Two-layer ensemble of deep learning models for medical image segmentation.

DANG, T., NGUYEN, T.T., MCCALL, J., ELYAN, E. and MORENO-GARCÍA, C.F.

2024

Cognitive Computation (2024) 16:1141–1160
https://doi.org/10.1007/s12559-024-10257-5

Two-layer Ensemble of Deep Learning Models for Medical Image Segmentation

Truong Dang[1] · Tien Thanh Nguyen[1] · John McCall[1] · Eyad Elyan[1] · Carlos Francisco Moreno-García[1]

Received: 11 April 2023 / Accepted: 11 January 2024 / Published online: 31 January 2024

Abstract

One of the most important areas in medical image analysis is segmentation, in which raw image data is partitioned into structured and meaningful regions to gain further insights. By using Deep Neural Networks (DNN), AI-based automated segmentation algorithms can potentially assist physicians with more effective imaging-based diagnoses. However, since it is difficult to acquire high-quality ground truths for medical images and DNN hyperparameters require significant manual tuning, the results by DNN-based medical models might be limited. A potential solution is to combine multiple DNN models using ensemble learning. We propose a two-layer ensemble of deep learning models in which the prediction of each training image pixel made by each model in the first layer is used as the augmented data of the training image for the second layer of the ensemble. The prediction of the second layer is then combined by using a weight-based scheme which is found by solving linear regression problems. To the best of our knowledge, our paper is the first work which proposes a two-layer ensemble of deep learning models with an augmented data technique in medical image segmentation. Experiments conducted on five different medical image datasets for diverse segmentation tasks show that proposed method achieves better results in terms of several performance metrics compared to some well-known benchmark algorithms. Our proposed two-layer ensemble of deep learning models for segmentation of medical images shows effectiveness compared to several benchmark algorithms. The research can be expanded in several directions like image classification.

Keywords Image segmentation · Ensemble method · Ensemble learning · Deep learning · Medical image

Introduction

Medical image analysis refers to the science of examining visual representations acquired in clinical practice to help radiologists and clinicians with more efficient decision-making and treatment processes. One of the most important areas in medical image analysis is segmentation, in which raw image data is partitioned into structured and meaningful regions to gain further insights such as anatomy research, disease diagnosis, treatment planning, and prognosis monitoring [1]. With the global advancements in imaging techniques, the volume of medical image data is increasing substantially, which puts increased stress on the limited number of medical professionals [2]. In order to efficiently handle this ever-growing amount of data and exploit its rich information, Artificial Intelligence (AI) has been considered one of the most prominent solutions, promising to revolutionise medical research and practices. AI refers to computer algorithms that can perform human-level tasks, and AI-based automated segmentation algorithms can potentially assist physicians with more effective imaging-based diagnoses. Before the rise of deep learning, there have been many works on medical image segmentation [3, 4]. However, these works relied on creating handcrafted features which are time-consuming and it is more difficult to extract discriminating features from medical images compared to RGB images, due to various noises, blur, and low contrast, among others [5].

In recent years, the field of deep learning has witnessed many successes, especially with Deep Neural Networks (DNNs) in many areas such as computer vision [6] or natural language processing [7]. An important advantage of deep learning compared to traditional machine learning techniques is its ability to automatically learn the representation of the data with multiple levels of abstraction [7], which relieves the practitioners from having to construct

✉ Tien Thanh Nguyen
t.nguyen11@rgu.ac.uk

[1] School of Computing, Robert Gordon University, Aberdeen, UK

handcrafted features. DNNs have also been widely used for medical image segmentation and have shown encouraging results. Popular medical segmentation tasks which have seen successful applications of deep learning include liver segmentation [8], brain-tumour segmentation [9], cardiac image segmentation [10], polyp segmentation [11], etc.

However, unlike in other fields where images are widely available, it is difficult to acquire high-quality ground truths for medical images due to the high-level medical knowledge required. For comparison, it is known that ImageNet, one of the most popular datasets, contains one million annotated images [6], while most medical image datasets have only around 1,000 instances [12]. The lack of labeled and high-quality data is an important obstacle in the application of deep learning to medical image analysis tasks [1]. Moreover, it is known that DNN hyperparameters are usually found via trial-and-error and the optimal set of hyperparameters is very difficult to find. A potential approach to solve these problems is to combine the results of multiple deep learning models to achieve better predictions.

Ensemble learning is a technique in which multiple classifiers are combined to make a collaborated decision. By combining the predictions of multiple classifiers, the poor results of some classifiers are likely to be compensated by more well-performing ones. Many studies have shown that ensemble learning can achieve much better results compared to just using a single classifier [13]. Ensemble learning has been applied in many areas, such as computer vision [14] and bioinformatics [15]. In recent years, the medical image analysis community has also applied ensemble learning to improve the results of deep learning models [14, 16–18].

Recently, Zhou et al. [19] noted that the success of deep learning was due to layer-by-layer processing and feature transformation between layers. Based on this observation, the authors proposed a deep ensemble of random forests and completely random trees. The output of each layer is used as the input to the next layer. Nguyen et al. [20] proposed a heterogeneous multi-layer ensemble learning framework in which each layer contains several different classifiers generated by training different learning algorithms on the layer input data. Considering the critical nature of medical applications, it is necessary to leverage the power of ensemble learning, particularly the multi-layer ensemble framework on deep learning for medical segmentation to achieve optimal results.

In this paper, we propose a novel two-layer ensemble of deep medical segmentation algorithms which achieves competitive results compared to benchmark methods. In the first layer, each model performs a prediction on each pixel, and then these predictions are used as additional channels of the training image for the second layer of the ensemble. This potentially increases the discriminative capability of the ensemble. The second layer's prediction will then be combined via a weight-based scheme. To the best of our knowledge, our paper is the first work which proposes a two-layer ensemble for deep learning-based medical image segmentation via augmenting the input images using the predictions of the first layer as additional channels in the second layer. Our proposed two-layer ensemble is a general model and can be potentially extended to other deep learning tasks as well. Our contributions are as follows:

- We propose a heterogeneous ensemble of deep segmentation models for the medical image segmentation problem.
- We propose to use the predictions of each deep segmentation model in the ensemble as additional channels of the original training image to create the training data for the second layer. A second layer of the ensemble will use this new training data as input to perform the predictions.
- We propose a weight-based scheme for the combination of predictions in the second layer. The weights are found by solving linear regression problems based on the relationship between the predictions and ground truth labels of training observations.
- Experiments conducted on five medical image segmentation datasets demonstrate the effectiveness of our approach.

The paper is organised as follows. In "Background and Related Works" section, we briefly review the existing approaches relating to segmentation in medical image analysis and ensemble learning. The proposed ensemble is introduced in "Proposed Ensemble" section. The details of experimental studies on five medical image segmentation datasets are described in "Experimental Details" section. Finally, the conclusion is given in "Conclusion" section.

Background and Related Works

Deep Learning for Medical Image Segmentation

With the success of [6] in applying DNNs to the problem of image classification, deep learning has become the most popular approach in computer vision. Since then, many notable deep architectures have been proposed to solve vision problems. For example, VGG16 [21] was a deep CNN for image classification using a stack of convolution layers with small receptive fields in the first layers instead of a few layers with big receptive fields like previous models. This allows the model to have much fewer parameters and more non-linearity, which makes the decision function more discriminative and the model easier to train. VGG16 managed to achieve a top-5 accuracy of 92.7% on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC)-2013 dataset. Another notable model is ResNet [21], which was motivated by the problem of training a really deep architecture. The network uses shortcut connections for identity mapping, i.e. instead of learning a function, the layers having shortcut connections learn the residual mapping. This allows ResNet to have a very deep network at 152 layers while achieving 96.4% accuracy in the ILSVRC-2016 competition.

Most deep learning-based segmentation architectures are inspired by Fully Convolutional Network (FCN) [22], which creates a segmentation network by using an existing classification network and replacing the fully connected layers with convolutional ones to output spatial maps instead of classification scores. Those maps are then upsampled to produce dense pixel-level output. The design of FCN has influenced many popular deep learning architectures for segmentation [23]. Another notable example is DeepLab [24] which makes use of Conditional Random Fields (CRF) [25] as a post-processing step for the refinement of the segmentation result. The proposed architecture models each pixel as a node in the random field and employs a fully connected factor graph in which one pairwise term is used for each pixel pair irrespective of their distance. This allows the model to incorporate both short-range and long-range information into account, facilitating the restoration of detailed structures in the segmentation process that were lost due to the spatial invariance of CNN.

Reliable computer-assisted segmentation of anatomical structures in medical images is considered very important for the diagnosis and monitoring of diseases. This has motivated considerable research efforts in applying deep learning to medical image segmentation. Compared to image segmentation in general, each human organ has its specific challenge concerning medical segmentation. For example, the segmentation area of the brain and lung required for diagnosis is relatively large, while blood vessels require higher segmentation accuracy [26]. This has led to the development of new deep learning architectures, such as UNet, which was developed to overcome the limits of previous architectures due to the loss of spatial resolution in the case of small or irregular objects [27]. Due to its excellent performance, UNet has been widely used in medical image segmentation and other areas of computer vision with many variants [26]. UNet consists of a contracting path and an expanding path designed symmetrically. To help with localization, high-resolution features from the contracting path are combined with the upsampled output. An important difference between UNet compared to previous architectures is that the upsampling part also has a large number of feature channels, which allow the network to propagate context information to higher resolution layers. This makes UNet suitable for medical image segmentation and UNet is a winner of the ISBI 2015 bioimage segmentation challenge. Other notable examples are LinkNet [28] which takes the sum of the upsampled output and the corresponding features in the convolutional path, and Feature Pyramid Network (FPN) [29] which uses the concatenation of features of all levels in the upsampling part to help with the final prediction. V-Net, which is a 3D extension of UNet, was proposed in [30]. A cascade of V-Net for brain tumour segmentation was proposed in [31] to segment each region separately before combining the results. Another novel deep learning-based image segmentation model is UNet + + [32], a variant of the popular UNet architecture. Unlike UNet, which uses plain skip connections in its architecture, UNet + + consists of an encoder and a decoder which are connected through a series of nested dense convolutional blocks, effectively reducing the semantic gap between the feature maps of the encoder and decoder. UNet + + has achieved good results on many benchmark datasets in recent years and has become an algorithm of choice in medical image segmentation. Nie et al. [33] used 3D FCN to integrate contextual information and features of different scales to segment multimodal infant brain MRI images. Zhang et al. [34] noted that retinal vessel segmentation is challenging due to various imaging conditions, low image contrast, and the presence of pathologies. The authors proposed an edge-sensing mechanism to add additional boundary labels to segment blood vessels and achieved competitive results on three retinal segmentation datasets. Jue et al. [35] proposed cross-modality educed deep learning segmentation (CMEDL), which combines CT and pseudo-MR images produced from CT to improve chest X-ray segmentation. The study in [36] proposed SSNet to handle spatial variations for MRI spleen segmentation by integrating a variant of Generative Adversarial Network (GAN) [37] to create synthetic spleen labels to improve predictions.

Ensemble Learning

Ensemble learning is a popular approach in machine learning for combining a collection of classifiers for a collaborative decision. Designing an ensemble system requires two stages, namely ensemble generation, and ensemble integration. In the ensemble generation, multiple classifiers are generated by using either a homogeneous strategy (training a learning algorithm on multiple training sets generated from the original training data) [37, 38] or a heterogeneous strategy (training different learning algorithms on the original training data) [39–41]. A combining method is then used to aggregate the predictions of the constituent classifiers in the ensemble integration stage to obtain the collaborated prediction.

There have been many applications of ensemble of DNNs to medical image segmentation. For example, [16] used an ensemble of 2D and 3D segmentation models with a meta-learner to segment 3D cardiac MRI data, while [42] used an ensemble of 5 CNNs for brain MRI lesion segmentation. In [14], the authors proposed a bagging ensemble of deep segmentation models to train multiple UNets to segment dense nuclei pathological images. [17] used Dirichlet distribution and Mahalanobis distance to learn dynamic weights for an ensemble of deep learning models with online ensemble learning and achieved good results on 2 out of 4 medical datasets. [43] used a pre-trained CNN to extract features that were used to train an ensemble of classifiers to demonstrate competitive results on the ImageCLEF 2016 medical image public dataset. In [18], the authors proposed a two-stage selective ensemble of medical image classifiers based on accuracy and diversity criteria. Dang et. al. proposed a weighted ensemble of deep learning-based segmentation algorithms for cardiographic segmentation and achieved competitive results in the CAMUS competition [44].

Recently, there has been increasing interest in the ensemble generation inspired by the success of DNNs. 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 and Feng [19] (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. Utkin et al. [45] proposed a weighted average approach for gcForest by associating each tree with a weighted vector for its class distribution vector. The optimal weight vectors of each tree in one layer are found by minimising the distance between the class label vector in a binary encoding scheme and the weighted prediction vector of this forest. The authors proposed to set only a weight vector for each group to reduce the computational overhead. Nguyen et al. [20] 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 optimisation problem in which the two objectives to be maximised are classification accuracy and diversity of the ensemble in each layer.

Proposed Ensemble

Our proposed method is inspired by multi-layer ensemble learning architectures, in which the segmentation algorithms in one layer train the segmentation model of that layer, while also generating the new training data generated by the preceding layer [19]. This facilitates the successive refinement of medical image segmentation results through each layer. It is recognised that the most successful segmentation algorithms in recent years have been based on DNNs [46], and even though deep learning models can be trained in parallel using a Graphics Processing Unit (GPU), a multi-layer ensemble model of deep learning-based segmentation algorithms would require a lot of computational resources. Therefore, an important question arises: How many layers should a deep ensemble model extend? In [20], the authors showed that on some datasets, the number of layers obtained on ensembles was two or three only. Based on this observation, we introduce a novel two-layer ensemble model for the segmentation of medical images. Figure 1 shows the high-level overview of our proposed ensemble.

Two-layer Ensemble for Segmentation

Let $\mathbf{D} = \{\mathbf{I}_n, \mathbf{Y}_n\}_{n=1}^{N}$ be the training set where N is the number of images, $\mathbf{I}_n$ is an input image of size (H, W, C) in which H is the image height, W is the image width, and C is the number of channels ($C = 1$ for grayscale, $C = 3$ for colour images). The mask $\mathbf{Y}_n$ is also an image of size (W, H), in which each entry shows which group the pixel belongs to, i.e. $\mathbf{Y}_n(i,j) \in \mathcal{Y}$ where $\mathcal{Y} = \left\{y_m\right\}, m = 1, \dots M$ is the set of all classes and M is the number of classes.

We aim to learn a hypothesis $\mathbf{h} : \mathbf{I}_n \rightarrow \mathbf{Y}_n$ (i.e. segmentation model) to approximate the unknown relationship between each image and its corresponding mask, and then use this hypothesis to assign a label for each unsegmented image. We also denote $\{\mathcal{K}_k\}_{k=1}^{K}$ by the set of K segmentation algorithms. Each segmentation algorithm $\mathcal{K}_k$ learns from the dataset $\mathbf{D}$ to obtain a trained segmentation model $\mathbf{h}_k$. In ensemble learning, we train K segmentation algorithms $\{\mathcal{K}_k\}_{k=1}^{K}$ on $\mathbf{D}$ to get K segmentation models $\{\mathbf{h}_k\}_{k=1}^{K}$.

In the next step, we generate the training data for the second layer of the ensemble. Based on the results of [20] and the stacking generalisation model [40], we propose a two-layer deep ensemble architecture for segmentation in medical image analysis (Fig. 1). Firstly, the training set $\mathbf{D}$ is divided into T disjointed parts $\{\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_T\}$, where $\mathbf{D} = \mathbf{D}_1 \cup \mathbf{D}_2 \cup \cdots \cup \mathbf{D}_T, \mathbf{D}_{t_i} \cap \mathbf{D}_{t_j} = \emptyset t_i, t_j = 1, \dots, T, t_i \neq t_j$. Then for each part $\mathbf{D}_t (t = 1, \dots, T)$, the segmentation algorithms $\{\mathcal{K}_k\}_{k=1}^{K}$ will learn on its complimentary $\mathbf{D} \backslash \mathbf{D}_t$ to obtain segmentation models $\mathbf{h}_{k,t}$. The images in $\mathbf{D}_t$ are then segmented by using these segmentation models. Let $P_k(y_m | \mathbf{I}_n(i,j))$ be probability prediction that $\mathbf{h}_{k,t}$ assigns pixel $\mathbf{I}_n(i,j)$ to be in class y_m. The prediction of $\mathbf{h}_{k,t}$ showing the probability all pixels of the image $\mathbf{I}_n$ belonged to class y_m is given by a matrix:

$$\mathbf{P}_k\left(y_m | \mathbf{I}_n\right) = \begin{bmatrix} P_k(y_m | \mathbf{I}_n(1,1)) & P_k(y_m | \mathbf{I}_n(1,2)) & \dots & P_k(y_m | \mathbf{I}_n(1,H)) \\ \dots & \dots & \dots & \dots \\ P_k(y_m | \mathbf{I}_n(W,1)) & P_k(y_m | \mathbf{I}_n(W,2)) & \dots & P_k(y_m | \mathbf{I}_n(W,H)) \end{bmatrix} \quad (1)$$

For each image $\mathbf{I}_n$ there will be $M \times K$ prediction matrices $\mathbf{P}_k\left(y_m | \mathbf{I}_n\right)$. An example is shown in Fig. 2, in which two segmentation models, UNet-VGG16 and LinkNet-ResNet34, predict an image of the CAMUS datasets for three classes. In this case $M = 3, K = 2$, and the first three images on the left show

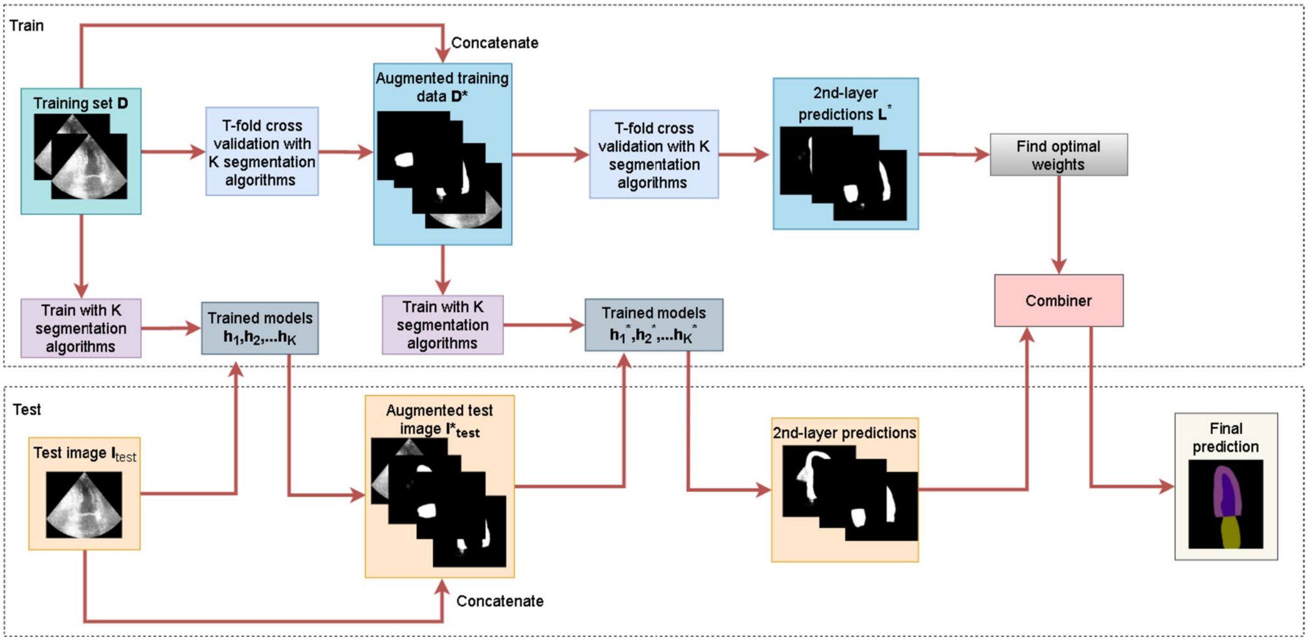
Train
Concatenate
Training set **D**
T-fold cross validation with K segmentation algorithms
Augmented training data **D***
T-fold cross validation with K segmentation algorithms
2nd-layer predictions **L***
Find optimal weights
Train with K segmentation algorithms
Trained models h1,h2,...hK
Train with K segmentation algorithms
Trained models h1*,h2*,...hK*
Combiner
Test
Test image I_test
Augmented test image I*test
2nd-layer predictions
Final prediction
Concatenate

Fig. 1 High-level overview of the proposed ensemble

the predictions by UNet-VGG16 for three classes and the images on the right show the predictions by LinkNet-ResNet34. The predictions have been multiplied by 255 for visualisation.

In this study, we propose to augment the training data for the second layer of the ensemble by concatenating these $M \times K$ prediction matrices to the original training images to create new images $\mathbf{I}_n^*$. The prediction matrix $\{\mathbf{P}_k(y_m|\mathbf{I}_n)\}$ serves as an additional channel of the original image$\mathbf{I}_n$. In total, the new images $\mathbf{I}_n^*$ will have $C + M \times K$ channels:

$$\mathbf{I}_n^* = \mathbf{I}_n \cup \{\mathbf{P}_k(y_m|\mathbf{I}_n)\}, k = 1, \dots, K; m = 1, \dots, M \tag{2}$$

The new training data for the second layer of the ensemble will be given as follows:

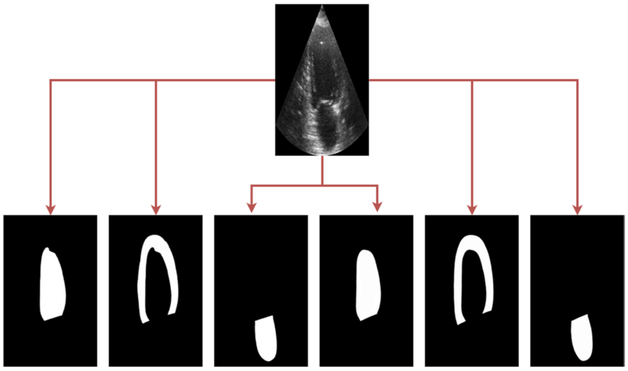

Fig. 2 Example of prediction results on CAMUS dataset. Top: Original image. The bottom is the predictions for the Left ventricle, Myocardium, and Left atrium classes, made by UNet and LinkNet with backbones ResNet34 and VGG16, respectively. The result has been multiplied by 255 for visualization

$$\mathbf{D}^* = \{\mathbf{I}_n^*, \mathbf{Y}_n\}, n = 1, \dots, N \tag{3}$$

For the second layer of the ensemble, we train $\{\mathcal{K}_k\}_{k=1}^K$ on $\mathbf{D}^*$ to get trained segmentation models $\{\mathbf{h}_k^*\}_{k=1}^K$. We then train a combiner $\mathbf{C}$ to merge the trained models $\mathbf{C}(\{\mathbf{h}_k^*\}_{k=1}^K)$ for the final decision making. The training of a combiner will be conducted on the predictions for all pixels of training images in $\mathbf{D}^*$. Once again, the new training data $\mathbf{D}^*$ is divided into disjoint parts $\{\mathbf{D}_1^*, \mathbf{D}_2^*, \dots, \mathbf{D}_T^*\}$. Then, for each part $\mathbf{D}_t^* (t = 1, \dots, T)$, the segmentation algorithms $\{\mathcal{K}_k\}_{k=1}^K$ will learn on $\mathbf{D}^* \backslash \mathbf{D}_t^*$ to obtain segmentation models $\mathbf{h}_{k,t}^*$. Therefore, these models will now predict on $\mathbf{D}_t^*$. The second layer probability prediction for all images in $\mathbf{D}^*$ is given as follows:

$$\mathbf{L}^* = \begin{bmatrix} P_1(y_1|\mathbf{I}_1^*(1,1)) & P_1(y_2|\mathbf{I}_1^*(1,1)) & \dots & P_K(y_M|\mathbf{I}_1^*(1,1)) \\ P_1(y_1|\mathbf{I}_1^*(1,2)) & P_1(y_2|\mathbf{I}_1^*(1,2)) & \dots & P_K(y_M|\mathbf{I}_1^*(1,2)) \\ \dots & \dots & \dots & \dots \\ P_1(y_1|\mathbf{I}_1^*(W,H)) & P_1(y_2|\mathbf{I}_1^*(W,H)) & \dots & P_K(y_M|\mathbf{I}_1^*(W,H)) \\ P_1(y_1|\mathbf{I}_2^*(1,1)) & P_1(y_2|\mathbf{I}_2^*(1,1)) & \dots & P_K(y_M|\mathbf{I}_2^*(1,1)) \\ \dots & \dots & \dots & \dots \\ P_1(y_1|\mathbf{I}_2^*(W,H)) & P_1(y_2|\mathbf{I}_2^*(W,H)) & \dots & P_K(y_M|\mathbf{I}_2^*(W,H)) \\ & \dots & \dots & \\ P_1(y_1|\mathbf{I}_N^*(W,H)) & P_1(y_2|\mathbf{I}_N^*(W,H)) & \dots & P_K(y_M|\mathbf{I}_N^*(W,H)) \end{bmatrix}. \tag{4}$$

Normally, a learning algorithm trains the combiner on $\mathbf{L}^*$ with given labels of each pixel to combine the prediction of segmentation models for the final prediction. It is noted that each row in $\mathbf{L}^*$ is the probability prediction by K segmentation models on a pixel of each training image. Therefore $\mathbf{L}^*$ will be a matrix of $N \times W \times H$

rows and $M \times K$ columns. With a large training set and large image sizes, the size of $\mathbf{L}^*$ will be very large. For instance, on the Kvasir-SEG dataset of 800 training images with an image size of (640,544), the matrix $\mathbf{L}^*$ will have 800*640*544 = 278,528,000 rows. The large size of $\mathbf{L}^*$ causes a challenge for conventional machine learning algorithms to train the combiner on all data at once. In this paper, we use a weight-based combining method for the segmentation algorithms $\{\mathbf{h}_k^*\}_{k=1}^K$, in which each segmentation algorithm has its own weight in the combiner. The weights are found via an optimisation method, which will be discussed in the next section. This approach is practical to train the combiner on the whole $\mathbf{L}^*$ at once.

Combining Method

Let $\mathbb{W} = \{w_{k,m}\}$ be the weight matrix, in which $w_{k,m}$ is the weight associated with the segmentation model $\mathbf{h}_k^*$ and class $y_m (k = 1, \dots, K, m = 1, .., M)$. Since the class labels of the training observations are known in advance, the weights $\mathbb{W}$ can be obtained by exploring the relationship between the second-layer probability predictions in $\mathbf{L}^*$ and the class labels of the training pixels. The weight matrix is found by minimizing the difference between the prediction for pixel $\mathbf{I}_n(i,j)$ and its true class label. From the second-layer probability prediction matrix $\mathbf{L}^*$, we extract the probabilities associated with class y_m to create a matrix of size $(N \times W \times H, K)$:

$$\mathbf{L}_m^* = \begin{bmatrix} P_1(y_m|\mathbf{I}_1^*(1,1)) & P_2(y_m|\mathbf{I}_1^*(1,1)) & \dots & P_K(y_m|\mathbf{I}_1^*(1,1)) \\ P_1(y_m|\mathbf{I}_1^*(1,2)) & P_2(y_m|\mathbf{I}_1^*(1,2)) & \dots & P_K(y_m|\mathbf{I}_1^*(1,2)) \\ \dots & \dots & \dots & \dots \\ P_1(y_m|\mathbf{I}_1^*(W,H)) & P_2(y_m|\mathbf{I}_1^*(W,H)) & \dots & P_K(y_m|\mathbf{I}_1^*(W,H)) \\ P_1(y_m|\mathbf{I}_2^*(1,1)) & P_2(y_m|\mathbf{I}_2^*(1,1)) & \dots & P_K(y_m|\mathbf{I}_2^*(1,1)) \\ \dots & \dots & \dots & \dots \\ P_1(y_m|\mathbf{I}_2^*(W,H)) & P_2(y_m|\mathbf{I}_2^*(W,H)) & \dots & P_K(y_m|\mathbf{I}_2^*(W,H)) \\ \dots & \dots & \dots & \dots \\ P_1(y_m|\mathbf{I}_N^*(W,H)) & P_2(y_m|\mathbf{I}_N^*(W,H)) & \dots & P_K(y_m|\mathbf{I}_N^*(W,H)) \end{bmatrix}. \tag{5}$$

We also define a crisp label vector (i.e. belonging to $\{0, 1\}$) of size $(N \times W \times H, 1)$ associated with class y_m as follows:

$$\mathbb{Y}_m = \begin{bmatrix} \mathbb{I}[\mathbf{Y}_1(1,1) = y_m] \\ \dots \\ \mathbb{I}[\mathbf{Y}_N(1,1) = y_m] \\ \dots \\ \mathbb{I}[\mathbf{Y}_N(W,H) = y_m] \end{bmatrix}. \tag{6}$$

where $\mathbb{I}[.]$ is the indicator function. The weight vector $\mathbb{W}_m = \{w_{k,m}\}, k = 1, \dots, K$ of size $(K, 1)$ for class y_m is then found by solving a linear regression problem:

$$\min_{\mathbb{W}_m} ||\mathbf{L}_m^* \mathbb{W}_m - \mathbb{Y}_m||_2 \tag{7}$$

$\mathbb{W}_m$ can be imposed with different constraints, such as Non-Negative Least Squares, i.e. $w_{k,m} \geq 0$ [46, 47], Bounded Variable Least Squares, i.e. $l_{k,m} \leq w_{k,m} \leq u_{k,m}$ in which $l_{k,m}$ and $u_{k,m}$ are lower and upper bounds [48, 49], respectively, and Bounded Variable with Constant Sum, i.e. $-1 < w_{k,m} < 1, \sum_{k=1}^K w_{k,m} = 1$ [50]. In this study, we simply constrain the weights between 0 and 1, i.e. $0 \leq w_{k,m} \leq 1$. By solving M different linear regression problems, we will get the optimal weight matrix $\mathbb{W} = \{\mathbb{W}_m\}_{m=1}^M$.

Given an unsegmented image $\mathbf{I}_{test}$, it is segmented firstly by $\{\mathbf{h}_k\}_{k=1}^K$ to get the prediction matrices $\{\mathbf{P}_k(y_m|\mathbf{I}_{test})\}(k = 1, \dots, K, m = 1, \dots, M)$. Then, the augmented data of $\mathbf{I}_{test}$ is created by concatenating it with $\{\mathbf{P}_k(y_m|\mathbf{I}_{test})\}$ which are considered as additional image channels by using the following equation (with $k = 1, \dots, K, m = 1, \dots, M$):

$$\mathbf{I}_{test}^* = \mathbf{I}_{test} \cup \{\mathbf{P}_k(y_m|\mathbf{I}_{test})\} \tag{8}$$

The trained segmentation models of the second layer $\{\mathbf{h}_k^*\}_{k=1}^K$ are then applied on $\mathbf{I}_{test}^*$ to get the prediction matrices $\{\mathbf{P}_k(y_m|\mathbf{I}_{test}^*)\}$ $(k = 1, \dots, K, m = 1, \dots, M)$. The weighted combining of predicted probabilities by linear combination for each pixel $\mathbf{I}_{test}^*(i,j)$ is performed as follows:

$$CM_m(\mathbf{I}_{test}(i,j)) = \sum_{k=1}^K w_{k,m} P_k(y_m|\mathbf{I}_{test}^*(i,j)) = \mathbb{P}_m(\mathbf{I}_{test}^*(i,j))\mathbb{W}_m \tag{9}$$

in which $\mathbb{P}_{\boldsymbol{m}}(\mathbf{I}_{\boldsymbol{test}}^*(\boldsymbol{i},\boldsymbol{j}))$ and $\mathbb{W}_{\boldsymbol{m}}$ are defined as:

$$\mathbb{P}_m(\mathbf{I}_{test}^*(i,j)) = [P_1(y_m|\mathbf{I}_{test}^*(i,j)), \dots, P_K(y_m|\mathbf{I}_{test}^*(i,j))] \tag{10}$$

$$\mathbb{W}_m = [w_{1,m}, w_{2,m}, \dots, w_{K,m}]^T \tag{11}$$

Finally, the predicted class label is obtained by getting the label corresponding to the maximum value of the weighted combination:

$$\mathbf{I}_{test}(i,j) \in y_{\hat{m}} \text{where} \hat{m} = argmax_{m=1,\dots,M} CM_m\{\mathbf{I}_{test}(i,j)\} \tag{12}$$

Algorithm 1 Two-layer ensemble for segmentation

```
Input: Training set $\boldsymbol{D} = \{\mathbf{I}_n, \mathbf{Y}_n\}_{n=1}^{N}$, segmentation algorithms $\{\boldsymbol{\mathcal{K}}_k\}_{k=1}^{K}$
Output: Trained segmentation models $\{\mathbf{h}_k\}_{k=1}^{K}$, $\{\mathbf{h}_k^*\}_{k=1}^{K}$ and optimal weights $\mathbb{W}$
(Prediction probability generation)
$\{\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_T\} = \boldsymbol{T-partition}(\mathbf{D})$
for $\boldsymbol{t} \leftarrow \mathbf{1}$ to $\boldsymbol{T}$ do
    for $\boldsymbol{k} \leftarrow \mathbf{1}$ to $\boldsymbol{K}$ do
        $\mathbf{h}_{k,t} = \boldsymbol{Learn}(\boldsymbol{\mathcal{K}}_k, \mathbf{D} \backslash \mathbf{D}_t)$
        for $\mathbf{I}$ in $\mathbf{D}_t$ do
            $\{\mathbf{P}_k(y_m|\mathbf{I})\}_{m=1}^{M} = \boldsymbol{Segment}(\mathbf{h}_{k,t}, \mathbf{I})$
$\mathbf{D}^* = \{\mathbf{I}_n^*, \mathbf{Y}_n\}$, where $\mathbf{I}_n^*$ is defined as Eq. (2)
2nd-level probability generation
$\mathbf{L}^* = \emptyset, \{\mathbf{D}_1^*, \mathbf{D}_2^*, \dots, \mathbf{D}_T^*\} = \boldsymbol{T-partition}(\mathbf{D}^*)$
for $\boldsymbol{t} \leftarrow \mathbf{1}$ to $\boldsymbol{T}$ do
    for $\boldsymbol{k} \leftarrow \mathbf{1}$ to $\boldsymbol{K}$ do
        $\mathbf{h}_{k,t}^* = \boldsymbol{Learn}(\boldsymbol{\mathcal{K}}_k, \mathbf{D}^* \backslash \mathbf{D}_t^*)$
        $\mathbf{L}^* = \mathbf{L}^* \cup \boldsymbol{Segment}(\mathbf{h}_{k,t}^*, \mathbf{D}_t^*)$
(Weight vector generation)
for $\boldsymbol{m} \leftarrow \mathbf{1}$ to $\boldsymbol{M}$ do
    Get $\mathbf{L}_m^*$ by Eq. (5)
    Get $\mathbb{Y}_m$ by Eq. (6)
    Find $\mathbb{W}_m = \{w_{k,m}\}, \boldsymbol{k} = \mathbf{1}, \dots, \boldsymbol{K}$ by solving Eq. (7)
$\mathbb{W} = \{\mathbb{W}_m\}_{m=1}^{M}$
(Base segmentation models generation)
for $\boldsymbol{k} \leftarrow \mathbf{1}$ to $\boldsymbol{K}$ do
    $\mathbf{h}_k = \boldsymbol{Learn}(\boldsymbol{\mathcal{K}}_k, \mathbf{D})$
    $\mathbf{h}_k^* = \boldsymbol{Learn}(\boldsymbol{\mathcal{K}}_k, \mathbf{D}^*)$
return $\{\mathbf{h}_k\}_{k=1}^{K}$, $\{\mathbf{h}_k^*\}_{k=1}^{K}$ and $\mathbb{W}$
```

Algorithm 2 Test process for the two-layer ensemble for segmentation

```
Input: Test image $\mathbf{I}_{test}$, trained segmentation models $\{\mathbf{h}_k\}_{k=1}^{K}, \{\mathbf{h}_k^*\}_{k=1}^{K}$ and the weight matrix $\mathbb{W}$
Output: Prediction for $\mathbf{I}_{test}$
1: for $k \leftarrow 1$ to $K$ do
2:     $\{\mathbf{P}_k(y_m|\mathbf{I}_{test})\}_{m=1}^{M} = Segment(\mathbf{h}_k, \mathbf{I}_{test})$
3: $\mathbf{I}_{test}^*$ is created from $\mathbf{I}_{test}$ and $\{\mathbf{P}_k(y_m|\mathbf{I}_{test})\}_{m=1}^{M}$ using Eq. (8)
4: for $k \leftarrow 1$ to $K$ do
5:     $\mathbf{P}_k(y_m|\mathbf{I}_{test}^*) = \mathbf{P}_k(y_m|\mathbf{I}_{test}) \cup Segment(\mathbf{h}_k^*, \mathbf{I}_{test}^*)(m = 1, \dots, M)$
6: Use Eq. (9) to combine the predictions $\{\mathbf{P}_k(y_m|\mathbf{I}_{test}^*)\}$ with $m = 1, \dots, M; k = 1, \dots, K$
7: Use Eq. (12) to get the final prediction
8: return The final prediction for $\mathbf{I}_{test}$.
```

The combining and training procedure is described in Algorithm 1. This algorithm receives inputs including training set $\mathbf{D} = \{\mathbf{I}_n, \mathbf{Y}_n\}_{n=1}^{N}$ and segmentation algorithms $\{\mathcal{K}_k\}_{k=1}^{K}$. Lines 2–7 create the probability matrices by using T-fold cross-validation procedure. Line 8 creates the augmented input data for the second layer by using Eq. 2. Lines 10–14 create the second-level predictions for all training pixels $\mathbf{L}^*$ by using the T-fold cross-validation procedure. Lines 16–20 find the optimal weight matrix by using Eq. 7. In lines 21–24, the segmentation models are generated. Lines 23 trains the models for the first layer while line 24 trains the models for the second layer. Line 25 returns the trained models and the optimal weight matrix.

The testing procedure receives as input an image $\mathbf{I}_{test}$, the trained models, and the optimal weight matrix (see Algorithm 2). In lines 1–2, the segmentation models perform predictions on the image to create the probability matrix, while in line 3, the augmented input to the second layer is created by using Eq. 8. Lines 4–5 create the second-level probability matrix from augmented input. Line 6–7 uses Eq. 9 and Eq. 12 to combine the second-level prediction predictions of segmentation models by using the weight matrix $\mathbb{W}$. Finally, line 8 returns the final segmentation result.

Experimental Details

Experimental Settings

For the experimental validation, we used UNet [51], LinkNet [28], and Feature Pyramid Network (FPN) [29], which are three popular segmentation architectures for medical image analysis. The backbones used were VGG16 [52], ResNet34 and ResNet101 [53], pre-trained on the ImageNet dataset [54]. In total, we started with nine segmentation models. In this paper, the number of cross-validation folds was set to 5, the batch size was set to 8, each segmentation model was run for 300 epochs, the learning rate was set to 0.0001, and the Adam optimizer [55] was used in the experiments. We compared the performance of the proposed ensemble to the nine segmentation algorithms and three other benchmark algorithms:

- The first one is one layer ensemble system in which the outputs of segmentation models are combined by using a weight-based combining algorithm (denoted by OLE-9 in the tables) [56].
- The second one is Decision Template [57], denoted by DT-9 in the tables. In this algorithm, each class is repre-

sented by a decision template, which is calculated by taking the average of the predictions of all training instances associated with that class. For each test instance, the class having the smallest distance between the predictions and the corresponding decision template is chosen.

- The third one is the weighted ensemble of deep learning medical segmentation models [44], in which the predictions of each segmentation model are combined via weighted summation. The weights are found by solving an optimisation problem using Comprehensive Learning Particle Swarm Optimisation (CLPSO). This benchmark algorithm is denoted by WE-CLPSO in the next sections.
- The fourth one is UNet+ +[32], one of the state-of-the-art deep learning-based segmentation models introduced recently.

Performance Metrics

The performance of our proposed ensemble and the related benchmarks were evaluated using four popular segmentation metrics. Suppose there are M classes, and there are N images each having size (W, H). Let $\mathbf{P}$ and $\mathbf{G}$ be the prediction of a segmentation model on these images and the corresponding ground truth:

$$\mathbf{P} = \left[\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M\right], \mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_\mathrm{M}] \tag{13}$$

where $\mathbf{p}_m$ is a vector with size $(N \times W \times H, 1)$ associated with class label y_m in which its element is the prediction for each pixel in the form of crisp label i.e. belonging to $\{0, 1\}$. Likewise, $\mathbf{g}_m$ is a vector with size $(N \times W \times H, 1)$ associated with class label y_m in which each element is the ground truth of each pixel in the form of a crisp label. Hence, the Dice coefficient for the m^{th} class is then defined as follows [58]:

$$DC_m = \frac{2\mathbf{p}_m^T\mathbf{g}_m}{||\mathbf{p}_m||^2 + ||\mathbf{g}_m||^2} \tag{14}$$

Another well-known overlap-based metric is Intersection-over-Union (IoU) (also known as Jaccard index) [59] which is defined for the m^{th} class as follows:

$$IoU_m = \frac{\mathbf{p}_m^T\mathbf{g}_m}{||\mathbf{p}_m||^2 + ||\mathbf{g}_m||^2 - \mathbf{p}_m^T\mathbf{g}_m} \tag{15}$$

Notice that the measures the intersection between the prediction and the ground truth pixels divided by their union. The IoU coefficient is the average of all IoU coefficients associated with the class labels.

$$IoU_{avg} = \frac{1}{M}\sum_{m=1}^{M} IoU_m \tag{16}$$

In the context of medical image analysis, local discrepancies between contours are often of interest as well. For example, radiation treatment planning applications require quantified errors in geometric displacement to ensure target coverage, normal tissue avoidance, and similar analyses [60]. Overlap-based metrics such as the Dice coefficient and IoU usually do not account for spatial distribution. For example, a segmentation with “leaks” has the same score as a leak-free one with a separate disconnected region of false positives having the same size as the leaked area [61]. Therefore, we reported two additional measures based on distances between geometrical contours. Let GT_m and PR_m be the set of coordinate vectors of the ground truth contour and prediction contour for class y_m respectively. The Hausdorff distance HD_m associated with class y_m is defined as the maximum of all distances between a point in a contour to the closest point in the other one and is calculated as follows [59]:

$$HD_m = \max(d\left(GT_m, PR_m\right), d(PR_m, GT_m)) \tag{17}$$

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

$$d(A, B) = \frac{1}{|A|}\sum_{a\in A}\min_{b\in B}||a - b|| \tag{18}$$

Meanwhile, the Mean Absolute Distance (MAD) for class $\boldsymbol{y_m}$[61] is the average of all the distances from points of the prediction contour to the ground truth contour, and vice versa and is calculated as follows:

$$MAD_m = \frac{1}{|GT_m| + |PR_m|}(\sum_{gt\in GT_m}\min_{pr\in PR_m})||gt - pr|| + \sum\nolimits_{pr\in PR_m}\min_{gt\in GT_m})||pr - gt||. \tag{19}$$

It is noted that a low Hausdorff distance and MAD, or a high Dice coefficient and IoU indicate good segmentation results.

Datasets

Several public medical image segmentation datasets were used in these experiments, which are shown in Table 1. The first dataset is CVC-ColonDB [11], a public polyp dataset consisting of 300 images, each containing polyps and background, selected from 15 short colonoscopy videos such that variation in scale and view angles of the polyps are maximised. The second dataset is CVC-EndoSceneStill [62], a four-class dataset of 912 images obtained from 44 video sequences acquired from 36 patients for endoluminal scene object segmentation. This dataset contains some information like lumen and specular highlights which are essential

Table 1 The information of experimental datasets

Dataset	Number of instances	Number of classes	Image size
CVC-ColonDB	300	2 (polyp, background)	512×576
CVC-EndoSceneStill-2017	912	4 (polyp, lumen, specular, background)	224×224
MICCAI2015	808	2 (polyp, background)	288×384
CAMUS-ED	1000	4 (left ventricle, myocardium, left atrium, background)	928×576
CAMUS-ES	1000	4 (left ventricle, myocardium, left atrium, background)	928×576

for helping clinicians navigate through the colon during the inspection procedure. The third dataset in our experiment is from the MICCAI 2015 Endoscopic Vision Challenge [63], which is a colorectal polyp detection and localisation challenge. The dataset contains 612 training images and 196 test images. Each image contains at least one polyp and has been selected to have shots in which polyp appearance can be mistaken with other elements of the scene. There are two classes: polyp and background. The final two datasets are from the Cardiac Acquisitions for Multi-structure Ultrasound Segmentation (CAMUS) challenge [10], a competition for accurate segmentation of 2D echocardiographic images. These datasets are well-known for cardiographic [64], consisting of cardiographic images of 500 patients. The data of 50 patients are withheld for testing in which the submission link for evaluation is available,[1] and the results are reported for End Diastolic (ED) and End Systolic (ES) cases separately, which are denoted by CAMUS-ED and CAMUS-ES respectively. For the CAMUS dataset, we submitted our results to the organisers' server to get the Dice and Hausdorff result, while for the CVC-EndoSceneStill and MICCAI2015 dataset we used the pre-specified train and test set. For the CVC-ColonDB we used 20% of the total data as the test set.

Influence of Using Different Number of Segmentation Algorithms

Figure 3 shows the performance of the proposed ensemble for two cases:

- Proposed ensemble (6): Using 6 segmentation models generated by VGG16 and ResNet34 backbone.
- Proposed ensemble (9): Using all 9 segmentation models

From this figure, it can be seen that the proposed ensemble (9) obtains better results, with noticeable improvements in Hausdorff and MAD scores, while the results for CAMUS-ED and CAMUS-ES are generally similar for the two cases. For the Dice metric (top left), the proposed ensemble (9) achieves a score of 0.956 as compared to just 0.939 by the proposed ensemble (6), which is an improvement of 1.7%. On the MICCAI2015 dataset, there is also an improvement from 0.772 to 0.838. For the remaining datasets, namely CVC-EndoSceneStill, CAMUS-ED, and CAMUS-ES, the results for both cases are identical. For the Hausdorff distance (top right), the proposed ensemble (9) achieves much better results on CVC-ColonDB, CVC-EndoSceneStill, and MICCAI2015.

For CVC-ColonDB, the proposed ensemble (9) reduces the Hausdorff distance by 1.58 times compared to the proposed ensemble (6), from 23.293 to 14.751, while for the MICCAI2015 dataset, there is also a reduction of 11.397. For the MAD metric (bottom left), there is a small improvement from around 0.5 to 1.5 for the CVC-ColonDB and CVC-EndoSceneStill dataset, while for MICCAI2015, the decrease in MAD is 4.308. Concerning the IoU measure (bottom right), the results for CAMUS-ED and CAMUS-ES are not available, since the evaluation server only returns Dice, Hausdorffx, and MAD metrics. For CVC-ColonDB and MICCAI2015, there is an improvement of 3% and 7.5% respectively, while for CVC-EndoSceneStill, the results for both cases are just slightly above 0.6. From the results discussed above, it can be observed that the results for the proposed ensemble (9) are better, especially on Hausdorff and MAD scores.

Results and Discussions

Table 2 shows the Dice results, in which the first nine rows denote the results of the base segmentation models, and the next five rows denote the results of four selected benchmark algorithms and the proposed ensemble respectively. It can be seen that the proposed ensemble achieves better results compared to the benchmark algorithms on all datasets. For the CVC-ColonDB dataset, the proposed ensemble obtains 0.95618 while the segmentation models only gain a score from 0.81227 (UNet-VGG16) to 0.94794 (FPN-ResNet101). The proposed ensemble also yields higher performance compared to the other benchmark algorithms for the remaining datasets. WE-CLPSO, DT-9, and UNet + + gained slightly lower results than the proposed ensemble, at 0.95382, 0.95210, and 0.94591 respectively, while the score for OLE-9 is lower by 1.897%. For CVC-EndoSceneStill and MICCAI2015, DT-9 performs much less effective compared

[1] https://www.creatis.insa-lyon.fr/Challenge/camus/scientificInterests.html

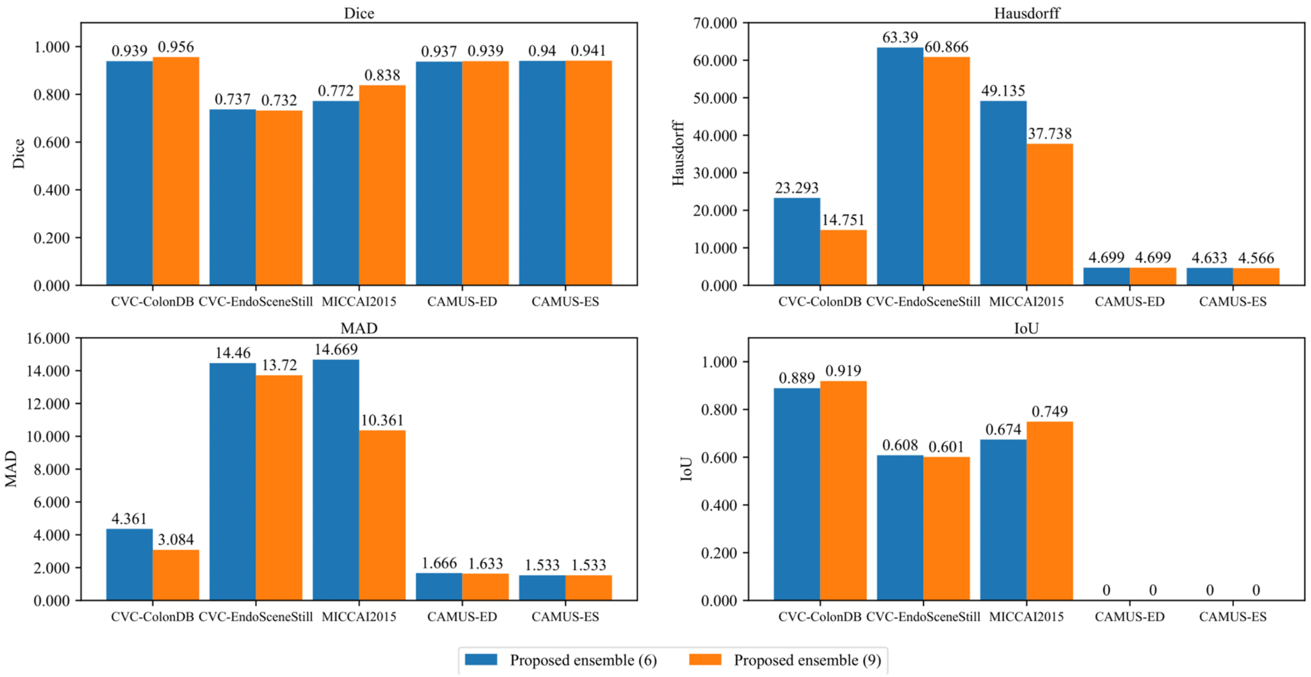

Fig. 3 The performance of the proposed ensemble using 6 and 9 segmentation algorithms

to the proposed ensemble (lower than 8.72% and 7.55% respectively). The proposed ensemble obtained a Dice coefficient of 0.73238 on CVC-EndoSceneStill as opposed to just around 0.68, 0.71, and 0.72 by UNet++, OLE-9, and WE-CLPSO, respectively. Concerning the MICCAI2015 dataset, the proposed ensemble obtained a score of 0.83776, which is higher than OLE-9 by 1.596%. The score of WE-CLPSO is only slightly higher than 0.72 while the score of UNet++ is just around 0.6. For the CAMUS-ED and CAMUS-ES datasets, the proposed ensemble is higher than the other benchmark algorithms by a small margin (from 0.16% to 0.27% for CAMUS-ED and from 0.27% to 0.3% for CAMUS-ES).

The Hausdorff results are shown in Table 3. For CVC-ColonDB, the scores of segmentation models range from 119.6171 (LinkNet-VGG16) to 16.02841 (FPN-ResNet101). The proposed ensemble obtains the best result at 14.75129 which is better than FPN-ResNet101 by 1.277. The Hausdorff distances achieved by DT-9, OLE-9, WE-CLPSO, and UNet++ are higher than that of the proposed ensemble, which indicates their worse performance (16.69677 of WE-CLPSO, 21.97803 of OLE-9, and 22.62884 of UNet++ vs. 14.75129 of the proposed ensemble). For CVC-EndoSceneStill, the predictions by FPN-ResNet34 obtain the best Hausdorff distance at 60.13216, followed by FPN-ResNet101 (around 60.55597) and the proposed ensemble (at 60.86643). In contrast, both WE-CLPSO and OLE-9 obtained a score of around 68, UNet++ had a score of around 67.78 while DT-9 is much worse at 103.45. The proposed ensemble obtains the best results on the remaining three datasets. For the MICCAI2015 dataset, the proposed ensemble obtains a score of 37.73843 which is better than the best segmentation model (FPN-ResNet101) by a difference of 2.0675. The Hausdorff scores of WE-CLPSO, DT-9, and OLE-9 are much worse compared to that of the proposed ensemble (from around 45.84 by OLE-9 to around 63.37 by WE-CLPSO), while the score of UNet++ is twice as large (at around 111.68). For both CAMUS-ED and CAMUS-ES, the scores by the base models generally range from around 5.0 to 10.33. The Hausdorff distances obtained by WE-CLPSO and OLE-9 are very similar (4.866 for CAMUS-ED and 4.8 for CAMUS-ES) while DT-9 performed slightly worse with Hausdorff distances of 4.899 and 4.833 respectively. The score of UNet++ is 5.966 (CAMUS-ED) and 5.6 (CAMUS-ES) which is not as good as the other benchmark algorithms. Compared to these benchmark algorithms, the best results are obtained by the proposed ensemble at 4.699 (CAMUS-ED) and 4.566 (CAMUS-ES).

Table 4 shows the MAD results by the benchmark algorithms and the proposed ensemble. It can be seen that the proposed ensemble obtains the best score for all datasets except MICCAI2015, in which FPN-ResNet101 has the best results. The results by segmentation models on CVC-ColonDB range from 25.9 (LinkNet-VGG16) to 3.17290 (FPN-ResNet101). The results by WE-CLPSO (3.30906), DT-9 (3.34359), OLE-9 (4.43871), and UNet++ (6.00943) are worse than that of the proposed ensemble (rank first with 3.08421 of MAD). For the CVC-EndoSceneStill dataset,

Table 2 Comparison of Dice score between 9 base segmentation algorithms, benchmark algorithms, and the proposed ensemble

Model	CVC-ColonDB	CVC-EndoSceneStill	MICCAI2015	CAMUS-ED	CAMUS-ES
UNet-VGG16	0.81227	0.62501	0.59604	0.9093	0.9103
LinkNet-VGG16	0.88822	0.62978	0.52461	0.874	0.8576
FPN-VGG16	0.88119	0.65474	0.59909	0.8533	0.84933
UNet-ResNet34	0.92281	0.70467	0.75645	0.93467	0.93467
LinkNet-ResNet34	0.94722	0.6845	0.77767	0.93233	0.93467
FPN-ResNet34	0.93055	0.73016	0.76523	0.934	0.93533
UNet-ResNet101	0.91962	0.68474	0.72566	0.93067	0.93
LinkNet-ResNet101	0.93651	0.64733	0.71433	0.87633	0.87767
FPN-ResNet101	0.94794	0.72395	0.83399	0.913	0.90767
WE-CLPSO	0.95382	0.72573	0.72587	0.93666	0.93766
DT-9	0.9521	0.64518	0.76226	0.93733	0.938
OLE-9	0.93721	0.71371	0.82179	0.93633	0.938
UNet+ +	0.94591	0.68695	0.60791	0.925	0.92533
Proposed ensemble	**0.95618**	**0.73238**	**0.83776**	**0.939**	**0.94066**

Bold values indicates the best result among all methods on each dataset

the proposed ensemble obtained a score of 13.72005 which is better than WE-CLPSO, OLE-9, and UNet+ +by 2.56, 3.02, and 3.953 respectively. DT-9 meanwhile performed much worse with 31.19257 of MAD. The best-performing method for the MICCAI2015 dataset is FPN-ResNet101 at 9.13451 followed by the proposed ensemble at 10.3611. The remaining segmentation models range from 14.877 (FPN-ResNet34) to 46.04682 (LinkNet-VGG16), while WE-CLPSO has a slightly better MAD score than FPN-VGG16 at 21.64301, which is twice the distance as that of the proposed ensemble. Both DT-9 and OLE-9 are slightly worse than the proposed ensemble by a difference of 5.44 and 2.91 respectively. On the other hand, the performance of UNet+ +for this dataset is very poor, at around 42.76 of MAD. For CAMUS-ED, all four methods: WE-CLPSO, DT-9, OLE-9, and the proposed ensemble yielded the same MAD score at 1.63, which is better than the best-performing method among the segmentation models by 0.167, and better than UNet+ +by 0.3. For CAMUS-ES, the MAD score by the proposed ensemble is 1.533 which is the best result, followed by DT-9 at 1.566, WE-CLPSO and OLE-9 which are both at 1.599, and UNet+ +at 1.9.

Table 5 shows the IoU results by the benchmark algorithms and the proposed ensemble. The results for CAMUS are not available since the evaluation server only returns Dice, Hausdorff, and MAD scores. It can be seen that the proposed ensemble achieves the best results on all three datasets. For CVC-ColonDB, the proposed ensemble gains a score of 0.91879, which is better than the highest-performing model (FPN-ResNet101) by 1.39%. WE-CLPSO, DT-9,

Table 3 Comparison of Hausdorff score between 9 base segmentation algorithms, benchmark algorithms, and the proposed ensemble

Method	CVC-ColonDB	CVC-EndoSceneStill	MICCAI2015	CAMUS-ED	CAMUS-ES
UNet-VGG16	85.00296	97.06509	93.1726	8.13	9.9
LinkNet-VGG16	119.6171	96.71459	114.6831	9.766	10.2
FPN-VGG16	39.2416	87.709	76.44703	8.466	10.33
UNet-ResNet34	28.9293	75.34879	65.70021	5.1	5.1
LinkNet-ResNet34	19.30709	80.66761	62.4856	5.23333	5.1
FPN-ResNet34	23.28578	**60.13216**	50.53678	5.2	5.13333
UNet-ResNet101	26.61305	70.04111	73.42515	5.83333	5.43333
LinkNet-ResNet101	22.81325	73.01387	74.4993	5.06667	4.83333
FPN-ResNet101	16.02841	60.55597	39.80595	5.6	5.4
WE-CLPSO	16.69677	68.1574	63.36689	4.866	4.8
DT-9	17.13916	103.4518	50.85759	4.899	4.833
OLE-9	21.97803	68.68501	45.84166	4.866	4.8
UNet+ +	22.62884	67.78712	111.68482	5.966	5.6
Proposed ensemble	**14.75129**	60.86643	**37.73843**	**4.699**	**4.566**

Bold values indicates the best result among all methods on each dataset

Table 4 Comparison of MAD score between 9 base segmentation algorithms, benchmark algorithms, and the proposed ensemble

Method	CVC-ColonDB	CVC-EndoSceneStill	MICCAI2015	CAMUS-ED	CAMUS-ES
UNet-VGG16	25.56598	29.2703	36.31338	2.566	2.8333
LinkNet-VGG16	25.90662	28.61254	46.04682	2.766	2.9333
FPN-VGG16	8.72541	23.77789	27.254	2.56	2.83
UNet-ResNet34	7.69548	18.47708	23.69092	1.73333	1.66667
LinkNet-ResNet34	4.0483	22.6211	20.62498	1.8	1.7
FPN-ResNet34	4.71461	13.85228	14.87703	1.73333	1.66667
UNet-ResNet101	5.2808	17.28837	25.7939	1.83333	1.76667
LinkNet-ResNet101	5.24861	18.97017	26.90045	1.73333	1.63333
FPN-ResNet101	3.1729	14.81169	**9.13451**	1.8	1.76667
WE-CLPSO	3.30906	16.28011	21.64301	**1.633**	1.599
DT-9	3.34359	31.19257	15.80289	**1.633**	1.566
OLE-9	4.43871	16.73726	13.27487	**1.633**	1.599
UNet+ +	6.00943	17.67296	42.76071	1.933	1.9
Proposed ensemble	**3.08421**	**13.72005**	10.3611	**1.633**	**1.533**

Bold values indicates the best result among all methods on each dataset

and UNet+ +obtained slightly lower scores, while the score of OLE-9 is only around 0.88. For the CVC-EndoSceneStill dataset, the highest score is achieved by the proposed ensemble at 0.6015, which is higher than OLE-9 by 2.23%, while the result by DT-9 is only around 0.5. For MICCAI2015, the proposed ensemble also yields the highest result at 0.74947, while OLE-9 only scores around 0.72. Both WE-CLPSO and DT-9 have much lower results at just around 0.62 and 0.66 respectively.

From the results discussed above, it can be seen that:

- The proposed ensemble achieves higher results compared to the segmentation models, especially for the Hausdorff and MAD metrics. This demonstrates that the proposed ensemble of segmentation models is effective.
- The proposed ensemble is better than DT-9 on all datasets, especially on CVC-EndoSceneStill and MICCAI2015 with a difference of 8.72% and 7.55% for Dice score respectively. This is because DT-9 takes the average of the predictions of all training instances of each class as the representation of that class, while the proposed ensemble uses the predictions as additional information for the second ensemble layer. Since the average might not be the best representative for each class, the performance of DT-9 is worse than the proposed ensemble.
- The proposed ensemble is better than OLE-9 on all datasets and all metrics. The best results were obtained on CVC-EndoSceneStill and MICCAI2015 in which the proposed ensemble is better by 1.867% and 1.596%

Table 5 Comparison of IoU score between 9 base segmentation algorithms, benchmark algorithms, and the proposed ensemble

Method	CVC-ColonDB	CVC-EndoSceneStill	MICCAI2015	CAMUS-ED	CAMUS-ES
UNet-VGG16	0.7197	0.48681	0.50829	-	-
LinkNet-VGG16	0.8134	0.49107	0.4356	-	-
FPN-VGG16	0.80448	0.51218	0.50998	-	-
UNet-ResNet34	0.86437	0.56898	0.6595	-	-
LinkNet-ResNet34	0.90359	0.54707	0.67967	-	-
FPN-ResNet34	0.87653	0.60004	0.66542	-	-
UNet-ResNet101	0.85966	0.55178	0.62661	-	-
LinkNet-ResNet101	0.88605	0.50592	0.61908	-	-
FPN-ResNet101	0.90482	0.58973	0.74507	-	-
WE-CLPSO	0.91472	0.59457	0.62379	-	-
DT-9	0.91181	0.50204	0.66277	-	-
OLE-9	0.88722	0.57911	0.72961	-	-
UNet+ +	0.90147	0.54866	0.51274	-	-
Proposed ensemble	**0.91879**	**0.6015**	**0.74947**	-	-

Bold values indicates the best result among all methods on each dataset for Dice score, respectively. This is because the proposed ensemble uses a two-layer ensemble in contrast to OLE-9 which only uses one ensemble layer.

- The proposed ensemble achieves higher results compared to WE-CLPSO on most datasets and metrics. For example, on the MICCAI2015 dataset, the Hausdorff distance by the proposed ensemble is only half of the distance by WE-CLPSO. This can also be observed for the MAD metric on both CVC-EndoSceneStill and MICCAI2015. This shows that the proposed ensemble which uses the two-layer ensemble performs better than WE-CLPSO.
- The proposed ensemble achieved higher results compared to UNet + + on all datasets and metrics. For example, on the CVC-EndoSceneStill dataset, the proposed ensemble achieved a MAD score of 13.72 while UNet + + only obtained a score of 17.67296. On the CVC-ColonDB dataset, the Dice value of the proposed ensemble is 0.95618 compared to only 0.94591 by UNet + +. This shows that the proposed ensemble performed better than UNet + + on all datasets in our experiments.

Figure 4 (left) shows two example results for the CVC-ColonDB dataset. From left to right, top to bottom are: UNet-VGG16, LinkNet-VGG16, FPN-VGG16, UNet-ResNet34, LinkNet-ResNet34, FPN-ResNet34, UNet-ResNet101, LinkNet-ResNet101, FPN-ResNet101, WE-CLPSO, DT-9, OLE-9, UNet + +, proposed ensemble, input image and the ground truth. It can be seen that the VGG16-based models and UNet-ResNet34 failed to predict the area in the middle of the polyp, while the polyps predicted by LinkNet-ResNet34 and FPN-ResNet34 are much thinner compared to the ground truth. The ResNet101-based models (2nd row, 3rd-4th columns, and 3rd row, 1st column) give better predictions but still contain many rough edges, especially on the left side. The predictions by WE-CLPSO (3rd row, 2nd column) and DT-9 (3rd row, 3rd column) leave a small black area on the right. Meanwhile, the polyp prediction by OLE-9 (3rd row, 4th column) and UNet + + (4th row, 1st column) is divided into two separate areas. The prediction by the proposed ensemble agrees the most with the ground truth. The bottom left figure provides another example for CVC-ColonDB. The predictions by the VGG16-based models and FPN-ResNet101 contain significant deformations from the ground truth, while LinkNet-ResNet34 and FPN-ResNet34 fail to predict a large bell-like area on the left of the polyp. The polyp predicted by LinkNet-ResNet101 (2nd row, 4th column) contains a spurious area at the bottom, while the predictions by both FPN-ResNet101 and WE-CLPSO (3rd row, first two columns) do not preserve the curvature in the upper area as compared to the ground truth. For DT-9 and OLE-9 (3rd row, last two columns), there are many rough edges, and some areas on the middle left of the polyp are considered background (especially for OLE-9). On the other hand, for UNet + + (4th row, 1st column), the prediction has a spurious area on the lower left of the polyp, while the prediction for the upper area is smaller compared to the ground truth. In contrast, the prediction by the proposed ensemble (4th row, 2nd column) resembles the ground truth the most, although there are still some rough edges present.

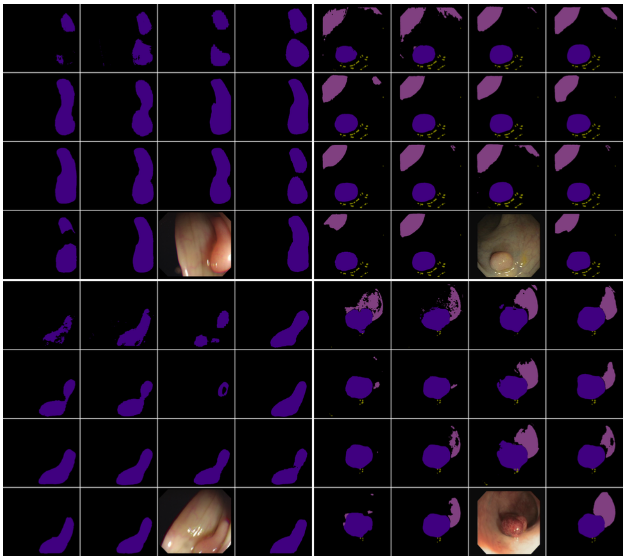

Fig. 4 Two examples of CVC-ColonDB (left) and CVC-EndoSceneStill (right). In each example, from left to right, top to bottom are the results of UNet-VGG16, LinkNet-VGG16, FPN-VGG16, UNet-ResNet34, LinkNet-ResNet34, FPN-ResNet34, UNet-ResNet101, LinkNet-ResNet101, FPN-ResNet101, WE-CLPSO, DT-9, OLE-9, UNet++, proposed ensemble, input image and the ground truth

On the right are two examples of the CVC-EndoSceneStill dataset, with indigo denoting polyp, pink denoting lumen, and olive denoting specular highlights. It can be seen that the predictions for polyp and specular highlights are similar and the most noticeable differences are seen in the predictions for lumen. For the first example (top right), the VGG16-based models, UNet-ResNet34 and LinkNet-ResNet34 (1st row and 2nd row, 1st column) wrongly predict lumen on the top right while there is none. On the second row, the lumen prediction by FPN-ResNet34, UNet-ResNet101, and FPN-ResNet101 (2nd, 3rd, and 4th column respectively) have a pointed area at the bottom part which is not present in the ground truth, while the prediction by LinkNet-ResNet101 for lumen contains an area at the bottom part of the lumen which is not present in the ground truth.

WE-CLPSO, DT-9, and OLE-9 wrongly predict an additional lumen area on the right which is actually the background while UNet + + (4th row, 1st column) fails to predict the lower lumen area. It is recognised that only the proposed ensemble segments the lumen correctly. For the second example (bottom right), the models with VGG16 and ResNet34 as backbone, such as UNet-VGG16 fail to predict many areas within the lumen. The prediction for polyp by UNet-ResNet101 contains several small, disjointed areas on the left while LinkNet-ResNet101 and FPN-ResNet101 either fail to predict the left part of the lumen or wrongly predict only a small dot of the lumen. The prediction by WE-CLPSO and OLE-9 both contain holes within the lumen, while the lumen prediction by DT-9 is more correct. However, it should be noted that DT-9 wrongly predicts the existence of specular highlights within the polyps, and its prediction for polyp overshoots a sizable area on the top left. On the other hand, UNet+ +only predicts some small areas of the lumen, while wrongly considering some small areas on the upper left of the polyp as lumen. Even though the proposed ensemble does not correctly predict the left part of the lumen, the predicted lumen does not contain holes like other methods and the polyp and specular highlights predictions are generally correct.

Figure 5 shows the examples for CAMUS-ED (left) and CAMUS-ES (right), with indigo representing the left ventricle, pink representing the myocardium, and olive green representing the left atrium. From left to right, top to bottom are: UNet-VGG16, LinkNet-VGG16, FPN-VGG16, UNet-ResNet34, LinkNet-ResNet34, FPN-ResNet34, UNet-ResNet101, LinkNet-ResNet101, FPN-ResNet101, WE-CLPSO, DT-9, OLE-9, UNet+ +, proposed ensemble, input image (the ground truths are not available for these datasets, and the metrics were evaluated by submitting the predictions to an evaluation system). For the top left example, it can be seen that the UNet-VGG16 and FPN-VGG16 fail to predict the lower-left area of the left ventricle, while

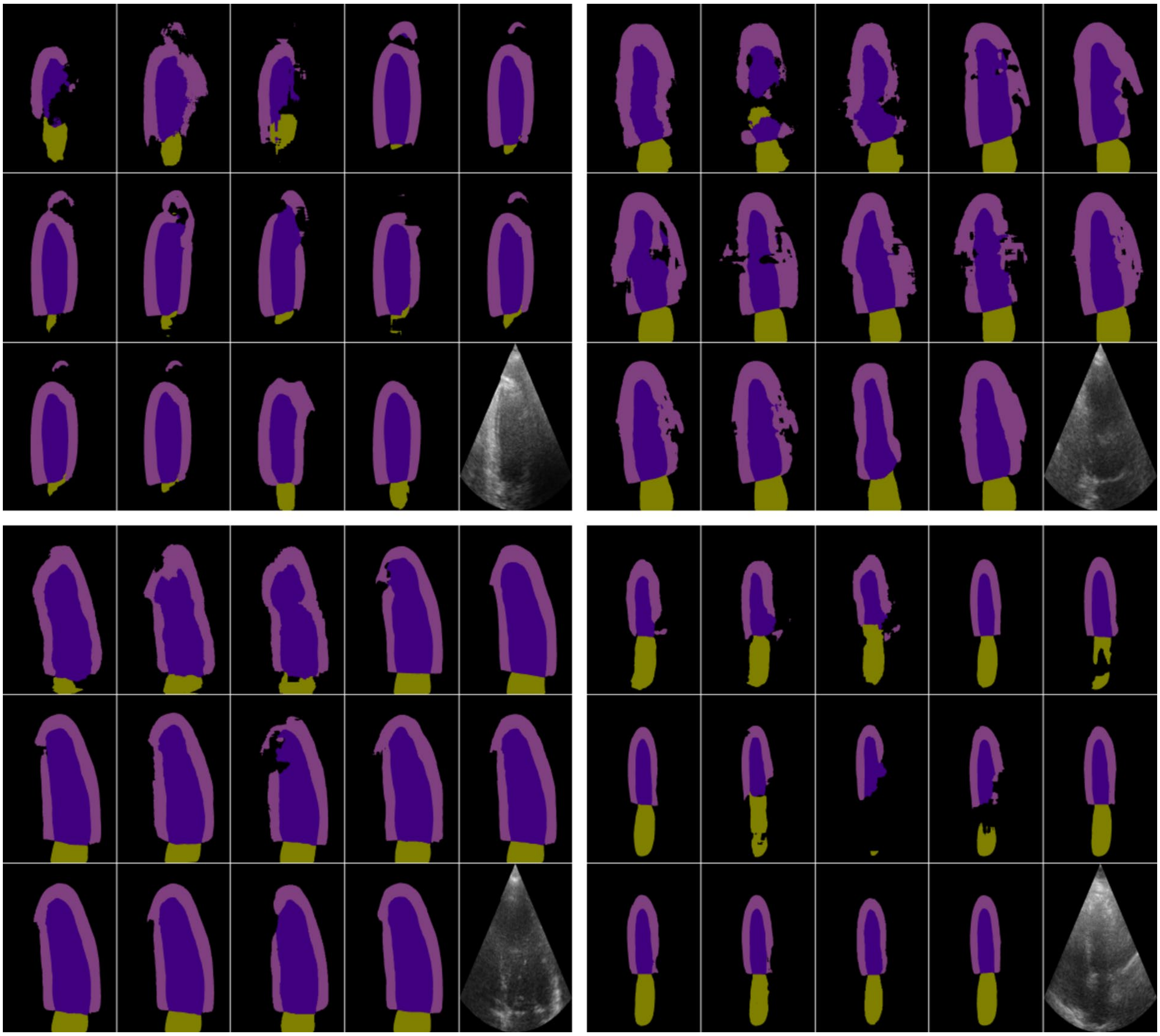

Fig. 5 Two examples of CAMUS-ED (left) and CAMUS-ES (right). In each example, from left to right, top to bottom are the results of UNet-VGG16, LinkNet-VGG16, FPN-VGG16, UNet-ResNet34, LinkNet-ResNet34, FPN-ResNet34, UNet-ResNet101, LinkNet-ResNet101, FPN-ResNet101, WE-CLPSO, DT-9, OLE-9, UNet+ +, proposed ensemble, and input image

LinkNet-VGG16 wrongly predicts a large area to the right as myocardium. The remaining segmentation models wrongly predict a separate area at the top as myocardium and also only manage to predict a small left atrium area. Even though WE-CLPSO, DT-9, and OLE-9 give better results compared to the segmentation models, they still predict a small area at the top as myocardium and the predicted left atrium is still small. On the other hand, UNet+ +correctly predicts the left atrium but fails to predict the lower right area of the myocardium. In contrast, the proposed ensemble predicts the correct shapes of the left ventricle, myocardium, and left atrium. The reason for the incorrect predictions by the benchmark algorithms can be seen by visually inspecting the image (3rd row, 5th column). It can be seen that the image quality is lower on the right, with a large black area present on the bottom right. For the second example (bottom left), UNet-VGG16 and FPN-VGG16 fail to predict a left atrium area on the right and on the top left respectively.

The remaining segmentation models generally predict the correct shapes, except for LinkNet-ResNet101 (2nd row, 3rd column) which fails to segment the left area of both myocardium and left ventricle. In comparison, WE-CLPSO, DT-9, and OLE-9 provide better results even though there is still a pointed myocardium area on the left, which is not present in the prediction by the proposed ensemble. The predictions for the left ventricle and left atrium by UNet+ +are generally correct, however, UNet+ +mistakes a myocardium on the top left as background. The two examples on the right are from CAMUS-ES. For the top right example, it can be seen that the predictions by segmentation models generally contain various deformations. For example, LinkNet-VGG16 fails to predict correctly the middle areas of the left ventricle and myocardium and mistake some part for left atrium. Other segmentation models also have many mistakes in segmenting the left and right areas of the myocardium, such as leaving non-predicted holes within the object (UNet-ResNet34, FPN-ResNet34, and the ResNet101-based models). WE-CLPSO, DT-9, and OLE-9 provide better segmentation, but their predictions still contain unpredicted areas on the right, especially for OLE-9 and DT-9, while the prediction by UNet+ +has many unstable curves and still contains a hole on the lower right area. In contrast, even though the proposed ensemble still predicts a redundant myocardium area on the right, the segmented areas are generally correct. With respect to the second example (bottom right), the VGG16-based models either fail to predict the right area of the myocardium (UNet-VGG16 and FPN-VGG16) or predict it as left ventricle instead (LinkNet-VGG16). LinkNet-ResNet34, LinkNet-ResNet101, and FPN-ResNet101 fail to predict the left atrium while the left atrium prediction by UNet-ResNet101 contains many unpredicted small areas, and only UNet-ResNet34 and FPN-ResNet34 provide overall correct result. The predictions by WE-CLPSO, DT-9, OLE-9, and UNet+ +are better than the segmentation models but still contain a small, unsegmented myocardium area on the middle right. It is noted that this problem is not present in the proposed ensemble.

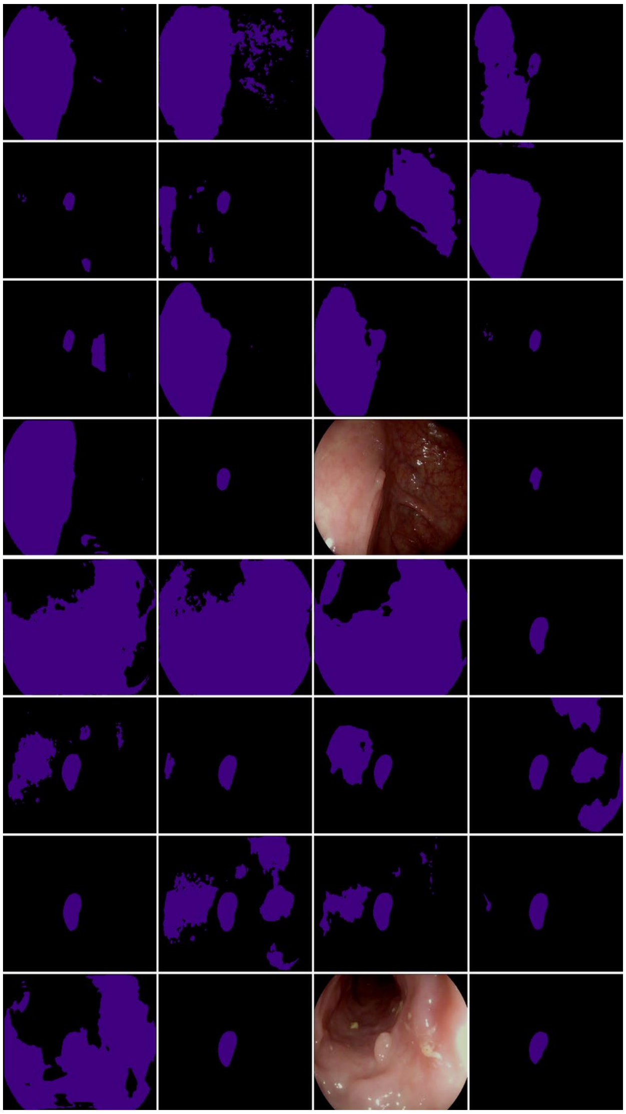

Fig. 6 Two examples of MICCAI2015. In each example, from left to right, top to bottom are the results of UNet-VGG16, LinkNet-VGG16, FPN-VGG16, UNet-ResNet34, LinkNet-ResNet34, FPN-ResNet34, UNet-ResNet101, LinkNet-ResNet101, FPN-ResNet101, WE-CLPSO, DT-9, OLE-9, UNet+ +, proposed ensemble, input image and the ground truth

Figure 6 shows two examples of MICCAI2015. With respect to the first example (top), all VGG16-based models, UNet-ResNet34 and LinkNet-ResNet101 wrongly segment the entire left region of the image as polyp. This can be explained by inspecting the image, in which the polyp is a very small region in the center and is nearly indistinguishable from the surrounding left area. The remaining segmentation models correctly identify the polyp region but still mistakenly predict other areas as polyp as well. For example, FPN-ResNet101 predicts a small region on the right as polyp in contrast to the ground truth. WE-CLPSO, DT-9, and UNet + + wrongly predict the entire left region as a polyp, while the result by OLE-9 contains several small areas to the left of the real polyp. In contrast, the proposed ensemble correctly identifies the polyp region and does not make incorrect predictions in other areas. For the second example (bottom), it can be seen that many of the benchmark algorithms incorrectly predict many regions as polyp, due to the fact that the input image contains many areas which on first glance can be mistaken for polyp. The VGG16-based models mistake a large area as polyp, while the remaining benchmark algorithms usually mistake a large region on the left or three separate regions on the right as polyp. Among the segmentation models, only UNet-ResNet34 and FPN-ResNet101 have generally correct predictions. WE-CLPSO, DT-9, and UNet + + have a lot of spurious predictions around the real polyp, while the prediction by OLE-9 has a small area on the left wrongly identified as a polyp. In contrast, the proposed ensemble provides the correct segmentation of the polyp.

Table 6 shows the weights found by OLE-9 (top) and the proposed ensemble (bottom) for the CVC-EndoSceneStill dataset. It can be seen that for the lumen class, the weights found by the proposed method for the less-performing segmentation models are much smaller compared to OLE-9. For example, FPN-VGG16's weight is 0.38013 for OLE-9 but for the proposed ensemble it is just 0.20808. In contrast, the weight by LinkNet-ResNet101 increases from 0 for OLE-9 to 0.26 for the proposed ensemble. This can be seen from the examples described in the previous section in which the predictions made by ResNet101-based models are better than the other models. For the Polyp class, since the predictions by the segmentation models are similar, the weights assigned by the proposed ensemble are more evenly distributed compared to those of OLE-9. The same can also be seen for the remaining classes. For both OLE-9 and the proposed ensemble, there exists a number of weights that are very small since their contribution to the final predictions is minimal and have been rounded to zero for readability. However, while OLE-9 has nine zero weights, there are only five zero weights for the proposed ensemble.

With respect to the computational time required by each algorithm, it can be seen that:

- Overall, the proposed ensemble required more time compared to OLE-9 and DT-9 since the proposed ensemble is a two-layer ensemble of deep learning segmentation models while the OLE-9 and DT-9 are one-layer ensembles. For the CVC-ColonDB dataset, the required time for the proposed ensemble was 12.81 h as opposed to 5.2 and 5.3 h by DT-9 and OLE-9. The proposed ensemble took 54.12 h on the CVC-EndoSceneStill dataset while DT-9 and OLE-9 took only 15.56 and 15.72 h respectively. For MICCAI2015, 13.21 h were required for the proposed method compared to

Table 6 The weights of OLE-9 and the proposed ensemble for CVC-EndoSceneStill

OLE-9	**Polyp**	**Lumen**	**Specular**	**Background**
UNet-VGG16	0.09459	0.0577	0.07159	0.13734
LinkNet-VGG16	0.05498	0	0.03219	0
FPN-VGG16	0.26744	0.38013	0.18295	0.3185
UNet-ResNet34	0.11722	0.11464	0.11573	0.13752
LinkNet-ResNet34	0.186	0.15637	0.20135	0.20735
FPN-ResNet34	0.08969	0.07533	0.10691	0.17958
UNet-ResNet101	0	0	0	0
LinkNet-ResNet101	0.11963	0	0.00796	0
FPN-ResNet101	0.19017	0	0.12135	0.00683
Proposed ensemble				
UNet-VGG16	0.31845	0	0.06505	0.07941
LinkNet-VGG16	0.24299	0	0.09584	0.124
FPN-VGG16	0	0.20808	0.07764	0.09951
UNet-ResNet34	0.04911	0.12985	0.13017	0.16163
LinkNet-ResNet34	0.10423	0.11275	0.12538	0.18732
FPN-ResNet34	0.07549	0.08996	0.07904	0.07822
UNet-ResNet101	0	0	0.157	0
LinkNet-ResNet101	0.02832	0.26	0.04756	0.138
FPN-ResNet101	0.20933	0.0097	0.03008	0.1111

around 5 h for DT-9 and OLE-9. The same can be seen for both CAMUS-ED and CAMUS-ES with the proposed ensemble taking 40 h, while DT-9 and OLE-9 took only around 20 h.

- The proposed ensemble requires roughly the same computational time as WE-CLPSO. For CVC-ColonDB and CVC-EndoSceneStill, WE-CLPSO required 10.42 and 58.96 h respectively compared to 12.81 and 54.12 h by the proposed ensemble. For the MICCAI2015, WE-CLPSO took 12.08 h which is lower than the proposed ensemble by 1.13 h. In contrast, for CAMUS-ED and CAMUS-ES, while the proposed ensemble requires only around 38 h, WE-CLPSO took 58.5 h. It is noted that the image size of the CAMUS dataset is much bigger than those of other experimental datasets, resulting in a larger prediction matrix. It is noted that WE-CLPSO searches the optimal weights for classifiers in $\mathbf{L}^*$ and the larger $\mathbf{L}^*$ is, the more computational time required. The computational time required for the proposed ensemble can be further reduced by parallelizing the T-fold cross-validation procedure to create the predictions in the first and second layers.

Conclusion

In this paper, we presented a two-layer ensemble of deep learning models for the segmentation of medical images. The key idea is to use the probability prediction by the constituent models in the first layer as augmented data for the second layer. The output probability prediction by the second layer is combined by using a weight-based scheme which is not only an effective combiner but also computationally efficient. The weights are found by solving a linear regression problem associated with each class label. Our results on five benchmark datasets show that the proposed ensemble method is able to combine the strengths and mitigate the drawbacks of the constituent segmentation methods, resulting in an overall improvement.

Several directions can be conducted to improve the proposed ensemble in the future. Firstly, parallelization implemented in the cross-validation procedures at the first and second layers is a potential approach to reduce computational requirements. Secondly, it is noted that the presence of some models might degrade the ensemble performance because of their poor performance or less diversity in predictions. Therefore, a potential solution is to perform ensemble selection to find the optimal subset of models which would not only reduce the computational complexity of the ensemble but also increase its performance as well. Finally, due to the general structure of the two-layer ensemble, the proposed ensemble could be extended to solve other supervised learning tasks such as image classification.

Author Contributions Truong Dang: methodology, writing—original draft preparation, experimental study; Tien Thanh Nguyen: conceptualization, methodology, writing—original draft preparation; John McCall: supervision, writing—reviewing and editing; Eyad Elyan: conceptualization, validation, writing—reviewing and editing; Carlos Francisco Moreno-Garcia: conceptualization, supervision, writing—reviewing and editing. All authors have participated in (a) conception and design, analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.

Funding Funding was provided by the Newton Fund Institutional Links program, project 527639907, in collaboration with Universidad Nacional Autonoma de Mexico (UNAM), granted by The British Council, UK, and Secretaría de Tecnología e Innovación (SECTEI), Mexico City, Mexico.

Data Availability Data publicly available in a repository: The datasets used in our study are available in the repository at: https://onedrive.live.com/?authkey=%21AHKKCcwmkz7UGco&id=EB98BC9D8217F78A%212206&cid=EB98BC9D8217F78A

Declarations

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

Participation Statement All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.

Conflict of Interest All author declares that they have no conflict of interest.

Financial Interest The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript.

Disclosure This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.

References

1. Wang S, Li C, Wang R, et al. Annotation-efficient deep learning for automatic medical image segmentation. Nat Commun. 2021;12.
2. Diaz O, Kushibar K, Osuala R, et al. Data preparation for artificial intelligence in medical imaging: A comprehensive guide to open-access platforms and tools. Physica Med. 2021;83:25–37.
3. Yu-Qian Z, Wei-Hua G, Zhen-Cheng C, et al. Medical images edge detection based on mathematical morphology. IEEE Engineering in Medicine and Biology Society. 2005;6:6492–5.
4. Chen W, Smith R, Ji S-Y, et al. Automated ventricular systems segmentation in brain CT images by combining low-level segmentation and high-level template matching. BMC Med Inform Decis Mak. 2009;9:S4.
5. WangR, Lei T, Cui R, et al. Medical image segmentation using deep learning: A survey. IET Image Process. 2022.
6. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM. 2012;60(6):84–90.
7. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521 (7553):436–44.
8. Chlebus G, Schenk A, Moltz JH, et al. Automatic liver tumor segmentation in CT with fully convolutional neural networks and object-based postprocessing. Sci Rep. 2018;8.
9. Cherukuri V, Ssenyonga P, Warf B, et al. Learning based segmentation of CT brain images: Application to post-operative hydrocephalic scans. IEEE Trans Biomed Eng. 2018;65:1871–84.
10. Leclerc S, Smistad E, Pedrosa J, et al. Deep Learning for Segmentation Using an Open Large-Scale Dataset in 2D Echocardiography. IEEE Trans Med Imaging. 2019;38:2198–210.
11. Bernal J, Sanchez J, Vilarino F. Towards automatic polyp detection with a polyp appearance model. Pattern Recogn. 2012;45:3166–82.
12. Shen D, Wu G, Suk H-I. Deep Learning in Medical Image Analysis. Annu Rev Biomed Eng. 2017;19:221–48.
13. Tang EK, Suganthan PN, Yao X. An analysis of diversity measures. Mach Learn. 2006;65:247–71.
14. Li X, Yang H, He J, et al. Beds: Bagging ensemble deep segmentation for nucleus segmentation with testing stage stain augmentation. IEEE 18th Int Symp Biomed Imaging (ISBI). 2021;659–662.
15. Yang P, Yang J, Zhou B, Zomaya A. A review of ensemble methods in bioinformatics. Curr Bioinform. 2010;5.
16. Zheng H, Zhang Y, Yang L, et al. A New Ensemble Learning Framework for 3D Biomedical Image Segmentation. Proceedings of AAAI. 2019;33:5909–16.
17. Pacheco AGC, Trappenberg T, Krohling RA. Learning dynamic weights for an ensemble of deep models applied to medical imaging classification. Int Jt Conf Neural Net (IJCNN). 2020;1–8.
18. Yang Y, Hu Y, Zhang X, et al. Two-stage selective ensemble of CNN via deep tree training for medical image classification. IEEE Trans Cybern. 2021;1–14.
19. Zhou Z-H, Feng J. Deep Forest: Towards An Alternative to Deep Neural Networks. Proceedings of IJCAI. 2017;3553–3559.
20. Nguyen TT, Van Pham N, Dang MT, et al. Multi-layer heterogeneous ensemble with classifier and feature selection. Proceedings of GECCO. 2020;725–733.
21. Lateef F, Ruichek Y. Survey on semantic segmentation using deep learning techniques. Neurocomputing. 2019;338:321–48.
22. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. Proceedings of CVPR. 2015;3431–3440.
23. Garcia-Garcia A, Orts-Escolano S, Oprea S, et al. A survey on deep learning techniques for image and video semantic segmentation. Appl Soft Comput. 2018;70:41–65.
24. Chen L, Papandreou G, Kokkinos I, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell. 2018;40:834–48.
25. Krahenbuhl P, Koltun V. Parameter learning and convergent inference for dense random fields. Proceedings of ICML. 2013;28.
26. Liu X, Song L, Liu S, et al. A review of deep-learning-based medical image segmentation methods. Sustainability. 2021;13.
27. Baccouche A, Garcia-Zapirain B, Castillo Olea C, et al. Connected-UNets: a deep learning architecture for breast mass segmentation. Breast Cancer. 7 (2021).
28. Chaurasia A, Culurciello E. Linknet: Exploiting encoder representations for efficient semantic segmentation. IEEE Visual Communications and Image Processing (VCIP). 2017;2017:1–4.
29. Lin TY, Dollár P, Girshick R, et al., Feature pyramid networks for object detection. Proc CVPR. 2017;936–944.
30. Milletari F, Navab N, Ahmadi SA. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth Int Conf 3D Vision (3DV). 2016;565–571.
31. Casamitjana A, Catà M, Sánchez I, et al. Cascaded V-Net Using ROI Masks for Brain Tumor Segmentation, Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. 2018;381–391.
32. Zhou Z, Siddiquee MMR, Tajbakhsh N, Liang J. UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation. IEEE Trans Med Imaging. 2020;39(6):1856–67.
33. Nie D, Wang L, Adeli E, et al. 3-D fully convolutional networks for multimodal isointense infant brain image segmentation. IEEE Transactions on Cybernetics. 2019;49:1123–36.
34. Zhang Y, Chung AC Deep supervision with additional labels for retinal vessel segmentation task. MICCAI. 2018;83–91.
35. Jue J, Jason H, Neelam T, et al. Integrating cross-modality hallucinated MRI with CT to aid mediastinal lung tumor segmentation. MICCAI. 2019;221–229.
36. Huo Y, Xu Z, Bao S, et al. Splenomegaly segmentation using global convolutional kernels and conditional generative adversarial networks, Medical Imaging. Image Processing. 2018;2018:45–51.
37. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. NIPS. 2014;27.
38. Fernandez-Delgado M, Cernadas E, Barro S, et al. Do we need hundreds of classifiers to solve real world classification problems? J Mach Learn Res. 2014;15:3133–81.
39. Nguyen TT, Dang MT, Liew AW-C, et al. A weighted multiple classifier framework based on random projection. Inf Sci. 2019;490:36–58.
40. Nguyen TT, Nguyen MP, Pham XC, et al. Combining heterogeneous classifiers via granular prototypes. Appl Soft Comput. 2018;73:795–815.
41. Nguyen TT, Luong AV, Dang MT, et al. Ensemble selection based on classifier prediction confidence. Pattern Recogn. 2020;100.
42. Winzeck S, Mocking SJT, Bezerra R, et al. Ensemble of Convolutional Neural Networks Improves Automated Segmentation of Acute Ischemic Lesions Using Multiparametric Diffusion-Weighted MRI. Am J Neuroradiol. 2019;40:938–45.
43. Kumar A, Kim J, Lyndon D, et al. An ensemble of fine-tuned convolutional neural networks for medical image classification. IEEE J Biomed Health Inform. 2017;21:31–40.
44. Dang T, Nguyen TT, Moreno-García CF, et al. Weighted ensemble of deep learning models based on comprehensive learning particle swarm optimization for medical image segmentation. IEEE Cong Evol Comput (CEC). 2021;744–751.
45. Utkin LV, Kovalev MS, Meldo AA. A deep forest classifier with weights of class probability distribution subsets. Knowl-Based Syst. 2019;173:15–27.

46. Litjens G, Kooi T, Bejnordi BE, et al. A survey on deep learning in medical image analysis. Med Image Anal. 2017;42:60–88.
47. Lawson C, Hanson R. Solving least squares problems, Classics in applied mathematics. 1995.
48. Stark P. Bounded-variable least-squares: an algorithm and applications. In: Comput Stat. 2008.
49. Bro R, Sde Jong S. A fast non-negativity-constrained least squares algorithm. J Chemometrics 11 (1997).
50. Zhang L, Zhou W. Sparse ensembles using weighted combination methods based on linear programming. Pattern Recognit. 2011;44:97–106.
51. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. CoRR abs/1505.04597. 2015.
52. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556. 2015.
53. He K, Zhang X, Ren S, et al., Deep residual learning for image recognition. Proc CVPR. 2016;770–778.
54. Deng J, Dong W, Socher R, et al. ImageNet: A Large-Scale Hierarchical Image Database. Proc CVPR. 2009;248–255.
55. Kingma D, Ba J. Adam: A Method for Stochastic Optimization. Proc 3rd Int Conf Learn Represent (ICLR). 2015.
56. Do DT, Nguyen TT, Nguyen TT, et al. Confidence in prediction: an approach for dynamic weighted ensemble. ACIIDS. 2020;12033:358–70.
57. Kuncheva L, Bezdek J, Duin R. Decision templates for multiple classifier fusion. Pattern Recogn. 2001;34:299–314.
58. Liu Q, Tang X, Guo D, et al. Multi-class Gradient Harmonized Dice Loss with Application to Knee MR Image Segmentation. MICCAI. 2019;86–94.
59. Taha AA, Hanbury A. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. BMC Med Imag. 2015;15.
60. Kim HS, Park SB, Lo SS, et al. Bidirectional local distance measure for comparing segmentations. Med Phys. 2012;39(11):6779–90.
61. Yeghiazaryan V, Voiculescu I. Family of boundary overlap metrics for the evaluation of medical image segmentation. J Med Imag. 2018;5.
62. Vazquez D, Bernal J, Sanchez FJ, et al. A Benchmark for Endoluminal Scene Segmentation of Colonoscopy Images. J Healthcare Eng. 2017.
63. Bernal J, Tajkbaksh N, Sanchez FJ, Comparative validation of polyp detection methods in video colonoscopy: Results from the miccai, et al. endoscopic vision challenge. IEEE Trans Med Imaging. 2015;36(2017):1231–49.
64. Chen C, Qin C, Qiu H, Tarroni G, Duan J, Bai W, Rueckert D. Deep learning for cardiac image segmentation: A review. Front Cardiovascular Med. 2020;7.