Ensemble of deep learning models with surrogate-based optimization for medical image segmentation.

DANG, T., LUONG, A.V., LIEW, A.W.C., MCCALL, J. and NGUYEN, T.T.

2022

Ensemble of deep learning models with surrogate-based optimization for medical image segmentation

Truong Dang[1], Anh Vu Luong[2], Alan Wee Chung Liew[3], John McCall[4], Tien Thanh Nguyen[5]
[145]School of Computing, Robert Gordon University, Aberdeen, UK
[23]School of Information and Communication Technology, Griffith University, Queensland, Australia

***Abstract*—Deep Neural Networks (DNNs) have created a breakthrough in medical image analysis in recent years. Because clinical applications of automated medical analysis are required to be reliable, robust and accurate, it is necessary to devise effective DNNs based models for medical applications. In this paper, we propose an ensemble framework of DNNs for the problem of medical image segmentation with a note that combining multiple models can obtain better results compared to each constituent one. We introduce an effective combining strategy for individual segmentation models based on swarm intelligence, which is a family of optimization algorithms inspired by biological processes. The problem of expensive computational time of the optimizer during the objective function evaluation is relieved by using a surrogate-based method. We train a surrogate on the objective function information of some populations and then use it to predict the objective values of each candidate in the subsequent populations. Experiments run on a number of public datasets indicate that our framework achieves competitive results within reasonable computation time.**

***Index Terms*—image segmentation, deep learning, ensemble learning, particle swarm optimization, surrogate models, surrogate-assisted evolutionary algorithms**

I. Introduction

Over the last decades, medical imaging technologies such as computed tomography (CT), X-ray and magnetic resonance (MR) have become important in the prevention and treatment of diseases [29]. These techniques provide non-invasive yet much more powerful means to investigate the human body compared to traditional medical imaging. This has led to a massive amount of medical data being collected and stored in databases. Manual processing of such a large amount of medical data is burdensome. Artificial intelligence has great potential in automatic processing of medical data.

It is known that image segmentation could provide quantitative and qualitative evidence for early diagnosis [30] and is therefore considered very important. However, manual segmentation by experts is time-consuming, error prone and subject to variability between different clinicians. In recent years, deep learning methods have achieved state-of-the-art results on many medical image segmentation benchmarks. Most of the state-of-the-art segmentation architectures are inspired by Fully Convolutional Network (FCN) [13], which consists of a conventional Convolutional Neural Network (CNN) architecture followed by a number of upsampling layers, which increase the feature map resolution to that of the original image. Even though deep learning models have achieved remarkable results on medical image analysis [7], the performance of these models varies due to a number of reasons, such as weight initialization and hyperparameters [14]. In order to improve the results of deep learning models on medical datasets, a simple and effective approach is to combine the results from multiple models, known in the literature as ensemble learning. By combining different models, each having different predictions, the ensemble will be able to outperform each constituent model.

Creating an effective combining method for an ensemble is an important stage when designing an ensemble system. The combining method can be obtained by using an optimization method [1]. Evolutionary Computation (EC) which is a family of optimization algorithms based on natural evolution, for example, has been widely used for ensemble optimization [16]. EC starts with a population, and at each generation/iteration, a fitness value for each individual is calculated from which to create the next generation/iteration. The fitness value helps the population to converge towards the optimal solution. Despite many advantages of EC over traditional optimization algorithms such as its capability to solve non-differentiable, discontinuous or multi-modal problems which are common in real-life applications [1], applying EC to deep learning is a challenge caused by the expensive computational complexity. If the fitness evaluation time t is high then the optimization process will take a lot of time. For example, [17] took 17 days to evolve a three-layer CNN on the CIFAR-10 dataset, due to the time-consuming evaluation process for each candidate CNN architecture. An approach we can use to reduce the computational time is Surrogate-Assisted Evolutionary Algorithm (SAEA). This approach uses a surrogate model $G(.)$ as an approximation for the fitness function $F(.)$ to reduce the computation time.

In this paper, we propose a novel ensemble and SAEA method for the problem of medical image segmentation. A number of deep learning-based segmentation models output the predictions of the image, and the outputs are sent to a combiner. The combiner used in this paper is based on Decision Template [18], however instead of using the pre-specified method used in the original paper, here we seek to optimize the Decision Template using Particle Swarm Optimization (PSO), a swarm intelligence-based optimization method. Since the objective function evaluation for each candidate takes a long time to run, we propose to use a surrogate model to reduce the computation time. The surrogate model trained from previous objective function evaluation results is used to predict the objective value of each candidate in subsequent iterations. Our contributions are as follows:

- We propose an ensemble of deep learning architectures for the medical image segmentation problem
- We propose to use SAEA to search for the optimal Decision Template as a combining model within reasonable computation time. Particle Swarm Optimization (PSO) is used as the optimization method.
- Experiments conducted on several medical image segmentation datasets demonstrate the effectiveness of our proposed method.

The paper is organized as follows. In Section 2, a brief review of the related works is provided. Our proposed ensemble is introduced in Section 3. The details of experimental studies on several medical segmentation datasets are described in Section 4. Finally, the conclusion is given in Section 5.

II. Background and Related Work

A. *Deep learning for medical image segmentation*

Since 2012 when deep learning first achieved state-of-the-arts results on ImageNet [6], there have been many works applying deep learning for medical image segmentation. UNet [7] is one of the most popular medical segmentation architectures. It has an U-shaped structure, consisting of a symmetric contracting path and an expanding path. Skip connections are used to connect upsampling results with corresponding features in the contracting path, allowing UNet to combine high-level and low-level information to improve accuracy. In recent years, there have been many works that seek to improve further the results of medical image segmentation using deep learning. An example is LinkNet [8] which uses residual modules (res-block) in place of UNet and performs summation between the upsampling results and the contracting path. Feature Pyramid Network (FPN) [9] provides a top-down pathway to construct higher resolution layers from a semantic rich layer, and lateral connections are added to improve results. [12] proposed V-Net, an extension of UNet to 3D medical datasets. [10] introduced SegNet, in which the upsampling path uses pooling indices from the contracting path in order to improve segmentation results. Another notable work is [21] in which the authors proposed attention UNet for pancreas segmentation, achieving 2-3% higher Dice scores compared to other methods. [19] applied FCN for optic disc and cupped area segmentation in fundus images for glaucoma diagnosis. [20] noted that for MRI spleen segmentation, many false positive and false negative labeling are caused by the shape and size of the spleen. The authors proposed a network called SSNet, integrating a variant of Generative Adversarial Network (GAN) to create synthetic spleen labels to improve predictions.

B. *Ensemble Learning*

Ensemble learning is a popular approach in machine learning in which a collection of base models is combined for the collaborative decision. In recent years, there has been great interest in ensemble of deep learning models. [31] used an ensemble of 2D and 3D segmentation models with a meta-learner for 3D cardiac MRI segmentation. In [32], the authors used a number of CNN models to extract the histology image features at different scales, then the optimal subset of CNN models was selected to create the ensemble. Anonymous et al. proposed a weighted ensemble of deep learning-based segmentation algorithms for cardiographic segmentation and achieved good results on the CAMUS competition [3]. Besides, there are some novel ensemble generation approaches inspired by the success of deep neural networks. Instead of using only one layer like in traditional ensemble models, the ensemble systems were made to train deeply through multiple layers. The first deep ensemble system was proposed by Zhou et al. [4] (called gcForest), containing multiple layers of two Completely-Random Tree Forests and two Random Forests in each layer. Each forest in a layer outputs a class vector, which is then concatenated to the original data as the input data to the next layer. Anonymous et al. [2] proposed MULES, a deep ensemble system with classifier and feature selection in each layer. The optimal configuration of each layer is found by using a bi-objective optimization problem in which the two objectives to be maximized are classification accuracy and diversity of the ensemble in each layer.

C. *Surrogate-assisted Evolutionary Algorithm (SAEA)*

In most real-world problems, evaluating a candidate solution with high accuracy usually involves a lot of computational time. For example, in evolutionary optimization of aerodynamic structures, computational fluid dynamic (CFD) simulations is usually used, however such simulations usually take hours or days to complete [22]. This seriously limits the application of EC to solving these problems, since EC requires fitness evaluation to guide the population toward the optimal solution. In order to circumvent this, researchers try to incorporate low-cost surrogate models with EC to solve the expensive problems. This approach is known as Surrogate-assisted evolutionary algorithm (SAEA). There exists a number of popular surrogate models, such as Radial Basis Functions (RBF) models, polynomial approximation (PR), Gaussian processes (GP) or Kriging, extreme learning machines, artificial neural networks (ANN), and support vector regression [23]. An important problem in SAEA is the choice of update for the surrogate model, or model management. There are three types of model management: individual-based, generation-based and their hybrids [22]. In generation-based management, all candidates are used for real fitness evaluation (FE) and the surrogate is updated after a number of generations, which can be fixed or adaptive [24]. In contrast, individual-based methods only choose a small number of individuals for real FE at each generation. There are a number of approaches to choose individuals. The random approach chooses some

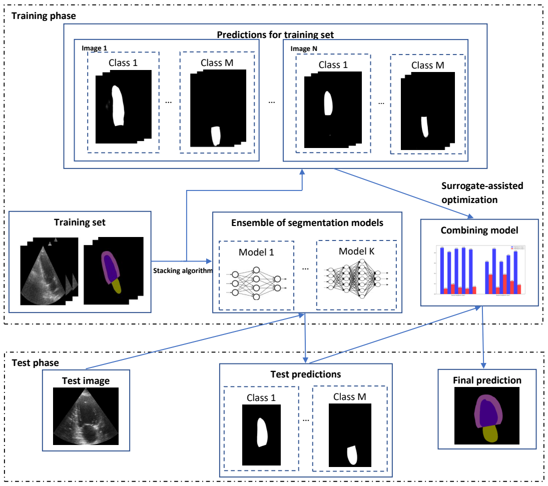


Fig. 1. Overview of the proposed framework.

random individuals for real FE [25], while [26] updated the surrogate using the best candidate at each generation to evolve deep learning architectures. Another approach is to cluster the population into a number of clusters, and the representative member of each cluster is used to update the surrogate model [28].

III. Proposed method

Figure 1 gives an overview of the proposed framework which consists of three main steps:

- Train ensemble of segmentation models and generate prediction for the training set: We develop an ensemble of K segmentation models to solve the medical image segmentation problem. The Stacking algorithm [1] is applied to the training set with segmentation algorithms to create the predictions for each image in the training set.
- Combining algorithm: A combining algorithm is applied to the predictions of the training set to obtain the combining model. We use Decision Template as the combining algorithm and PSO to search the optimal model for combining.
- Surrogate-assisted optimization of Decision Template: A surrogate model is developed to predict the objective value when using PSO in order to reduce computation time.

Suppose the training set $\mathbf{D} = \{(\mathbf{I}_n, \mathbf{Y}_n)\}_{n=1}^N$ consists of N images and ground truths where $\mathbf{I}_n$ is the n^{th} image and $\mathbf{Y}_n$ is the corresponding ground truth. Each image has height H and width W, and the ground truth has the same size as the image. The pixel at position (i, j) with $1 \le i \le W, 1 \le j \le H$ is denoted as $\mathbf{I}_n(i, j)$, while its ground truth $\mathbf{Y}_n(i, j)$ belongs to the set $\mathcal{Y}$, where $\mathcal{Y} = \{y_m\}_{m=1}^M$ is a set of M classes. For the semantic segmentation problem, we aim to segment each pixel of an image $\mathbf{I}$ into classes by using a segmentation model $\mathbf{Q}$ trained by a segmentation algorithm $\mathbf{h}$ on the training set $\mathbf{D}$. In this study, we train an ensemble of K segmentation models denoted by $\{\mathbf{Q}_k\}$ and then use a combining algorithm C to combine $\{\mathbf{Q}_k\}$ i.e. $C\{\mathbf{Q}_k\}$ to obtain prediction of the ensemble.

Algorithm 1 Training ensemble of segmentation models and generating predictions for training set

Input: Training images $\mathbf{D}$, K segmentation algorithms $\{\mathbf{h}_k\}_{k=1}^K$

Output: The predictions $\mathcal{P}$ and the trained segmentation models $\{\mathbf{Q}_k\}_{k=1}^K$

```
Learn $K$ segmentation models $\{\mathbf{Q}_k\}_{k=1}^K$ on $\mathbf{D}$ using $\{\mathbf{h}_k\}_{k=1}^K$
$\mathcal{P} = \emptyset$
$\mathbf{D} = \mathbf{D}_1 \cup ... \cup \mathbf{D}_T, \mathbf{D}_i \cap \mathbf{D}_j = \emptyset (i \ne j)$
for each $\mathbf{D}_t$ do
    $\tilde{\mathbf{D}}_t = \mathbf{D} - \mathbf{D}_t$
    Learn ensemble of segmentation models $\{\mathbf{Q}_k^t\}_{k=1}^K$ on $\tilde{\mathbf{D}}_t$ using $\{\mathbf{h}_k\}_{k=1}^K$
    Segment images in $\mathbf{D}_t$ by $\{\mathbf{Q}_k^t\}_{k=1}^K$
    Add outputs on images in $\mathbf{D}_t$ to $\mathcal{P}$ (Equation 1)
return $\mathcal{P}$ and $\{\mathbf{Q}_k\}_{k=1}^K$
```

A. *Prediction of training data - Ensemble of segmentation models*

In the first step, we aim to train K segmentation models and the predictions for training data so as to train the combining model. The K segmentation models $\{\mathbf{Q}_k\}$ are obtained by training K segmentation algorithms $\{\mathbf{h}_k\}$ on the training set $\mathbf{D}$. Meanwhile, the predictions for instances in the training set are generated by using the Stacking algorithm [1]. In this algorithm, $\mathbf{D}$ is divided into T dis-joined parts $\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_T$. The segmentation algorithms $\{\mathbf{h}_k\}_{k=1}^K$ train segmentation models $\{\mathbf{Q}_k^t\}, k = 1, ..., K$ on the part of $\mathbf{D} - \mathbf{D}_t$. The trained model $\{\mathbf{Q}_k^t\}$ will predict for images in the part $\mathbf{D}_t$ to output the probabilities that each pixel of an image belongs to the classes. This procedure runs through all T parts of $\mathbf{D}$ so that we can obtain the probability predictions for pixels of all training images. The predictions will be concatenated in the form of matrix:

$$\mathcal{P} = \begin{bmatrix} P_1(1,1)\cdots P_1(1,M) & P_1(2,1)\cdots P_1(2,M)\cdots P_1(K,1)\cdots P_1(K,M) \\ P_2(1,1)\cdots P_2(1,M) & P_2(2,1)\cdots P_2(2,M)\cdots P_2(K,1)\cdots P_2(K,M) \\ \cdots & \\ P_L(1,1)\cdots P_L(1,M) & P_L(2,1)\cdots P_L(2,M)\cdots P_L(K,1)\cdots P_L(K,M) \end{bmatrix} \tag{1}$$

where $L = N \times W \times H$ is the total number of pixels in the training set, $P_l(k, m)$ is the probability prediction that the l^{th} pixel belongs to class y_m given by k^{th} segmentation model $(1 \le l \le L, 1 \le k \le K, 1 \le m \le M)$. Predictions $\mathcal{P}$ and ground truth $\{\mathbf{Y}_n\}$ will be use as training data to train the combining model. The procedure of training ensemble of segmentation models and generating prediction for training set is described in the Algorithm 1.

It is noted that $\mathcal{P}$ is usually very big. Suppose we have a dataset of 100 images ($N = 100$), with $H = 480, W = 480$, for a three-class segmentation problem using three segmentation algorithms ($M = 3, K = 3$), the matrix $\mathcal{P}$ would have $L = N \times W \times H = 23,040,000$ rows and $M \times K = 3 \times 3 = 9$

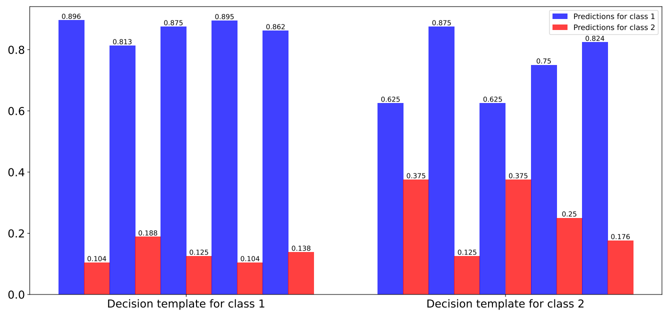


Fig. 2. Example of decision template

columns. It can be seen that the matrix size is very big even for a moderate-size image segmentation dataset, and any combining algorithm applied on this matrix would present great computational and memory requirements.

B. Combining algorithm

The next step is to use a combining algorithm C to train a combining model on the training data $(\mathcal{P}, \{\mathbf{Y}_n\}_{n=1}^N)$. There are many combining algorithms introduced and one of the most popular ones is Decision Template [18]. In this algorithm, a representation for each class called decision template is calculated by taking the average of the predictions $\mathcal{P}$ of all training instances associated with that class. The aim of decision templates is to create discriminative representations for all classes on the predictions of K models from which we can get the collaborated prediction. The decision template $\mathcal{DT} = \{DT_j\}_{j=1}^M$ where DT_j is a $(M \times K, 1)$ vector called decision template for class y_j which is given by the following formula:

$$DT_j = \begin{bmatrix} dt_j(1,1) & \cdots & dt_j(1,M) & \cdots & dt_j(K,1) & \cdots & dt_j(K,M) \end{bmatrix} \quad (2)$$

where the calculation of each entry is given as follows $(1 \leq l \leq L, 1 \leq j \leq M, 1 \leq k \leq K, 1 \leq m \leq M)$:

$$dt_j(k,m) = \frac{\sum_{l=1}^{L} \mathbb{I}[y_j == y_l] P_l(k,m)}{\sum_{l=1}^{L} \mathbb{I}[y_j == y_l]} \quad (3)$$

in which y_l is the true class label of the l^{th} pixel in the training data, $\mathbb{I}[.]$ is the indicator function. Equation 3 is the average value of the predictions of the pixels belonging to class y_j associated with segmentation algorithm $\mathbf{h}_k$.

In the testing phase, given an image $\mathbf{I}$, the segmentation models Q_k will first segment the image, giving the probability prediction $P_x(k,m)$ for each pixel $x \in \mathbf{I}$:

$$\mathcal{P}_x = \begin{bmatrix} P_x(1,1) \cdots P_x(1,M) \cdots P_x(K,1) \cdots P_x(K,M) \end{bmatrix} \quad (4)$$

Class y_x which is assigned to pixel x will be the class associated with the shortest Euclidean distance $||.||_2$ amongst M decision templates and $\mathcal{P}_x$.

$$y_x = argmax_m ||DT_m - \mathcal{P}_x||_2 \quad (5)$$

We observed that there are cases when the decision templates do not provide enough discrimination for classes. Figure 2 shows the decision templates of the Fertility dataset from the UCI machine learning repository. There are two classes in this dataset, and five classifiers were used to generate the predictions before computing the decision templates. It can be observed that the 2 templates for the 2 classes look similar as the predictions for class 1 always predominate over those of class 2, causing the poor discrimination of representation. A better discriminative representation is expected to obtain a better combining model.

Instead of applying Equation 3, we find the optimal decision template $\mathcal{DT} = \{DT_j\}_{j=1}^M$ which provides better discriminative representation for each class. We consider this problem in terms of searching for the optimal decision template which maximizes the Dice coefficient, which is one of the most popular segmentation metrics [33]. Let **pred** and **ground** be the predictions and ground truths of all training pixels:

$$\mathbf{pred} = \{pred_1, pred_2 ..., pred_M\} \quad (6)$$

$$\mathbf{ground} = \{ground_1, ground_2 ..., ground_M\} \quad (7)$$

in which $pred_m$ is the vector of size $(N \times W \times H, 1)$ in which its element is the prediction for each pixel belonging to the class label y_m in the form of crisp label i.e. in $\{0,1\}$. Likewise $ground_m$ is the vector of size $(N \times W \times H, 1)$ associated with the class label y_m which is the ground truth of each pixel in the form of crisp label i.e. in $\{0,1\}$. $pred_m$ is obtained based on the segmentation rule in Equation 5 while $ground_m$ is obtained from the ground truths $\{\mathbf{Y}_n\}$. The Dice coefficient is given by the following equation:

$$DC = \frac{1}{M} \sum_{m=1}^{M} \frac{2 \times pred_m^T ground_m}{||pred_m||^2 + ||ground_m||^2} \quad (8)$$

Thus the optimization problem is formulated as follows:

$$\begin{aligned} \max_{\mathcal{DT}} \quad & DC \\ \text{s.t.} \quad & 0 \leq dt_j(k,m) \leq 1 \end{aligned} \quad (9)$$

C. Surrogate-assisted optimization of Decision Template

In this study, the non-differentiable optimization problem in Equation 9 is solved by using Particle Swarm Optimization (PSO) [5], a swarm intelligence-based method which is known to be simple, easy to implement and computationally efficient [27]. PSO maintains a swarm of $popSize$ particles in which each particle with size $M \times K$ represents a candidate for optimal decision templates. The positions of a particle are defined by $x_i = (x_i^1, x_i^2, ..., x_i^{M \times K}), i = 1, ..., popSize$. A velocity $v_i = (v_i^1, v_i^2, ..., v_i^{M \times K})$ is associated with each particle x_i. At each iteration, the quality of each candidate is assessed based on its objective value $F(x_i)$, computed by Equation 8 with the predictions based on Equation 5, and the local best and global best position is stored. Then the velocity is updated as follows:

$$v_i^u \leftarrow a \times v_i^u + c_1 \times r_1 \times (pbest_i^u - x_i^u) + c_2 \times r_2 \times (gbest^u - x_i^u) \quad (10)$$

where a is the inertia weight which controls the velocity speeding rate, c_1 and c_2 are acceleration constants used to control the learning rate of the particle's local best and the swarm global best, respectively. $pbest_i^u$ is the u^{th} dimension of i^{th} particle's best position ($1 \leq u \leq M \times K$) and $gbest^u$ is the u^{th} dimension of the swarm's best position. r_1 and r_2 are two random number drawn from a uniform distribution over $[0, 1]$. The inertial weight is calculated via the following equation [1]:

$$a = w_{max} - \frac{(w_{max} - w_{min}) \times t}{nIter} \tag{11}$$

where t is the current iteration, $nIter$ is the total number of iterations. The position of each particle is updated in each iteration to search for the optimal solution:

$$x_i^u \leftarrow x_i^u + v_i^u \tag{12}$$

The search algorithm terminates after $nIter$ iterations and we can obtain the solution for the optimization problem.

As discussed in the previous section, the matrix $\mathcal{P}$ has a very large size. Considering that PSO requires the calculation of objective values for each particle over a number of iterations, it would be prohibitively expensive to apply PSO to search for the optimal decision template.

In this paper, we propose using a surrogate-assisted method to predict the objective value of a candidate decision template instead of always calculating its objective function $F(.)$ at each iteration. Let x be the candidate decision template, then $F(x)$ will output the objective value. A surrogate model $G(.)$ is a function which, given the input x, will output the predicted objective value $G(x)$ in less computation time compared to $F(x)$. The procedure for training and applying the surrogate model is designed as follows:

- We first run PSO normally for n_1 iterations ($n_1 < nIter$). That means the objective value $F(x_i)$ of each candidate x_i is computed on entire $\mathcal{P}$. The data with size $n_1 \times popSize$ including candidate decision templates like x_i and their associated objective values $F(x_i)$ is used to train the surrogate model.
- The surrogate model G is trained on the data $(x_i, F(x_i))$ to approximate the relationship between a candidate and its objective value. G then will be used to predict the objective value of candidate decision templates in the subsequent iterations.
- The surrogate model G is updated to adapt with the changes of candidate's objective value relationship in the search process. In this study, after n_2 iterations, the objective value of each candidate is calculated normally using $F(.)$ and the surrogate model G is updated based on the new data $(x_i, F(x_i))$.

Let t_F be the time needed to calculate the objective function on the entire $\mathcal{P}$ and t_G is the time by using the surrogate model G to predict the objective function, and $t_G < t_F$. For the first n_1 iterations, the time taken would be:

$$T_{Initial} = n_1 \times popSize \times t_F \tag{13}$$

If surrogate model $G(.)$ is not used, the time taken for the remaining $(nIter - n_1)$ iterations would be:

$$T_F = (nIter - n_1) \times popSize \times t_F \tag{14}$$

Algorithm 2 PSO optimization with surrogate model

```
Input: Predictions $\mathcal{P}$, maximum number of iterations $nIter$, population size $popSize$, $c_1, c_2$, number of initial generations before the surrogate is used $n_1$, number of generations before update is performed $n_2$, surrogate model $G(.)$, original objective function $F(.)$
Output: The optimal candidate $x$
1:  Initialize population $x_1, ..., x_{popSize}$ and velocity $v_1, ..., v_{popSize}$
2:  for $n$ from 1 to $n_1$ do
3:      for $i$ from 1 to $popSize$ do
4:          $obj = F(x_i)$
5:      Use the objective values to evaluate $gbest$ (global best) and $pbest$ (local best)
6:      Update velocity for each candidate using Equation 10
7:      Update the candidates using Equation 12
8:  Use the objective values calculated to initialize the surrogate model $G(.)$
9:  for $n$ from $n_1 + 1$ to $nIter$ do
10:     if $(n - n_1)\%n_2 == 0$ then
11:         for $i$ from 1 to $popSize$ do
12:             $obj = F(x_i)$
13:         Use the objective values calculated to update the surrogate model $G(.)$
14:     else
15:         for $i$ from 1 to $popSize$ do
16:             $obj = G(x_i)$ // Use the surrogate function
17:     Use the objective values to evaluate $gbest$ (global best) and $pbest$ (local best)
18:     Update velocity for each candidate using Equation 10
19:     Update the candidates using Equation 12
20: return $x = gbest$
```

Otherwise for the case when surrogate model $G(.)$ is used, the number of iterations where $F(.)$ is used would be $\frac{(nIter - n_1)}{n_2}$ (since the update is performed every n_2 iterations) and the number of iterations where the surrogate model $G(.)$ is used is:

$$(nIter - n_1) - \frac{(nIter - n_1)}{n_2} = \frac{(nIter - n_1)(n_2 - 1)}{n_2} \tag{15}$$

The time for this case would then be:

$$\begin{aligned} T_G &= \frac{(nIter - n_1) popSize \times t_F}{n_2} + \frac{(nIter - n_1)(n_2 - 1) popSize \times t_G}{n_2} \\ &= \frac{(nIter - n_1) \times popSize[t_F + t_G \times (n_2 - 1)]}{n_2} \end{aligned} \tag{16}$$

The time saved by using the surrogate model would be:

$$\Delta T = T_F - T_G = (n_2 - n_1) \times popSize \times \frac{(t_F - t_G) \times (n_2 - 1)}{n_2} \tag{17}$$

Our algorithm is described in detail in Algorithm 2. In line 1, the population is initialized. From line 2 to line 7, the first n_1 iterations are run, in which at each iteration the objective values are calculated (line 3-4), $gbest$ and $pbest$ are chosen (line 5), the velocity and candidates are updated using Equation 10 and Equation 12 (line 6-7). At line 8, the objective values are used to initialize the surrogate model $G(.)$. After that, from iteration $n_1 + 1$ to iteration $nIter$, after each n_2 iteration, the original objective function is used, then the surrogate model is updated (line 10-13) otherwise the surrogate function is used (line 14-16). The updates of $gbest$,

TABLE I
DICE COEFFICIENT RESULTS

	CPM-17	EAD-19	Promise12	Red Lesion
UNet-VGG16	0.89349	0.60122	0.75572	0.95826
LinkNet-VGG16	0.86906	0.53730	0.75911	0.93427
FPN-VGG16	0.91857	0.52466	0.89467	0.95215
UNet-ResNet34	0.91401	0.65588	0.90906	0.96074
LinkNet-ResNet34	0.90116	0.63333	0.88222	0.96048
FPN-ResNet34	0.90833	0.65705	0.90696	0.96324
UNet-ResNet101	0.89729	0.54014	0.88954	0.94494
LinkNet-ResNet101	0.89335	0.51021	0.87153	0.94739
FPN-ResNet101	0.89344	0.51892	0.88962	0.95250
Weighted ensemble	0.91986	0.66353	0.91887	0.96411
DT	0.91426	0.63033	0.91494	0.96136
ODTwS	0.91467	0.65685	**0.92096**	0.96485
Proposed method	**0.92008**	**0.67548**	0.91864	**0.96486**

TABLE II
MAD RESULTS

	CPM-17	EAD-19	Promise12	Red Lesion
UNet-VGG16	3.43312	31.59863	5.73269	2.31072
LinkNet-VGG16	4.21508	36.17425	9.71780	5.21094
FPN-VGG16	2.09656	20.80811	2.87209	2.57937
UNet-ResNet34	2.21969	18.49515	2.27919	1.97219
LinkNet-ResNet34	2.68086	21.01394	3.22702	1.97697
FPN-ResNet34	2.34745	17.09293	2.43242	1.76366
UNet-ResNet101	2.69950	24.42896	2.65853	2.87523
LinkNet-ResNet101	2.98517	32.34771	3.15007	2.91733
FPN-ResNet101	2.91055	24.60246	2.84251	2.48759
Weighted ensemble	2.03470	**16.46748**	2.43193	1.71864
DT	2.20465	23.74325	2.34031	1.89920
ODTwS	2.26373	19.84421	**2.15538**	1.69317
Proposed method	**2.00719**	19.78338	2.18445	**1.68530**

$pbest$, velocity and the candidates are still the same (line 17-19). Finally, $gbest$ is returned which is associated with the optimal decision template for the combining model.

IV. EXPERIMENTAL STUDIES

A. Experimental Settings

In this experiment, three popular segmentation architectures were used: UNet [7], LinkNet [8] and Feature Pyramid Network (FPN) [13]. The backbones used were VGG16, ResNet34 and ResNet101 [11], pretrained on the ImageNet dataset [6]. Thus in total, 9 segmentation algorithms were used in the experiments. Each algorithm was trained for 300 epochs to obtain the segmentation model. We set $T = 5$ for the T-fold cross-validation procedure in the Stacking algorithm. For the PSO algorithm, $popSize$ was set to 10, $nIter = 500$, while $c_1 = c_2 = 1.494$ and $w_{max} = 0.9, w_{min} = 0.5$ and were set according to [1]. The surrogate model used in this paper is Radial Basis Function (RBF) which is one of the most popular surrogate models in the literature [23]. The parameters n_1 and n_2 were set to 100 and 3 respectively. The proposed method were compared with the 9 base segmentation models and three additional benchmarks, weighted ensemble [3], Decision Template [18] (denoted by DT) and the optimal Decision Template found via PSO method without using surrogate model (denoted by ODTwS).

Three performance metrics were used for the evaluation of the base segmentation algorithms and the proposed ensemble: Dice coefficient, Intersection-over-Union (IoU), and MAD. Dice coefficient, defined in Equation 8, is one of the most popular metrics for medical image segmentation. However, its shortcoming is that it is a measure for total volume difference, without taking into account local discrepancies between contours, which is important in the context of medical image analysis [15]. Therefore, we also used another distance measure between geometrical contours for the evaluation. Let GT_m and PR_m be the set of coordinate vectors of the ground truth contour and prediction contour with respect to class y_m respectively. The Mean Absolute Distance (MAD) [37] is defined as follows:

$$MAD = \frac{1}{M}\sum_{m=1}^{M} \frac{\sum_{gt \in GT_m} \min_{pr \in PR_m} ||gt - pr|| + \sum_{pr \in PR_m} \min_{gt \in GT_m} ||pr - gt||}{|GT_m| + |PR_m|} \tag{18}$$

Intersection-over-Union (IoU) (also known as Jaccard index) [37] is defined as follows:

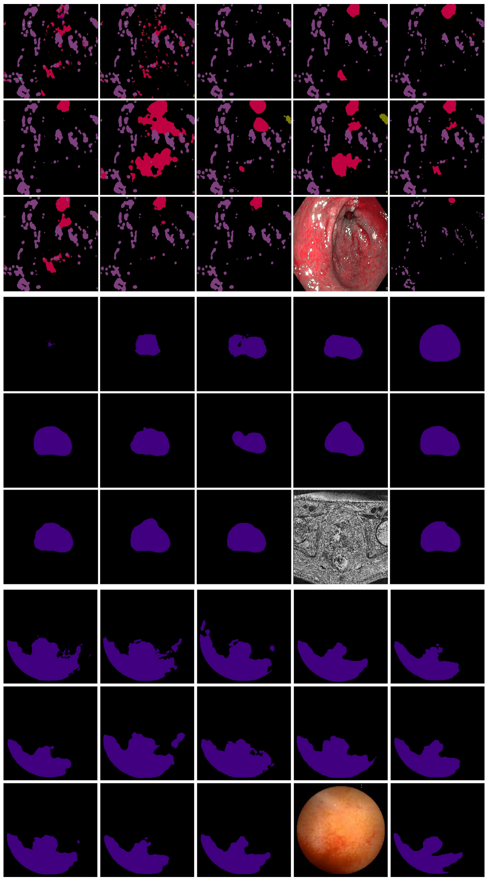
$$IoU = \frac{1}{M}\sum_{m=1}^{M} \frac{pred_m^T ground_m}{||pred_m||^2 + ||ground_m||^2 - pred_m^T ground_m} \tag{19}$$

It is noted that low MAD or high Dice coefficient/IoU corresponds to good segmentation results.

A number of public medical image segmentation datasets were used in the experiments. The first dataset is CPM-17 [36], a nucleus segmentation dataset consisting of whole slide tissue images among which there are 32 images for training and 32 images for testing. The second dataset is Endoscopy Artefact Detection (EAD-19) [34], a dataset created to address the problem of detection of artefacts in video endoscopy. There are 475 images and six classes: specularity, saturation, artifact, bubbles, instrument and background. The third dataset used in this paper is Prostate MR Image Segmentation 2012 (Promise12) dataset [12], which contains MRI data acquired in various conditions typically encountered in clinical settings. The final dataset is the Red Lesion dataset [35], a dataset containing images and ground truths of red lesion in the small bowel. The dataset contains 1,570 frames with red lesion and 2,325 frames without lesion.

B. Results and discussion

Table I shows the results concerning the Dice coefficient. The first 9 rows denote the results of the 9 base segmentation models, while the next four rows the results of 3 selected benchmark algorithms and the proposed method. It can be seen that the proposed method achieved overall better results on most datasets except Promise12. For the CPM-17 dataset, the 9 base segmentation models achieved from 0.86906 to 0.91857 while the proposed method obtained the highest score at 0.92008. This can also be observed for the three other datasets (higher than 0.16% and 1.84% on Red Lesion, and EAD-19). Compared with DT, the proposed method achieved better results on all datasets, especially on EAD-19 (a difference of 4.52%). On this dataset, the Dice coefficient of the proposed method was also higher than that of ODTwS by 1.86%. For the other datasets, the proposed method also obtained better results compared with ODTwS except for Promise12 in which ODTwS obtained a higher score by a small margin. The



From left to right, top to bottom (for each dataset): UNet-VGG16, UNet-ResNet34, UNet-ResNet101, LinkNet-VGG16, LinkNet-ResNet34, LinkNet-ResNet101, FPN-VGG16, FPN-ResNet34, FPN-ResNet101, Weighted ensemble, DT, ODTwS, Proposed method, Original image, Ground truth.

Fig. 3. Several examples from three datasets (from top to bottom): EAD-19, Promise12 and Red Lesion dataset.

TABLE III
IoU RESULTS

	CPM-17	EAD-19	Promise12	Red Lesion
UNet-VGG16	0.81250	0.46412	0.66822	0.92121
LinkNet-VGG16	0.77524	0.40507	0.66978	0.87967
FPN-VGG16	0.85266	0.39417	0.82491	0.91035
UNet-ResNet34	0.84514	0.51406	0.84522	0.92563
LinkNet-ResNet34	0.82443	0.49280	0.80780	0.92519
FPN-ResNet34	0.83608	0.51469	0.84221	0.93015
UNet-ResNet101	0.81857	0.39915	0.81783	0.89788
LinkNet-ResNet101	0.81233	0.37872	0.79402	0.90217
FPN-ResNet101	0.81234	0.38055	0.81798	0.91098
Weighted ensemble	0.85471	0.52246	0.85964	0.93172
DT	0.84546	0.49262	0.85381	0.92675
ODTwS	0.84609	0.51626	**0.86280**	**0.93307**
Proposed method	**0.85515**	**0.53540**	0.85933	**0.93307**

TABLE IV
COMPARE TIME BETWEEN ODTWS, THE PROPOSED METHOD AND WEIGHTED ENSEMBLE (IN HOURS)

	ODTwS	Proposed method	Weighted ensemble
Red Lesion	39.4	**19.02**	65.98
CPM-17	3.6	**1.83**	5.83
EAD-19	41.67	**20.04**	67.43
Promise12	21.07	**10.19**	16.16

proposed method also achieved better MAD (Table II) and IoU (Table III) scores as well. For the EAD-19 dataset, the proposed method obtained a MAD score of 19.78338 while the best score is obtained by the weighted ensemble at 16.748. It should be noted that even in this case, the proposed method still obtained better result than ODTwS which is at 19.84421. For the Promise12 dataset, ODTwS achieved the best result at 2.15538 compared to the proposed method at 2.18445. For the IoU score (Table III), the proposed method also achieved the best results on CPM-17 and EAD-19, while on Promise12, ODTwS obtained better score than the proposed method and on Red Lesion both the proposed method and ODTwS achieved the best result at 0.93307.

Figure 3 shows three prediction examples by the proposed method and the benchmark algorithms for the EAD-19, Promise12, and Red Lesion datasets (from top to bottom). For each image, from left to right, top to bottom, are the predictions of the 9 base segmentation models, weighted ensemble, DT, ODTwS, the proposed method, image, and ground truth respectively. The top image shows the example for EAD-19 dataset, in which violet color denotes specularity, red denotes bubble, olive denotes artifact, indigo denotes instrument and medium green represents saturation class. The base segmentation models predicted larger areas of specularity than the ground truth, and some models, such as FPN-VGG16, predicted two large areas of the bubble while there was in fact only a small part on the top right. The predictions by weighted ensemble (2^{nd} row, 5^{th} column) and DT (3^{rd} row, 1^{st} column) still contained bubble in three small areas. Compared to them, the predicted bubbles by ODTwS and the proposed method were more in agreement with the ground truth. For the Promise12 dataset, the predictions by the benchmark algorithms (such as UNet-VGG16, FPN-ResNet34 and ODTwS) had deformations compared to the ground truth, while weighted ensemble, DT and the proposed method provided generally correct shape. For the Red Lesion dataset, the ground truth consisted of a large crescent area on the bottom left and a small adjacent circle area. The base segmentation models either predicted a larger circle area (UNet-VGG16 and LinkNet-VGG16) or wrongly predicted a small separate area on the right (FPN-VGG16). DT's prediction for the circle area was too big, while weighted ensemble and ODTwS (3^{rd} row, 2^{nd} column) failed to predict an adjacent area at the right. The proposed method achieved the best result among all the presented methods.

Table IV shows the comparison between the run time of the proposed method which uses the surrogate model, ODTwS which does not use any surrogate in its calculation, and finally the weighted ensemble method. The reason for choosing these methods is because they are optimization-based methods. It can be seen that the results on all four datasets show that the proposed method achieved significant savings compared with the case when the surrogate model was not used. For the Red Lesion dataset, the original run time was around 39.4 hours, while the surrogate run time was just 19.02 hours, which is a savings of 2.07 times. The same results can be observed on the other datasets. This can be explained from the results of Equation 17. The proposed method achieved a saving of 1.5-3 times compared to the weighted ensemble.

The results discussed above demonstrate that (i) The proposed method achieved better results compared to the base segmentation models on all datasets. This shows the effectiveness of the proposed ensemble of segmentation models (ii) The proposed method obtained a higher score than DT for all datasets, especially for EAD-19 (higher than 4.52%). It can be seen that finding the optimal DT provides significantly better results compared to using the original method. (iii) The proposed method outperforms ODTwS on most datasets except on Promise12 in which ODTwS obtained a slightly higher score. This demonstrates that using surrogate model to predict the objective value provides competitive results while achieving significant computational savings. (iv) The proposed method is better than the weighted ensemble on most cases except on Promise12 for Dice and IoU score and EAD-19 for MAD score by a small margin. However, the computation time is reduced from 1.5-3 times compared to the weighted ensemble, and this demonstrates further the effectiveness of the proposed method.

V. Conclusion

In this paper, we proposed a novel ensemble of different deep learning-based segmentation models for medical image segmentation. The combining model working on the outputs of these segmentation models based on the basic idea of the Decision Template method is optimized using PSO. Dice coefficient, a common performance metric for medical image segmentation is used as the objective function criteria. Since the evaluation of each candidate in the optimization algorithm is computationally expensive, we propose to use a surrogate model in order to predict the objective values of each candidate. The surrogate model is trained on initialized data and then is updated after several iterations. The proposed method was evaluated on 4 popular medical image segmentation datasets with Dice coefficient, Mean Absolute Distance and Intersection-over-Union. The experiments show that our proposed method achieves competitive results compared to the individual segmentation models and the selected benchmark algorithms while drastically reducing computational time.

References

[1] T.T. Nguyen, M.T. Dang, V.A. Baghel et al., Evolving interval-based representation for heterogeneous classifier fusion, Knowl Based Syst., 2020.

[2] T. T. Nguyen, N. V. Pham, M. T. Dang et al., Multi-layer heterogeneous ensemble with classifier and feature selection. In GECCO, 2020.

[3] T. Dang, T. T. Nguyen et. al., Weighted Ensemble of Deep Learning Models based on Comprehensive Learning Particle Swarm Optimization for Medical Image Segmentation. In IEEE CEC, 2021, pp. 744-751.

[4] Z.-H. Zhou, J. Feng. Deep Forest: Towards An Alternative to Deep Neural Networks. In IJCAI, 2017, pp. 3553-3559.

[5] J. Kennedy, R. Eberhart, Particle Swarm Optimization. In: Proceedings of ICNN'95, 1995, pp. 1942-1948.

[6] A. Krizhevsky, S. Ilya, H. Geoffrey. ImageNet classification with deep convolutional neural networks, in Commun. ACM, 2017, pp. 84-90.

[7] O. Ronneberger, P. Fischer, T. Brox, U-Net. Convolutional Networks for Biomedical Image Segmentation, in Proceedings of MICCAI, 2015.

[8] A. Chaurasia, E. Culurciello. LinkNet: Exploiting encoder representations for efficient semantic segmentation. In IEEE Int. Conf. Vis. Commun., 2017, pp. 1-4.

[9] T. Lin, P. Dollar, R. Girshick et al.. Feature Pyramid Networks for Object Detection, in IEEE CVPR, 2017, pp. 936-944.

[10] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. In IEEE TPAMI, 2017, pp. 2481-2495.

[11] F. Lateef, Y. Ruichek. Survey on semantic segmentation using deep learning techniques. Neurocomputing, Elsevier, 2019, 338, pp.321-348.

[12] F. Milletari, N. Navab, S. Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation, in Fourth International Conference on 3D Vision, 2016, pp. 565-571.

[13] J. Long, E. Shelhamer, T. Darrell. Fully convolutional networks for semantic segmentation, in IEEE CVPR, 2015, pp. 3431-3440.

[14] A.G.C. Pacheco, T. Trappenberg, R.A. Krohling. Learning dynamic weights for an ensemble of deep models applied to medical imaging classification, in IJCNN, 2020, pp. 1-8.

[15] H. Kim, S. Park, S. Lo et al. Bidirectional local distance measure for comparing segmentations, in Med. Phys., 2019, pp. 6779-6790.

[16] K.-J. Kim, S.-B. Cho. An evolutionary algorithm approach to optimal ensemble classifiers for DNA microarray data analysis, IEEE Trans. Evo. Comp., 2008, pp. 377-388.

[17] L. Xie, A. Yuille, Genetic CNN. In: ICCV, 2017, pp. 1388-1397.

[18] L.I. Kuncheva, J.C. Bezdek, R.P.W. Duin. Decision templates for multiple classifier fusion: an experimental comparison, Pattern Recognit. 34 (2001) 299-314.

[19] V.G. Edupuganti, A. Chawla, K. Amit. Automatic optic disk and cup segmentation of fundus images using deep learning. In IEEE International Conference on Image Processing (ICIP), 2018, pp. 2227-2231.

[20] Z. Huo, Z. Xu, S. Bao et. al. Splenomegaly segmentation using global convolutional kernels and conditional generative adversarial networks. Med. Imaging, 2018.

[21] O. Oktay, J. Schlemper, L.L. Folgoc et al. Attention U-Net: Learning where to look for the pancreas. arXiv2018, arXiv:1804.03999.

[22] Y. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm Evol, 2011.

[23] P. Jeng-Shyang. An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems. In: Inf. Sci, 2021.

[24] Y. Jin et. al. Data-driven evolutionary optimization: an overview and case studies, IEEE Trans. Evo. Comp. 2019, pp. 442–458.

[25] Y. Jin, M. Olhofer, B. Sendhoff. On evolutionary optimization with approximate fitness functions. In GECCO 2000.

[26] Y. Sun, H. Wang, B. Xue et al. Surrogate-Assisted Evolutionary Deep Learning Using an End-to-End Random Forest-Based Performance Predictor. In IEEE Trans. Evo. Comp. 2020.

[27] Y. Sun, B. Xue, M. Zhang et al. A particle swarm optimization-based flexible convolutional autoencoder for image classification, IEEE Trans. Neural Netw. Learn. Syst. (2018) pp. 1–15.

[28] F. Mota, F. Gomide. Fuzzy clustering in fitness estimation models for genetic algorithms and applications. In IEEE Int. Conf. Fuzzy Syst., 2006.

[29] H. Brody. Medical imaging. Nature 502, S81 (2013).

[30] Z. Liu, H. Wang, S. Zhang et al. NAS-SCAM: Neural Architecture Search-Based Spatial and Channel Joint Attention Module for Nuclei Semantic Segmentation and Classification. In MICCAI 2020, pp. 263-272.

[31] H. Zheng, Y. Zhang, L. Yang et. al. A New Ensemble Learning Framework for 3D Biomedical Image Segmentation. In AAAI, 2019.

[32] Z. Yang, L. Ran, S. Zhang et. al. EMS-Net: Ensemble of Multiscale Convolutional Neural Networks for Classification of Breast Cancer Histology Images, Neurocomputing, 2019.

[33] Z. Kelly, W. Simon, B. Aditya et al. (2004). Statistical Validation of Image Segmentation Quality Based on a Spatial Overlap Index. Academic radiology. 11. 178-89. 10.1016/S1076-6332(03)00671-8.

[34] S. Ali, F. Zhou, C. Daul et al. Endoscopy artifact detection (EAD 2019) challenge dataset. arXiv preprint arXiv:1905.03209.

[35] P. Coelho, A. Pereira, A. Leite et al. A Deep Learning Approach for Red Lesions Detection in Video Capsule Endoscopies. In ICIAR 2018.

[36] Q.D Vu et al. Methods for segmentation and classification of digital microscopy tissue images. Front Bioeng. Biotechnol., 2019.

[37] A. Taha, A. Hanbury. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. In BMC Med. Imaging, 2015.