Article

Not peer-reviewed version

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

Loris Nanni * , Alessandra Lumini , Carlo Fantozzi

Posted Date: 10 October 2023

doi: 10.20944/preprints202310.0572.v1

Keywords: deep learning; ensembles; segmentation; transformers doi:10.20944/preprints202310.0572.v1

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, Padova, Italy; {loris.nanni,carlo.fantozzi}@unipd.it
[2] Department of Computer Science and Engineering, University of Bologna, 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, data augmentation method or the learning rate strategy. Our proposed ensemble, which is based on the simple average rule, demonstrates exceptional performance on several datasets. All the resources used in this study are available online at the following GitHub repository: https://github.com/LorisNanni.

Keywords: deep learning; ensembles; segmentation; transformers

1. Introduction

Image semantic segmentation [1] involves dividing an image into distinct, non-overlapping 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 vehicles 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 were 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 which utilizes atrous (dilated) convolutions to capture multi-scale 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 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.

The structure of the paper is as follows. In Section 2, we present a review on ensemble approaches. In Sections 3 and 4, we introduce and test an ensemble composed of different convolutional and transformers topologies that achieves SOTA performance. Section 5 concludes the discussion with some final remarks.

2. 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 ([13] and references therein). In other words, base learners should generalize differently [14] 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 the 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 [13]. The last advantage is particularly important for deep neural networks, which are prone to overfitting [9].

Ensemble approaches were born well before deep learning, with the first scientific works dating to the 1990s [14]. Over more than three decades, several methods, both supervised and semi-supervised [15], have been proposed to build ensembles while ensuring diversity, and combine the answers of the base classifiers themselves. As far as building strategies are concerned, two renowned methods are *boosting* [16], where different base learners are trained on the same data, and *bagging* [17], where a single base learner is trained multiple times on different data. In [16], 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 [18,19] 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 poly-logarithmic in n. The analysis for bagging in [17] 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* [20]. If the training process is “unstable”, that is, bootstrapped sets produce quite different classifiers, then the combined output of such classifiers exhibits higher accuracy.

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 non-binary 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 [21]. 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.

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 [22].

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 the following networks:

- As CNN-based architectures, we use the well-known DeepLabV3+ [23] and HarDNet-MSEG [10].
- As transformer-based architecture, we use Polyp-PVT [11];

- As hybrid CNN/transformer-based architecture, we use HSNet [12].

Regarding optimization techniques, we use 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.

For a more detailed description of the set of loss functions see [24] and [10], the interested reader can refer to them.

- The Generalized Dice Loss $L_{GD}(Y,T)$ is a multiclass variant of the Dice Loss.
- The Tversky Loss $L_T(Y,T)$ is a weighed version of the Twersky index designed to deal with unbalanced classes.
- The Focal Tversky Loss $L_{FT}(Y,T)$ is a variant of the Twersky loss where a modulating factor is used to ensure that the model focuses on hard samples instead of properly classified examples.
- The Focal Generalized Dice Loss $L_{FGD}(Y,T)$ is the focal version of the Generalized Dice Loss.
- 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.
- 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.
- 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.
- The losses described above can be combined in different ways:
  - $Comb_1(Y,T) = L_{FGD}(Y,T) + L_{FT}(Y,T)$,
  - $Comb_2(Y,T) = L_{lcGD}(Y,T) + L_{FGD}(Y,T) + L_{lcFT}(Y,T)$,
  - $Comb_3(Y,T) = L_S(Y,T) + L_{GD}(Y,T)$,
  - $Comb_4(Y,T) = L_{MS}(Y,T) + L_{FGD}(Y,T)$.
- 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. 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 use a weight of 2 for it: $L_{STR} = L_{wIoU} + 2L_{wbce}$.

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 apply the data augmentation techniques investigated in [12] and [24].

- Data Augmentation 1 (DA1) [24] is obtained through horizontal flip, vertical flip, and 90° rotation;
- Data Augmentation 2 (DA2) [24] 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 in using multi-scale 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 remove all images with fewer than 100 pixels belonging to the foreground class. Moreover, we also discard images that do not contain background pixels.

3.3. Performance Metrics

As performance indicators, we have used two standard metrics: the Dice score and the Intersection over Union (IoU). 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

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 720x576 pixels to 1920x1072 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 is also available for light reflections. The images in this dataset are uniformly sized at 576x768 pixels.

Our training set comprises 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, constitute the test set for our experiments (Table 1).

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

According to previous works [10–12], we use mean Dice (mDic), mean IoU (mIoU) as performance indicators on this problem. The polyp datasets are available at https://github.com/james128333/HarDNet-MSEG.

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 adopt the framework introduced in [25], which relies on a small training set consisting of 2000 images from the ECU dataset [26]. Additionally, we evaluate the performance of the framework on ten diverse testing datasets, as outlined in Table 2. Following the testing protocol outlined in [25], we calculate the Dice score at the pixel level, not at the image level, and then compute the average score across each dataset; finally the average Dice score on the 10 test sets is considered.

Table 2. Test set for SKIN. The ECU dataset is 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 use the freely available LISC database [27], which is a collection of 250 hematological images extracted from the peripheral blood of eight healthy people. Images have been acquired at high resolution (720×576 pixels) and manually labelled to segment 10 different types of leukocytes. In this work, we do not perform classification, therefore we consider only segmentation performance. The testing protocol, as suggested by the authors of the dataset, is a 10-fold cross-validation. LISC is available at https://users.cecs.anu.edu.au/~hrezatofighi/Data/Leukocyte%20Data.htm.

3.4.4. Butterfly Identification (BFLY)

As already done in the literature, for butterfly identification we adopt the public Leeds Butterfly dataset [28]. For a fair comparison with previous results, we use the same testing protocol proposed by the authors of the dataset, that is, a 4-fold cross-validation, where each fold includes 208 test images and 624 training images. The dataset is available at https://www.josiahwang.com/dataset/leedsbutterfly/.

3.4.5. Microorganism Identification (EMICRO)

For the task of identifying microorganisms we select the Environmental Microorganism Image Dataset Version 6 (EMDS-6). Proposed in [29], it is a public dataset with 840 images. Following the original paper, we assign 37.5% of the images to the test set. EMDS-6 is available at https://figshare.com/articles/dataset/EMDS-6/17125025/1.

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 [30], 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 corresponding masks, split in a training and a test set by the original authors of the dataset, which were created by human experts.

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 [31] has been 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 [32], which includes 1447 images for training and 289 images for validation. This dataset can be accessed at https://github.com/HYOJINPARK/ExtPortraitSeg.

3.4.9. Camouflaged Segmentation (CAM)

The CAMO dataset [33] is specifically created for identifying and separating 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 aims to assess the performance of our ensembles. Evaluation is carried out on several real-world datasets. We perform 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.

The experiments between the various topologies are not symmetrical, given the different computation times for training. The experiments involved modifying the size of images based on the input size requirements of the models. However, the predicted masks were always returned to their original dimensions.

doi:10.20944/preprints202310.0572.v1

4.1. Experiments: DeepLabV3+

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

- initial learning rate = 0.01;
- number of epoch = 10 or 15 (it depends 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 have tested some backbones to be coupled with DeepLabV3+: ResNet18 (RN18) pretrained on ImageNet; ResNet50 (RN50) pretrained on ImageNet; ResNet101 (RN101) pretrained on the VOC segmentation dataset. DeepLabV3+ is trained for 10 epochs if it is coupled with DA1 or for 15 epochs if DA2 is used as 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 are 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, $Comb1$, $Comb2$, $Comb3$) are trained once using DA1 and once using DA2.
- EM2(10) is similar to the previous ensemble, but $LDiceBES$ is used instead of L_T.
- In EM2(5)_DAx, five RN101 networks are trained using the loss of EM2(10). All five networks are trained using data augmentation DAx.
- EM3(10) is similar to the previous ensemble, but L_{STR} is used as a loss function.

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

Table 3. Performance of the proposed DeepLabV3+ ensembles on various benchmark datasets, as measured by Dice scores.

	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

- Among stand-alone networks, RN101 obtains the best average performance, but in RIBS (small training set) it works worse than the others. This probably happens because it is a larger network with respect to RN18 and RN50, thus requiring a larger training set for better tuning.
- ERN101(10) always outperforms RN101(1).
- E101(10) outperforms ERN101(10) with a p-value of 0.0078 (Wilcoxon signed rank test) and EM(10) outperforms 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 is no winner, the various losses lead to similar performance.
- EM3(10) obtains the highest average performance but the p-value is quite high: it outperforms EM(10) with a p-value of 0.1406 and EM2(10) with a p-value of 0.2812.
- There is no statistical difference between the performance of EM2(5)_DA1 and EM2(5)_DA2.

The IoU performance indicator is reported in Table 4 only for the best ensembles. Using IoU confirms the conclusions obtained with Dice. EM3(10) obtains the highest average performance but the p-value is quite high: it outperforms EM(10) with a p-value of 0.1484 and EM2(10) with a p-value of 0.2656.

Table 4. IoU obtained by the DeepLabV3+ based ensembles.

	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 are obtained using a range of diverse datasets, so we are pretty confident that these results are reliable.

4.2. Experiments: Combining Different Topologies

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

- LRa: 10^{-4}.
- LRb: $5 \cdot 10^{-4}$ decaying to $5 \cdot 10^{-5}$ after 10 epochs.
- LRc: $5 \cdot 10^{-5}$ decaying to $5 \cdot 10^{-6}$ 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, 6 and 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 Table 6 and Table 7, the SM column indicates whether we are 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, 6 and 7 report the performance of the following ensembles.

- Fusion: the combination of all the nets varying DA and LR strategy.
- Baseline Ensemble: fusion between 9 networks (same size of Fusion) obtained retraining DA3-LRc nine times.
- Previous: the best ensemble previously reported among [34–36], where SOTA performance was obtained for the given segmentation problem.

Table 5. Dice score obtained by the HardNet based ensembles.

	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**
Previous			0.863	**0.886**	0.916	—

Table 6. Dice score obtained by the PVT based ensembles.

	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**
Previous				0.877	0.883	0.922	—

Table 7. Dice score obtained by the HSN based ensembles.

	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**
Previous				0.879	0.879	—	—

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

- Fusion obtains the best performance, outperforming (on average) the stand-alone approaches and previous ensemble.
- There is no clear winner among the different data augmentation approaches and learning rate strategies.
- The proposed Fusion always improves the Baseline Ensemble except in CAMO. In this dataset there is a significant difference in performance between LRc and the other learning strategy, combining only the 3 networks based on LRc (i.e., using the three data augmentations coupled with LRc) both HS and PVT get a Dice of 0.830, outperforming the Baseline Ensemble.

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

In Table 8, our ensembles are compared with the SOTA reported in the literature. In our final proposed ensembles, the methods are 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) ⊖ Fusion(FH) ⊖ Fusion(PVT) ⊖ Fusion(HSN).
- Ens2: Fusion(FH) ⊖ Fusion(PVT) ⊖ Fusion(HSN).
- Ens3: Fusion(PVT) ⊖ Fusion(HSN).

Table 8. Comparison with previous SOTA ensembles.

	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
[34]	0.874	0.893	0.926	—
[35]	—	0.895	—	—
[36]	0.885	—	—	—

It is clear that combining different network architectures leads to higher performance than with a single topology. Moreover, we obtain new SOTA performance.

5. Conclusions

Many interesting results have been obtained in this work. However, we cannot state that the results obtained from these tests can be generalized to other application domains. For this reason, more tests will be run in the future to evaluate the conclusions reported here, namely, to prove that:

- a fusion of different convolutional and transformer networks can achieve SOTA performance;
- applying different approaches to learning rate strategy is a feasible method to build a set of segmentation networks;
- a better way to add the transformers (HSN and PVT) in an ensemble is to modify the way the final segmentation map is obtained, avoiding excessively sharp masks.

As future work, we aim through techniques such as pruning, quantization, low-ranking factorization and distillation, to decrease the complexity of the ensembles.

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.

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.
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.
3. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition, 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.
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 Transactions on Pattern Analysis and Machine Intelligence* **2017**, *40*, 834–848.
6. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2017**, *39*, 2481–2495.
7. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 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. IEEE/CVF International Conference on Computer Vision, 2021, pp. 568–578.
9. Mohammed, A.; Kora, R. A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University-Computer and Information Sciences* **2023**.
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, 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, 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. *Computers in Biology and Medicine* **2022**, *150*, 106173.

13. Rokach, L. Ensemble-based classifiers. *Artificial Intelligence Review* **2010**, *33*, 1–39. doi:10.1007/s10462-009-9124-7.
14. Polikar, R. Ensemble Based Systems in Decision Making. *IEEE Circuits and Systems Magazine* **2006**, *6*, 21–45. doi:10.1109/MCAS.2006.1688199.
15. Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; Ma, Q. A survey on ensemble learning. *Frontiers of Computer Science* **2020**, *14*, 241–258. doi:10.1007/s11704-019-8208-z.
16. Schapire, R.E. The strength of weak learnability. *Machine learning* **1990**, *5*, 197–227. doi:10.1007/BF00116037.
17. Breiman, L. Bagging Predictors. *Machine learning* **1996**, *24*, 123–140. doi:10.1007/BF00058655.
18. Valiant, L.G. A Theory of the Learnable. *Communications of the ACM* **1984**, *27*, 1134–1142. doi:10.1145/1968.1972.
19. Kearns, M.; Valiant, L.G. Cryptographic Limitations on Learning Boolean Formulae and Finite Automata. *Journal of the ACM* **1994**, *41*, 67–95. doi:10.1145/174644.174647.
20. Efron, B. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics* **1979**, *7*, 1–26. doi:10.1214/aos/1176344552.
21. Alexandre, L.A.; Campilho, A.C.; Kamel, M. On combining classifiers using sum and product rules. *Pattern Recognition Letters* **2001**, *22*, 1283–1289. Selected Papers from the 11th Portuguese Conference on Pattern Recognition - RECPAD2000, doi:10.1016/S0167-8655(01)00073-3.
22. Ganaie, M.A.; Hu, M.; Malik, A.K.; Tanveer, M.; Suganthan, P.N. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence* **2022**, *115*, 105151. doi:10.1016/j.engappai.2022.105151.
23. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. Computer Vision – ECCV 2018: 15th European Conference; Springer-Verlag: Berlin, Heidelberg, 2018; pp. 833–851. doi:10.1007/978-3-030-01234-2_49.
24. Nanni, L.; Lumini, A.; Loreggia, A.; Formaggio, A.; Cuza, D. An Empirical Study on Ensemble of Segmentation Approaches. *Signals* **2022**, *3*, 341–358. doi:10.3390/signals3020022.
25. Lumini, A.; Nanni, L. Fair comparison of skin detection approaches on publicly available datasets. *Expert Systems with Applications* **2020**, *160*, 113677. doi:https://doi.org/10.1016/j.eswa.2020.113677.
26. Phung, S.L.; Bouzerdoum, A.; Chai, D. Skin segmentation using color pixel classification: analysis and comparison. *IEEE transactions on pattern analysis and machine intelligence* **2005**, *27*, 148–154.
27. Liu, Y.; Cao, F.; Zhao, J.; Chu, J. Segmentation of White Blood Cells Image Using Adaptive Location and Iteration. *IEEE Journal of Biomedical and Health Informatics* **2017**, *21*, 1644–1655. doi:10.1109/JBHI.2016.2623421.
28. Filali, I.; Achour, B.; Belkadi, M.; Lalam, M. Graph ranking based butterfly segmentation in ecological images. *Ecological Informatics* **2022**, *68*, 101553. doi:10.1016/j.ecoinf.2022.101553.
29. 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. *Frontiers in Microbiology* **2022**, *13*. doi:10.3389/fmicb.2022.829027.
30. 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, 2021, [arXiv:eess.IV/2107.01327].
31. Liu, L.; Liu, M.; Meng, K.; Yang, L.; Zhao, M.; Mei, S. Camouflaged locust segmentation based on PraNet. *Computers and Electronics in Agriculture* **2022**, *198*, 107061. doi:10.1016/j.compag.2022.107061.
32. Park, H.; Sjösund, L.L.; Yoo, Y.; Kwak, N. ExtremeC3Net: Extreme Lightweight Portrait Segmentation Networks using Advanced C3-modules, 2019, [arXiv:cs.CV/1908.03093].
33. 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. doi:10.1109/ACCESS.2021.3064443.
34. Nanni, L.; Lumini, A.; Loreggia, A.; Formaggio, A.; Cuza, D. An Empirical Study on Ensemble of Segmentation Approaches. *Signals* **2022**, *3*, 341–358. doi:10.3390/signals3020022.
35. Nanni, L.; Loreggia, A.; Lumini, A.; Dorizza, A. A Standardized Approach for Skin Detection: Analysis of the Literature and Case Studies. *Journal of Imaging* **2023**, *9*. doi:10.3390/jimaging9020035.
36. Nanni, L.; Fantozzi, C.; Loreggia, A.; Lumini, A. Ensembles of Convolutional Neural Networks and Transformers for Polyp Segmentation. *Sensors* **2023**, *23*. doi:10.3390/s23104688.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.