information

Article

Exploring the Potential of Ensembles of Deep Learning Networks for Image Segmentation

Loris Nanni [1,*], Alessandra Lumini [2] and Carlo Fantozzi [1]

[1] Department of Information Engineering, University of Padova, 35122 Padova, Italy; carlo.fantozzi@unipd.it
[2] Department of Computer Science and Engineering, University of Bologna, 40126 Cesena, Italy; alessandra.lumini@unibo.it
* Correspondence: loris.nanni@unipd.it

Abstract: To identify objects in images, a complex set of skills is needed that includes understanding the context and being able to determine the borders of objects. In computer vision, this task is known as semantic segmentation and it involves categorizing each pixel in an image. It is crucial in many real-world situations: for autonomous vehicles, it enables the identification of objects in the surrounding area; in medical diagnosis, it enhances the ability to detect dangerous pathologies early, thereby reducing the risk of serious consequences. In this study, we compare the performance of various ensembles of convolutional and transformer neural networks. Ensembles can be created, e.g., by varying the loss function, the data augmentation method, or the learning rate strategy. Our proposed ensemble, which uses a simple averaging rule, demonstrates exceptional performance across multiple datasets. Notably, compared to prior state-of-the-art methods, our ensemble consistently shows improvements in the well-studied polyp segmentation problem. This problem involves the precise delineation and identification of polyps within medical images, and our approach showcases noteworthy advancements in this domain, obtaining an average Dice of 0.887, which outperforms the current SOTA with an average Dice of 0.885.

Keywords: deep learning; ensembles; segmentation; transformers

Citation: Nanni, L.; Lumini, A.; Fantozzi, C. Exploring the Potential of Ensembles of Deep Learning Networks for Image Segmentation. *Information* **2023**, *14*, 657. https://doi.org/10.3390/info14120657

Academic Editor: Gholamreza Anbarjafari (Shahab)

Received: 23 November 2023
Accepted: 6 December 2023
Published: 12 December 2023

1. Introduction

Image semantic segmentation [1] involves dividing an image into distinct, nonoverlapping sections with similar properties. It is a fundamental task in computer vision and image processing. The development of convolutional neural networks (CNNs) has significantly advanced deep learning-based image semantic segmentation, finding applications in various domains like autonomous driving, medical imaging, indoor navigation, virtual reality, and augmented reality. For example, image semantic segmentation plays a vital role in autonomous vehicle driving by segmenting the different elements in the scene, such as roads, vehicles, pedestrians, traffic signs, and obstacles. This information helps the autonomous system make accurate decisions and navigate safely. In medical imaging, image semantic segmentation is employed to identify and segment different anatomical structures or abnormalities in images, including organs, tumors, lesions, blood vessels, and tissues. This assists in diagnosis, treatment planning, and monitoring of diseases; for example, clinical practice often involves using object identification to detect polyps, while, in skin and blood analysis, it can help identify diseases.

Semantic segmentation involves grouping similar components of an image that belong to the same class. Traditional methods for image segmentation are pixel-based, edge-detection-based, or region-based, but they have limitations: for instance, edge-detection-based methods encounter challenges in forming closed regions, while region-based methods struggle to accurately segment edges [2].

For a long time, the ability to recognize and segment objects in images has been a unique trait of humans. The growth of deep learning, particularly convolutional neural

networks (CNNs), has greatly improved the performance of semantic image segmentation, enabling accurate and efficient segmentation in various application domains. The fully convolutional network (FCN) [3] was one of the first attempts to create a CNN-based image segmentation network, where the traditional fully connected layer was replaced by a fully convolutional layer. U-Net [4] is another popular DNN architecture for image segmentation. It consists of an encoder–decoder structure with skip connections that help preserve spatial information. U-Net is widely used in medical image segmentation tasks due to its ability to handle limited training data effectively. DeepLab [5] is a family of DNN architectures designed for semantic image segmentation that utilizes atrous (dilated) convolutions to capture multiscale contextual information effectively. SegNet [6] is an encoder–decoder style DNN architecture for semantic segmentation. It utilizes pooling indices during the encoding phase to efficiently upsample feature maps during decoding. SegNet achieves good results while being computationally efficient.

These and other deep learning approaches [1] based on convolutional neural networks (CNNs) have demonstrated remarkable accuracy in various semantic segmentation tasks. However, CNNs have limitations in capturing global relationships in images due to their localized convolutional operations. As a result, alternative methods such as vision transformers (ViT) [7] and pyramid vision transformers (PVT) [8] have been developed. ViTs and PVT are advanced computer vision techniques that have revolutionized image understanding and achieved state-of-the-art (SOTA) performance in visual recognition tasks. ViTs utilize self-attention mechanisms within the transformer architecture to process images divided into fixed-size patches. This enables capturing global dependencies and long-range relationships between patches. On the other hand, PVT combines CNNs and ViTs by employing a hierarchical approach with multiscale feature pyramids. PVT uses transformers to model relationships between features at different scales, integrating local details and the global context through innovative attention modules. PVT is trained with a combination of supervised and self-supervised learning methods to enhance its robustness and generalization capabilities.

Despite the significance of the methods mentioned, there is still room for enhancing their segmentation capabilities by combining them into an ensemble. Ensemble learning is a machine learning strategy that combines multiple models, called base learners, to achieve more accurate predictions or decisions than any individual model can achieve alone [9]. The concept behind ensemble learning is to leverage the collective intelligence of diverse models to enhance overall performance. In ensemble learning, base learners can be trained on the same dataset using different algorithms, parameters, or training sets. Each base learner learns from the data and generates its prediction, which is then aggregated with the others to produce the final prediction. Ensemble learning offers several advantages, including improved prediction accuracy, reduced overfitting, and increased robustness to noisy data. It is particularly effective when the base learners are diverse and make uncorrelated errors [9].

Given the significance of the aforementioned methods and the potential enhancement in segmentation accuracy through their combination, we propose an investigation into the applicability of different topologies of networks (CNN [10], PVT [11], and mixed CNN and transformer [12]) for semantic image segmentation. Additionally, we explore the performance improvement of an ensemble that integrates these methods to evaluate its impact on segmentation performance. This research builds upon prior work [13–15], which focused on a limited number of case studies and models. The novelty of our research with respect to previous works lies in the proposal of ensembles that encompass a wide range of network architectures, models, and data augmentation techniques. Our experimental results demonstrate that, through these strategies, we can construct robust and efficient ensembles for a diverse array of segmentation problems without the need for extensive hyperparameter tuning.

The structure of this paper is as follows. In Section 2, we present a review of ensemble approaches. In Sections 3 and 4, we introduce and test an ensemble composed

of different convolutional and transformers topologies that achieves SOTA performance. Sections 5 and 6 conclude the discussion with some final remarks.

2. Literature Review

2.1. Ensemble Approaches

As anticipated in Section 1, ensemble methods combine the outputs of multiple classifiers to improve classification performance. Component classifiers are called *base learners* or, sometimes, weak learners, thus highlighting that the performance of the individual components of the ensemble is not decisive. What has been experimentally proven to be crucial is the degree of *diversity* among the ensemble components ([16] and the references therein). In other words, base learners should generalize differently [17] and, first of all, their right and wrong answers on training samples should not be correlated. This key aspect of ensemble learning creates an advantage out of finding that no single classifier works well on all datasets, a fact known as the "no free lunch" theorem. In addition to improved prediction accuracy, other advantages of ensemble methods include the ability to increase performance without additional training data, which are notoriously difficult to obtain in many practical applications, increased robustness to noisy data, and a reduced tendency to overfit the training set [16]. The last advantage is particularly important for deep neural networks, which are prone to overfitting [9].

Ensemble approaches were proposed well before deep learning, with the first scientific works dating to the 1990s [17]. Over more than three decades, several methods, both supervised and semisupervised [18], have been proposed to build ensembles while ensuring diversity, and combining the answers of the base classifiers themselves. As far as building strategies are concerned, two renowned methods are *boosting* [19], where different base learners are trained on the same data, and *bagging* [20], where a single base learner is trained multiple times on different data. In [19], boosting is theoretically analyzed and it is proved that, by "filtering" the data used to learn the classifiers, the error of the ensemble classifier as defined in the PAC model [21,22] can be made smaller than ε with probability $1 - \delta$ for any $0 < \varepsilon < 1/2$. A consequence of the constructive proof is that a labeled sample of size n of any learnable concept can be compressed into a rule of size only polylogarithmic in n. The analysis for bagging in [20] is not entirely quantitative. The fundamental idea is to build multiple training sets of size n by sampling the available n data multiple times, with replacement. This procedure was first introduced in statistics with the name *bootstrapping* [23]. If the training process is "unstable", that is, bootstrapped sets produce quite different classifiers, then the combined output of such classifiers exhibits higher accuracy.

2.2. Ensemble Combination Strategies

As mentioned earlier, different methods have also been proposed to combine the answers of the base classifiers, a crucial step known as *voting*. Popular fusion strategies that are easy to implement in practice are *majority voting* and the *average rule* [9]. The former dictates that the final output of the ensemble is the class on which the maximum number (for nonbinary problems, not necessarily the majority) of base learners agree. For semantic segmentation, majority voting implies that a pixel is assigned to the predicted mask if the majority of the base learners predict so. The average rule, which is applicable when the classification result is a continuous value, stipulates that the final output is the mean of the outputs of the base learners. This strategy is attractive for semantic segmentation, where the output of the learners is typically a per-pixel probability of that pixel belonging to the mask. The average rule is the simplest member in a family of strategies based on the output of the base learners [24]. A prominent variant of the average rule is the *weighted average rule*, where the sum is performed with weights assigned to the base learners according to their performance on the training or validation set.

2.3. Ensembles in Deep Learning

In recent years, ensemble strategies have been successfully applied in deep learning:

- For different tasks, including image classification, detection, and segmentation;
- In several application domains, including healthcare, speech analysis, forecasting, fraud prevention, and information retrieval.

This paper addresses the task of image segmentation in multiple application domains: healthcare, detection of skin and camouflaged objects, gesture recognition, human activity recognition, and portrait segmentation. SOTA results in such domains are reported in Section 4 as baselines for our experiments. For a broader review of ensembles in deep learning, we refer the interested reader to the recent survey in [25].

3. Materials and Methods

In this section, we will outline the methods and techniques used in creating our ensemble models.

In our experimentation, we examine various ensembles constructed from four distinct network architectures. These networks were selected to diversify our feature representations for semantic segmentation. Each network was carefully chosen based on its unique characteristics:

- DeepLabV3+ [26] and HarDNet-MSEG [10] are both convolutional neural network (CNN)-based architectures with different encoder structures, offering distinct feature representations. DeepLabV3+ excels in semantic segmentation, while HarDNet-MSEG provides unique multiscale feature extraction;
- Polyp-PVT [11] represents a transformer-based architecture, offering a different approach to feature extraction and context modeling, which complements CNNs;
- HSNet [12] is a hybrid architecture that combines CNN and transformer components, exploiting the advantages of both, resulting in a broader range of feature representations and contextual information.

These networks are state-of-the-art within their respective categories. By ensembling networks from diverse architectural backgrounds, we harnessed the richness of feature representations, mitigating biases and errors inherent to a single model. This diversity allowed us to capture a wide range of patterns and contexts, enhancing segmentation performance and robustness.

Regarding optimization techniques, we used Adam for HarDNet-MSEG, AdamW for Polyp-PVT and HSNet, and stochastic gradient descent (SGD) for DeepLabV3+, in line with the original papers.

3.1. Loss Functions

The type of loss function used can affect the training and performance of a model in semantic segmentation tasks. One common loss function used is pixel-wise cross-entropy, which evaluates the accuracy of predicted labels at the pixel level. However, this approach can be problematic when the dataset is unbalanced in terms of labels, which can be addressed by using counterweights. In this work, we employed a variety of loss functions for semantic segmentation, each chosen for specific reasons based on its appropriateness in addressing different challenges in the segmentation task. Our primary objective was to establish a diverse array of loss functions rooted in various underlying principles. This approach was pursued with the aim of optimizing the overall performance of our ensemble model. The types of loss functions used in this study can be categorized into the following groups:

- Dice-Based Loss Functions:
  - The Generalized Dice Loss $L_{GD}(Y, T)$ is a multiclass variant of the Dice Loss;
  - The Focal Generalized Dice Loss $L_{FGD}(Y, T)$ is the focal version of the Generalized Dice Loss, emphasizing hard-to-segment regions while downplaying well-segmented areas;

  - The Log-Cosh Dice Loss $L_{lcGD}(Y,T)$ is a combination of the Dice Loss and the Log-Cosh function, applied with the purpose of smoothing the loss curve.
- Tversky-Based Loss Functions:
  - The Tversky Loss $L_T(Y,T)$ is a weighted version of the Tversky index designed to deal with unbalanced classes;
  - The Focal Tversky Loss $L_{FT}(Y,T)$ is a variant of the Tversky loss where a modulating factor is used to ensure that the model focuses on hard samples instead of properly classified examples;
  - The Log-Cosh Focal Tversky Loss $L_{lcFT}(Y,T)$ is based on the same idea of smoothing, here applied to the Focal Tversky Loss.
- Structural Similarity-Based Loss Functions:
  - The SSIM Loss $L_S(Y,T)$ is obtained from the Structural similarity (SSIM) index, usually adopted to evaluate the quality of an image;
  - The MS-SIM Loss $L_{MS}(Y,T)$ is a variant of $L_S(Y,T)$ defined using the multiscale structural similarity (MS-SSIM) index.
- Boundary-Based Loss Functions:
  - The Boundary Enhancement Loss (L_{BE}) explicitly focuses on the boundary areas during training. The Laplacian filter $\mathscr{L}(\cdot)$ is used to generate strong responses around the boundaries and zero everywhere else; see [13] for details. We gather Dice Loss, Boundary Enhancement loss, and the Structure Loss together, weighted appropriately: $L_{DiceBES} = \lambda_1 L_{Dice} + \lambda_2 L_{BE} + L_{Str}$. We set $\lambda_1 = 1$ and $\lambda_2 = 0.01$;
  - The Structure Loss is a combination of the weighted Intersect over Union (L_{wIoU}) and the weighted binary-crossed entropy loss L_{wbce}. We refer the reader to [10] for details. The weights in this loss function are determined by the importance of each pixel, which is calculated from the difference between the center pixel and its surrounding pixels. To give more importance to the binary-crossed entropy loss, we used a weight of 2, as suggested in [10], for it: $L_{STR} = L_{wIoU} + 2L_{wbce}$.
- Combined Loss Functions:
  The losses described above can be combined in different ways; notice that each component has the same weight equal to 1:
  - $Comb_1(Y,T) = L_{FGD}(Y,T) + L_{FT}(Y,T)$,
  - $Comb_2(Y,T) = L_{lcGD}(Y,T) + L_{FGD}(Y,T) + L_{lcFT}(Y,T)$,
  - $Comb_3(Y,T) = L_S(Y,T) + L_{GD}(Y,T)$,
  - $Comb_4(Y,T) = L_{MS}(Y,T) + L_{FGD}(Y,T)$.

These loss functions were selected based on their specific characteristics and suitability for addressing various segmentation challenges. For example, Dice-based loss functions are known for their ability to capture fine details, making them suitable for high-resolution image segmentation. Tversky-based loss functions, on the other hand, are effective in handling class imbalance, making them valuable for datasets with uneven class distributions. Boundary-based losses are designed to focus on the accurate delineation of object boundaries within segmented regions. These losses aim to penalize errors in boundary localization and promote precise edge detection. Lastly, SSIM-based loss functions offer a different perspective by evaluating the structural similarity between the predicted and ground truth masks, which can be beneficial for certain types of segmentation tasks. The choice of these diverse loss functions allows us to take advantage of their unique strengths to effectively address different segmentation problems. This flexibility in loss function selection enhances the robustness and performance of our semantic segmentation model.

For a more detailed description of the set of loss functions, the interested reader can refer to [10,13].

3.2. Data Augmentation

The training of the segmentation network and the final performance of the system are strongly affected by the size of the training set. In order to increase the amount of data available for training a system, various techniques can be applied to the original dataset. In this work, we applied the data augmentation techniques investigated in [12,13]. The choice of these particular data augmentation techniques was informed by both empirical evidence from previous experiments, which demonstrated their effectiveness in enhancing segmentation performance, and the need to enhance diversity among classifiers by incorporating various types of data augmentation.

- Data Augmentation 1 (DA1) [13] is obtained through horizontal flip, vertical flip, and 90° rotation;
- Data Augmentation 2 (DA2) [13] consists of 13 operations, some changing the color of an image and some changing its shape;
- Data Augmentation 3 (DA3) is a variant of the approach used in [12]. It consists of using multiscale strategies (i.e., 1.25, 1, 0.75) to alleviate the sensitivity of the network to scale variability. Simultaneously, random perspective technology is adopted to process the input image with a probability of 0.5, together with random color adjustment with a probability of 0.2 for data augmentation. While DA1 and DA2 do not include randomness, DA3 uses a different training set for each network. The application of this data augmentation technique substantially amplifies result variability within the network, consequently fostering greater diversity among ensemble constituents.

Some artificial images, mainly produced by the DA2 method, contain only background pixels. To discard them, we simply removed all images with fewer than 100 pixels belonging to the foreground class. Moreover, we also discarded images that did not contain background pixels.

3.3. Performance Metrics

As performance indicators, we used two standard metrics: the Dice score and the intersection over union (IoU). These metrics ensure comparability with other works, provide insight into segmentation accuracy, and are suitable for a variety of datasets. The true positives, true negatives, false positives, and false negatives in the formulas below are represented by TP, TN, FP, and FN, respectively. A is the predicted mask and B is the ground truth mask. The Dice score is defined as:

$$F1Score = Dice = \frac{|A \cap B|}{|A| + |B|} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}.$$

The intersection over union (IoU) is defined as:

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}.$$

3.4. Datasets and Testing Protocols

We conducted experiments with our ensembles on nine datasets, selected for our study because they present diverse and well-documented image segmentation challenges. These datasets serve as a demonstration of the applicability and versatility of our ensemble approach to a wide range of image types and applications, offering valuable insights into the model's performance across various scenarios. Additionally, we exclusively chose freely downloadable datasets to establish a benchmark that can be accessed and used by the entire community.

The following subsections include a brief description of the nine datasets used in this work (Figure 1).

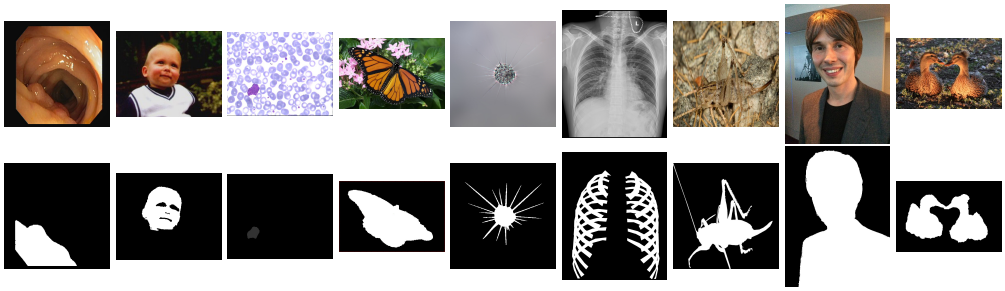

Figure 1. Samples from the nine datasets. Top row: original images. Bottom row: corresponding masks.

3.4.1. Polyp Segmentation (POLYP)

Polyp segmentation in colonoscopy images is a challenging task that involves distinguishing between two classes: polyp pixels and the low-contrast background of the colon. In our study, we conducted experiments on five different datasets widely used [12] for polyp segmentation.

- The Kvasir-SEG dataset comprises medical images that have been meticulously labeled and verified by medical professionals. These images depict various segments of the digestive system, showcasing both healthy and diseased tissue. The dataset encompasses images with varying resolutions, ranging from 720 × 576 pixels to 1920 × 1072 pixels, organized into folders based on their content. Some of these images also include a small picture-in-picture display indicating the position of the endoscope within the body;
- The CVC-ColonDB dataset consists of images designed to offer a diverse range of polyp appearances, maximizing dataset variability;
- CVC-T serves as the test set of a larger dataset named CVC-EndoSceneStill;
- The ETIS-Larib dataset comprises 196 colonoscopy images;
- CVC-ClinicDB encompasses images extracted from 31 videos of colonoscopy procedures. Expert annotations identify the regions affected by polyps, and ground truth data are also available for light reflections. The images in this dataset are uniformly sized at 576 × 768 pixels.

Our training set comprised 1450 images sourced from the largest datasets, with 900 images from Kvasir and 550 images from ClinDB. The remaining images, including 100 from Kvasir, 62 from ClinDB, and all images from ColDB, CVC-T, and ETIS, constituted the test set for our experiments (Table 1). According to previous works [10–12], we used mean Dice (mDic) and mean IoU (mIoU) as performance indicators on this problem.

Table 1. Test set for POLYP.

Short Name	Name	#Samples
Kvasir	Kvasir-SEG dataset	100
ColDB	CVC-ColonDB	380
CVC-T	CVC-EndoSceneStill	300
ETIS	ETIS-Larib	196
ClinicDB	CVC-ClinicDB	612

The polyp datasets are available at https://github.com/james128333/HarDNet-MSEG, (accessed on 5 December 2023).

3.4.2. Skin Segmentation (SKIN)

In the context of skin detection, the segmentation task involves identifying parts of an image that correspond to "skin" or "non-skin", which makes it essentially a binary classification problem. In this paper, we employed the framework introduced in [27] for a fair comparison of skin detection approaches. This framework is based on a small training

set consisting of 2000 images from the ECU dataset [28] and 10 diverse testing datasets, as outlined in Table 2. Following the testing protocol outlined in [27], we calculated the Dice score at the pixel level, not at the image level, and then computed the average score across each dataset; finally, the average Dice score on the test sets was considered.

Table 2. Test set for SKIN. The ECU dataset was split into 2000 images for training and 2000 as a further test set.

Short Name	Name	#Samples
Prat	Pratheepan	78
MCG	MCG-skin	1000
UC	UChile DB-skin	103
CMQ	Compaq	4675
SFA	SFA	1118
HGR	Hand Gesture Recognition	1558
Sch	Schmugge dataset	845
VMD	Human Activity Recognition	285
ECU	ECU Face and Skin Detection	2000
VT	VT-AAST	66

3.4.3. Leukocyte Segmentation (LEUKO)

Leukocyte recognition is the task of segmenting the white blood cells from the background, with the aim of diagnosing many diseases such as leukemia and infections. In our experiments, we used the freely available LISC database [29], which is a collection of 250 hematological images extracted from the peripheral blood of eight healthy people. Images were acquired at a high resolution (720 $\times$ 576 pixels) and manually labeled to segment different types of leukocytes; notice that there is no imbalance in the number of images between different types of leukocytes. In this work, we did not perform classification; therefore, we only consider segmentation performance. The testing protocol, as suggested by the authors of the dataset, is a 10-fold cross-validation approach. LISC is available at https://users.cecs.anu.edu.au/~hrezatofighi/Data/Leukocyte%20Data.htm (accessed on 5 December 2023).

3.4.4. Butterfly Identification (BFLY)

As already proposed in the literature, for butterfly identification, we adopted the public Leeds Butterfly dataset [30]. For a fair comparison with previous results, we used the same testing protocol proposed by the authors of the dataset, that is, a 4-fold cross-validation approach, where each fold includes 208 test images and 624 training images. The dataset is available at https://www.josiahwang.com/dataset/leedsbutterfly/ (accessed on 5 December 2023).

3.4.5. Microorganism Identification (EMICRO)

For the task of identifying microorganisms, we selected the Environmental Microorganism Image Dataset Version 6 (EMDS-6). Proposed in [31], it is a public dataset with 840 images. Following the original paper, we assigned 37.5% of the images to the test set. EMDS-6 is available at https://figshare.com/articles/dataset/EMDS-6/17125025/1 (accessed on 5 December 2023).

3.4.6. Ribs Segmentation (RIBS)

The goal of this application is the semantic segmentation of ribs from chest radiographs. The training and testing samples come from the VinDr-RibCXR dataset [32], which is a small, publicly available dataset for the segmentation and labeling of the anterior and posterior ribs. The dataset contains 245 anteroposterior/posteroanterior chest X-ray images and the corresponding masks, created by human experts. We split the dataset into a training and a test set in the same way as that used by the original authors of [32].

3.4.7. Locust Segmentation (LOC)

The detection and segmentation of locusts is crucial for plant protection robots to effectively capture and eliminate them. However, locusts often have colors and textures that blend in with their surroundings, making it difficult for common segmentation methods to accurately distinguish them. This poses a challenge for efficient locust control. The same dataset used in [33] was tested. There are 874 images in the training set and 120 images in the test set.

3.4.8. Portrait Segmentation (POR)

Portrait segmentation is widely used as a preprocessing step in various applications such as security systems, entertainment, and video conferences. For this study, we utilized the EG1800 dataset [34], which includes 1447 images for training and 289 images for validation. In addition, 62.63% of the pixels belong to the foreground class; thus, it is a fairly balanced dataset. POR can be accessed at https://github.com/HYOJINPARK/ExtPortraitSeg (accessed on 5 December 2023).

3.4.9. Camouflaged Segmentation (CAM)

The CAMO dataset [35] was specifically created to identify and separate camouflaged objects in images. It includes two categories: those that are naturally camouflaged, such as animals, and those that are artificially camouflaged, often corresponding to humans. The dataset contains a total of 1250 images, with 1000 reserved for training and 250 for testing.

4. Experimental Results

Our extensive empirical evaluation aimed to assess the performance of our ensembles. The evaluation was carried out on several real-world datasets, as described in Section 3.4. All networks were trained by resizing the images to a consistent input size. For the test set, we resized the input images to the input dimensions of the network and resized the output masks to the original image size to calculate the performance metrics.

We performed two different sets of tests.

- In Section 4.1, different methods for building an ensemble of DeepLabV3+ models are tested and compared;
- In Section 4.2, the ensemble of different topologies is tested and the different methods for building the output mask of HArdNet, HSN and PVT are compared.

We selected the specific ensemble architectures in our work for their ability to complement each other, combining different strengths and mitigating weaknesses. The diversity of the ensemble members was a key factor in our selection, as it contributes to the overall robustness and adaptability of the ensemble. The experiments between the various topologies are not symmetrical, given the different computation times for training.

4.1. Experiments: DeepLabV3+

In this section, we compare various methods to create a DeeplabV3+ ensemble. The fusion was performed by the average rule if not specified otherwise. The optimization parameters were not modified (i.e., they were the same in all the tested datasets) to prevent overfitting phenomena.

- Initial learning rate = 0.01;
- Number of epoch = 10 or 15 (it depended on data augmentation: see below);
- Momentum = 0.9;
- L2Regularization = 0.005;
- Learning Rate Drop Period = 5;
- Learning Rate Drop Factor = 0.2;
- Shuffle training images at every epoch;
- Optimizer = SGD (stochastic gradient descent).

We tested some backbones for coupling with DeepLabV3+: ResNet18 (RN18) pretrained on ImageNet; ResNet50 (RN50) pretrained on ImageNet; ResNet101 (RN101) pretrained on the VOC segmentation dataset. DeepLabV3+ was trained for 10 epochs if it was coupled with DA1 or for 15 epochs if DA2 was used as the data augmentation approach. Data augmentation approaches are described in Section 3.2. Each ensemble is made up of N models ($N = 1$ denotes a stand-alone model); if not specified, each network differs only for the randomization in the training process (i.e., N different trainings were run).

- ERN18(N) is an ensemble of N RN18 networks trained with DA1;
- ERN50(N) is an ensemble of N RN50 networks trained with DA1;
- ERN101(N) is an ensemble of N RN101 networks trained with DA1;
- E101(10) is an ensemble of 10 RN101 models trained with DA1 and five different loss functions. The final fusion is determined by the formula: $2 \times L_{GD} + 2 \times L_T + 2 \times Comb1 + 2 \times Comb2 + 2 \times Comb3$, where $2 \times L_x$ indicates two RN101 models trained using the loss function L_x;
- EM(10) is a similar ensemble, but the two networks using the same loss (as in E101(10), the five losses are L_{GD}, L_T, $Comb1$, $Comb2$, $Comb3$) were trained once using DA1 and once using DA2;
- EM2(10) is similar to the previous ensemble, but $LDiceBES$ was used instead of L_T;
- In EM2(5)_DAx, five RN101 networks were trained using the loss of EM2(10). All five networks were trained using data augmentation DAx;
- EM3(10) is similar to the previous ensemble, but L_{STR} was used as a loss function.

The results of the experiments are provided in Table 3 and can be summarized as follows:

- Among stand-alone networks, RN101 obtained the best average performance, but in RIBS (a small training set), it performed worse than the others. This probably happened because it is a larger network than RN18 and RN50, thus it requires a larger training set for better tuning;
- ERN101(10) always outperformed RN101(1);
- E101(10) outperformed ERN101(10) with a p-value of 0.0078 (Wilcoxon signed rank test) and EM(10) outperformed E101(10) with a p-value of 0.0352. For the sake of space, we have not reported the performance obtained from individual losses. In any case, there was no winner: the various losses led to similar performances;
- EM3(10) obtained the highest average performance, but the p-value was quite high: it outperformed EM(10) with a p-value of 0.1406 and EM2(10) with a p-value of 0.2812;
- There was no statistical difference between the performance of EM2(5)_DA1 and EM2(5)_DA2. Instead, EM2, using both data augmentation methods, achieved better performance (on average) than EM2(5)_DA1 and EM2(5)_DA2.

Table 3. Dice scores for the proposed DeepLabV3+ ensembles on the nine benchmark datasets. The best performance metrics for each dataset are highlighted in bold.

	POLYP	SKIN	LEUKO	BFLY	EMICRO	RIBS	LOC	POR	CAM
RN18(1)	0.806	0.865	0.897	0.960	0.908	0.827	0.812	0.980	0.624
RN50(1)	0.802	0.871	0.895	0.968	0.909	0.818	0.835	0.979	0.665
RN101(1)	0.808	0.871	0.915	0.976	0.918	0.776	0.830	0.981	0.717
ERN18(10)	0.821	0.866	0.913	0.963	0.913	0.842	0.830	0.981	0.672
ERN50(10)	0.807	0.872	0.897	0.969	0.918	0.839	0.840	0.980	0.676
ERN101(10)	0.834	0.878	0.925	0.978	0.919	0.779	0.838	0.982	0.734
E101(10)	0.842	0.880	0.925	0.980	0.921	0.785	0.841	0.984	**0.747**
EM(10)	0.851	**0.883**	0.936	0.983	0.924	0.833	0.854	0.985	0.740
EM2(10)	0.851	**0.883**	0.943	0.984	**0.925**	0.846	0.859	**0.986**	0.731
EM2(5)_DA1	0.836	0.881	0.928	0.982	0.921	0.800	0.841	0.985	0.742
EM2(5)_DA2	0.847	0.869	**0.948**	**0.985**	0.920	**0.860**	0.842	0.983	0.700
EM3(10)	**0.852**	**0.883**	0.945	**0.985**	**0.925**	0.856	**0.860**	**0.986**	0.728

The IoU performance indicator is only reported in Table 4 for the best ensembles. Using IoU, we confirmed the conclusions obtained with Dice, i.e., EM3(10) obtained the highest average performance but the p-value was quite high; it outperformed EM(10) with a p-value of 0.1484 and EM2(10) with a p-value of 0.2656.

Table 4. IoU for the best DeepLabV3+ ensembles on the nine benchmark datasets. The best performance metrics for each dataset are highlighted in bold.

	POLYP	SKIN	LEUKO	BFLY	EMICRO	RIBS	LOC	POR	CAM
EM(10)	0.787	0.798	0.887	0.966	0.869	0.714	0.769	0.971	**0.630**
EM2(10)	0.790	**0.799**	0.897	0.969	0.870	0.734	0.778	**0.972**	0.621
EM3(10)	**0.791**	0.798	**0.899**	**0.970**	**0.872**	**0.749**	**0.780**	**0.972**	0.617

All these conclusions were obtained using a range of diverse datasets, so we are fairly confident that these results are reliable.

4.2. Experiments: Combining Different Topologies

Each network was trained end-to-end for 50 epochs, with a batch size of 20. HardNet-MSEG, PVT, and HSNet were trained using the structure loss function and the following learning rates:

- LRa: 10^{-4};
- LRb: 5×10^{-4} decaying to 5×10^{-5} after 10 epochs;
- LRc: 5×10^{-5} decaying to 5×10^{-6} after 30 epochs.

We removed the normalization layer from the HardNet, PVT, and HSN models. In the original versions of these models, the segmentation maps are normalized between 0 and 1 before being output, even though there are no foreground pixels in the image. However, this assumption may not hold for all datasets. As a result, the segmentation results obtained using the modified HardNet, PVT, and HSN models may differ slightly from the original results. Additionally, we changed the way the final segmentation maps are processed in the PVT and HSN models. In the original versions, the maps are summed and then passed through a sigmoid function: this saturates the sigmoid and the network output is very sharp; hence, the average rule among outputs of HSNs and PVTs is almost like a voting rule. In our modified versions (named SM), we pass each map separately through the sigmoid and average the results. Our output is given by:

$$\sum_{i=1}^{n_S} sigmoid(P_i)/n_S,$$

where P_i is a segmentation map and n_S is the number of segmentation maps of the topology.

Tables 5–7 report the performance of the three networks (that is, HardNet-MSEG, PVT, and HSNet) by varying the data augmentation (DA) and the learning rate (LR) on four problems. For Tables 6 and 7, the SM column indicates whether we were using the original output of HSN and PVT (SM = No) or the segmentation maps we previously described (SM = Yes). The last rows of Tables 5–7 report the performance of the following ensembles:

- Fusion: the combination of all the nets while varying the DA and LR strategy;
- Baseline Ensemble: fusion between nine networks (the same size of Fusion) obtained via retraining DA3-LRc nine times;
- SOTAEns: The best ensemble, related to a given topology, previously reported in [13–15]. It is important to note that, in this way, we show the comparison with the best previous results of that network in that dataset, on average, improving on the previous results.

Table 5. Dice scores obtained by the HardNet based ensembles. The best performance metrics for each dataset are highlighted in bold.

	DA	LR	POLYP	SKIN	EMICRO	CAM
HardNet	DA1	LRa	0.828	0.873	0.912	0.700
		LRb	0.821	0.858	0.905	0.667
		LRc	0.795	0.869	0.909	0.712
HardNet	DA2	LRa	0.852	0.870	0.912	0.715
		LRb	0.826	0.854	0.905	0.665
		LRc	0.846	0.872	0.910	0.710
HardNet	DA3	LRa	0.828	0.853	0.907	0.653
		LRb	0.832	0.839	0.904	0.613
		LRc	0.828	0.865	0.904	0.694
Fusion	DA1,2,3	LRa,b,c	**0.868**	0.883	**0.921**	**0.726**
SOTAEns			0.863	**0.886**	0.916	—

Table 6. Dice scores obtained by the PVT based ensembles. The best performance metrics for each dataset are highlighted in bold.

	DA	LR	SM	POLYP	SKIN	EMICRO	CAM
PVT	DA1	LRa	No	0.857	0.874	0.919	0.788
		LRb	No	0.850	0.844	0.914	0.743
		LRc	No	0.861	0.877	0.919	0.810
PVT	DA2	LRa	No	0.862	0.845	0.917	0.742
		LRb	No	0.847	0.854	0.912	0.743
		LRc	No	0.862	0.876	0.917	0.813
PVT	DA3	LRa	No	0.855	0.875	0.917	0.765
		LRb	No	0.851	0.856	0.916	0.718
		LRc	No	0.871	0.883	0.918	0.817
Fusion	DA1,2,3	LRa,b,c	No	0.884	**0.892**	0.925	0.813
Fusion	DA1,2,3	LRa,b,c	Yes	**0.885**	**0.892**	**0.926**	0.814
Baseline Ensemble	DA3	LRc		0.880	0.886	0.921	**0.829**
SOTAEns				0.877	0.883	0.922	—

Table 7. Dice scores obtained by the HSN based ensembles. The best performance metrics for each dataset are highlighted in bold.

	DA	LR	SM	POLYP	SKIN	EMICRO	CAM
HSN	DA1	LRa	No	0.847	0.873	0.919	0.776
		LRb	No	0.852	0.816	0.916	0.742
		LRc	No	0.860	0.873	0.919	0.817
HSN	DA2	LRa	No	0.857	0.873	0.921	0.742
		LRb	No	0.849	0.850	0.918	0.743
		LRc	No	0.873	0.873	0.919	0.814
HSN	DA3	LRa	No	0.866	0.863	0.922	0.782
		LRb	No	0.854	0.856	0.913	0.697
		LRc	No	0.866	0.876	0.924	0.800
Fusion	DA1,2,3	LRa,b,c	No	0.881	0.885	**0.926**	0.813
Fusion	DA1,2,3	LRa,b,c	Yes	**0.882**	**0.886**	**0.926**	0.812
Baseline Ensemble	DA3	LRc		0.876	0.879	0.923	**0.820**
SOTAEns				0.879	0.879	—	—

The conclusions that can be drawn from the results in the tables are as follows:

- Fusion obtained the best performance, outperforming (on average) the stand-alone approaches and previous best ensemble (SOTAEns);
- There was no clear winner among the different data augmentation approaches and learning rate strategies;
- The proposed Fusion ensemble always improved the Baseline Ensemble except in CAMO. In this dataset, there was a significant difference in the performance between LRc and the other learning strategies; combining only the three networks based on LRc (i.e., using the three data augmentations coupled with LRc), both HS and PVT obtained a Dice of 0.830, outperforming the Baseline Ensemble.

In summary, the data in the aforementioned tables suggest that using the proposed ensemble segmentation method improved the performance of previous HSN and PVT ensembles.

In Table 8, our ensembles are compared with the state of the art (SOTA) reported in the literature. In our final proposed ensembles, the methods were combined with the weighted average rule: weight 1 for EM3 and Fusion(FH); weight 2 for Fusion(PVT) and Fusion(HSN). We report the performance of the following ensembles:

- Ens1: EM3(10) $\ominus$ Fusion(FH) $\ominus$ Fusion(PVT) $\ominus$ Fusion(HSN). See Figure 2;
- Ens2: Fusion(FH) $\ominus$ Fusion(PVT) $\ominus$ Fusion(HSN);
- Ens3: Fusion(PVT) $\ominus$ Fusion(HSN).

It is clear that combining different network architectures led to higher performance than with a single topology. Moreover, we obtained SOTA performance. For instance, in EMicro, the authors of the dataset reported a Dice score of 0.884, and our ensembles obtained a higher 0.927 Dice score. Compared with previous work, we standardized the data augmentation step, which was previously implemented in different languages for CNNs (Matlab) and transformers (Python); in this work, we only used the data augmentation created by Matlab. This led to small differences in performance, e.g., the implementation used in this paper of the data augmentation method detailed in [14] obtained an average Dice of 0.891 instead of 0.895, so the method proposed in this work is our suggested ensemble. In addition, this ensemble was tested on four datasets, so we are more confident that the proposed approach will perform well in other datasets.

Some inference masks are shown in Figure 3. They demonstrate that our ensemble model produces better boundary results and makes more accurate predictions with respect to the best stand-alone net (PVT). Finally, Table 9 presents a performance comparison between the proposed ensemble Ens2 and recent models developed and available in the literature for the polyp segmentation problem. Compared to the literature, considering average performance, the proposed ensemble outperformed the methods proposed in the literature. In Table 10, the skin segmentation performance for each of the 10 skin datasets is reported for Ens2 and stand-alone networks; each stand-alone network was coupled with DA3 and LRc. It is clear that Ens2 strongly outperformed the stand-alone approaches.

Table 8. Comparison with previous SOTA ensembles: Dice scores.

	POLYP	SKIN	EMICRO	CAM
Ens1	0.886	0.892	0.927	0.817
Ens2	0.887	0.893	0.927	0.812
Ens3	0.886	0.894	0.927	0.805
[13]	0.874	0.893	0.926	—
[14]	—	0.895	—	—
[15]	0.885	—	—	—

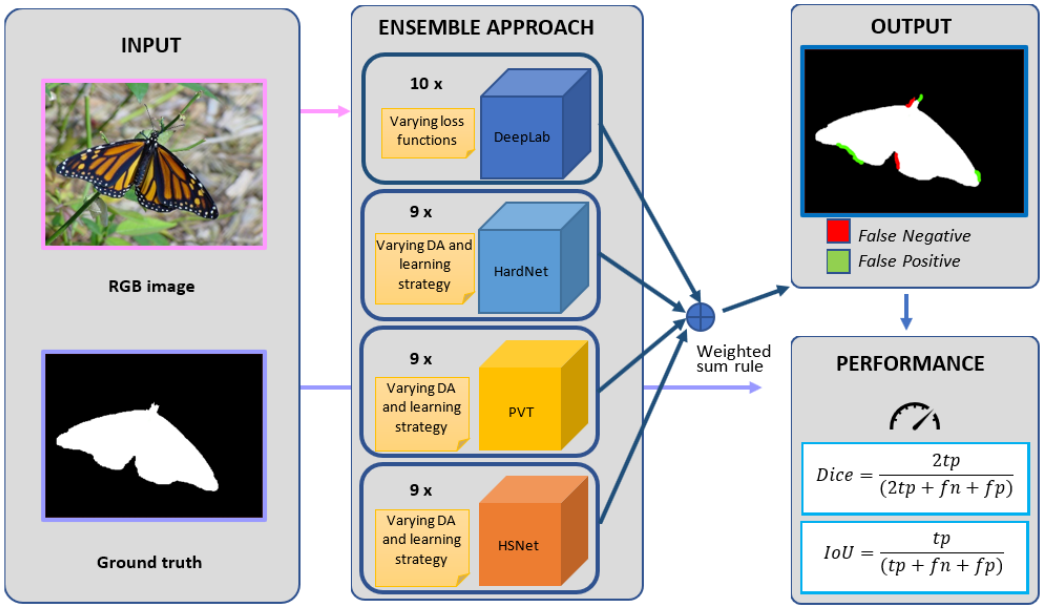

Figure 2. Schema of ensemble Ens1.

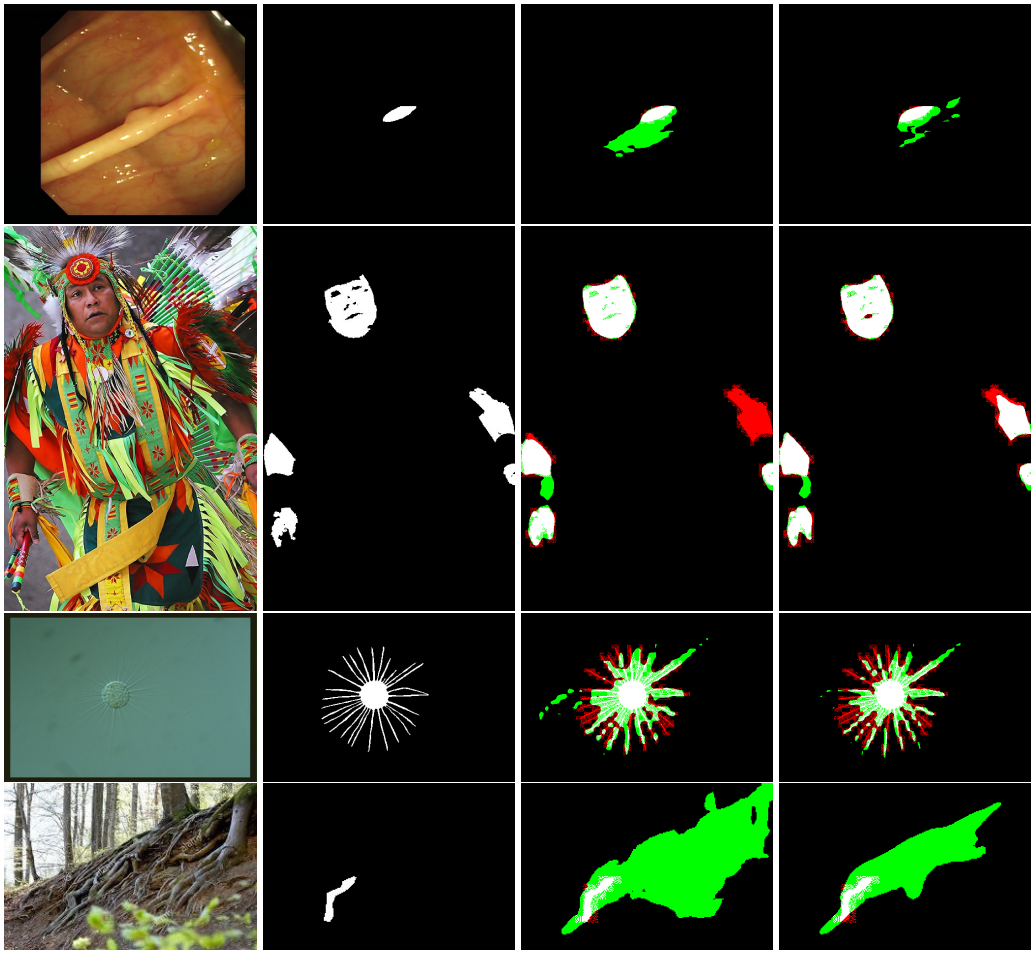

Figure 3. Segmentation results on the polyp, skin, EMICRO, and CAMO datasets; each line contains original images, ground truth, result from PVT_DA3_LRc, and Ens2. False-positive pixels are in green, while the false negatives are in red.

Table 9. Comparison of performance between the proposed ensembles and recent models developed and available in the literature for the polyp segmentation problem. The best performance metrics for each dataset are highlighted in bold.

Method	Kvasir		ClinDB		ColDB		ETIS		CVC-T		*Average*	
	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice
Ens2	0.883	0.927	0.893	0.935	**0.766**	**0.840**	**0.762**	**0.833**	0.834	0.899	**0.828**	**0.887**
HSNet [12]	0.877	0.926	0.905	0.948	0.735	0.81	0.734	0.808	0.839	0.903	0.818	0.879
MIA-Net [36]	0.876	0.926	0.899	0.942	0.739	0.816	0.725	0.8	0.835	0.9	0.815	0.877
P2T [37]	0.849	0.905	0.873	0.923	0.68	0.761	0.631	0.7	0.805	0.879	0.768	0.834
DBMF [38]	**0.886**	**0.932**	0.886	0.933	0.73	0.803	0.711	0.79	**0.859**	**0.919**	0.814	0.875
HarDNet [10]	0.857	0.912	0.882	0.932	0.66	0.731	0.613	0.677	0.821	0.887	0.767	0.828
PraNet, from [10]	0.84	0.898	0.849	0.899	0.64	0.709	0.567	0.628	0.797	0.871	0.739	0.801
SFA, from [10]	0.611	0.723	0.607	0.7	0.347	0.469	0.217	0.297	0.329	0.467	0.422	0.531
U-Net++, from [10]	0.743	0.821	0.729	0.794	0.41	0.483	0.344	0.401	0.624	0.707	0.57	0.641
U-Net, from [10]	0.746	0.818	0.755	0.823	0.444	0.512	0.335	0.398	0.627	0.71	0.581	0.652
SETR [39]	0.854	0.911	0.885	0.934	0.69	0.773	0.646	0.726	0.814	0.889	0.778	0.847
TransUnet [40]	0.857	0.913	0.887	0.935	0.699	0.781	0.66	0.731	0.824	0.893	0.785	0.851
TransFuse [41]	0.87	0.92	0.897	0.942	0.706	0.781	0.663	0.737	0.826	0.894	0.792	0.855
UACANet [42]	0.859	0.912	0.88	0.926	0.678	0.751	0.678	0.751	0.849	0.91	0.789	0.85
SANet [43]	0.847	0.904	0.859	0.916	0.67	0.753	0.654	0.75	0.815	0.888	0.769	0.842
MSNet [44]	0.862	0.907	0.879	0.921	0.678	0.755	0.664	0.719	0.807	0.869	0.778	0.834
Polyp-PVT [11]	0.864	0.917	0.889	0.937	0.727	0.808	0.706	0.787	0.833	0.9	0.804	0.869
SwinE-Net [45]	0.87	0.92	0.892	0.938	0.725	0.804	0.687	0.758	0.842	0.906	0.803	0.865
AMNet [46]	0.865	0.912	0.888	0.936	0.69	0.762	0.679	0.756	-	-	-	-
MGCBFormer [47]	0.885	0.931	**0.915**	**0.955**	0.731	0.807	0.747	0.819	0.851	0.913	0.826	0.885

Table 10. Comparison of performance between the proposed ensembles and recent models for the skin detection problem. The best performance metrics for each dataset are highlighted in bold.

Method	Prat	MCG	UC	CMQ	SFA	HGR	Sch	VMD	ECU	VT	*AVG*
Ens2	**0.928**	0.896	**0.913**	**0.870**	**0.956**	**0.972**	0.804	**0.770**	**0.956**	**0.861**	**0.893**
HardNet	0.908	0.881	0.911	0.832	0.948	0.962	0.772	0.661	0.942	0.832	0.865
PVT	0.919	0.891	0.906	0.860	0.950	0.970	**0.806**	0.726	0.950	0.849	0.883
HSN	0.921	**0.898**	0.908	0.854	0.954	0.966	0.778	0.659	0.951	0.860	0.876

5. Discussion

The results presented in Section 4 offer valuable insight into the effectiveness of our proposed ensemble segmentation approach. In this dedicated discussion, we dive deeper into these findings and their implications.

5.1. Performance and Ensemble Comparison

Our experiments demonstrate that the fusion-based ensemble consistently outperformed both the stand-alone approaches and the previous ensemble method. This robust performance improvement is a key highlight of our research. The fusion approach, which combines diverse network architectures, models, and data augmentation techniques, showcased its potential as a powerful tool to improve segmentation accuracy across a wide range of problems. This finding underlines the adaptability and utility of ensemble methods, particularly those that draw upon multiple sources of variation.

5.2. Data Augmentation and Learning Rate Strategies

While we explored various data augmentation techniques and learning rate strategies, there was no definitive standout approach across all datasets. The choice of data augmentation and learning rate strategy did not yield a clear winner, indicating the complex and dataset-specific nature of these decisions. However, our ensemble approach was demon-

strated to be effective regardless of these variations, emphasizing its robustness and ability to mitigate the need for extensive hyperparameter tuning.

5.3. Comparative Analysis with the State of the Art

In Table 8, we provide a comprehensive comparison of our ensembles with the SOTA results reported in the literature. Our final proposed ensembles combine multiple methods using a weighted average rule, leading to significant performance gains. It is evident that incorporating different network architectures within our ensembles results in superior performance compared to single-topology ensembles. Furthermore, our approach achieved new SOTA performance in multiple segmentation tasks, underlining the contributions and advancements made by our research.

5.4. Overall Contribution

In summary, our research introduces a novel ensemble segmentation method that leverages the fusion of diverse network architectures, models, and data augmentation techniques. The results illustrate the method's potential for improving segmentation accuracy without extensive hyperparameter tuning. The ability of our ensembles to consistently outperform previous ensembles and achieve new SOTA performance is a significant contribution to the field of image segmentation. Moreover, the findings support the notion that combining diverse network topologies enhances segmentation outcomes. The results reported in this study provide valuable insights and practical guidance for researchers and practitioners when selecting and composing ensembles for image segmentation tasks. Our work emphasizes the adaptability and robustness of ensemble methods and underscores the potential for their broader application in various domains.

6. Conclusions

Many interesting results were obtained in this work. However, it is essential to acknowledge the limitations of our method to provide a more balanced perspective on the research outcomes. While our experiments produced promising results, we recognize that the generalization of these findings to other application domains may have constraints. To address these limitations, further tests and evaluations will be conducted in the future, with the aim of confirming the following:

- The fusion of different convolutional and transformer networks can achieve state-of-the-art (SOTA) performance;
- The application of diverse approaches to the learning rate strategy is a viable method to build a set of segmentation networks;
- The integration of transformers (HSN and PVT) in an ensemble can be enhanced by modifying the way the final segmentation map is obtained, thereby avoiding excessively sharp masks.

As part of our future work, we also plan to explore techniques such as pruning, quantization, low-ranking factorization, and distillation to reduce the complexity of the ensembles and address potential scalability issues.

Author Contributions: Conceptualization, L.N., C.F. and A.L.; software, L.N. and A.L.; writing—original draft preparation, C.F., A.L. and L.N.; writing—review and editing, C.F., L.N. and A.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All the resources required to replicate our experiments are available at https://github.com/LorisNanni (accessed on 5 December 2023).

Acknowledgments: We would like to acknowledge the support that NVIDIA provided us through the GPU Grant Program. We used a donated TitanX GPU to train the neural networks discussed in this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hao, S.; Zhou, Y.; Guo, Y. A brief survey on semantic segmentation with deep learning. *Neurocomputing* **2020**, *406*, 302–321. [CrossRef]
2. Wang, S.; Mu, X.; Yang, D.; He, H.; Zhao, P. Attention guided encoder-decoder network with multi-scale context aggregation for land cover segmentation. *IEEE Access* **2020**, *8*, 215299–215309. [CrossRef]
3. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
4. Siddique, N.; Paheding, S.; Elkin, C.P.; Devabhaktuni, V. U-Net and its variants for medical image segmentation: A review of theory and applications. *IEEE Access* **2021**, *9*, 82031–82057. [CrossRef]
5. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef] [PubMed]
6. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
7. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:cs.CV/2010.11929.
8. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 568–578.
9. Mohammed, A.; Kora, R. A comprehensive review on ensemble deep learning: Opportunities and challenges. *J. King Saud Univ. Comput. Inf. Sci.* **2023**, *35*, 757–774. [CrossRef]
10. Huang, C.H.; Wu, H.Y.; Lin, Y.L. HarDNet-MSEG: A Simple Encoder-Decoder Polyp Segmentation Neural Network that Achieves over 0.9 Mean Dice and 86 FPS. *arXiv* **2021**, arXiv:cs.CV/2101.07172.
11. Dong, B.; Wang, W.; Fan, D.P.; Li, J.; Fu, H.; Shao, L. Polyp-PVT: Polyp Segmentation with Pyramid Vision Transformers. *arXiv* **2023**, arXiv:eess.IV/2108.06932.
12. Zhang, W.; Fu, C.; Zheng, Y.; Zhang, F.; Zhao, Y.; Sham, C.W. HSNet: A hybrid semantic network for polyp segmentation. *Comput. Biol. Med.* **2022**, *150*, 106173. [CrossRef]
13. Nanni, L.; Lumini, A.; Loreggia, A.; Formaggio, A.; Cuza, D. An Empirical Study on Ensemble of Segmentation Approaches. *Signals* **2022**, *3*, 341–358. [CrossRef]
14. Nanni, L.; Loreggia, A.; Lumini, A.; Dorizza, A. A Standardized Approach for Skin Detection: Analysis of the Literature and Case Studies. *J. Imaging* **2023**, *9*, 35. [CrossRef] [PubMed]
15. Nanni, L.; Fantozzi, C.; Loreggia, A.; Lumini, A. Ensembles of Convolutional Neural Networks and Transformers for Polyp Segmentation. *Sensors* **2023**, *23*, 4688. [CrossRef] [PubMed]
16. Rokach, L. Ensemble-based classifiers. *Artif. Intell. Rev.* **2010**, *33*, 1–39. [CrossRef]
17. Polikar, R. Ensemble Based Systems in Decision Making. *IEEE Circuits Syst. Mag.* **2006**, *6*, 21–45. [CrossRef]
18. Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; Ma, Q. A survey on ensemble learning. *Front. Comput. Sci.* **2020**, *14*, 241–258. [CrossRef]
19. Schapire, R.E. The strength of weak learnability. *Mach. Learn.* **1990**, *5*, 197–227. [CrossRef]
20. Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
21. Valiant, L.G. A Theory of the Learnable. *Commun. ACM* **1984**, *27*, 1134–1142. [CrossRef]
22. Kearns, M.; Valiant, L.G. Cryptographic Limitations on Learning Boolean Formulae and Finite Automata. *J. ACM* **1994**, *41*, 67–95. [CrossRef]
23. Efron, B. Bootstrap Methods: Another Look at the Jackknife. *Ann. Stat.* **1979**, *7*, 1–26. [CrossRef]
24. Alexandre, L.A.; Campilho, A.C.; Kamel, M. On combining classifiers using sum and product rules. *Pattern Recognit. Lett.* **2001**, *22*, 1283–1289. [CrossRef]
25. Ganaie, M.A.; Hu, M.; Malik, A.K.; Tanveer, M.; Suganthan, P.N. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105151. [CrossRef]
26. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the Computer Vision—ECCV 2018: 15th European Conference, Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 833–851. [CrossRef]
27. Lumini, A.; Nanni, L. Fair comparison of skin detection approaches on publicly available datasets. *Expert Syst. Appl.* **2020**, *160*, 113677. [CrossRef]
28. Phung, S.L.; Bouzerdoum, A.; Chai, D. Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 148–154. [CrossRef] [PubMed]
29. Liu, Y.; Cao, F.; Zhao, J.; Chu, J. Segmentation of White Blood Cells Image Using Adaptive Location and Iteration. *IEEE J. Biomed. Health Inform.* **2017**, *21*, 1644–1655. [CrossRef]
30. Filali, I.; Achour, B.; Belkadi, M.; Lalam, M. Graph ranking based butterfly segmentation in ecological images. *Ecol. Inform.* **2022**, *68*, 101553. [CrossRef]

31. Zhao, P.; Li, C.; Rahaman, M.M.; Xu, H.; Ma, P.; Yang, H.; Sun, H.; Jiang, T.; Xu, N.; Grzegorzek, M. EMDS-6: Environmental Microorganism Image Dataset Sixth Version for Image Denoising, Segmentation, Feature Extraction, Classification, and Detection Method Evaluation. *Front. Microbiol.* **2022**, *13*, 829027. [CrossRef]
32. Nguyen, H.C.; Le, T.T.; Pham, H.H.; Nguyen, H.Q. VinDr-RibCXR: A Benchmark Dataset for Automatic Segmentation and Labeling of Individual Ribs on Chest X-Rays. *arXiv* **2021**, arXiv:eess.IV/2107.01327.
33. Liu, L.; Liu, M.; Meng, K.; Yang, L.; Zhao, M.; Mei, S. Camouflaged locust segmentation based on PraNet. *Comput. Electron. Agric.* **2022**, *198*, 107061. [CrossRef]
34. Park, H.; Sjösund, L.L.; Yoo, Y.; Kwak, N. ExtremeC3Net: Extreme Lightweight Portrait Segmentation Networks using Advanced C3-modules. *arXiv* **2019**, arXiv:cs.CV/1908.03093.
35. Yan, J.; Le, T.N.; Nguyen, K.D.; Tran, M.T.; Do, T.T.; Nguyen, T.V. MirrorNet: Bio-Inspired Camouflaged Object Segmentation. *IEEE Access* **2021**, *9*, 43290–43300. [CrossRef]
36. Li, W.; Zhao, Y.; Li, F.; Wang, L. MIA-Net: Multi-information aggregation network combining transformers and convolutional feature learning for polyp segmentation. *Knowl.-Based Syst.* **2022**, *247*, 108824. [CrossRef]
37. Wu, Y.H.; Liu, Y.; Zhan, X.; Cheng, M.M. P2T: Pyramid Pooling Transformer for Scene Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *11*, 12760–12771. [CrossRef] [PubMed]
38. Liu, F.; Hua, Z.; Li, J.; Fan, L. DBMF: Dual Branch Multiscale Feature Fusion Network for polyp segmentation. *Comput. Biol. Med.* **2022**, *151*, 106304. [CrossRef] [PubMed]
39. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H.; et al. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 6877–6886. [CrossRef]
40. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *arXiv* **2021**, arXiv:2102.04306. [CrossRef]
41. Zhang, Y.; Liu, H.; Hu, Q. TransFuse: Fusing Transformers and CNNs for Medical Image Segmentation. In *Proceedings of the Medical Image Computing and Computer Assisted Intervention—MICCAI 2021*; de Bruijne, M., Cattin, P.C., Cotin, S., Padoy, N., Speidel, S., Zheng, Y., Essert, C., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 14–24. [CrossRef]
42. Kim, T.; Lee, H.; Kim, D. UACANet: Uncertainty Augmented Context Attention for Polyp Segmentation. In Proceedings of the 29th ACM International Conference on Multimedia, MM'21, Virtual Event, 20–24 October 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 2167–2175. [CrossRef]
43. Wei, J.; Hu, Y.; Zhang, R.; Li, Z.; Zhou, S.K.; Cui, S. Shallow Attention Network for Polyp Segmentation. In Proceedings of the Lecture Notes in Computer Science, Strasbourg, France, 27 September–1 October 2021; Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics; Springer: Berlin/Heidelberg, Germany, 2021; Volume 12901. [CrossRef]
44. Zhao, X.; Zhang, L.; Lu, H. Automatic Polyp Segmentation via Multi-scale Subtraction Network. *arXiv* **2021**, arXiv:2108.05082. [CrossRef]
45. Park, K.B.; Lee, J.Y. SwinE-Net: Hybrid deep learning approach to novel polyp segmentation using convolutional neural network and Swin Transformer. *J. Comput. Des. Eng.* **2022**, *9*, 616–632. [CrossRef]
46. Song, P.; Li, J.; Fan, H. Attention based multi-scale parallel network for polyp segmentation. *Comput. Biol. Med.* **2022**, *146*, 105476. [CrossRef]
47. Xia, Y.; Yun, H.; Liu, Y.; Luan, J.; Li, M. MGCBFormer: The multiscale grid-prior and class-inter boundary-aware transformer for polyp segmentation. *Comput. Biol. Med.* **2023**, *167*, 107600. [CrossRef]