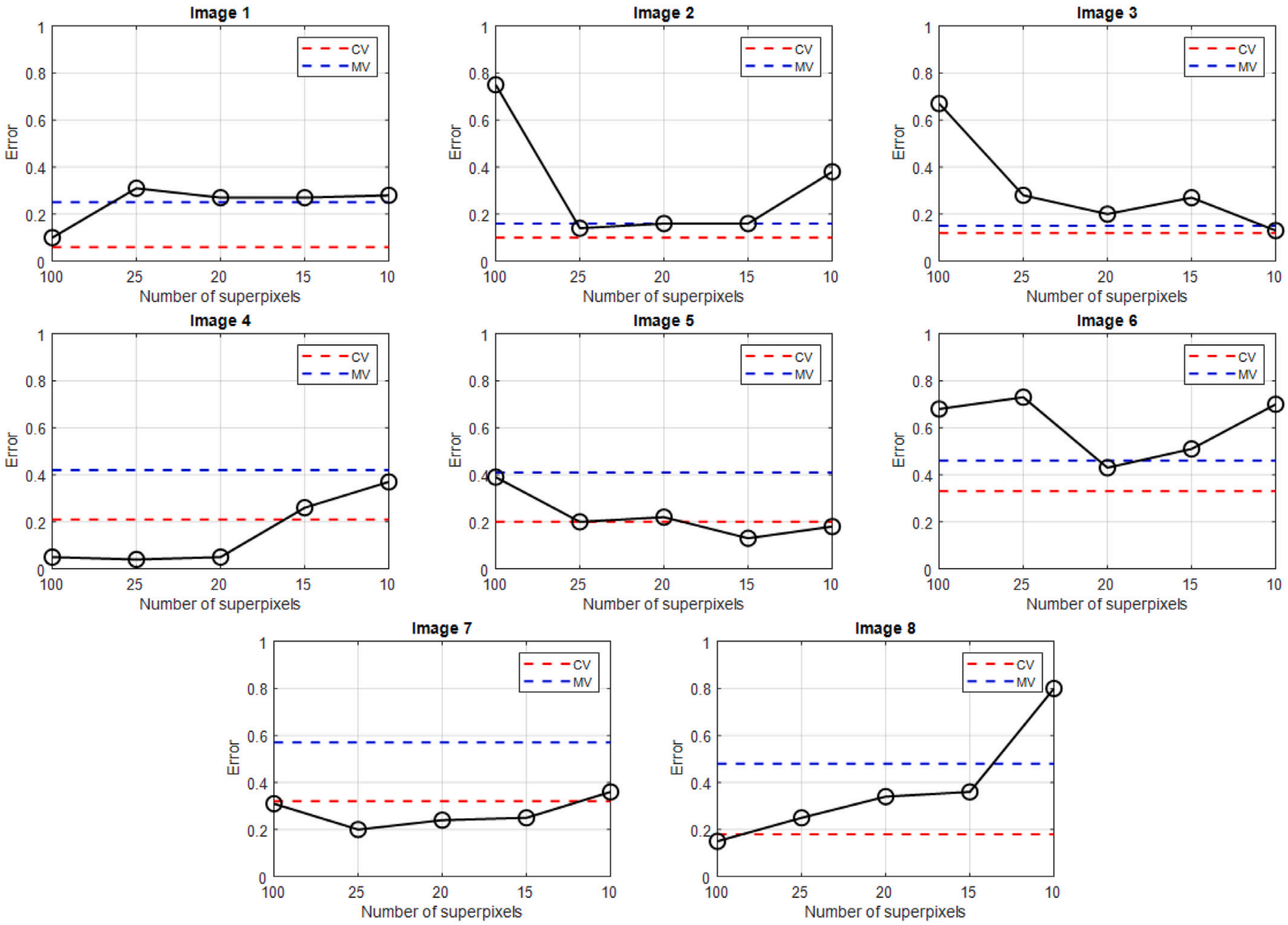

Fig. 7. Normalized explanation error of LIME (manual investigation to select the best number of superpixels using SLIC segmentation algorithm for the input image and the number of top features to be seen for that number of superpixels) compared with the automatic EGAE.

that make convergence difficult for GA. Fig. 9 shows the convergence guarantee with $\varphi = 0.9$ in all images despite the increase in NFE of images 1, 2, and 3. In contrast, setting $\varphi = 0.5$ may lead to premature convergence with a solution entirely far from the optimal solution, as in images 4, 5, and 6. Images 7 and 8 in Fig. 9 also show the cases in which GA could escape from premature convergence via its operators with $\varphi = 0.5$. All in all, Fig. 9 shows that in the case of having a large number of segmentations in the input image (100, for example), $\varphi = 0.9$ guarantees the convergence but may increase the NFE as a side effect which is not considerable at all.

5.4. Discussions

The discussion section outlines three main characteristics of EGAE. First, the advantages of EGAE over LIME are itemized. Second, the difference between consensus voting and majority voting is clarified. Finally, the justification for using GA is stated.

5.4.1. EGAE vs. LIME

Generally, EGAE outperforms LIME in the following criteria:

1. **Automation:** EGAE has an automatic image segmentation. Meanwhile, LIME needs the intervention of the user to manually set the number of superpixels in the input image through a segmentation algorithm and the appropriate number of top features ($num\_features$) to be seen. This is useful because it is not always straightforward for the expert to specify the number of superpixels in the input image.
2. **Accuracy of explanation:** EGAE generally achieves greater accuracy in explaining the prediction class (as experimented with

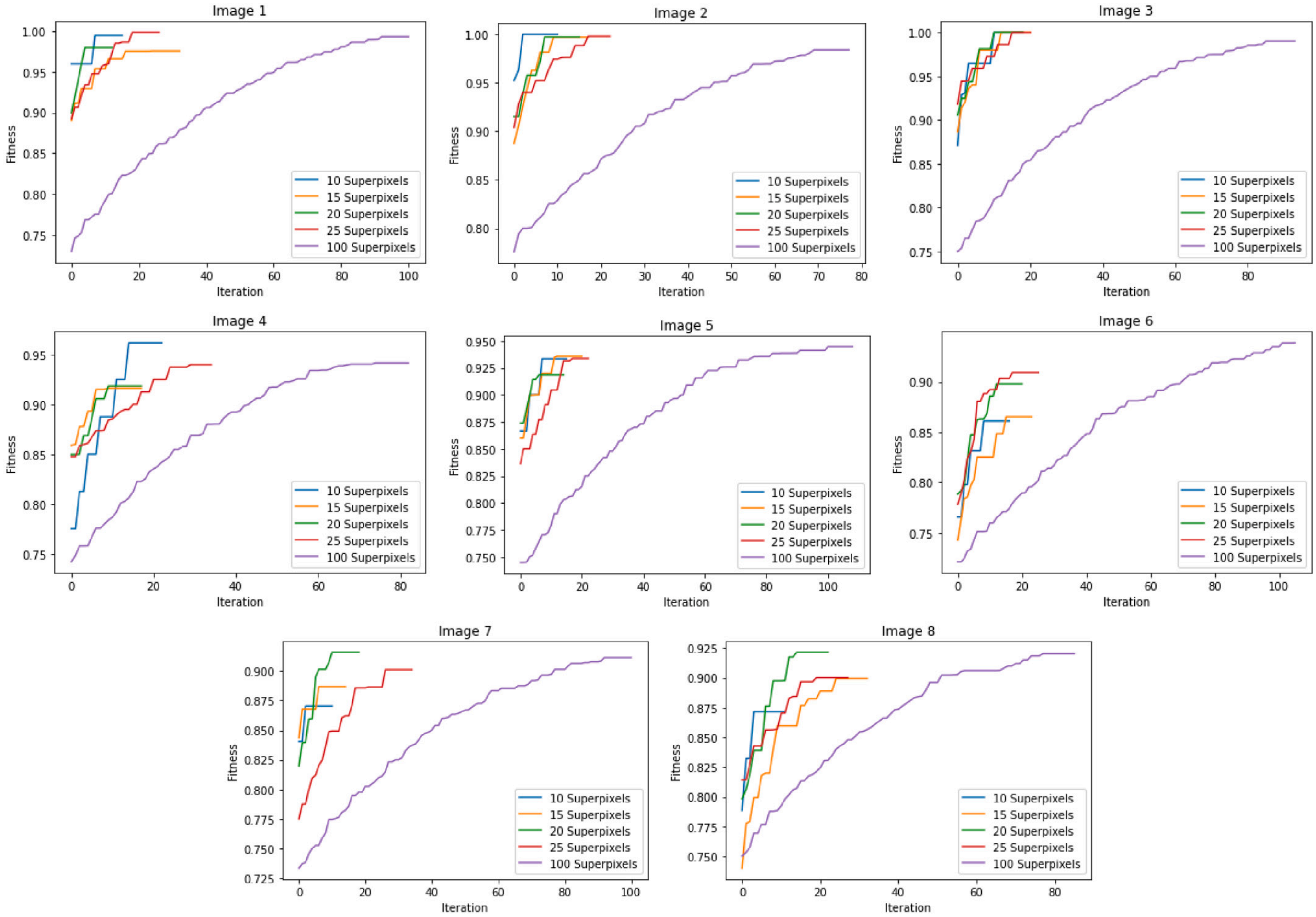

Fig. 8. Performance graph of EGAE.

melanoma detection dataset) because of ensembling multiple GAs (each with a different number of segmentations in the input image) and the embedded voting strategies. EGAE achieves acceptable accuracies using considerably fewer images (in comparison with the search space), even when the search space is huge.

3. **Fewer hyperparameters to be determined:** EGAE does not use a linear surrogate model as LIME does. Therefore, LIME needs to determine more parameters in advance. Besides the number of superpixels in the input image and top features (discussed in detail in previous sections how EGAE automatically explains without needing them) distance metric, $num\_samples$, $model\_regressor$, and $feature\_selection$ are among the most important parameters that EGAE is independent of them. LIME uses distance metric for calculating the weights of images in the vicinity of the image that needs explanation. There are well-known distance metrics to be used and LIME, by default, exploits the cosine distance metric. EGAE does not need to calculate the distance metric since it directly emulates the prediction model via multiple consecutive GAs. The $num\_samples$ parameter shows the neighborhood size to learn the linear model in LIME which the user should specify in advance. In other words, $num\_samples$ specifies the number of images LIME needs for evaluation. The greater $num\_samples$ yields both a more accurate explanation and better reproducibility. However, increasing $num\_samples$ conflicts with execution time. The user must compromise between $num\_samples$ and execution time to achieve an accurate explanation within a reasonable time. EGAE implicitly calculates $num\_samples$ using the Number of Function Evaluations (NFE) in GAs during execution. $Model\_regressor$ is a regressor that is used in the explanation provided by LIME. However, EGAE is entirely independent of $model\_regressor$ as it uses GA for optimization. $Feature\_selection$ specifies the methodology to select the number of features required for explanation that the user should specify in advance for LIME. In contrast, EGAE defines the best features using its voting strategies. $Segmentation\_fn$ is the segmentation algorithm used to divide the input image into superpixels. There are well-known segmentation algorithms, including quick shift, SLIC, and fenzelswalb. Both EGAE and LIME need to determine the segmentation algorithm in advance. Unfortunately, even optimizing these parameters (which may result in higher computational cost), inconsistencies can still occur in LIME. Although the expert can also investigate the appropriate number of GAs in EGAE and the segmentation function of the input image, experiments reveal that the existing setting works well.

It was mentioned that between execution time and the number of images evaluated (accessible via NFE in Eq. (9)), the NFE is a fairer criterion. However, the existing LIME image explainer is almost ten times faster than EGAE, assuming the execution time as a measurement criterion. For example, the computational times of EGAE and LIME for image 1 are 1130 and 110 seconds, respectively, using the same number of images (We assigned the average NFE of image 1 in Fig. 6 (5172) to the $num\_samples$ parameter of LIME image explainer with the input image that is segmented into 100 superpixels) recalling that the more superpixels the input image has, the more time EGAE spends for the respective GA. This also holds for the rest of the images. Despite higher execution time, EGAE offers an explanation with automatic segmentation of input image, which is worthy.

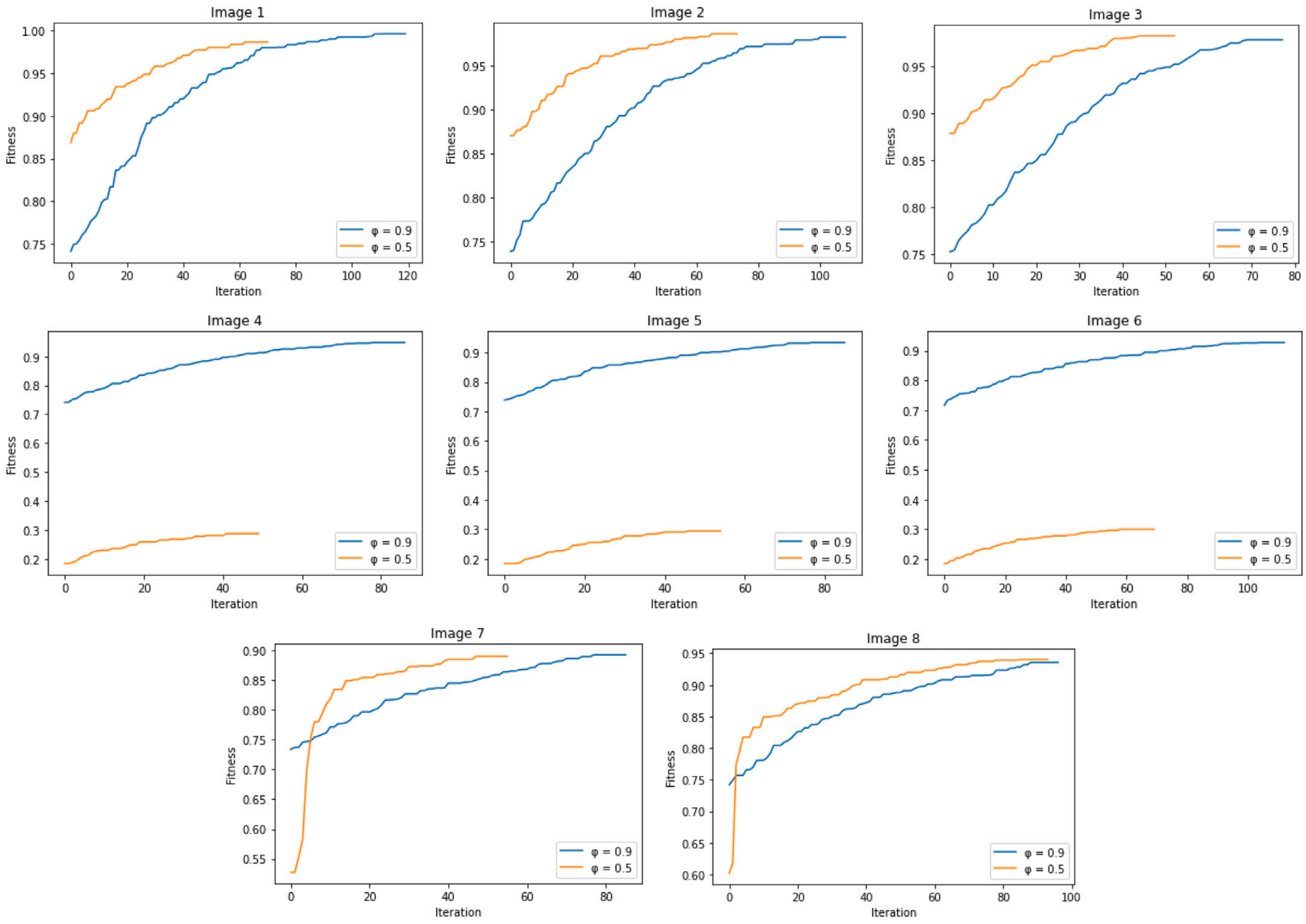

Fig. 9. The effect of determination of chromosomes' sparsity.

5.4.2. *Consensus voting vs. majority voting*

It is intuitively shown in Fig. 4 and numerically discussed in Table 3 and Fig. 7 that Consensus Voting (CS) generally has better accuracy of calculated explanation than Majority Voting (MV). However, CV can be compromised by interpretability and reproducibility in minority of cases. First, the interpretability of CV can be compromised when it contains only few pixels. The concept of superpixels will guarantee the interpretability of the results. It is difficult for the user to interpret CV with few pixels. In such cases, MV with more pixels could keep EGAE to remain interpretable. Second, the reproducibility of the results can be compromised due to getting stuck in local optima by GA initially. This happens because even by controlling parameters, it is inevitable for evolutionary algorithms to get stuck in local optima. Likewise, some anomalies can occur and affect voting. One anomaly scenario is that segmentizing an image into a certain number of superpixels may generate many global optima or ridge. Ridge refers to local optima close to each other. It is difficult for evolutionary algorithms to escape from the ridge. Additionally, GA could also converge to different global optima in each run in case of having multiple global optima. Furthermore, such anomalies do not necessarily happen with larger superpixels, which shows the inevitability of the problem. In both cases, the reproducibility of CV and MV could be affected. Thus, it cannot conclude which of the voting approaches has better reproducibility. For example, Fig. 4 shows better reproducibility of MV in images 2 and 3, recalling that there is no guarantee to have 100% reproducibility for existing explainers as well. Even in LIME, increasing the $num\_samples$ (which increases the time accordingly) increases the degree of reproducibility, but there is no guarantee of reaching complete reproducibility. MV helps EGAE to remain both interpretable and with better reproducibility in case of an anomaly. Meanwhile, the accuracy of explanation in CV generally outperforms MV.

5.4.3. *Why single-objective GA works in EGAE?*

This section justifies the usage of GA with its single-objective function for automatic explanation initially. Then, the process of explanation in EGAE is shown using an illustrative example. The evolutionary algorithms can be divided into two major groups: those appropriate for discrete problems and those suitable for solving continuous problems. Almost the majority of evolutionary algorithms are intrinsically continuous. However, they can be converted to their respective discrete versions (Particle Swarm Optimization [30], Whale Optimization Algorithm [31], Gravitational Search Algorithm [32], Harmony Search Algorithm [33], etc.). In contrast, GA belongs to the category of intrinsically discrete solutions. The problem of image-based automatic model-agnostic explanation can be formulated as a discrete problem so that each solution can be encoded as a binary string. Thus, GA is accordingly selected for optimization as a well-known and intrinsically discrete algorithm. The fitness function of EGAE is defined based on the accuracy of prediction and the number of active superpixels in a solution (image), as discussed in Eq. (3) of Section 4.1. Furthermore, there is neither a conflict nor a direct relationship between prediction accuracy and number of active superpixels. In other words, the accuracy of prediction does not always grow when the number of superpixels increases and vice versa. This also confirms that the problem can be formulated as a single-objective optimization with a defined fitness function.

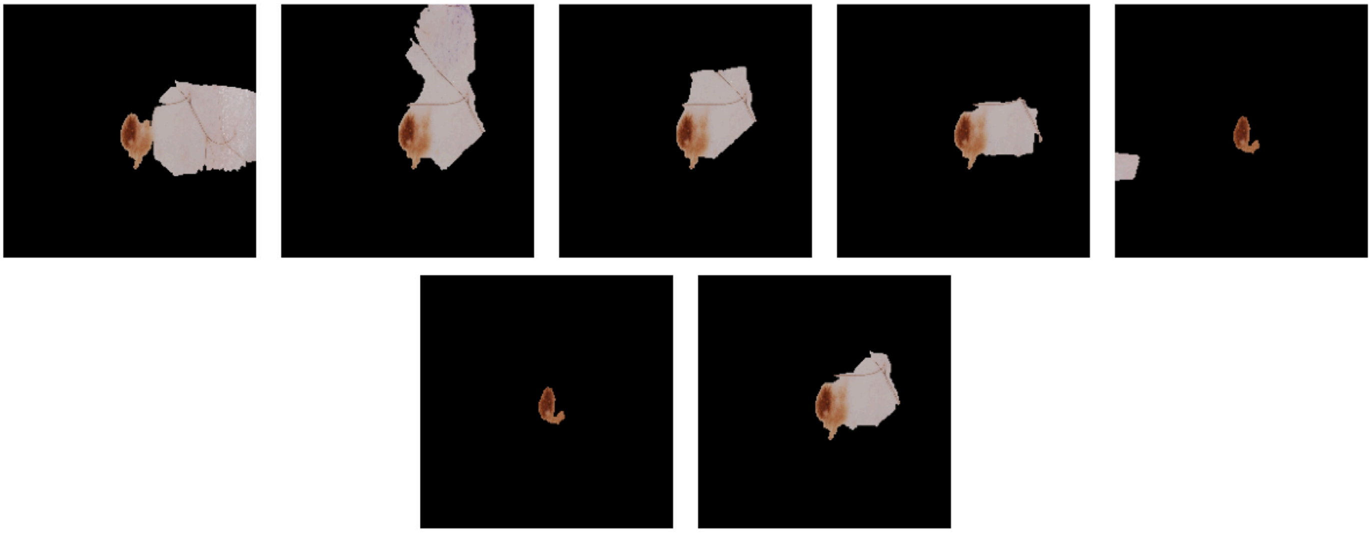

Fig. 10. Mechanism of EGAE for image 1 in a sample run. First row from left to right: The explanations of GAs on image 1 segmentized into 10, 15, 20, 25, and 100 superpixels. Second row from left to right: The explanations of Consensus Voting and Majority Voting.

GAs work by discovering and combining good building blocks (schema) of the initial population in each iteration. GAs iteratively identify the good building blocks solely by the fitness function and tend to improve the schema if possible. As such, GAs guarantee optimization. Fig. 10 shows the process of EGAE intuitively for image 1 from Fig. 2. EGAE (with 10, 15, 20, and 25 superpixels) converged into the global optimum in Fig. 10. However, EGAE converged to a local optimum, close to the global optimum, with 100 superpixels. The global optimum is an image that only contains the lesion based on the fitness function. Finding the global optimum is difficult even for evolutionary algorithms when the search space is massive. EGAE can sometimes overcome this problem through its voting strategies. The consensus voting in Fig. 10 discards the irrelevant superpixel. This is obviously another advantage of EGAE in case of getting stuck in a local optimum in one of the GAs (recalling that the investigation of phase 3 in EGAE does not recognize any image that corrupts consensus voting, as shown in Fig. 5).

6. Conclusion

This paper proposes an automatic Ensemble-based Genetic Algorithms Explainer (EGAE), which attempts to improve the existing LIME image explainer by eliminating the user's interventions in determining the number of superpixels in the input image as well as the top features for the automatic explanation. EGAE has three phases, so that the sparsity of chromosomes is initially calculated using a heuristic algorithm. Second, multiple GAs are executed consecutively. Thus, in each GA, images that constitute the initial population have a distinguished number of superpixels compared to other GAs. The result of each GA is an image that explains the prediction. Finally, the images from GAs are ensembled using consensus and majority voting to construct two images to show simultaneously to the user for an explanation. EGAE has been tested on melanoma detection dataset, and EGAE has three advantages over LIME:

1. First, EGAE is automatic and eliminates the user's intervention in determining two parameters. These parameters are the number of superpixels in the input image (it is now specified by the expert manually through a segmentation algorithm in LIME) and the number of top features (it is now specified by the expert intuitively with the $num\_features$ parameter in LIME).
2. Second, EGAE generally achieves greater accuracy of explanation than LIME while using very few images for explanation compared with the search space. The reason is that EGAE discards more non-informative sections of the image while concentrating on the informative parts.
3. Third, EGAE is not surrogate model dependent and thus needs fewer hyperparameters to be tuned in advance.

In general, EGAE tries to emulate the classifier to unveil the black box architecture and is close to the users' decision-making processes (in this case, clinicians' point of view for the melanoma detection dataset). This can be investigated through a new evaluation metric to calculate the explanation accuracy using the Euclidean distance of actual explanation delineated by clinicians from the calculated explanation by the explainer. Further investigation on the performance graph of EGAE confirms the use of GA as an optimization technique for the automatic explanation. In addition, the effect of determining the sparsity of chromosomes in proper convergence of GAs is also discussed. However, the main limitation of EGAE is that even though it has an acceptable level of reproducibility in some cases, it is a non-reproducible explainer like most existing explainers, including LIME and SHAP explainers. Future research will look into testing EGAE on other datasets, such as plant disease datasets. In addition, it can be investigated how intelligent segmentation algorithms (such as those introduced in Section 2.2) could enhance the explanation.

Declaration of competing interest

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

Acknowledgments

This work has been partially funded by grant PID2020-112540RB-C41 (funded by MCIN/AEI/10.13039/501100011033/, Spain), AETHER-UMA, Spain (A smart data holistic approach for context-aware data analytics: semantics and context exploitation). Funding for open access charge: Universidad de Málaga/CBUA. Additionally, we thank Dr. Miguel Ángel Berciano Guerrero from Unidad de Oncología Intercentros, Hospitales Univesitarios Regional *y* Virgen de la Victoria de Málaga, and Instituto de Investigaciones Biomédicas (IBIMA), Málaga, Spain, for his support in images selection and general medical orientation in the particular case of Melanoma.

References

[1] A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.

[2] J. Meena, Y. Hasija, Application of explainable artificial intelligence in the identification of squamous cell carcinoma biomarkers, Comput. Biol. Med. 146 (2022) 1–12.

[3] A. Holzinger, R. Goebel, R. Fong, T. Moon, K.-R. Müller, W. Samek, xxAI - Beyond Explainable AI, Springer Nature, 2022.

[4] G. Yang, Q. Ye, J. Xia, Unbox the black-box for the medical explainable AI via multi-modal and multi-centre data fusion: A mini-review, two showcases and beyond, Inf. Fusion 77 (2022) 29–52.

[5] L.A. de Souza Jr., R. Mendel, S. Strasser, A. Ebigbo, A. Probst, H. Messmann, J. P. Papa, C. Palm, Convolutional neural networks for the evaluation of cancer in Barrett's esophagus: Explainable AI to lighten up the black-box, Comput. Biol. Med. 135 (2021) 1–14.

[6] G. Ahmed Tahir, C. Kiong Loo, Explainable deep learning ensemble for food image analysis on edge devices, Comput. Biol. Med. 139 (2021) 1–18.

[7] S. Hurtado, H. Nematzadeh, J. Garcıa-Nieto, M.-A. Berciano-Guerrero, I. Navas-Delgado, On the use of explainable artificial intelligence for the differential diagnosis of pigmented skin lesions, in: Springer (Ed.), Bioinformatics and Biomedical Engineering: 9th International Work-Conference, IWBBIO 2022, 2022, pp. 319–329.

[8] W. Hryniewska, A. Grudzień, P. Biecek, LIMEcraft: Handcrafted superpixel selection and inspection for visual explanations, Mach. Learn. (2022).

[9] W. Abbes, D. Sellami, Deep neural networks for melanoma detection from optical standard images using transfer learning, Procedia Comput. Sci. 192 (2021) 1304–1312.

[10] The American cancer society medical and editorial content team, melanoma skin cancer early detection, diagnosis, and staging, 2022, https://www.cancer.org/content/dam/CRC/PDF/Public/8825.00.pdf. (Accessed 21 December 2022).

[11] P. Bansal, R. Garg, P. Soni, Detection of melanoma in dermoscopic images by integrating features extracted using handcrafted and deep learning models, Comput. Ind. Eng. 168 (2022) 1–15.

[12] J. Jaworek-Korjakowska, A. Brodzicki, B. Cassidy, C. Kendrick, M. Hoon Yap, Interpretability of a deep learning based approach for the classification of skin lesions into main anatomic body sites, Cancers 13 (2021) 1–14.

[13] F. Stieler, F. Rabe, B. Bauer, Towards domain-specific explainable AI: Model interpretation of a skin image classifier using a human approach, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW.

[14] M. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision model-agnostic explanations, in: IEEE (Ed.), Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[15] T. Peltola, Local interpretable model-agnostic explanations of Bayesian predictive models via Kullback-Leibler projections, in: Workshop on Explainable Artificial Intelligence - Stockholm, Sweden, 2018.

[16] I. Ahern, A. Noack, L. Guzman-Nateras, D. Dou, B. Li, J. Huan, NormLime: A new feature importance metric for explaining deep neural networks, 2019.

[17] J. Rabold, H. Deininger, M. Siebers, U. Schmid, Enriching visual with verbal explanations for relational concepts – Combining LIME with Aleph, in: P. Cellier, K. Driessens (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, 2020, pp. 180–192.

[18] S. Shi, X. Zhang, W. Fan, A modified perturbed sampling method for local interpretable model-agnostic explanation, 2020, CoRR abs/2002.07434.

[19] L. Deng, The MNIST database of handwritten digit images for machine learning research [best of the web], IEEE Signal Process. Mag. 29 (6) (2012) 141–142.

[20] B. He, W. Hu, K. Zhang, S. Yuan, X. Han, C. Su, J. Zhao, G. Wang, G. Wang, L. Zhang, Image segmentation algorithm of lung cancer based on neural network model, Expert Syst. 39 (3) (2022).

[21] A. Qi, D. Zhao, F. Yu, A. Heidari, Z. Wu, Z. Cai, F. Alenezi, R. Mansour, H. Chen, M. Chen, Directional mutation and crossover boosted ant colony optimization with application to COVID-19 X-ray image segmentation, Comput. Biol. Med. 148 (2022) 1–19.

[22] W. Wang, T. Tu, F. Bergholm, Improved minimum spanning tree based image segmentation with guided matting, KSII Trans. Internet Inf. Syst. 16 (1).

[23] H. Su, D. Zhao, H. Elmannai, A. Heidari, S. Bourouis, Z. Wu, Z. Cai, W. Gui, M. Chen, Multilevel threshold image segmentation for COVID-19 chest radiography: A framework using horizontal and vertical multiverse optimization, Comput. Biol. Med. 146 (2022) 1–33.

[24] S. Katoch, S. Chauhan, V. Kumar, A review on genetic algorithm: Past, present, and future, Multimedia Tools Appl. 80 (2021) 8091–8126.

[25] S. Sharma, V. Jain, Solving N-queen problem by genetic algorithm using novel mutation operator, IOP Conf. Ser.: Mater. Sci. Eng. 1116 (1) (2021) 1–6.

[26] H. Nematzadeh, R. Enayatifar, H. Motameni, F. Gadelha Guimarães, V. Nazário Coelho, Medical image encryption using a hybrid model of modified genetic algorithm and coupled map lattices, Opt. Lasers Eng. 110 (2018) 24–32.

[27] J. García, C. Acosta, M. Mesa, Genetic algorithms for mathematical optimization, J. Phys.: Conf. Ser. 1448 (1) (2020) 1–5.

[28] M. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[29] I. Palatnik de Sousa, M. Maria Bernardes Rebuzzi Vellasco, E. Costa da Silva, Local interpretable model-agnostic explanations for classification of lymph node metastases, Sensors 19 (2019) 1–18.

[30] H. Zhang, M. Yuan, Y. Liang, Q. Liao, A novel particle swarm optimization based on prey–predator relationship, Appl. Soft Comput. 68 (2018) 202–218.

[31] S. Chakraborty, A. Saha, S. Nama, S. Debnath, A novel particle swarm optimization based on prey–predator relationship, Comput. Biol. Med. 139 (2021) 1–24.

[32] J. Jiang, R. Jiang, X. Meng, K. Li, SCGSA: A sine chaotic gravitational search algorithm for continuous optimization problems, Expert Syst. Appl. 144 (2020) 1–18.

[33] A. Moayedikia, K. Ong, Y. Boo, W. Yeo, R. Jensen, Feature selection for high dimensional imbalanced class data using harmony search, Eng. Appl. Artif. Intell. 57 (2017) 38–49.