Ensemble learning based on classifier prediction confidence and comprehensive learning particle swarm optimisation for medical image segmentation.

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

2022

Ensemble Learning based on Classifier Prediction Confidence and Comprehensive Learning Particle Swarm Optimisation for Medical Image Segmentation

Truong Dang[1], Tien Thanh Nguyen[2], and John McCall[3]
National Subsea Centre, Robert Gordon University
Aberdeen, UK
Email: [1]t.dang1@rgu.ac.uk,[2]t.nguyen11@rgu.ac.uk,[3]j.mccall@rgu.ac.uk

Alan Wee-Chung Liew [4]
School of ICT, Griffith University
Queensland, Australia
Email: [4]a.liew@griffith.edu.au

***Abstract*—LVSegmentation, a process of partitioning an image into multiple segments to locate objects and boundaries, is considered one of the most essential medical imaging process. In recent years, Deep Neural Networks (DNN) have achieved many notable successes in medical image analysis, including image segmentation. Due to the fact that medical imaging applications require robust, reliable results, it is necessary to devise effective DNN models for medical applications. One solution is to combine multiple DNN models in an ensemble system to obtain better results than using each single DNN model. Ensemble learning is a popular machine learning technique in which multiple models are combined to improve the final results and has been widely used in medical image analysis. In this paper, we propose to measure the confidence in the prediction of each model in the ensemble system and then use an associate threshold to determine whether the confidence is acceptable or not. A segmentation model is selected based on the comparison between the confidence and its associated threshold. The optimal threshold for each segmentation model is found by using Comprehensive Learning Particle Swarm Optimisation (CLPSO), a swarm intelligence algorithm. The Dice coefficient, a popular performance metric for image segmentation, is used as the fitness criteria. The experimental results on three medical image segmentation datasets confirm that our ensemble achieves better results compared to some well-known segmentation models.**

***Index Terms*—image segmentation, deep learning, ensemble selection, ensemble method, particle swarm optimization**

I. Introduction

Over the last decades, due to technological advancement, the amount of medical data such as MRI computed tomography (CT), X-ray and magnetic resonance (MR) have grown rapidly. They provide important information for clinical decision making processes. One of the most important tasks of medical image analysis is segmentation, which is the process in which an image is partitioned into a number of segments which delineate different kind of objects. However it is very time-consuming and laborious for experts to segment medical images manually, especially on a large amount of data. In order to efficiently handle this ever-growing amount of data, Artificial Intelligence (AI) has been considered as one of the most prominent solutions, where AI refers to computer algorithms which has the capabilities to perform human-level tasks like automatically segment an image. Since 2012, there have been many applications of deep learning to segmentation. However, compared with other problems like image classification in which there are many datasets having millions of examples (for example ImageNet [10]) to efficiently train the deep models, the amount of publicly available medical images is still limited. Considering that the breakthrough of deep learning was achieved by training on ImageNet [15], this means that deep learning models for medical images are still not exploited to their full potential. Another problem is that deep learning models generally require careful parameter tuning to achieve good results. These shortcomings create challenges in choosing a suitable and robust deep learning model for clinical applications. One solution for these challenges is to exploit the strength of multiple segmentation models to provide an improved result.

Ensemble learning is a popular technique in which a number of machine learning methods are combined to create a collaborated decision. However, it is observed that not any combination gives the desired results. The presence of some models may downgrade the ensemble performance and they should be removed from the ensemble. The idea of this paper is based on the real-life observation that when a committee of experts consults on a problem, each of them usually has different background and level of expertise. If an expert is known to be very knowledgeable in a field, his/her recommendation would be trusted even though he/she might not be sure about the current recommendation. In contrast, if an expert is not knowledgeable about the issue being discussed then we would not trust his/her recommendation even if he/she is very sure of it. We apply this idea to select the optimal subset of deep segmentation models for medical image segmentation. The expertise level of each segmentation model is encoded by using a threshold. The confidence of the prediction of each model is measured and then compared with the corresponding threshold to determine whether this model should be included in the ensemble. We propose using

Shannon entropy to measure confidence in the prediction. The optimal threshold for each segmentation model is found by maximizing the Dice coefficient, a popular performance metric for image segmentation, using Comprehensive Learning Particle Swarm Optimisation (CLPSO), a swarm intelligence algorithm.

The paper is organised as follows. In section 2, we provide a brief review of the existing approaches relating to medical image segmentation, ensemble learning, and PSO. Our proposed ensemble is introduced in section 3. The details of experimental studies on three medical image segmentation datasets are described in section 4. Finally, the conclusion is given in section 5.

II. Background and Related Work

A. Medical Image Segmentation

Before the rise of deep learning, the majority of works on medical segmentation relied on hand-crafting low-level image processing methods to obtain candidate boundaries [36], [12]. However, handcrafted features are difficult to create and it is more difficult to extract discriminating features from medical images compared to other types of images due to various noises, low contrast etc. [34]. Since its success in image classification in 2012, deep learning has been widely applied to segmentation. One of the first successful architectures was the Fully Convolutional Network (FCN) [29]. This architecture uses an existing classification network, such as VGG16 [30], as the backbone and replaces the fully connected layers with upsampling layers to produce pixel-level segmentation result. There have also been deep networks specifically designed for the segmentation of medical images. A notable example is UNet [28], a deep segmentation network designed for the problem of segmentation of neuronal structures in electron microscopic stacks. Building upon FCN, the authors combined high resolution features from the convolutional layers with the upsampled output, which facilitates more precise segmentation based on this information. An important contribution of this method is that in the upsampling part there is also a large number of feature channels which allow the network to propagate context information to successive layers. The network is therefore largely symmetric. Other notable examples are LinkNet [5] which takes the sum of the upsampled output and the corresponding features in the convolutional path, and Feature Pyramid Network (FPN) [18] which uses the concatenation of features of all levels in the upsampling part to help with the final prediction. [22] proposed V-Net, which is an extension of UNet to 3D medical segmentation data. [4] proposed a cascade of V-Net for the problem of brain tumor segmentation by segmenting each region separately before combining. In [13] the authors integrated cross-modality MRI generated from the CT in order to improve segmentation quality. Another notable work is [26] in which the authors proposed attention UNet for pancreas segmentation, achieving 2-3% higher Dice scores compared to other benchmarks. [9] proposed a weighted ensemble of deep learning-based segmentation models in which weighted summation is used to combine the predictions of each segmentation model. The authors used the weights found by solving an optimisation problem using CLPSO for the summation.

B. Ensemble Learning and Ensemble Selection

Ensemble learning is a popular machine learning technique in which multiple learners i.e. classifiers are combined to improve the overall performance. Typically, ensemble systems are built by either training a learning algorithm on multiple training sets generated from the original training data or training different learning algorithms on the original training data to generate the ensemble [24], [25]. Afterwards, a combining method is then applied to the predictions of the generated classifiers for the final decision. There are some techniques concerning the combining methods. Nguyen et al. [24] searched for the weights of classifiers in the combining by minimizing the distance between these combinations computed on the training data and the class label of training observations. Pacheco et al. [27] modelled the output probabilities as a Dirichlet distribution and optimised the weights of classifiers using a loss function based on Mahalanobis distance. In [16], the authors proposed Decision Template method to combine classifiers in which the decision templates associated with class labels are calculated by taking the average of the predictions of all training instances. For each new observation, the distance between each decision template and the prediction for this observation is calculated and the class having the smallest distance is chosen.

Meanwhile, based on the observation that the presence of some classifiers might lower the performance of the ensemble, there have been many research efforts into *Ensemble Selection* (ES) (also known as ensemble pruning) which aims to select a subset of classifiers which performs better than the whole ensemble. There are two approaches to ensemble selection: static or dynamic approach. The static approach selects a subset of classifiers during the training phase and uses it for the testing phase. The static approach can be further divided into ordering-based methods and optimisation-based methods. The ordering-based methods try to order the classifiers according to ranking criteria e.g. validation error [20] or margin [21], among which only the top classifiers are selected. Optimisation-based methods formulate ensemble selection as an optimisation problem which can be solved by heuristic optimisation or mathematical programming. For example, Ant Colony Optimisation (ACO) was used in [6] to find the optimal set of classifiers and combining method in the ensemble systems. In [23], the authors introduced a novel encoding to simultaneously search for the optimal set of classifiers and the associated features using Genetic Algorithm (GA). It is recognized that the static approach limits the flexibility of the selection procedure [3]. In contrast, the dynamic approach selects a different subset of classifiers for each test instance. In the dynamic approach, a classifier is selected based on its performance in a local region of the feature space called Region of Competence (RoC). A comparative review of dynamic methods can be found in [3].

III. Proposed Method

Let $\mathbf{D}$ be the training set of N observations $\{(\mathbf{I}_n, \mathbf{Y}_n)\}_{n=1}^{N}$ where $\mathbf{I}_n$ is the n^{th} training image with size $W \times H$, and $\mathbf{Y}_n$ is the corresponding ground truth. The ground truth $\mathbf{Y}_n$ has the same size as $\mathbf{I}_n$ in which each position denotes the class label of the corresponding image pixel. Each class label belongs to a set of labels $\mathcal{Y} = \{y_m\}_{m=1}^{M}$ i.e. $\mathbf{Y}_n(i,j) \in \mathcal{Y}(1 \leq i \leq W, 1 \leq j \leq H)$. Let $\mathbf{K} = \{\mathcal{K}_k\}_{k=1}^{K}$ be the set of K segmentation algorithms and each learning algorithm $\mathcal{K}_k$ trains the segmentation model $\mathbf{C}_k$ on the training data $\mathbf{D}$. For an image $\mathbf{I}$, let $P_{k,m}(\mathbf{I}(i,j))$ denote the prediction probability by the model associated with $\mathcal{K}_k$ that the pixel $\mathbf{I}(i,j)(1 \leq i \leq W, 1 \leq j \leq H)$ belongs to class y_m. There are several constraints on $\{P_{k,m}(\mathbf{I}(i,j))\}$ as $0 \leq P_{k,m}(\mathbf{I}(i,j)) \leq 1$ and $\sum_{m=1}^{M} P_{k,m}(\mathbf{I}(i,j)) = 1$ for each k. In ensemble learning, the prediction probabilities $\{P_{k,m}(\mathbf{I}(i,j))\}$ of the K models are combined to obtain the final prediction.

In ensemble learning, the predictions from all models are usually used for combination to create the final prediction. However, it is possible that the presence of some models degrades the ensemble performance. Our idea for models selection is based on the observation in real-life when consultation from an expert committee is required. Experts' answers have different levels of confidence so that we should treat them differently when conducting the aggregation. Applying this idea to our problem, it can be seen that for optimal selection of deep segmentation models, each model should have a particular evaluation criteria for selection into the ensemble. We compute the *Shannon entropy* of the prediction by $\mathbf{C}_k$ on pixel $\mathbf{I}(i,j)$ as follows:

$$\mathbf{E}_k(\mathbf{I}(i,j)) = -\sum_{m=1}^{M} P_{k,m}(\mathbf{I}(i,j)) * log(P_{k,m}(\mathbf{I}(i,j))) \tag{1}$$

It can be seen that more confident in the prediction of a model is associated with lower entropy. For example, suppose a model returns a probability prediction $P_1 = [0.9, 0.05, 0.05]$, then the entropy would be $E_1 = 0.39$. Another method with prediction $P_2 = [0.35, 0.35, 0.3]$, which is less confident than the previous method i.e. the decision is difficult to get from the prediction of the second method, would have entropy $E_2 = 1.09$. Based on this observation, we define θ_k as the entropy threshold for $\mathcal{K}_k$. Only the predictions having entropy lower than the corresponding threshold are added into the ensemble. In this way, our approach takes into consideration the confidence of each segmentation model on each pixel:

$$\begin{cases} \mathbf{E}_k(\mathbf{I}(i,j)) < \theta_k : \mathbf{C}_k \text{ is selected} \\ \mathbf{E}_k(\mathbf{I}(i,j)) \geq \theta_k : \mathbf{C}_k \text{ is not selected} \end{cases} \tag{2}$$

The selected segmentation models will have their predictions combined via summation:

$$P_m^*(\mathbf{I}(i,j)) = \frac{\sum_{k=1}^{K} \mathbb{I}[\mathbf{E}_k(\mathbf{I}(i,j)) < \theta_k] P_{k,m}(\mathbf{I}(i,j))}{\sum_{k=1}^{K} \mathbb{I}[\mathbf{E}_k(\mathbf{I}(i,j)) < \theta_k])} \tag{3}$$

where $P_m^*(\mathbf{I}(i,j))$ is the combined prediction probability for class y_m and $\mathbb{I}[.]$ denotes the indicator function, which is equal to 1 if the condition inside the bracket is true, otherwise it is equal to 0. The class label associated with the maximum value among the combined probabilities is assigned to the pixel $\mathbf{I}(i,j)$:

$$\mathbf{I}(i,j) \in y_s \text{ if } s = argmax_{m=1,...,M} P_m^*(\mathbf{I}(i,j)) \tag{4}$$

In the proposed selection method in Equation 2, a particular subset of segmentation models is selected for each image based on the confidence in the prediction for that image. Thus, the different images will be predicted by different subsets of segmentation models.

We formulate an optimisation problem to find the optimal thresholds $\{\theta_k\}_{k=1}^{K}$ by exploring the ground-truth information of given training data. In this study, we apply the Stacking algorithm to generate the predictions of pixels in the training images [25]. The training set $\mathbf{D}$ is divided into T disjoint parts $\{\mathbf{D}_1, ..., \mathbf{D}_T\}$, where $\mathbf{D} = \mathbf{D}_1 \cup ... \cup \mathbf{D}_T, \mathbf{D}_i \cap \mathbf{D}_j = \emptyset (i \neq j), |\mathbf{D}_1| \approx ... \approx |\mathbf{D}_T|$, and their corresponding remainder $\{\tilde{\mathbf{D}}_1, ..., \tilde{\mathbf{D}}_T\}$ in which $\tilde{\mathbf{D}}_t = \mathbf{D} - \mathbf{D}_t$. Each segmentation algorithm $\mathcal{K}_k$ trains on $\tilde{\mathbf{D}}_t$ to obtain a model $\mathbf{C}_k^t$. Afterwards, $\mathbf{C}_k^t$ will segment each image in $\mathbf{D}_t$. For a pixel at (i,j) of image $\mathbf{I}$ in the training set $\mathbf{D}$, these models will output a probability vector $P_{k,m}(\mathbf{I}(i,j))$. The predictions for an image $\mathbf{I}$ is an $(W \times H) \times (M \times K)$ matrix $\mathbf{P}(\mathbf{I})$:

$$\mathbf{P}(\mathbf{I}) = \begin{bmatrix} P_{1,1}(\mathbf{I}(1,1)) & \cdots & P_{1,M}(\mathbf{I}(1,1)) & \cdots & P_{K,1}(\mathbf{I}(1,1)) & \cdots & P_{K,M}(\mathbf{I}(1,1)) \\ P_{1,1}(\mathbf{I}(1,2)) & \cdots & P_{1,M}(\mathbf{I}(1,2)) & \cdots & P_{K,1}(\mathbf{I}(1,2)) & \cdots & P_{K,M}(\mathbf{I}(1,2)) \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ P_{1,1}(\mathbf{I}(W,H)) & \cdots & P_{1,M}(\mathbf{I}(W,H)) & \cdots & P_{K,1}(\mathbf{I}(W,H)) & \cdots & P_{K,M}(\mathbf{I}(W,H)) \end{bmatrix} \tag{5}$$

The prediction for all N images in the training set $\mathbf{D}$ is given by a $(N \times W \times H) \times (M \times K)$ matrix:

$$\mathcal{P} = \begin{bmatrix} \mathbf{P}(\mathbf{I}_1) \\ \mathbf{P}(\mathbf{I}_2) \\ \cdots \\ \mathbf{P}(\mathbf{I}_N) \end{bmatrix} \tag{6}$$

Next we search for the optimal thresholds $\{\theta_k\}_{k=1}^{K}$ by optimising with respect to a fitness measure. In this study, we use Dice coefficient which is a popular measure to evaluate segmentation results [19]. Let **pred** and **ground** denote the final predictions and ground truths of all training pixels:

$$\textbf{pred} = \{pred_1, pred_2 ... pred_M\} \tag{7}$$

$$\textbf{ground} = \{ground_1, ground_2 ..., ground_M\} \tag{8}$$

in which $pred_m$ is the vector of size $(N \times W \times H, 1)$ with each element having a value of either 0 or 1 denoting whether the corresponding pixel is predicted to belong to class y_m. 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. belonging to $\{0, 1\}$. $ground_m$ is obtained from the ground truth $\{\mathbf{Y}_n\}$ while $pred_m$ is obtained based on Equation 3 and 4 for each row of $\mathcal{P}$. The Dice coefficient is calculated as follows:

$$DC = \frac{1}{M} \sum_{m=1}^{M} \frac{2 \times pred_m^T ground_m}{||pred_m||^2 + ||ground_m||^2} \tag{9}$$

We maximize the Dice coefficient to find the optimal $\{\theta_k\}_{k=1}^{K}$:

$$\begin{aligned} \max_{\{\theta_k\}_{k=1}^{K}} \quad & DC_{avg} \\ \text{s.t.} \quad & 0 \leq \theta_k \leq \log M (1 \leq k \leq K) \end{aligned} \tag{10}$$

where the inequality conditions come from the definition of entropy. We will maximize $\mathbf{D}$ given the ground truth labels **ground** and the probability predictions $\mathcal{P}$ (to calculate the crisp prediction **pred** based on $\{\theta_k\}_{k=1}^K$) of all pixels in the training images.

We use Comprehensive Learning PSO (CLPSO) [17], a variant of PSO which has been successfully applied to real-world problems in multiple works [35], [33], to solve this optimisation problem. Although PSO has attracted a high level of interest in many applications [32], the main deficiency of PSO is premature convergence [17]. Each particle in PSO only learns from its best position so far (*pbest*) and global best position (*gbest*) which makes it converge quickly. However, if the *gbest* gets trapped in a local optimum then other particles might be attracted to it, leading to premature convergence. [17] introduced Comprehensive Learning PSO (CLPSO) to mitigate this problem by having each particle learn from all particles' local best position. The authors compared CLPSO with eight PSO variants on 16 benchmark problems and found that the CLPSO makes use of the information in swarm more effectively to generate better quality solutions.

In CLPSO, the position $\{\theta_k\}_{k=1}^K$ of i^{th} particle will also be associated with a K-dimension vector $\mathbf{e}_i = (e_{1,i}, e_{2,i}, ..., e_{K,i})$ called *exemplar vector* for comprehensive learning. The exemplar vector is introduced for a particle to learn from the local best $(pbest)$ of itself as well as all the other particles. For example, a particle with the position $(0.13, 0.43, 0.22, 0.74, 0, 11)$, the velocity $(0.48, 0.25,\ 0.52, 0.13, 0.15)$, and the exemplar $(6, 8, 4, 8, 4)$, would learn/update the 3rd dimension position value based on the 3rd dimension position value of the 4th particle's $pbest$. A particle is assigned randomly with an exemplar vector at initialization. The exemplar will be updated after a number of iterations in which a particle's $pbest$ does not improve. In order to choose which particle to learn from for each dimension, two random particles are selected and the one with higher fitness value will be assigned as the exemplar for the updated particle on the corresponding dimension [17], [31]. Therefore, only one acceleration of constant c is needed. The updated equation for the velocity in the CLPSO is given by:

$$v_{k,i} \leftarrow a \times v_{k,i} + c \times r_1 \times (pbest_{k,e_{k,i}} - \theta_{k,i}) \tag{11}$$

in which a is the inertia weight which controls the velocity speeding rate, updated after each iteration according to the approach of [17], c is an acceleration constant used to control the learning rate of the exemplars' local best, $pbest_{k,e_{k,i}}$ is the k^{th} dimension of particle's best position referring to the k^{th} dimension of exemplar $\mathbf{e}_i$, and r_1 is a random number drawn from a uniform distribution over $[0, 1]$. The k-dimension of the i^{th} particle's position will be updated as follows:

$$\theta_{k,i} \leftarrow \theta_{k,i} + v_{k,i} \tag{12}$$

The pseudo-code of the training process of the proposed system is present in Algorithm 1. Firstly, **ground** is generated from $\mathbf{D}$ using Equation 7 and K segmentation algorithms $\{\mathcal{K}_k\}_{k=1}^K$ are first trained on $\mathbf{D}$ to create models $\{\mathbf{C}_k\}_{k=1}^K$.

Algorithm 1 Training process

Input: Training images $\mathbf{D}$, K segmentation algorithms $\{\mathcal{K}_k\}_{k=1}^K$, parameters for the CLPSO: maximum number of iteration $nIter$, number of candidates $nPop$, c, a.

Output: The optimal threshold $\{\hat{\theta}_k\}_{k=1}^K$, segmentation models $\{\mathbf{C}_k\}_{k=1}^K$

1: Generate **ground** from $\mathbf{D}$
2: Train K models $\{\mathbf{C}_k\}_{k=1}^K$ on $\mathbf{D}$ using $\{\mathcal{K}_k\}_{k=1}^K$
3: $\mathcal{P} = \emptyset$
4: $\mathbf{D} = \mathbf{D}_1 \cup ... \cup \mathbf{D}_T, \mathbf{D}_i \cap \mathbf{D}_j = \emptyset (i \neq j)$
5: **for** each $\mathbf{D}_t$ **do**
6: $\quad \tilde{\mathbf{D}}_t = \mathbf{D} - \mathbf{D}_t$
7: $\quad$ Train ensemble of segmentation models on $\tilde{\mathbf{D}}_t$ using $\{\mathcal{K}_k\}_{k=1}^K$
8: $\quad$ Segment images in $\mathbf{D}_t$ by these models
9: $\quad$ Add outputs on samples in $\mathbf{D}_t$ to $\mathcal{P}$ using Equation 6
10: Use the CLPSO method [17]: for each candidate $\{\theta_k\}_{k=1}^K$, compute the associated Dice coefficient using Algorithm 2
11: Select the optimal $\{\hat{\theta}_k\}_{k=1}^K$ with the best Dice coefficient
12: **return** $\{\hat{\theta}_k\}_{k=1}^K$, $\{\mathbf{C}_k\}_{k=1}^K$

Algorithm 2 Compute the Dice coefficient for each candidate generated in the CLPSO

Input: Candidate $\{\theta_k\}_{k=1}^K$, predictions $\mathcal{P}$, and **ground**.

Output: The Dice coefficient associated with $\{\theta_k\}_{k=1}^K$

1: **for** each row $\mathbf{I}_n(i, j)$ of $\mathcal{P}$ **do**
2: $\quad$ **for** $m \leftarrow 1$ to M **do**
3: $\quad\quad$ Compute $P_m^*(\mathbf{I}(i, j))$ by using Equation 3
4: $\quad$ Assign class label to $\mathbf{I}_n(i, j)$ by using Equation 4
5: Generate **pred**
6: Compute DC by Equation 9
7: **return** DC

Afterwards the prediction $\mathcal{P}$ for all pixels of training images are generated by using the Stacking algorithm (Step 3-9). Algorithm 2 is called in Algorithm 1 for each candidate $\{\theta_k\}_{k=1}^K$ generated in the CLPSO to calculate its associated Dice coefficient. In Algorithm 2, for each row of $\mathcal{P}$ i.e. the predictions of K algorithms for a pixel, the combined probabilities associated with the class labels are calculated by applying Equation 3 with the use of the candidate $\{\theta_k\}_{k=1}^K$ and then a class label for this pixel is assigned by using Equation 4. On the prediction result for all pixels of $\mathcal{P}$, the final predictions **pred** can be obtained in the form of crisp labels, then the Dice coefficient can be calculated. The CLPSO runs until it reaches the number of iterations. From the last generation, the candidate $\{\hat{\theta}_k\}_{k=1}^K$ which is associated with the best Dice coefficient is selected as the final solution.

The segmentation process for a test image is described in Algorithm 3. Given an unsegmented image $\mathbf{I}$, we first obtain the predictions $\mathbf{P}(\mathbf{I})$ for all pixels of $\mathbf{I}$ by using the $\{\mathbf{C}_k\}_{k=1}^K$ (Step 1). The M combined probabilities of each pixel then are calculated by using the optimal weight $\{\hat{\theta}_k\}_{k=1}^K$ and the predictions (Step 3-4). Equation 4 is applied to these combined probabilities of this pixel to give the final prediction (Step 5). The predictions for all pixels of $\mathbf{I}$ constitute its segmentation result.

Algorithm 3 Segmentation process

Input: Unsegmented image $\mathbf{I}$, the optimal weights $\{\hat{\theta}_k\}_{k=1}^{K}$ and $\{\mathbf{C}_k\}_{k=1}^{K}$

Output: Segmented result for $\mathbf{I}$

Obtain the prediction $\mathbf{P}(\mathbf{I})$ by using $\{\mathbf{C}_k\}_{k=1}^{K}$
for each pixel of $\mathbf{I}$ **do**
  for $m \leftarrow 1$ to M **do**
    Compute $P_m^*(\mathbf{I}(i,j))$ by using Equation 3 with $\{\hat{\theta}_k\}_{k=1}^{K}$
  Assign label to $\mathbf{I}(i,j)$ by using Equation 4
return Segmented result for $\mathbf{I}$

IV. Experimental Studies

A. Datasets and Performance Metrics

In this experiment, we used three popular deep learning-based segmentation methods UNet [28], LinkNet [5] and Feature Pyramid Network (FPN) [18] with three backbone VGG16 [30], ResNet34 and ResNet101 [11] to create an ensemble of $K = 9$ segmentation algorithms. Thus we need to search 9 real number thresholds $\{\theta_k\}_{k=1}^{9}$, $0 \leq \theta_k \leq \log M$, $(1 \leq k \leq 9)$ for these 9 algorithms. These backbones were pretrained on the ImageNet dataset [10]. All segmentation models were run for 300 epochs. The 5-fold cross-validation was used in the experiments and was run using GPU. For the CLPSO algorithm, the iteration was set to 500 and the number of candidates $nPop$ was set to 10 based on [9]. Two performance metrics were used for the evaluation of the segmentation models and the proposed ensemble: Dice coefficient and Hausdorff distance. Dice coefficient, defined in Equation 9, is one of the most popular metrics for medical image segmentation. However, the Dice coefficient measures total volume difference, without taking into account local contours discrepancies [14]. Therefore, we also used another performance metric which measures based on geometrical contour for the performance evaluation. Let GT_m and PR_m be the set of coordinate vectors of the ground truth and prediction contour with respect to class y_m respectively. The Hausdorff distance is defined as follows:

$$HD = \frac{1}{M}\sum_{m=1}^{M} max(d(GT_m, PR_m), d(PR_m, GT_m)) \quad (13)$$

where $d(A,B)$ is the directed Hausdorff distance:

$$d(A,B) = \frac{1}{|A|}\sum_{a\in A}\min_{b\in B}||a-b|| \quad (14)$$

It is noted that the low Hausdorff Distance or high Dice coefficient shows the good segmentation result. We compared the proposed ensemble with 9 segmentation models and two benchmarks algorithms: Weighted ensemble of deep learning segmentation models using CLPSO [9] (denoted as WE-CLPSO), and ensemble with Decision Template combining using 9 segmentation models (denoted DT-9).

Three datasets were used in the experiments namely BUSI-19, EAD-19 and Red Lesion dataset. BUSI-19 is a dataset of breast ultrasound images from women aged 25-75 [1], containing 780 images and four classes: normal, benign, malignant and background. Endoscopy Artefact Detection (EAD-19) [2] is a dataset created to address the problem of detection of artefacts in video endoscopy. There are 475 images and six class labels: specularity, saturation, artifact, bubbles, instrument and background. Red Lesion dataset [7] is a dataset which contains 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 Dice coefficients results by the benchmark algorithms and the proposed ensemble. It can be seen that the proposed ensemble obtains better results compared to the benchmark algorithms on all three datasets. For the BUSI-19 dataset, the scores of the 9 segmentation models range from 0.62180 (LinkNet-VGG16) to 0.76996 (FPN-ResNet34). In contrast, the proposed ensemble obtains a Dice score of 0.77317, which is higher than the best among the 9 segmentation models by 0.3%, followed by WE-CLPSO at 0.77150 while DT-9 is only at 0.76821. With respect to the EAD-19 dataset, the proposed ensemble attain a Dice coefficient of 0.67948 which is higher than both DT-9 and WE-CLPSO by 4.92% and 1.59% respectively. The best of the 9 segmentation models (FPN-ResNet34) obtain only 0.65705 which is lower than the proposed ensemble by 2.24%. For the Red Lesion dataset, the proposed ensemble also attains the highest result at 0.96569 followed by 0.96411 (WE-CLPSO) and 0.96324 (FPN-ResNet34) while the score of DT-9 is only 0.96136, while the remaining benchmark algorithms obtain slightly lower scores compared to the proposed ensemble.

Table II shows the Hausdorff results of the proposed ensemble and the benchmark algorithms. Overall, the proposed ensemble attains the best results on two out of three datasets and achieve better results compared to the two benchmark algorithms, WE-CLPSO and DT-9 on the remaining dataset. For the BUSI-19 dataset, the proposed ensemble obtains a Hausdorff score of 24.83183 which is better than both WE-CLPSO (25.03102) and DT-9 (30.06768). However, the best Hausdorff score for this dataset is 24.504 by FPN-ResNet101 while the scores of the remaining benchmark algorithms range from more than 24.77 (FPN-ResNet34) to over 69 (UNet-VGG16). For the EAD-19 dataset, the proposed ensemble achieves the best Hausdorff distance of 50.21533, which is better than WE-CLPSO by a margin of 3.683, while DT-9 is among the worst performing benchmark algorithm with respect to Hausdorff score at 72.49579, better only than UNet-ResNet101 and FPN-ResNet101. The proposed ensemble also obtains the best result on the Red Lesion dataset at 9.64297, which is better than WE-CLPSO and DT-9 by 0.42 and 0.957 respectively. The results in Table I and II demonstrate the outperformance of the proposed ensemble compared to 9 segmentation models, a traditional combining method namely Decision Template, a weighted ensemble system using CLPSO namely WE-CLPSO.

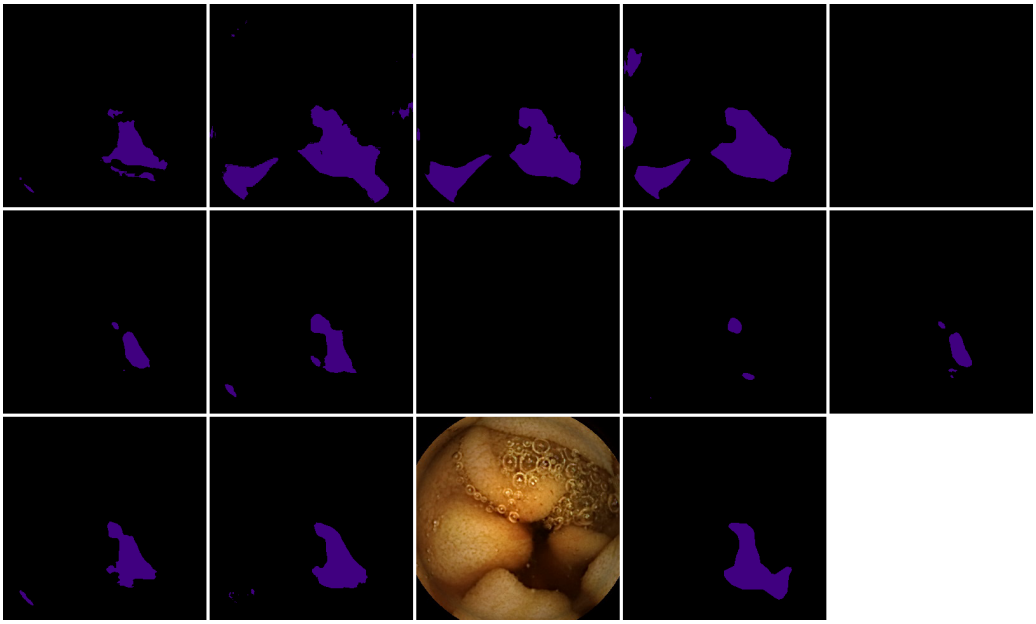
Figure 1 shows an example of the predictions by the proposed method and the benchmark algorithms for the Red

TABLE I
RESULT OF DICE COEFFICIENTS

Algorithm	BUSI-19	EAD-19	Red Lesion
UNet-VGG16	0.66107	0.60122	0.95826
LinkNet-VGG16	0.62180	0.53730	0.93427
FPN-VGG16	0.69506	0.52466	0.95215
UNet-ResNet34	0.67841	0.65588	0.96074
LinkNet-ResNet34	0.76164	0.63333	0.96048
FPN-ResNet34	0.76996	0.65705	0.96324
UNet-ResNet101	0.68962	0.54014	0.94494
LinkNet-ResNet101	0.71759	0.51021	0.94739
FPN-ResNet101	0.74456	0.51892	0.95250
WE-CLPSO	0.77150	0.66353	0.96411
DT-9	0.76821	0.63033	0.96136
Proposed ensemble	**0.77317**	**0.67948**	**0.96569**

TABLE II
RESULT OF HAUSDORFF DISTANCE

Algorithm	BUSI-19	EAD-19	Red Lesion
UNet-VGG16	69.80111	80.10919	12.50472
LinkNet-VGG16	56.44486	92.75078	28.28141
FPN-VGG16	38.80502	59.35158	14.90428
UNet-ResNet34	30.23913	56.65400	11.51804
LinkNet-ResNet34	28.49669	60.54548	10.92510
FPN-ResNet34	24.77195	53.31877	10.35638
UNet-ResNet101	42.67041	73.76465	16.06742
LinkNet-ResNet101	29.08265	85.97052	14.23599
FPN-ResNet101	**24.5040**	75.33468	14.22944
WE-CLPSO	25.03102	53.89836	10.06164
DT-9	30.06768	72.49579	10.59985
Proposed ensemble	24.83183	**50.21533**	**9.64297**



From left to right, top to bottom: UNet-VGG16, LinkNet-VGG16, FPN-VGG16, UNet-ResNet34, LinkNet-ResNet34, FPN-ResNet34, UNet-ResNet101, LinkNet-ResNet101, FPN-ResNet101, WE-CLPSO, DT-9, proposed ensemble, test image, and ground truth.

Fig. 1. Example result from Red Lesion dataset.

Lesion dataset (indigo color denotes red lesion). It can be seen that LinkNet-VGG16, FPN-VGG16 and Unet-ResNet34 wrongly predict a large area on the bottom left as red lesion, while the predictions for the bottom right area is too large compared to the ground truth. The result by UNet-VGG16 contains some small separate areas around the main red lesion area which is not present in the ground truth. FPN-ResNet34 and UNet-ResNet101 only manage to predict a small area of red lesion in the bottom right compared to the ground truth, while LinkNet-ResNet34, LinkNet-Resnet101 and FPN-ResNet101 fail to predict the red lesion area altogether. The prediction by WE-CLPSO is only a small area compared to the ground truth with several small adjacent dots. The predicted area by DT-9 is larger but contains many deformations and there is a sizable area in the bottom left which is segmented as red lesion. In contrast, the shape of the predicted red lesion by the proposed ensemble agrees more with the ground truth and there is just some dots on the bottom left area compared to DT-9.

To show the effectiveness of using CLPSO, we compare the Dice and Hausdorff results of CLPSO and PSO on the experimental datasets (Figure 2). With respect to Dice coefficient, on BUSI-19 the Dice result when CLPSO was used is 0.77317 which is higher by 1.53% compared to the result associated with PSO. In contrast, on both EAD-19 and Red Lesion the Dice scores obtained by CLPSO and PSO are similar, at around 0.679 and 0.96 respectively. On both BUSI-19 and Red Lesion datasets, the Hausdorff scores associated with CLPSO (24.8318 and 9.64) are better than those obtained by using PSO (26.2664 and 10.2575). Finally, the Hausdorff distance on EAD-19 when CLPSO was used is 50.21 which is slightly higher than the score associated with PSO by 0.27. It can be seen the proposed ensemble obtains better result when CLPSO is used instead of PSO.

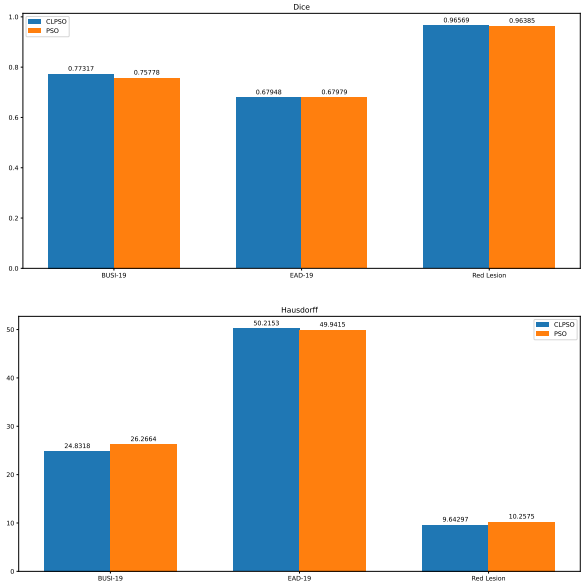


Fig. 2. Results of the proposed ensemble using CLPSO and PSO with respect to Dice (top) and Hausdorff (bottom) scores

Table III shows the optimal thresholds found by the proposed ensemble for the three datasets. For BUSI-19 dataset, the optimal threshold values are high for the ResNet-based models, with most having values from around 0.64 to around 1.07 and lower for the VGG16-based models (just around 0.001 to 0.004). This indicates that the ResNet-based models have more chance to be selected in the ensemble for this dataset. Meanwhile, for the EAD-19 and the Red Lesion datasets, the thresholds have high values mostly on the ResNet34-based models and FPN-VGG16, while the optimal thresholds for the remaining models are just around 0.002. Meanwhile, the computational time of the proposed ensemble on BUSI-19, EAD-19 and Red Lesion were 40.21, 59.15 and 60.82 hours for both training and testing process while these numbers of WE-CLPSO, the weight-based ensemble optimisation algorithm, are 42.54, 67.43 and 65.98 hours respectively. The results show that the required time for our proposed ensemble is slightly smaller than WE-CLPSO by around 2.33, 6.61 and 5.16 hours on BUSI-19, EAD-19 and Red Lesion respectively. This demonstrates that the computational time of our proposed ensemble is competitive to that of WE-CLPSO.

V. Conclusion

In this paper, we presented a selection approach for an ensemble of medical image segmentation algorithms. Our approach takes into consideration the fact that the presence of some segmentation algorithms might degrade ensemble performance, thus needing to be removed from the ensemble. We introduced a novel ensemble selection method which is based on the idea of measuring uncertainty in the prediction of each model with respect to its level of expertise. If the uncertainty is below its associate threshold, the prediction is confident and it is selected to calculate the combined prediction. Shannon entropy is used as the uncertainty measure. The optimal entropy threshold for each segmentation algorithm is found by using CLPSO, a swarm intelligence algorithm. Dice coefficient, which is a popular performance metric for image segmentation, is used as the fitness criteria. Our experiments on three medical image segmentation datasets showed that the proposed ensemble provides better results compared to not only the 9 single segmentation models but also two selected benchmark segmentation algorithms. It is noted that even though our proposed ensemble is competitive to other ensemble methods such as WE-CLPSO, the required computation time for training is still high. In the future, we will reduce the training time of the proposed ensemble by using some techniques such as surrogate models [8].

TABLE III
Entropy thresholds found by the proposed ensemble

Algorithm	BUSI-19	EAD-19	Red Lesion
UNet-VGG16	0.00181	0.052	0.001
LinkNet-VGG16	0.00357	0.005	0.0
FPN-VGG16	0.00449	0.177	0.636
UNet-ResNet34	0.71992	1.772	0.359
LinkNet-ResNet34	1.07032	0.816	0.47
FPN-ResNet34	0.82227	1.51	0.693
UNet-ResNet101	0.64662	0.002	0.002
LinkNet-ResNet101	0.99824	0.002	0.002
FPN-ResNet101	0.00327	0.0	0.0

References

[1] W. Al-Dhabyani, M. Gomaa, H. Khaled, A. Fahmy, Dataset of breast ultrasound images. Data in Brief, **28**, 104863 (2020).

[2] S. Ali, F. Zhou, C. Daul et al., Endoscopy artifact detection (EAD2019) challenge dataset. CoRR abs/1905.03209 (2019).

[3] A.S. Britto, R. Sabourin, L.E.S Oliveira, Dynamic selection of classifiers - A comprehensive review. Pattern Recognit. 47(11), 3665–3680 (2014).

[4] A. Casamitjana, M. Catà, I. Sánchez, et al., Cascaded V-Net Using ROI Masks for Brain Tumor Segmentation. In: Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. pp. 381-391 (2018).

[5] A. Chaurasia, E. Culurciello, LinkNet: Exploiting encoder representations for efficient semantic segmentation, in IEEE Visual Communications and Image Processing, 2017, pp. 1-4.

[6] Y. Chen, M.L. Wong, H. Li, Applying Ant Colony Optimization to configuring stacking ensembles for data mining. Expert Syst. Appl. 41(6), pp. 2688–2702 (2014).

[7] P. Coelho, A. Pereira, A. Leite et al., A deep learning approach for red lesions detection in video capsule endoscopies. In: Int. Conf. Image Process. Comput. Vis. Pattern Recognit., pp. 553–561 (2018).

[8] T. Dang, A.V. Luong, A.W.C. Liew, J. McCall, T.T. Nguyen, Ensemble of deep learning models with surrogate-based optimization for medical image segmentation. In: 2022 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8 (2022).

[9] T. Dang, T. T. Nguyen, C. Francisco Moreno-García, E. Elyan, J. McCall, Weighted Ensemble of Deep Learning Models based on Comprehensive Learning Particle Swarm Optimization for Medical Image Segmentation. In: 2021 IEEE Congress on Evolutionary Computation (CEC), pp. 744-751 (2021).

[10] J. Deng, W. Dong, R. Socher, L. -J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248-255 (2009).

[11] K. He, X. Zhang, S. Ren et al., Deep residual learning for image recognition. In: Proceedings of CVPR. pp. 770–778. IEEE (2016).

[12] S. Hwang, J. Oh, W. Tavanapong et al., Polyp detection in colonoscopy video using elliptical shape feature. In: IEEE Int. Conf. Image Process. vol. 2, pp. 465-468 (2007).

[13] J. Jue, H. Jason, T. Neelam et al., Integrating cross-modality hallucinated MRI with CT to aid mediastinal lung tumor segmentation. In: MICCAI. pp. 221–229 (2019).

[14] H. Kim, S. Park, S. Lo et al., Bidirectional local distance measure for comparing segmentations, in Medical Physics, 39 (11), pp. 6779–6790 (2012).

[15] A. Krizhevsky, S. Ilya, H. Geoffrey, ImageNet classification with deep convolutional neural networks, in Commun. ACM 60, 2017, pp. 84-90.

[16] L. Kuncheva, J. Bezdek, R. Duin, Decision templates for multiple classifer fusion. Pattern Recognit. 34, pp. 299–314 (2001).

[17] J.J. Liang, A.K. Qin, P. N. Suganthan, S. Baskar, Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions, IEEE Trans. on Evolutionary Computation. 10 (3), pp. 281-295 (2006).

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

[19] Q. Liu, X. Tang, D. Guo et al., Multi-class Gradient Harmonized Dice Loss with Application to Knee MR Image Segmentation, in MICCAI, pp. 86–94 (2019).

[20] D.D. Margineantu, T.G. Dietterich, Pruning adaptive boosting. In: Proceedings of ICML. p. 211–218 (1997).

[21] G. Martínez-Muñoz, A. Suárez, Aggregation ordering in bagging, pp. 258–263 (2004).

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

[23] T.T. Nguyen, A.W. Liew, M.T. Tran et al., A novel genetic algorithm approach for simultaneous feature and classifer selection in multi classifier system. In: IEEE Congr. Evol. Comput. (CEC). pp. 1698–1705 (2014).

[24] T.T. Nguyen, M.T. Dang, A.W.C. Liew et al., A weighted multiple classifier framework based on random projection, Information Sciences, 490, pp. 36-58 (2019).

[25] T.T. Nguyen, T.T.T. Nguyen, X.C. Pham et al., A novel combining classifier method based on Variational Inference, Pattern Recognition. 49, pp. 198-212 (2016).

[26] O. Oktay, J. Schlemper, L. Folgoc, et al., Attention U-Net: Learning where to look for the pancreas, CoRR abs/1804.03999 (2018).

[27] 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, pp. 1-8 (2020).

[28] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, in MICCAI, pp. 234–241 (2015).

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

[30] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556 (2015).

[31] B. Tran, B. Xue, M. Zhang, Variable-Length Particle Swarm Optimization for Feature Selection on High-Dimensional Classification, IEEE Trans Evo Comp. 23 (3) (2019), pp. 473-487.

[32] M.P Wachowiak, R. Smolikova, Y. Zheng et al., An approach to multimodal biomedical image registration utilizing particle swarm optimization. IEEE Trans. Evol. Comput. 8(3), pp. 289–301 (2004).

[33] J.J Wang, G.Y Liu, Saturated control design of a quadrotor with heterogeneous comprehensive learning particle swarm optimization. Swarm and Evolutionary Computation 46, pp. 84–96 (2019).

[34] R. Wang, T. Lei, R. Cui et al., Medical image segmentation using deep learning: A survey. IET Image Processing, 16(5), pp. 1243-1267 (2022).

[35] S. Yang, Y. Bao, Comprehensive learning particle swarm optimization enabled modeling framework for multi-step-ahead influenza prediction, Applied Soft Computing. 113 (2021) 107994.

[36] H. Zhu, Y. Fan, H. Lu et al., Improved curvature estimation for computer-aided detection of colonic polyps in CT colonography. Academic Radiology 18(8), pp. 1024–1034 (2011).